Scale-robust Deep Learning for Visual Recognition

Jie Zequn

(B.Eng., University of Science and Technology of China)

A THESIS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

December 2016

Declaration of Authorship

I, Jie Zequn, declare that this thesis titled, "Scale-robust Deep Learning for Visual Recognition" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Zequn Jie

Date: Dec 13 2016

Abstract

In recent years, deep learning has achieved great progress in almost all the visual recognition tasks. Nevertheless, deep learning lacks both image-level and object-level scale-robustness, making it difficult to handle the recognition tasks where testing images are in wide range of scales or contain objects with significantly diverse scales. In this thesis, we focus on improving both image-level and object-level scale-robustness for deep learning, leading to better recognition performance faced with the images and objects having large scale ranges.

First, scene recognition requires scale invariance for better recognizing the captured images of diverse scales. To achieve scale invariance for scene recognition, we proposed a framework integrating the recent powerful deep convolutional networks and locality-constrained linear coding. This framework first fine tunes multi-level convolutional neural networks (CNNs) in a cascaded way, then extracts multi-level CNN features to learn a cross-level universal codebook, and finally performs locality-constrained linear coding (LLC) and max-pooling on the patches of all levels to form the final representation.

Second, we proposed an end-to-end object detection framework based on fully convolutional networks (FCN) to detect vehicles and pedestrians. It integrates both hypothesis generation and hypothesis verification stages in conventional vehicle and pedestrian detection into a single process. Relied on the high-level semantic confidence of vehicles (or pedestrians) obtained by FCN, vehicle (or pedestrian) classification becomes more accurate. Benefited from FCN, time cost is significantly reduced compared to the one-by-one CNN pass strategy in other deep learning methods. Moreover, the positions of

vehicles (or pedestrians) can be directly decided by mapping the output neuron in the output confidence map back to its receptive field in the image. No position regression is needed like other deep learning methods.

Third, existing localization strategies generally fail in producing satisfying localization accuracy for small objects. We thus proposed a novel scale-aware pixel-wise object proposal network to tackle the challenges. A fully convolutional network is employed to predict location of object proposal for each pixel. The produced ensemble of pixel-wise object proposals enhances the chance of finding the object significantly without incurring heavy computational cost. To solve the challenge of localizing objects at small scale, two localization networks which are specialized for localizing objects with different scales are introduced, following the divide-and-conquer philosophy. Location outputs of these two networks are then adaptively combined to generate the final prediction of object locations by a large-/small-size weighting network.

Fourth, in object detection, it is common that multiple objects are shown in one captured image. Existing localization algorithms usually search for possible object regions over multiple locations and scales separately, which ignore the interdependency among different objects. To incorporate global interdependency between objects into localization, we propose an effective Tree-structured Reinforcement Learning (Tree-RL) approach to sequentially search for objects by fully exploiting both the current observation and historical search paths. The Tree-RL approach learns multiple searching policies through maximizing the long-term reward that reflects localization accuracy over all the objects. Starting with taking the entire image as a proposal, the Tree-RL approach allows the agent to sequentially discover multiple objects via a tree-structured traversing scheme.

Allowing multiple near-optimal policies, Tree-RL is able to find multiple objects with a single feed-forward pass and cover different objects with various scales, which is quite appealing in object detection.

We systematically conduct experiments on several benchmarks, and conclusively demonstrate the effectiveness of the above proposed methods in scene classification, object proposal generation, object localization and object detection.

Acknowledgements

First of all, I would like to express my deepest appreciation to my supervisors, Prof. Wen Feng Lu and Prof. Eng Hock Francis Tay, for their valuable guidance, scientific advices and encouragements throughout the entire duration of my research. This Ph.D. degree and dissertation would not have been possible to be completed without their great supports. It was my fortune to have this precious opportunity to study with these two greatest supervisors.

Besides my advisors, my sincere thanks also goes to Prof. Shuicheng Yan and Prof. Jiashi Feng, who provided me an opportunity to join their team, and who gave extensive valuable comments in project related issues . Without their precious support it would not be possible to conduct this research.

My sincere thanks also go to all the people in Vision and Machine Learning Lab: Ms. Xiaodan Liang, Mr. Yunchao Wei, Mr. Bo Zhao, Mr. Hao Liu, Mr. Wenhan Yang, Mr. Fang Zhao, Mr. Xiaojie Jin, Mr. Yunpeng Chen, Mr. Canyi Lu, Mr. Jianshu Li, Mr. Jian Zhao and Mr. Jianan Li for their kindest helps throughout my Ph.D study. I have to express my special thanks to Ms. Quanhong Fu who helped me a lot in academic writing and paper polishing as well. They are also my friends and make my graduate study in Singapore colorful and memorable.

Last but not least, I dedicate this small achievement to my family for their love, understanding and patience.

Publications

1. **Zequn Jie**, Xiaodan Liang, Jiashi Feng, Wen Feng Lu, Eng Hock Francis Tay, Shuicheng Yan, “Tree-Structured Reinforcement Learning for Sequential Object Localization,” Neural Information on Processing Systems (NIPS), 2016, Barcelona, Spain.

2. **Zequn Jie**, Xiaodan Liang, Jiashi Feng, Wen Feng Lu, Eng Hock Francis Tay, Shuicheng Yan, “Scale-aware Pixel-wise Object Proposal Networks,” in IEEE Transactions on Image Processing (TIP), 2016.

3. **Zequn Jie**, Wen Feng Lu, Siavash Sakhavi, Yunchao Wei, Eng Hock Francis Tay, Shuicheng Yan, “Object Proposal Generation with Fully Convolutional Networks,” in IEEE Transactions on Circuits and System for Video Technology (TCSVT), 2016.

4. **Zequn Jie**, Wen Feng Lu, Eng Hock Francis Tay, “On-road Vehicle Detection with Fully Convolutional Networks,” Conference on Machine Learning and Data Mining, 2016, New York, United States.

5. **Zequn Jie**, Jerry Y.H. Fuh, Eng Hock Francis Tay, Wen Feng Lu, Shuicheng Yan, “Robust Scene Classification with Cross-level LLC Coding on CNN Features,” Asian Conference on Computer Vision (ACCV), 2014, Singapore.

6. Yunchao Wei, Xiaodan Liang, Yunpeng Chen, **Zequn Jie**, Yanhui Xiao, Yao Zhao, Shuicheng Yan. Learning to Segment with Image-level Annotations. In Pattern Recongnition, 2016.

7. Xiaodan Liang, Yunchao Wei, Xiaohui Shen, **Zequn Jie**, Jiashi Feng, Liang Lin, Shuicheng Yan:, “Reversible Recursive Instance-level Object Segmentation,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, Las Vegas, United States.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Deep Learning for Visual Recognition

Deep learning [1] is a subfield of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing [2], transfer learning [3, 4], natural language processing [5], computer vision [6] and many more. There are mainly three important reasons for the booming of deep learning today: the dramatically increased chip processing abilities (e.g. GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms [7].

The Convolutional Neural Networks (CNN) is one of the most notable deep learning approaches where multiple layers are trained in a robust manner [8]. It has been found highly effective and is also the most commonly used in diverse computer vision applications. With the recent developments of CNN schemes in the computer vision domain, some well-known CNN models have emerged.

AlexNet [6] is a significant CNN architecture, which consists of five convolutional layers and three fully connected layers. After inputting one fixed-size (224×224) image, the network would repeatedly convolve and pool the activations, then forward the results into the fully-connected layers. The network was trained on ImageNet and integrated various regularization techniques, such as data augmentation, dropout, etc. AlexNet won the ILSVRC2012 competition [9], and set the tone for the surge of interest in deep convolutional neural network architectures. Nevertheless, there are two major drawbacks of this model: 1) it requires a fixed resolution of the image; 2) there is no clear understanding of why it performs so well.

In 2013, Zeiler et al. [10] introduced a novel visualization technique to give insight into the inner workings of the intermediate feature layers. These visualizations enabled them to find architectures that outperform AlexNet [6] on the ImageNet classification benchmark, and the resulting model, Clarifai, received top performance in the ILSVRC2013 competition.

As for the requirement of a fixed resolution, He et al. [11] proposed a new pooling strategy, i.e. spatial pyramid pooling, to eliminate the restriction of the image size. The resulting SPP-net could boost the accuracy of a variety of published CNN architectures despite their different designs.

In addition to the commonly used configuration of the CNN structure (five convolutional layers plus three fully connected layers), there are also approaches trying to explore deeper networks. In contrast to AlexNet, VGG [12] increased the depth of the network by adding more convolutional layers and taking advantage of very small convolutional filters in all layers. Similarly, Szegedy et al. [13] proposed a model, GoogLeNet, which also has quite a deep structure (22 layers) and has achieved leading performance in the ILSVRC2014 competition [9].

1.2 Image Scale and Object Scale

Three main sources of natural variation are the location, the viewpoint, and the size of an object or pattern. Variations in location are dealt with very well by a CNN [14], which follows naturally from the weight sharing employed in the convolution layers [15]. CNNs can also handle variations in viewpoint by creating filters that respond to viewpoint-invariant features [16]. Size variations pose a particular challenge in CNNs [17], especially when dealing with image corpora containing images of varying resolutions and depicting objects and patterns at different sizes and scales, as a result of varying distances from the camera and blurring by optical imperfections, respectively. This leads to variations in image resolution, object size, and image scale, which are two different properties of images. The relations between image resolution, object size, and image scale is formalized in digital image analysis using Fourier theory. Spatial frequencies are a central concept in the Fourier approach to image processing. Spatial frequencies are the two-dimensional analog of frequencies in signal processing. The fine details of an image are captured by high spatial frequencies, whereas the coarse visual structures are captured by low spatial frequencies. In what follows, we provide a brief

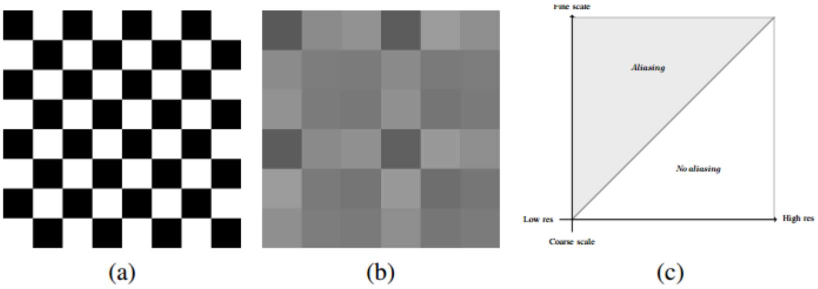


Figure 1.1: Illustration of aliasing. (a) Image of a chessboard. (b) Reproductions of the chessboard with an image of insufficient resolution (6 * 6 pixels). The reproduction is obtained by applying bicubic interpolation. (c) The space spanned by image resolution and image scale. Images defined by resolutionscale combinations in the shaded area suffer from aliasing.

intuitive discussion of the relation between resolution and scale, without resorting to mathematical formulations.

The scale of an image refers to its spatial frequency content. Fine scale images contain the range from high spatial frequencies (associated with small visual structures) down to low spatial frequencies (with large visual structures). Coarse scale images contain low spatial frequencies only. The operation of spatial smoothing (or blurring) of an image corresponds to the operation of a low-pass filter: high spatial frequencies are removed and low spatial frequencies are retained. So, spatial smoothing a fine scale image yields a coarser scale image.

The relation between the resolution and the scale of an image follows from the observation that in order to represent visual details, an image should have a resolution that is sufficiently high to accommodate the representation of the details. For instance, we consider the chessboard pattern shown in Figure 1.1a. Figure 1.1b shows a 6 × 6 pixel reproduction of the chessboard pattern. The resolution of the reproduction is insufficient to represent the fine structure of the chessboard pattern. The distortion of an original image due to insufficient resolution (or sampling) is called aliasing.

As this example illustrates, image resolution imposes a limit to the scale at which visual structure can be represented. Figure 1.1c displays the space spanned by resolution (horizontal axis) and scale (vertical axis). The limit is represented by separation of the shaded and unshaded regions. Any image combining a scale and resolution in the shaded area suffers from aliasing. The sharpest images are located at the shaded-unshaded boundary. Blurring an image corresponds to a vertical downward movement into the unshaded region.

Having discussed the relation between resolution and scale, we now turn to the discussion of the relation of object size to resolution and scale. Real-world images with a given scale and resolution contain objects and structures at a range of sizes [18].In addition, it may contain visual texture associated with the paper it was printed on and with the tools that were used used to create the artwork. Importantly, the same object may appear at different sizes. For instance, in the artwork shown there persons depicted at different sizes. The three persons in the middle are much larger in size than the one at the lower right corner. The relation between image resolution and object size is that the resolution puts a lower bound on the size of objects that can be represented in the image. If the resolution is too low, the smaller objects cannot be distinguished anymore. Similarly, the relation between image scale and object size is that if the scale becomes too coarse, the smaller objects cannot be distinguished anymore. Image smoothing removes the high-spatial frequencies associated with the visual characteristics of small objects.

1.3 State-of-the-art Research on Scale-invariant Image Representations

Training CNNs on large image collections that often exhibit variations in image resolution, depicted object sizes, and image scale, is a challenge. The convolutional filters, which are automatically tuned during the CNN training procedure, have to deal with these variations. Supported by the acquired filters the CNN should ignore task-irrelevant variations in image resolution, object size, and image scale and take into account task-relevant features at a specific scale. The filters providing such support are referred to as scale-invariant and scale-variant filters, respectively [19].

The importance of scale-variance was previously highlighted by Gluckman [19] and Park et al. [20], albeit for two different reasons. The first reason put forward by Gluckman arises from the observation that images are only partially described by scale invariance [19]. When decomposing an image into its scale-invariant components, by means of a scaleinvariant pyramid, and subsequently reconstructing the image based on the scale-invariant components the result does not fully match the initial image, and the statistics of the resulting image do not match those of natural images. For training a CNN this means that when forcing the filters to be scaleinvariant we might miss image structure which is relevant to the task. Gluckman demonstrated this, by means of his proposed space-variant image pyramids, which separate scalespecific from scale-invariant information in [19] and found that object recognition benefited from scale-variant information.

The second reason was presented by Park et al. in [20], where they argue that the need for scale-variance emerges from the limit imposed by image resolution, stating that

"Recognizing a 3-pixel tall object is fundamentally harder than recognizing a 300-pixel object or a 3000-pixel object." [20]. While recognising very large objects comes with it own challenges, it is obvious that the recognition task can be very different depending on the resolution of the image. Moreover, the observation that recognition changes based on the resolution ties in with the previously observed interaction between resolution and scale: as a reduction in resolution also changes the scale. Park et al. identify that most multi-scale models ignore that most naturally occurring variation in scale, within images, occurs jointly with variation in resolution, i.e. objects further away from the camera are represented at a lower scale and at a lower resolution. As such they implement a multi-resolution model and demonstrate that explicitly incorporating scale-variance boosts performance.

Standard CNN trained without any data augmentation will develop representations which are scale-variant. As such it is only capable of recognising the features it was trained on, at the scale it was trained on, such a CNN cannot deal with scale-variant features at different scales. A straightforward solution to this limitation is to expose the CNN to multiple scales during training, this approach is typically referred to as scale jittering [12, 13, 21]. It is commonly used as a data augmentation approach to increase the amount of training dataset, and as a consequence reduce overfitting. Additionally, it has been shown that scale jittering improves classification performance [12]. While part of the improved performance is due to the increase in training data and reduced overfitting, scale jittering also allows the CNN to learn to recognize more scale-variant features, and potentially develop scale-invariant filters. Scale-invariant filters might emerge from the CNN being exposed to scale variants of the same feature. However, standard CNN typically do not develop scale-invariant filters [22], and instead will require more filters to deal with the scaled variants of the same feature [6], in addition to the filters needed to capture scale-variant features. A consequence of this increase in parameters, which increases further when more scale variation is introduced, is that the CNN becomes more prone to overfit and training the network becomes more difficult in general. In practice this limits scale-jittering to small scale variations. Moreover, scale-jittering is typically implemented as jittering the resolution, rather than explicitly changing the scale, which potentially means that jittered versions are actually of the same scale.

1.4 Research Motivation

Based on the literature studied in the scale-invariant image representation for visual recognition, the following research gaps have been identified, which become the focus of the research effort presented in this thesis.

For image-level scale-robustness, there is lack of a method that can mitigate the influences brought by the image scale change and keep consistent high accuracy in image-level visual recognition, e.g. scene classification.

For object-level scale-robustness, there is lack of method that can achieve high object-level recognition recall and precision for objects in a wide range of scales, especially for small-scale objects. Therefore, the object-level recognition tasks including object proposal, localization and detection remain a large improvement space in terms of recall and accuracy.

1.5 Research Contributions

Our main contributions stem from the proposed solutions of the research motivations. We summarize them as follows:

- **Cross-level LLC Coding on CNN Features for Scene Recognition.** We proposed a cross-level Locality-constrained Linear Coding and pooling framework (*cross-level LLC-CNN*) on multi-level CNN features to enhance the discrimination and scale invariance of the image representation for scene classification problems. Based on the cascaded fine-tuning scheme, the CNN features gain stronger discrimination in scene classification. In addition, with cross-level Locality-constrained Linear Coding and pooling on these multi-level fine-tuned CNN features, robustness to scale transformation is improved.

- **Fully Convolutional Networks Based Object Detection for Vehicle and Pedestrian Detection.** We proposed a unified end-to-end system which can directly process the raw visual data captured by cameras to perform detection of vehicle and pedestrian simultaneously. The proposed method is based on fully convolutional networks (FCN) which adopts a novel object localization way that directly maps the output neuron in the objectness map to its receptive field in the image. FCN only requires to feed the image into the CNN with very few times thus runs much faster than R-CNN. Additionally, by training only on target objects (i.e. vehicles and pedestrians), the proposed system outperforms the 2-stage R-CNN which relies on class-agnostic object proposals in detection accuracy.

- **Scale-aware Object Proposal Networks for Object Detection.** We developed an effective scale-aware pixel-wise localization network (SPOP-net) for object proposal generation. The network fully exploits the available pixel-wise segmentation annotations and predicts the proposals pixel-wisely. Each proposal combines two proposals predicted by two networks specialized for different sizes respectively. The combination follows a weighting mechanism utilizing the weighting confidence produced by a large-/small-size object classification model. This strategy is shown to enhance the accuracy of localization on small objects. The proposals of the SPOP-net used in Fast-RCNN detector also provide superior detection accuracy, benefiting from the high recall rate of the proposed model. Such proposals achieving high recall and localization accuracy for both large and small objects are especially beneficial to the real-world detection tasks where objects are in a wide range of scales.

- **Tree-structured Reinforcement Learning for Sequential Object Localization.** We proposed a novel Tree-structured Reinforcement Learning (Tree-RL) approach to sequentially search for objects with the consideration of global interdependency between objects. It follows a top-down tree search scheme to allow the agent to travel along multiple near-optimal paths to discovery multiple objects. By incorporating the global interdependencies of objects, multiple objects could be detected with a small number of proposals in a sequential and intelligent fashion.

1.6 Thesis Organization

In the remaining chapters of the thesis, each chapter covers a topic of the research contributions. Chapter 2 is focused on the problem of scene recognition. A novel cross-level LLC coding on CNN features is proposed to solve the problem and shows to have better scale invariance and recognition accuracy than the previous methods. Chapter 3 solves the problem of vehicle and pedestrian detection with fully convolutional networks. First, a one-stage detection method is proposed for vehicle and pedestrian detection that unifying object localization and classification into an end-to-end system. Second, a two-stage detection system is utilized for vehicle and pedestrian detection. It first generates class-agnostic object proposals with fully convolutional networks and then perform classification on the proposals with fast R-CNN. Chapter 4 develops a novel scale-aware pixel-wise object proposal network to enhance the detection accuracy for a wide range of scales, especially for small objects. Combined with fast-RCNN, the novel method can achieve object detection with high accuracy. Chapter 5 proposes a novel tree-structured reinforcement learning to incorporate global interdependency of objects. It is useful in the cases that multiple objects are in one image. Chapter 6 draws the conclusions based on the works discussed in the thesis and gives suggestions for possible future works.

Chapter 2

Cross-level LLC Coding for Scene Recognition

2.1 Overview

The concept of scene recognition is to assign the images to be recognized to one of the candidate scene categories. The candidate scene categories must be within those categories that the training images belong to since the training process will provide the image content and appearance information of each category to the model. After training, the trained model must be able to generalize to the unseen images and predict their scene categories correctly.

Scene recognition outputs semantic scene type of the surrounding environment. However, it is not an easy task due to the great diversity of image contents as well as the variations in illumination and scale conditions. Early approaches [23–26] utilize hand-crafted features, e.g., SIFT [27] and HOG [28], which require designing lots of tricks and lack image representation power for different complex problems. Later, in contrast to hand-crafted features, image features learned from Convolutional Neural Network (CNN) [15] have been successfully applied to scene recognition [6, 29, 30]. However, CNN features retain too much global spatial information and lack invariance to scale transformation since raw pixels are filtered and pooled alternatively within their local neighborhoods in the network. Actually, as shown in [10], feature maps after each layer can be used to reconstruct the original image due to the high spatial order of CNN features. Although the max-pooling layer after each convolution layer provides a certain degree of invariance to local scale transformation, invariance to global scale transformation cannot be guaranteed. Based on the 4096-dimensional global CNN features, their variance to scale transformation will directly lead to the decrease of recognition accuracy when only scale transformed images are available for testing.

To improve the scale invariance of CNN features, a multi-level pooling frameworks has been proposed by [14]. Specifically, CNN features from patches with various sizes in different levels of the framework are extracted as mid-level image representations, followed by an intra-level pooling process over these patches. Within one level, densely distributed patch features cover the whole image and are pooled in an orderless way. By pooling the patch CNN features in each level, the final representation becomes patch-level orderless and scale invariant to a certain degree.

However, when the whole testing image is scaled, all the patches of its finer levels will be scaled by the same scaling ratio accordingly. In this case, CNN features of both

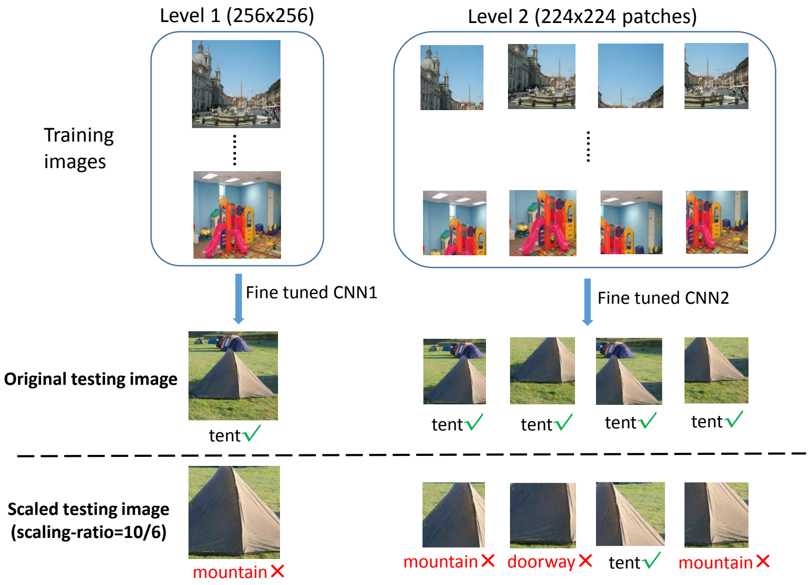


Figure 2.1: Predictions of each patch in level 1 and level 2 of both the original image and its scaled version (10/6 ratio) with the CNN trained on original training samples. It is shown that predictions of the original testing image are all correct, while there are many wrong predictions for all levels of the scaled image.

the whole image and the patches of all levels will not work well since CNN features of each level are learned in a supervised manner from the training patches in the same level. To demonstrate this, we conduct an experiment on an image from SUN 397 [31] with the model trained on original training samples. Figure 1 shows the prediction of each patch in level 1 and level 2 of both the original image and its scaled version (10/6 ratio). As can be seen, both the whole image (level 1) and patches in level 2 obtain the correct predictions – “tent” by the fine-tuned CNN of their own level. In contrast, the scaled testing image obtains a wrong prediction – “mountain” using the fine-tuned CNN trained on the original non-scaled training images. A similar situation also happens in level 2, where 3 patches of the total 4 obtain wrong predictions. In this case, even if orderless pooling is performed on top of the CNN features of patches, no scale invariance can be guaranteed since the features to be pooled, i.e., CNN features of each patch have changed due to the scaling of the whole testing image.

To address the problem of lacking scale invariance for CNN, we present a simple but effective framework, which we refer to as cross-level LLC coding and cascaded fine-tuned CNN (*cross-level LLC-CNN*), to provide CNN features more robust to scale transformation. The pipeline is illustrated in Figure 2. Details will be presented in Section 3. Our proposed framework first fine-tunes CNNs for each level in a cascaded way, which means the CNN parameters learned in the coarser level are utilized as the initialization of the finer level. Subsequently, CNN features of all the patches in multiple levels are extracted by their own fine-tuned CNNs. Then we learn a universal (cross-level) codebook on all the CNN features of multi-level patches by k-means. Based on this universal codebook, Locality-constrained Linear Coding (LLC) [32] is performed for all the CNN features. The locality-constrained nature of LLC ensures each patch to find its most similar patches among all the patches distributed in multiple levels, even if the image and its patches are scaled. This helps build a more robust representation to scale transformation. Finally, all the LLC features of patches in multiple levels are max-pooled together to build the final image representation.

Extensive experiments on two challenging scene classification datasets, i.e., MIT indoor scenes [33] and SUN 397 [31], verify the superiority of the cross-level LLC coding on the cascaded fine-tuned CNN features over other conventional methods. The rest of the chapter is organized as follows. First, we give a survey of typical methods for scene classification in Section 2.2. Then we elaborate on our framework, cross-level Locality-constrained Linear Coding (LLC) of CNN features in Section 2.3. After showing experimental results in Section 2.4, we draw a summary in Section 2.5.

2.2 Related Work

The approaches used for scene classification are based on low-level features at first. The problem of scene modeling for classification using low-level has been studied in image and video retrieval for several years [43]. Previous works used color, texture and shape features directly from the image combined with supervised learning methods to classify images into several semantic classes (e.g. indoor, outdoor, city, landscape, sunset, forest, etc.). On the other hand, the modeling of scene by a semantic intermediate representation was next proposed in order to overcome the drawbacks of low-level features based approaches when the categories become many and similar in each other, low-level features are not discriminative as a huge gap existed between low-level features and high-level semantics. Hence, current approaches dealing with this topic can be roughly divided into two major types: low-level based and intermediate semantic modeling.

2.2.1 Low-level scene modeling

The low-level features (e.g. color, texture, shape) based methods firstly extract low-level features from images then directly use them as input data to various classifiers (e.g. SVM, KNN, Bayes classifier) to obtain the final results category of the image. This type of approaches were motivated by the view that the type of scene can be directly described by the color/texture properties of the image. For example, a forest scene presents highly textured regions (trees), a mountain scene is described by a large amount of green and brown (plants and rocks), a coast scene includes considerable proportion of blue (water and sky), or the presence of straight horizontal and vertical edges denotes an urban scene. For low-level scene modeling, further division can be made as follows:

• Global: the scene is described by low-level features from the whole image.

Vailaya [44] consider the hierarchical classification of vacation images, and show that low-level features can successfully discriminate between many scenes types using a hierarchical structure. Using binary Bayesian classifiers, they attempt to classify the scenes into two types in each level, for each type, the classification can be further made into two sub-types. In this way, the classification can be described as a binary tree structure. Specifically, at the highest level, images are classified into indoor or outdoor; outdoor images are further classified as city or landscape; finally, landscape images are classified into sunset, forest and mountain classes. Different low-level features extracted from the whole image are used at different level depending on the classification problem: indoor/outdoor (using spatial color moments); city/landscape (using edge direction coherence vectors) and so on.

Also in [45], global features, including global color and textures are used to produce a set of semantic labels with a certain belief for each image. They trained k support vector machines (SVM) to classify images. Each test image is classified by the k classifiers and assigned a confidence score for the label that each classifier is attempting to predict. As a result, a k-dimension label-vector is generated for each image. This approach is specially useful for Content Based Image Retrieval (CBIR) and Relevance Feedback (RF) systems.

• Local: the image is first partitioned into several patches, and then features are extracted from each of those patches.

The scene can also be modeled by local low-level features, which are not from a single, whole image representation. Several proposals first split the image into a set of sub-regions, which can be independently described by their own low-level properties. These sub-regions are then classified, and finally the scene is categorized from the individual classification of each sub-region.

The origin of this approach can date back to 1997, when Szummer [46] proposed to independently classify image sub-regions to obtain a final result using a majority voting classifier. The goal of this work was to classify images as indoor or outdoor. The images are first partitioned into 16 patches, then for each patch color histogram and MSAR texture features are extracted. KNN classifiers were employed to classify each patch using the histogram intersection norm, which measures the amount of overlap between corresponding buckets in the two N-dimension histograms. Finally, the whole image is classified using a majority voting scheme from the sub-region classification results. They obtained a 90.3% of accuracy in indoor/outdoor scene classification, showing how scene types can be inferred from classification of low-level image features, especially for the indoor/outdoor scene retrieval problem.

Similar results were also obtained by Paek and Chang [47]. Moreover, they developed a framework to combine multiple probabilistic classifiers in a belief network. They trained classifiers for indoor/outdoor, sky/no sky and vegetation/no vegetation as secondary cues for the indoor/outdoor problem. The classification results for each one are then fed into a belief work to take the integrated decision.

Despite the good performance low-level scene modeling approaches to some extent, it is difficult for low-level based approaches to generalize to additional image data which are beyond the training set. More seriously, they lack an intermediate representation as bridge to fill the gap between low-level features and high-level semantics, thus hard to tackle the problem which contains large number of categories. Hence, later researches pay more attention to intermediate semantic modeling approaches.

2.2.2 Mid-level semantic scene modeling (bag-of-words)

Only depends on the low-level features, classification performance will decrease quickly as the number of categories increases for that many different categories may share very similar low-level features. For example, both street and highway include many straight lines and gray color; large amount of blue color may exist in coast and mountain; similar texture features and green color can be found in both grass and forest. Thus, intermediate representations (i.e. bag-of-words based methods) which can abridge the gap between low-level and high-level were proposed for scene classification.

Bag-of-words framework treats an image as a collection of appearance descriptors extracted from local patches. After obtaining low-level features distributed in feature space, clustering methods will be then used for generating a dictionary of visual words. Then the image can be represented as distributions of these semantic concepts/visual words. Usually, after obtaining visual words distributions, some topic models (e.g. pLSA, LDA, etc.) can be used to further discover topics in an image so that an image is modelled as mixtures of topics. The topics distributions of the image is used to classify an image as belong to a certain scene. This type of methods overcomes the weakness of semantic objects based approaches, which need to do difficult segmentation, object detection and recognition.

Early methods of this type adopted K-means Vector Quantization (VQ) to encode local features [34]. Later, Sparse Coding (SC) [35] was proposed to relax the cardinality constraint of VQ, which requires that only one coefficient of the code words is 1 while the rest are all 0. Later, some extensions of Sparse Coding, such as Locality-constraint linear coding (LLC) [32], which demands similar features should have similar codes over similar visual words, and Hierarchical Sparse Coding (HSC) [36], which builds a hierarchical structure of sparse coding, were also proposed. The performance of them showed obvious improvement compared to Bag-of-Words. To add spatial organization information to the orderless Bag-of-Features, Spatial Pyramid Matching (SPM) [26] partitions the entire image into multi-scale patches and performs VQ or SC on each patch. Also, Orientational Pyramid Matching (OPM) [37] was used to partition the image in a more discriminative way, with the consideration of the orientation information. In this type of framework, local scale invariant hand-crafted features are usually relied on, such as SIFT [27] and HOG [28]. The combination between low-level scale invariant features and mid-level orderless pooling builds a more robust representation to scale transformation. The main limitation of this type of framework lies in the designing of hand-crafted features, which needs lots of tricks and is not applicable to some specific complex problems.

2.2.3 Deep learning methods

The other type of framework, i.e., deep learning, tries to model high-level abstractions of visual data by using architectures containing multiple layers of non-linear transformations. Convolutional Neural Network (CNN), as a typical example of deep learning models, has achieved great success in image classification, including ILSVRC 2012, ILSVRC 2013, tiny image dataset CIFAR-10/100 [38] and hand-written digits recognition [8]. [10] later proved that CNN features do not have invariance to different kinds of geometric transformations, e.g., scale transformation and rotation transformation. To strengthen the representation power of CNN when scale transformation occurs, [14] proposed a multi-scale orderless pooling framework, which includes CNN feature extraction at multiple levels and VLAD [39] pooling over these features. Our approach differs from this work in the different CNN features extracted and the cross-level feature coding and pooling schemes.

2.3 Cross-level LLC Coding on Multi-level CNN Features

2.3.1 Multi-level Cascaded Fine-tuned CNNs

To capture the context information of various sizes of patches, similar to [14], we adopt a multi-level framework to extract fine-tuned CNN features in multiple levels. The patch sizes of level 1 to level 5 are chosen carefully as follows: 256*256, 224*224, 192*192, 160*160, 128*128. Intuitively, transferring the groundtruth label of the whole image to its patches requires the patches not to be too small. The reason is that in scene classification, the groundtruth label is the high-level semantic abstract on the whole image, and too small local patches usually cannot be summarized as the same abstract concept (groundtruth label) as that of the whole image. Actually, we have found that the single patch recognition accuracy of level 5 with patch size 128*128 only achieves 43.6%, while the recognition accuracy of level 1 is 61.46% on the MIT indoor scenes dataset. Fortunately, although in level 5, the single patch recognition accuracy is much lower than that of the whole image, the recognition accuracy using max-pooled features of this level can still obtain a satisfactory result of 64.97%. Thus, we set the smallest patch size as 128*128. The stride of all the 5 levels is 32 pixels, thus we have 1, 4, 9, 16, 25 patches from level 1 to level 5 respectively.

To improve the discrimination and adaptation power of off-the-shelf CNN features on scene classification datasets, we fine-tune the CNN model pre-trained on ImageNet for each level in a cascaded way. We choose the same CNN architecture with [10] for its proven great performance in ILSVRC 2013. It contains five convolutional layers and three fully-connected layers with 60 million parameters. Since the numbers of categories in scene classification datasets differ from that in ImageNet, we change the number of the outputs of the last fully-connected layer, which represents the predicted probability of each target category, from 1000 in ImageNet to 67 and 397 in MIT indoor scenes and SUN 397 datasets respectively. Before fed into this CNN model, all the patches are resized to 256*256. During the stochastic gradient descent training process, the parameters of the first seven layers are initialized by the parameters pre-trained on ImageNet and the parameters of the last fully-connected layer are randomly initialized with a Gaussian distribution. The learning rates of the convolutional layers, the first two fully-connected layers and the last fully-connected layer are initialized as 0.001, 0.002 and 0.01, respectively and reduced to one tenth of the current rates after every 20 epochs (50 epochs in total). By setting the different learning rates for different layers, the parameters in different layers are updated by different rates. The main reasons for this setting are as follows: the first few convolution layers mainly extract low-level invariant features, such as texture and shape, thus the parameters are more consistent from the pre-trained dataset to the target dataset, whose learning rates are set as a relatively low value (i.e., 0.001); for the final few layers, especially the last fully-connected layer which is specifically adapted to the new target dataset, a higher learning rate is required to guarantee its fast convergence to the new optimum.

To strengthen the connections between the fine-tuned CNNs of different levels and reduce the convergence time, we adopt a cascaded fine-tuning strategy. Specifically, we use the model pre-trained on ImageNet as our initialization when training the CNN of level 1. When training on other finer levels, we always use the model trained on the last coarser level as our initialization. For example, the CNN trained on level 1 will be the initialization when training CNN on level 2. In Section 4, we will show the superiority of the cascaded fine-tuned CNN over off-the-shelf CNN and CNN fine-tuned with the pre-trained model on ImageNet in recognition accuracy.

2.3.2 Cross-level LLC Coding and Pooling on CNN features

Although separate fine-tuning of CNN for each level enhances the discrimination power of CNN features, it is still unstable for scale transformation, as fine-tuned CNNs are trained on the original non-scaled training images and patches, thus naturally characterize the image spatial organization of these non-scaled samples better.

To solve this problem, we propose to use a cross-level feature coding and pooling scheme on the fine-tuned CNN features extracted from all patches of multiple levels. The pipeline is illustrated in Figure 2. Firstly, a 4096-dimensional feature is extracted in each patch with the fine-tuned CNN of their own level. Subsequently, a cross-level codebook is learned by clustering all these multi-level patch CNN features into 16384 clusters

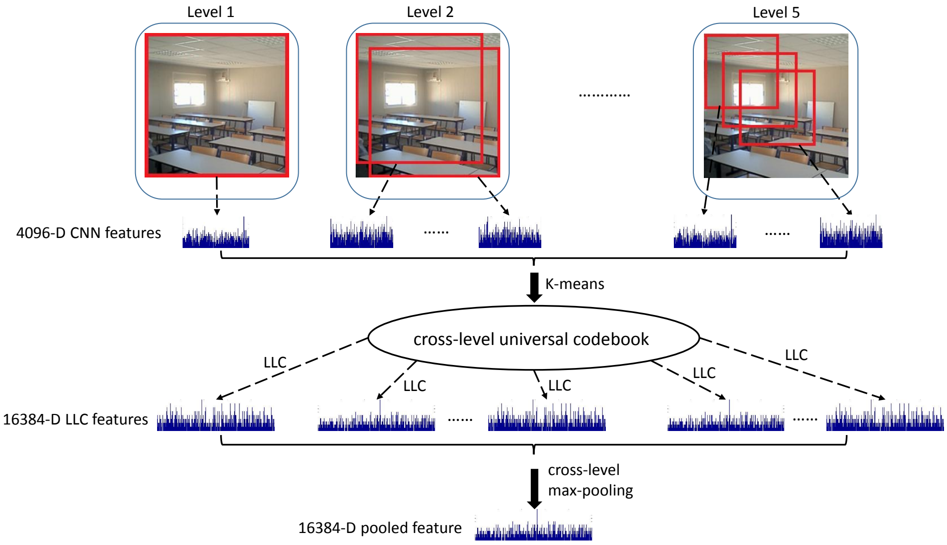


Figure 2.2: Pipeline of cross-level LLC coding and max-pooling on CNN features. First, patch CNN features of all the 5 levels are clustered to learn a cross-level codebook. Next, all these CNN features are encoded based on this codebook via LLC coding. Finally, max-pooling is performed on all the encoded features to form a cross-level pooled feature, as the new image representation.

(4 times as the 4096 dimensions) with the k-means algorithm. By doing this, different patch levels of CNN features can be found among the code words of this cross-level codebook such that the codebook gains multi-level representation power. Next, Locality-constrained Linear Coding (LLC) is performed on the multi-level CNN features based on the learned cross-level codebook. LLC coding enforces the corresponding encoding coefficients to be high if the code words are similar to the feature, and enforces the coefficients of other dissimilar code words to be zero [32]. The underlying hypothesis is that features approximately reside on a lower dimensional manifold in an ambient feature space [40]. Specifically, LLC coding uses the following criteria:

$$\begin{aligned} &\min_{C} \sum_{i=1}^{N} ||x_i - Bc_i||^2 + \lambda ||d_i \odot c_i||^2 \\ &\text{s.t.} \quad \mathbf{1}^T c_i = 1, \forall i. \end{aligned} \tag{2.1}$$

where N is the number of features to be encoded, x_i represents the ith encoded feature, B is the codebook matrix, c_i is the ith LLC coding result, $\odot$ denotes the element-wise multiplication, and $d_i \in R^M$ is the dissimilarity between the encoded feature and the code words with M denoting the codebook size. Specifically,

$$d_i = \exp\left(\frac{\mathrm{dist}(x_i, B)}{\sigma}\right) \tag{2.2}$$

where $dist(x_i, B) = [dist(x_i, b_1), ..., dist(x_i, b_M)]^T$, and $dist(x_i, b_j)$ is the Euclidean distance between x_i and b_j. The analytical solution of LLC is as follows:

$$\begin{aligned} \widetilde{c_i} &= (C_i + \lambda \mathrm{diag}(d)) \setminus \mathbf{1} \\ c_i &= \widetilde{c_i} \setminus \mathbf{1}^T \widetilde{c_i} \end{aligned} \tag{2.3}$$

where $C_i = (B - \mathbf{1}x_i^T)(B - \mathbf{1}x_i^T)^T$ denotes the data covariance matrix. Hence, LLC can be implemented very fast in practice.

By performing LLC on multi-level patch CNN features based on the cross-level codebook, different levels of CNN features extracted from patches of various sizes share a common codebook and can be encoded based on this codebook, regardless of their levels. This naturally enhances the scale invariance of the LLC features since no matter how the whole image and all of its patches are scaled, the CNN features can always find their similar code words in the cross-level codebook, either from the code words of their own levels or from other levels, and use these code words to represent them, leaving the reconstruction coefficients of all the rest dissimilar code words to be zero. As can be seen in Figure 3, CNN features of the original image will probably be represented by the

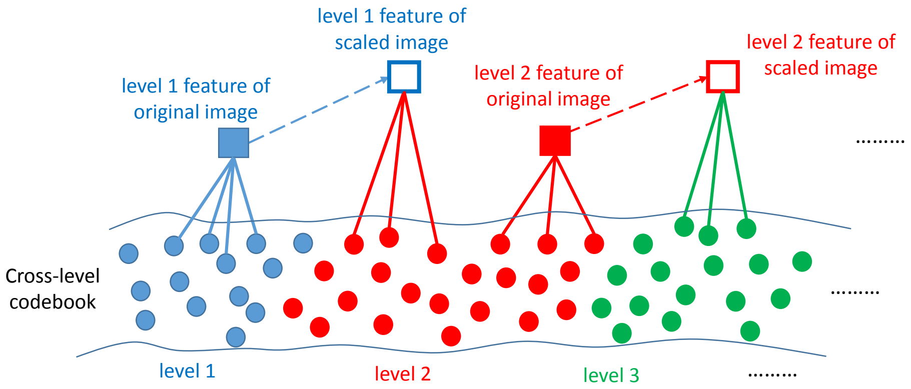

level 1 feature of scaled image
level 1 feature of original image
level 2 feature of original image
level 2 feature of scaled image
....
Cross-level codebook
....
level 1
level 2
level 3
....

FIGURE 2.3: Illustration of selected representation codewords of CNN features for original image and scaled image. Circles in different colors represent code words from different levels. Blue and red solid squares denote level 1 and level 2 CNN features of the original image, respectively, and blue and red hollow squares represent those of the scaled image respectively. (better viewed in color)

code words of their own level, while CNN features of the scaled image may be similar to the code words from other levels and represented by these code words.

After obtaining LLC features of all the patches from multiple levels, we max-pool these cross-level features together in a mid-level (patch-level) orderless manner to form the final image representation. Finally, a linear SVM is trained based on the cross-level pooled features to obtain the predictions. Experimental results on MIT indoor scenes and SUN 397 datasets shown in Section 4 verify the great discrimination and robustness to scale transformation of the proposed image representation.

2.4 Experiments

2.4.1 Datasets

We evaluate the proposed approach on the two currently largest scene classification datasets: MIT indoor scenes and SUN 397.

2.4.1.1 MIT indoor scenes

is the largest indoor scene dataset, which contains 67 categories and a total of 15620 images. The complex spatial layout of the indoor scene image makes the classification even more difficult than outdoor scene image classification. Therefore, this dataset is chosen as an important benchmark for the evaluation of our approach. The standard training/testing split for the MIT indoor scenes dataset consists of 80 training images and 20 testing images per category.

2.4.1.2 SUN 397

is the current largest scene classification dataset. It contains 397 scene categories, both indoor and outdoor, with at least 100 images per category. The 10 fixed splits for training and testing images are publicly available. For each category, there are 50 images for training and 50 images for testing. The accuracy is all averaged over all the 10 splits.

2.4.2 Multi-level Cascaded Fine-tuning

2.4.2.1 Baselines

We compare our cascaded fine-tuned CNN with two baselines: (a) off-the-shelf CNN features extracted by the pre-trained model on ImageNet (here we choose DeCAF$_6$ [41] as our off-the-shelf CNN feature for its better performance than DeCAF$_7$); (b) fine-tuned CNN initialized by the pre-trained model on ImageNet.

We conduct the comparison experiments on both the MIT indoor scenes dataset and the SUN 397 dataset. To be fair, we test all these 3 CNN features by simple max-pooling within their own level and training a linear SVM, without cross-level LLC coding and pooling. Please note that for level 1, since the whole image yields only one feature, there is no need to do pooling, and since no coarser level exists, there are no cascaded fine-tuned results. All the fine-tuned CNN features are obtained after 50 epochs of training. L2 normalization is performed on all the CNN features before used to train the SVMs. The SVM parameter (C) is all set as 0.5.

The results on MIT indoor scenes and SUN 397 are shown in Table 1 and Table 2 respectively for comparison. As can be seen, both on the MIT indoor scenes dataset and the SUN 397 dataset, fine-tuned CNN features, including those fine-tuned on ImageNet and cascaded fine-tuned ones on coarser levels of their own datasets, achieve higher accuracy than the off-the-shelf CNN features on all the levels. This is very natural since fine-tuned CNN features gain stronger discrimination power than generic off-the-shelf CNN features after the training on the specific datasets. The comparison between fine-tuned CNN on ImageNet and cascaded fine-tuned CNN shows that cascaded fine-tuned CNN features obtain higher accuracy than CNN fine-tuned on ImageNet by approximately 1% on all levels. This demonstrates that initialization by the trained model on the coarser level of a specific dataset helps the finer level model to converge to a better optimum than initialization by the model pre-trained on ImageNet.

Table 2.1: Classification accuracy on MIT indoor scenes for off-the-shelf CNN features, fine-tuned CNN features on ImageNet and cascaded fine-tuned CNN features of each level.

	level 1	level 2	level 3	level 4	level 5
off-the-shelf CNN	53.65	57.26	60.75	61.48	61.89
fine-tuned CNN on ImageNet	**61.46**	62.58	63.17	64.03	64.23
cascade fine-tuned CNN	—	**63.77**	**64.27**	**64.39**	**64.97**

Table 2.2: Classification accuracy on SUN 397 for off-the-shelf CNN features, fine-tuned CNN features on ImageNet and cascaded fine-tuned CNN features of each level.

	level 1	level 2	level 3	level 4	level 5
off-the-shelf CNN	40.53	41.25	41.68	42.07	42.64
fine-tuned CNN on ImageNet	**43.75**	44.88	45.17	45.54	45.81
cascade fine-tuned CNN	—	**45.61**	**46.33**	**46.58**	**46.87**

2.4.3 Cross-level LLC Coding and Pooling

2.4.3.1 Baselines

We compare our cross-level LLC and pooling approach (*cross-level LLC-CNN*) with multi-level pooled CNN features [14]. We choose multi-level max-pooling as the pooling method since we also perform max-pooling on our *cross-level LLC-CNN* features.

2.4.3.2 Classification Accuracy

We evaluate our cross-level LLC and pooling approach (*cross-level LLC-CNN*) on the MIT indoor scenes dataset and the SUN 397 dataset. The baseline method, multi-level pooled CNN is also tested for comparison with our *cross-level LLC-CNN*. The comparison results on each level and the combination of all levels are presented in Table 3 and Table 4. Here, cross-level LLC coding and pooling on a single level means that max-pooling is only performed within this level, while a cross-level codebook is still learned over all the levels. For the combination from level 1 to level 5, the final output of multi-level pooled CNN is obtained by concatenating the pooled result of each level together. Before all coding and pooling procedures, all the CNN features are extracted by the cascaded fine-tuned models. All the fine-tuned CNN features are obtained after 50 epochs of training. L2 normalization is performed on all the CNN features after extraction. The SVM parameter (C) is all set as 0.5. For reference, we also include some typical state-of-the-art results to compare with our approach.

In Table 3, from the comparison results between the baseline method and our approach, we can observe that on some finer levels, i.e., level 3, 4 and 5, our *cross-level LLC-CNN* works better than multi-level pooled CNN. The reason may be that more patches are available on these 3 levels (9 patches in level 3, 16 patches in level 4 and 25 patches in level 5), and pooling over more LLC features covers more information compared with original CNN features. Moreover, on the combination of all the 5 levels, our *cross-level LLC-CNN* also achieves higher accuracy than the baseline method, i.e., multi-level pooled CNN, which is 68.96% vs 67.87%, with a lower-dimensional feature. Compared to other state-of-the-arts, *cross-level LLC-CNN* also obtains the highest performance. It is worth mentioning that, to our best knowledge, the former best performance on this dataset is achieved by Multi-scale VLAD pooling on off-the-shelf CNN features, proposed by [14]. Compared to this MOP-CNN framework, our *cross-level LLC-CNN* obtains higher accuracy. Actually, the patches they used, i.e., 25 patches in level 2 and 49 patches in level 3, are much more than ours. The larger number of patches brings higher time cost in codebook learning and VLAD pooling. In contrast, the smaller number of patches utilized in our approach and the fast LLC performing make our *cross-level LLC-CNN* work much faster than their MOP-CNN. Table 4 shows the experimental

Table 2.3: Classification results on MIT indoor scenes for (a) baseline: multi-level pooled CNN; (b) *cross-level LLC-CNN*; (c) other state-of-the-arts.

methods		feature dimension	accuracy
(a) multi-level pooled CNN (baseline)	level1	4096	61.46
	level2	4096	63.77
	level3	4096	64.27
	level4	4096	64.39
	level5	4096	64.97
	level1+level2+$\cdots$+level5	20480	67.87
(b) ***cross-level LLC-CNN*** **(Ours)**	level1	16384	60.23
	level2	16384	62.47
	level3	16384	64.66
	level4	16384	65.48
	level5	16384	65.87
	level1+level2+$\cdots$+level5	16384	**68.96**
(c) state-of-the-arts	SPM [26]	5000	34.40
	FV+Bag of Parts [24]	221550	63.18
	Mode Seeking [42]	60000	64.03
	SPM+OPM [37]	—	63.48
	MOP-CNN [14]	12288	68.88

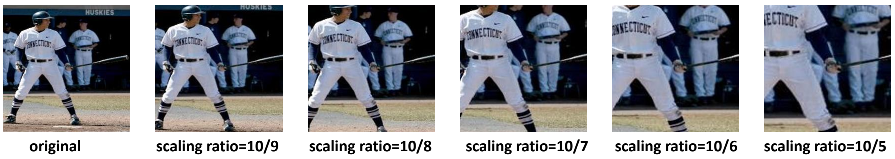
results on the SUN 397 dataset. Overall, the comparison results are similar with those on MIT indoor scenes dataset. On SUN 397, *cross-level LLC-CNN* outperforms multi-level pooled CNN on some finer levels (level 4 and level 5) and the combination of all the 5 levels. Compared to the state-of-the-arts, our approach achieves the best accuracy (50.87%) on the combination of all the 5 levels with a relatively low feature dimension.

2.4.3.3 Scale Invariance

To evaluate the scale invariance of our approach, we randomly select 670 testing images (half of the total) in MIT indoor scenes testing set and scale them by different scaling ratios, i.e., 10/9, 10/8, 10/7, 10/6, 10/5. For SUN 397, we use a random training/testing split (we choose the first split in the experiment) to evaluate the scale invariance. In this split, 1000 testing images randomly selected from the testing set are scaled by the same

Table 2.4: Classification results on SUN 397 dataset for (a) baseline: multi-level pooled CNN; (b) *cross-level LLC-CNN*; (c) other state-of-the-arts.

methods		feature dimension	accuracy
(a) multi-level pooled CNN (baseline)	level1	4096	43.75
	level2	4096	45.61
	level3	4096	46.33
	level4	4096	46.58
	level5	4096	46.87
	level1+level2+$\cdots$+level5	20480	49.23
(b) ***cross-level LLC-CNN*** **(Ours)**	level1	16384	40.48
	level2	16384	42.53
	level3	16384	45.89
	level4	16384	47.41
	level5	16384	48.53
	level1+level2+$\cdots$+level5	16384	**50.87**
(c) state-of-the-arts	Xiao et al.[31]	—	38.00
	Decaf [41]	4096	40.94
	Fisher Vector [43]	256000	47.20
	SPM+OPM [37]	—	45.91


original

scaling ratio=10/9

scaling ratio=10/8

scaling ratio=10/7

scaling ratio=10/6

scaling ratio=10/5

Figure 2.4: Illustration of the scaled testing image with different scaling ratios.

scaling ratios with those for MIT indoor scenes. Specifically, when scaling by a factor of ρ, we crop the image around the center with $1/\rho$ times of the original size, as illustrated in Figure 4. We compare the recognition accuracy over these scaled testing images of our *cross-level LLC-CNN* and the multi-level pooled CNN. Both methods are trained on non-scaled original training samples and the combination from level 1 to level 5 is adopted. Before all coding and pooling procedures, all the CNN features are extracted by the cascaded fine-tuned models. All the fine-tuned CNN features are obtained after 50 epochs of training. L2 normalization is performed on all the CNN features after extraction. The SVM parameter (C) is all set as 0.5.

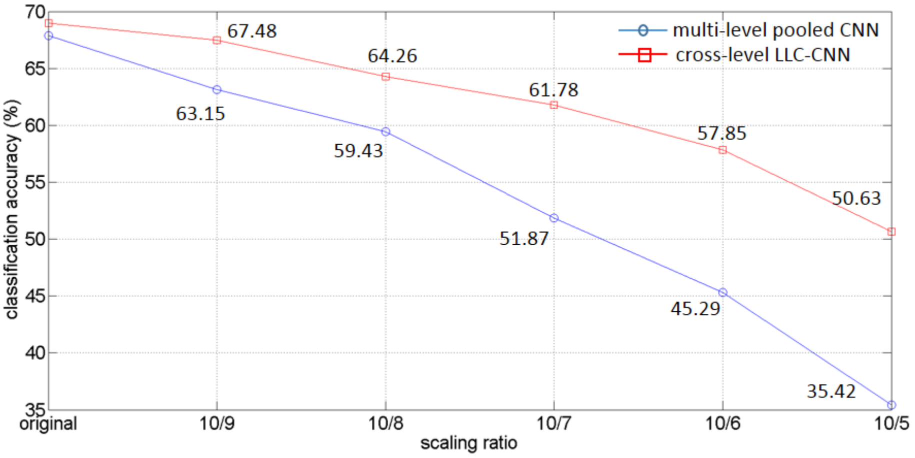


FIGURE 2.5: Classification accuracy comparison between multi-level pooled CNN features and our *cross-level LLC-CNN* for scaled images with different scaling ratios on the MIT indoor scenes dataset.

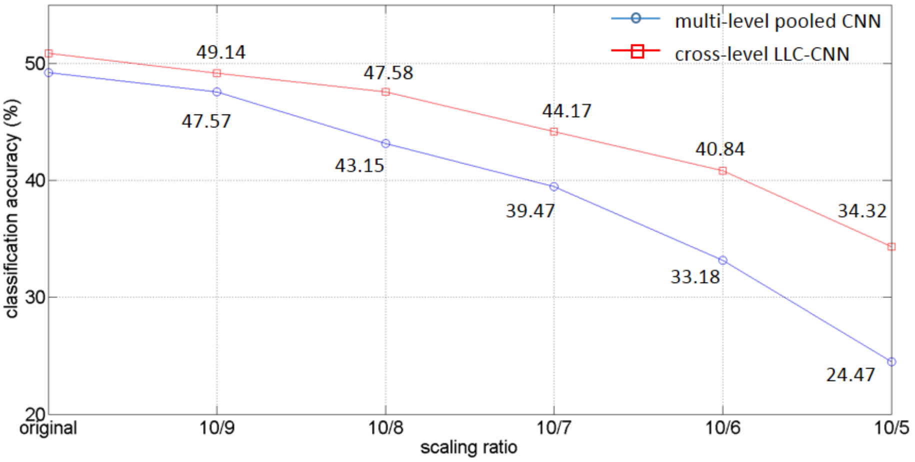


FIGURE 2.6: Classification accuracy comparison between multi-level pooled CNN features and our *cross-level LLC-CNN* for scaled images with different scaling ratios on the SUN 397 dataset.

The curves of recognition accuracy vs scaling ratio on the MIT indoor scenes dataset and the SUN 397 dataset are shown in Figure 5 and Figure 6, respectively. Both figures reflect the trend that the recognition accuracy decreases with the increase of the scaling ratio, whatever method is used. This shows that CNN features do not have scale invariance, as mentioned by lots of works [10, 14]. However, with our *cross-level LLC-CNN*, the classification accuracy decreases much more slowly than multi-level pooled CNN as the scaling ratio increases. As can be seen, from the original image to the 10/5 ratio scaled image, the difference in accuracy between our approach and the baseline approach is becoming increasingly big as the scaling ratio increases. Specifically, recognition accuracy with our approach when the testing image is scaled by 10/5 can still remain 50.63% and 34.32% for MIT indoor scenes and SUN 397 respectively. In comparison, the accuracy when the scaling ratio reaches 10/5 drops to 35.42% and 24.47% for MIT indoor scenes and SUN 397 respectively. The accuracy differences are all over 10%, showing the great superiority of our approach over the baseline approach. This superiority proves that LLC coding of CNN features on the cross-level codebook produces more robust features to the scale transformation, as LLC coding ensures that scaled CNN features can still be well represented by the cross-level codebook and their discrimination power is retained after scaling.

2.5 Summary

In this chapter, we proposed a cross-level Locality-constrained Linear Coding and pooling framework (*cross-level LLC-CNN*) on multi-level CNN features to enhance the discrimination and scale invariance of the image representation for scene classification problems. Based on the cascaded fine-tuning scheme, the CNN features gain stronger discrimination in scene classification. In addition, with cross-level Locality-constrained Linear Coding and pooling on these multi-level fine-tuned CNN features, robustness to scale transformation is improved. We evaluated our approach on the MIT indoor scenes dataset and the SUN 397 dataset. Experimental results demonstrated that significant improvements in classification accuracy are achieved for both original and scaled testing images. The improved recognition accuracy and scale invariance suggests the promising application potential of the proposed method in scene recognition for running vehicles.

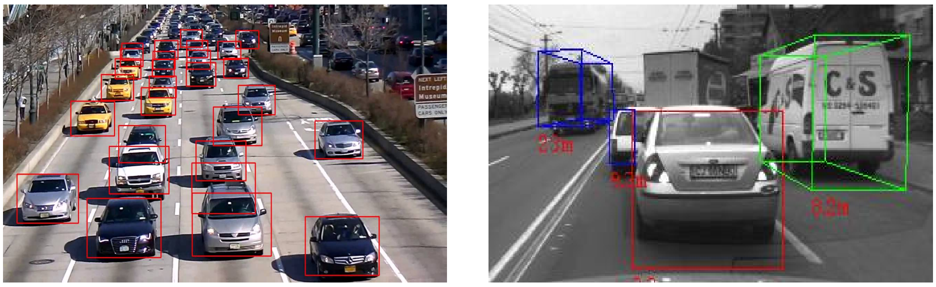
Chapter 3

Fully Convolutional Networks Based Object Detection for Vehicle and Pedestrian Detection

3.1 Overview

3.1.1 Vehicle detection

Vehicle accident statistics disclose that the main threats a driver is facing are from other vehicles. Consequently, developing on-board automotive driver assistance systems aiming to alert a driver about driving environments and possible collision with other vehicles has attracted a lot of attention. In these systems, robust and reliable vehicle detection is the first step. Vehicle detectionand tracking has many applications like



(A) (B)

Figure 3.1: Illustration of vehicle detection.

platooning (i.e., vehicles traveling in high speed and close distance in highways), stop and go (vehicles traveling in low speeds and close distance in cities).

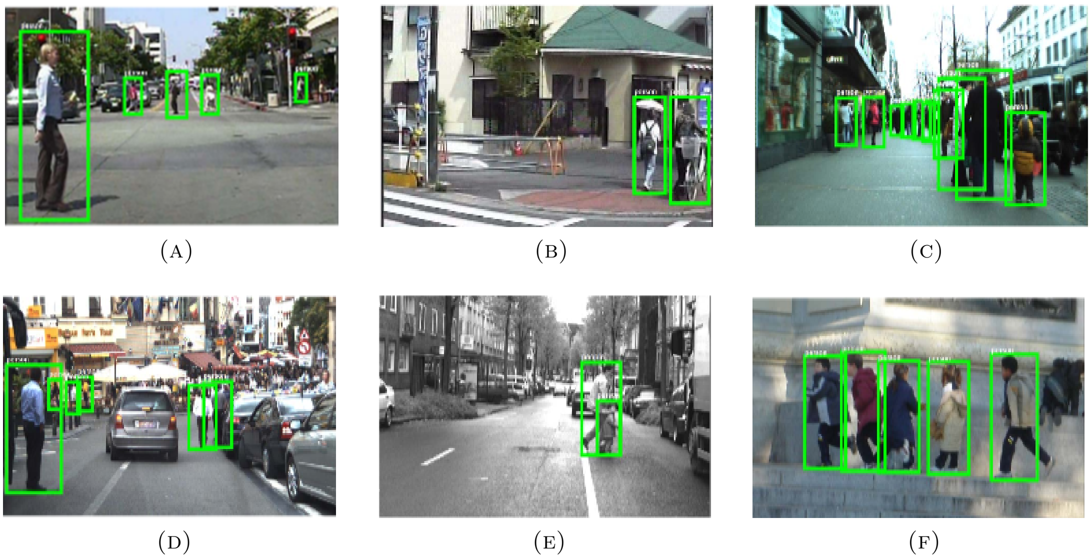
For vehicle detection, the goal is to detect whether there are cars (or other vehicles) ahead and localize the positions of the possible vehicles in the captured images (see Figure 3.1) so that their real positions from the view of ego-vehicles. The main difficulties of car detection comes from the following aspects. (i) The visual appearance of vehicles in images are in wide range of variations. (ii) The vehicles are also not easy to differentiate from non-vehicles objects in the images (e.g. buildings and other surrounding objects), as illustrated in Figure 3.1(B). As known, false detection that treating non-vehicles as vehicles will lower the driverless cars speed greatly because too many surrounding objects are seen as on-road vehicles. Missed detection that missing detecting a true vehicle will bring great dangers since front running vehicles are ignored. (iii) Under different viewing conditions, such as change in lighting, viewpoint, etc., the vehicles will also show different visual appearance.

3.1.2 Pedestrian detection

People are among the most important components of a driverless cars environment. Detecting and tracking pedestrians is thus an important area of research, and computer vision is bound to play a key role. Just in the US, nearly 5,000 of the 35,000 annual traffic crash fatalities involve pedestrians [44], hence the considerable interest in driverless car automated vision systems for detecting pedestrians [45].

For pedestrian detection, the goal is to detect whether there are pedestrian in front of the cars, their accurate positions in the image and their distance to the driverless car if yes (see Figure 3.2). This is to slow down the driverless car if there are pedestrian and adjust its movement direction according to the different pedestrian positions and distances.

Pedestrian detection as a hot computer vision research topic, draws much attention of researchers recently but still cannot be solved well dues to the following challenges. (1) Pedestrians are amorphous bodies without fixed patterns, and may keep static or move in any unpredictable direction (see Figure 3.2(F)); (2) Pedestrians are often mixed with the complex scene background (see Figure 3.2(E)); (3) Pedestrians maybe in great crowd in real-life (see Figure 3.2(C)); (4) Pedestrians may be small in the images resulting from the far distance between pedestrians and the driverless cars (see Figure 3.2(A)(C)(D)); (5) The illumination condition varies greatly dues to the various weather and time. (6) Pedestrians are often partially occluded by the cars, their bags, etc. (see Figure 3.2(B)).



(A) (B) (C)

(D) (E) (F)

Figure 3.2: Illustration of pedestrian detection.

3.1.3 Summary

In both vehicle and pedestrian detection, conventional methods primarily rely on hand-crafted image features like SIFT, HOG. Such hand-crafted local features require much efforts to design and may not generalize well in some complex detection problems. Another problem brought by the hand-crafted features is the feature computation time cost. Generally, due to the limitation that one single hand-crafted feature descriptor can only characterize the image in a certain aspect well, e.g., texture, shape or color, a combination of multiple hand-crafted feature descriptors is usually utilized to improve the recognition accuracy. Such combinations will lead to high time cost in feature computation (usually several seconds per image), thus is not feasible for real-time on-road vehicle detection.

Recently, deep learning based methods like R-CNN also achieve great success in generic object detection. However, R-CNN is still slow in implementation as it requires to classify all the object proposals (usually more than 1,000) independently using a very deep convolutional neural network. Additionally, R-CNN involves generating class-agnostic object proposals which are not necessary in vehicle and pedestrian detection task. It is because in vehicle and pedestrian detection problem, except for vehicles and pedestrian, other objects are not needed to be detected so that class-agnostic proposal generation brings too many obvious false positives to be classified by CNN. This significantly affects the running speed of R-CNN in vehicle and pedestrian detection. Another weakness of class-agnostic object proposal in vehicle and pedestrian detection is that using objects of all categories as positive training samples results in inferior detection performance than using only vehicles and pedestrian as positives for object proposal generators.

In this chapter, a data-driven learning framework based on fully convolutional networks (FCN) is proposed to directly process the original raw data from the images captured by cameras and produce a high-level semantic confidence score which shows to what extent a specific region may contain a vehicle. Briefly, we train a vehicle(pedestrian)/non-vehicle(pedestrian) binary classifier using a fully convolutional network (FCN) [46] on patches from images with annotated vehicles and pedestrians. The fully convolutional network can take an input image of arbitrary size and output a dense "confidence map" showing the probability of containing a vehicle and pedestrian for each corresponding box region in the original image. To predict the confidences for boxes of different scales, we rescale the original image into multiscales and feed them into the network to obtain the confidence maps of different scales correspondingly. Then, non-maximal suppression (NMS) is performed to remove redundant boxes. Finally, we train a support vector machine (SVM) on the image gradients to refine the boxes by finding the box with the highest confidence score among the neighboring boxes of each rough box obtained by FCN.

3.2 Related Works

3.2.1 Low-level feature based vehicle detection

In this section, we review the representative low-level feature based methods in vehicle detection. We first review the typical features that are used in vehicle detection and then review the part based detection models which are employed by the majority of works in vehicle detection.

3.2.1.1 Low-level feature

Representative features use coded descriptions to characterize the visual appearance of the vehicles. A variety of features have been used in vehicle detection such as local symmetry edge operators [47]. It is sensitive to size and illumination variations, thus a more spatial invariance edge based histogram was used in [48]. In recent years, these simple features evolves into more general and robust features that allow direct detection and classification of vehicles. Scale Invariant Feature Transformation (SIFT) [27], speeded up Robust Features (SURF) [49], Histogram of Oriented Gradient (HOG) [28] and Haar-like features [50] are extensively used in vehicle detection literature.

Scale Invariant Feature Transformation (SIFT) was first introduced in 1999 [27]. Features are detected through a staged filtering approach, which identifies local edge orientation around stable keypoints in scale space. A modified SIFT descriptor was used in [48], by introducing a rich representation for vehicle classes. In [51], SIFT interest points were re-identified as the initial particles to improve tracking performance. SIFT-based template matching technique was used in [52], to locate special marks in the license plate. SIFT and Implicit Shape Model (ISM) were combined in [53] to detect a set of keypoints and generate feature descriptors. In [54] an SIFT based mean shift algorithm was proposed. To compress the length of SIFT, Principal Component Analysis (PCA)SIFT was introduced in [55], through combining local features with global edge features using an adaptive boost classifier. However, it was slow and less distinctive [49]. Based on an enhanced SIFT feature-matching technique vehicle logo recognition algorithm was proposed in [56]. The SIFT matching algorithm was combined with SVM in [57], for multi-vehicle recognition and tracking. It perform tracking well in complex situations. Still, it consumes a lot of time, which restricts practical applications. The proposed method in [58], combined the advantages of SIFT and CAMSHIFT to track vehicle. Due to its distinctive representation, SIFT has wide applications. However, the high dimensionality and the use of Gaussian derivatives to extract feature points are timeconsuming and do not satisfy the real-time requirement [59]. Its low adaption to illumination variation is another drawback.

Speeded Up Robust Features (SURF) is a scale and rotation invariant interest point detector and descriptor that was introduced in [49]. Compared to SIFT its computational complexity was reduced by replacing Gaussian filter with a box of filters, which slightly affects the performance. SURF algorithm uses a Hessian matrix approximation on an integral image to locate the points of interest. The second-order partial derivatives of an image describe its local curvatures [60]. In [59], symmetrical SURF descriptor was proposed for vehicle detection with make and model recognition. Recently, symmetrical SURF was used in [60] for vehicle color recognition and in [61] to detect the central line of the vehicles. The proposed technique can process one vehicle per frame with 21 fps. A GPU based multiple camera system in [62] used Gabor filter as a directional filter with SURF matching for unique representation of vehicles. An on road vehicle detection in [63], uses cascade classifier and Gentle AdaBoost classifier with Haar-SURF mixed features. Because of its repeatability, distinctiveness, robustness and real-time capability, it has become one of the most commonly used features in computer vision [59]. Nevertheless, it is not stable under rotation and illumination variations.

The grid of Histogram of Oriented Gradient (HOG) [28] compute the image gradient directional histogram, which is an integrated presentation of gradient and edge information. It was originally proposed to detect pedestrian. HOG symmetry feature vectors was proposed in [64] and used together with the original HOG in hypothesis verification. A combination of a latent support vector machine (LSVM) and HOG was used in [65] to combines both local and global features of the vehicle as a deformable object model. HOG was combined with Disparity Maps in [66] to detect Airborne Vehicle in Dense Urban Areas. In [67] a relative discriminative extension to HOG (RDHOG) was proposed to enhance the descriptive ability. Illumination and geometric invariance together with the high computational efficiency are the main advantages of this feature, which outperform sparse representation in SIFT [68].

3.2.1.2 Part based detection models

ii. Part based detection models In this technique the vehicle is divided into a number of parts modeled by the spatial relation between them [47]. They consider the vehicle to be separated into front, side and rear parts which contains window, roof, wheels, and other parts [69]. The distinct parts are detected based on their appearance, edge and shape feature [70]. After that spatial relationship, motion cue and multiple models are used to identify vehicles.

In [71] part labelling was defined to cover the object densely. To ensure consistent layout of parts while allowing deformation they used Layout Consistent Random Field model. The method was expanded to 3-D models in [72] to learn physically localized part appearances. Also they combine object-level descriptions with pixel-level appearance, boundary, and occlusion reasoning. Deformable part based modelling was employed in [65] through the combination of a latent support vector machine (LSVM) and histograms of oriented gradients (HOGs). The algorithm combines vehicle global and local features as a deformable model composed of root filter and five parts filters to detect front, back, side, and front, back truncated.

Deformable part based model was used in [65], it consists of a global root filter, six part filters and a spatial model to detect and track vehicles on road using part-based transfer learning (PBTL). Vehicle detection by independent parts (VDIP) was introduced in [73] for urban driver assistance. Front, side, and rear parts were trained indecently using active learning. Part matching classification using a semisupervised approach form vehicles sideview from independently detected parts.A rear view vehicle detection was considered in [69] based on multiple salient parts that includes license plate and rear lamps. For part localization distinctive color, texture and region features were used. Then Markov random field model was used to construct probabilistic graph of the detected parts. Vehicle detection was accomplished by inferring the marginal posterior of each part using loopy belief propagation.

3.2.2 Low-level feature based pedestrian detection

In this section, we review the representative previous works of using low-level feature based methods to solve pedestrian detection. Specifically, we first review the typical low-level features that are used in pedestrian detection and then review the learning models of pedestrian detection.

3.2.2.1 Low-level features

Gradient-based features brought great progress to the pedestrian detection problem. Inspired by SIFT [27], Dalal and Triggs [28] popularized histogram of oriented gradient (HOG) features for detection by showing substantial gains over intensity based features. Zhu et al. [74] sped up HOG features by using integral histograms [75]. In earlier work, Shashua et al. [76] proposed a similar representation for characterizing spatially localized parts for modeling pedestrians. Since their introduction, the number of variants of HOG features has proliferated greatly with nearly all modern detectors utilizing them in some form.

Shape features are also a frequent cue for detection. Gavrila and Philomin [77, 78] employed the Hausdorff distance transform and a template hierarchy to rapidly match image edges to a set of shape templates. Wu and Nevatia [79] utilized a large pool of short line and curve segments, called edgelet features, to represent shape locally. Boosting was used to learn head, torso, leg and full body detectors; this approach was extended in [80] to handle multiple viewpoints. Similarly, shapelets [81] are shape descriptors discriminatively learned from gradients in local patches; boosting was used to combine multiple shapelets into an overall detector. Liu et al. [82] proposed granularity-tunable features that allow for representations with levels of uncertainty ranging from edgelet to HOG type features; an extension to the spatio-temporal domain was developed in [83].

While no single feature has been shown to outperform HOG, additional features can provide complementary information. Wojek and Schiele [84] showed how a combination of Haar-like features, shapelets [81], shape context [85] and HOG features outperforms any individual feature. Walk et al. [86] extended this framework by additionally combining local color self-similarity and the motion features discussed above. Likewise, Wu and Nevatia [87] automatically combined HOG, edgelet and covariance features. Wang et al. [88] combined a texture descriptor based on local binary patterns (LBP) [89] with HOG, additionally, a linear SVM classifier was modified to perform basic occlusion reasoning. In addition to HOG and LBP, [90] used local ternary patterns (variants of LBP). Color information and implicit segmentation were added in [91], with a performance improvement over pure HOG.

3.2.2.2 Learning models

Considerable effort has also been devoted to improving the learning framework. Tuzel et al. [92] utilized covariance matrices computed locally over various features as object descriptors. Since covariance matrices do not lie on a vector space, the boosting framework was modified to work on Riemannian manifolds, with improved performance. Maji et al. [93] proposed an approximation to the histogram intersection kernel for use with SVMs, allowing for substantial speed-ups and thus enabling a non-linear SVM to be used in sliding-window detection. Babenko et al. [94] proposed an approach for simultaneously separating data into coherent groups and training separate classifiers for each; [95] showed that both [93] and [94] gave modest gains over linear SVMs and AdaBoost for pedestrian detection, especially when used in combination [96].

A number of groups have attempted to efficiently utilize very large feature spaces. Feature mining was proposed by [97] to explore vast (possibly infinite) feature spaces using various strategies including steepest descent search prior to training a boosted classifier. These ideas were developed further by [98], who introduced a scheme for synthesizing and combining a rich family of part based features in an SVM framework. Schwartz et al. [99] represented pedestrians by edges, texture and color and applied partial least squares to project the features down to a lower dimensional space prior to SVM training.

To cope with articulation, the notion of parts and pose have been investigated by several authors. Notable early approaches for unsupervised part learning, including the constellation model [100, 101] and the sparse representation approach of [102], relied on keypoints. Leibe et al. [103] adapted the implicit shape model, also based on keypoints, for detecting pedestrians. However, as few interest points are detected at lower resolutions, unsupervised part based approaches that do not rely on keypoints have been proposed. Multiple instance learning (MIL) has been employed in order to automatically determine the position of parts without part-level supervision [104, 105]. And, in one of the most successful approaches for general object detection to date, Felzenszwalb et al. [106] proposed a discriminative part based approach that models unknown part positions as latent variables in an SVM framework. As part models seem to be most successful at higher resolutions, Park et al. [20] extended this to a multi-resolution model that automatically switches to parts only at sufficiently high resolutions.

3.2.3 Deep learning for object detection

Hand-crafted features usually involve lots of tricks to design thus are difficult to ensure good performance in various complex vision tasks. Recently, learned features by deep learning methods have shown great potentials in computer vision tasks [6, 12, 13]. Deep learning tries to model high-level abstractions of visual data by using architectures containing multiple layers of non-linear transformations.

Specifically, deep Convolutional Neural Network (CNN) has shown outstanding performance in large-scale image classification datasets, such as ImageNet [9] and CIFAR-10 [38]. Later lots of works [29, 30, 107] consider to transferring CNN features pre-trained on ImageNet to small-scale computer vision tasks in which only limited amount of task-specific training samples are available. Off-line CNN features extracted by the model pre-trained on ImageNet are successfully applied to various vision tasks, including object detection [30], image retrieval [14]. To further improve the adaptation and representation power of CNN features in the specific tasks, the fine-tuned CNN features based on the pre-trained ImageNet CNN features are also used and achieved better performance in these specific tasks [29, 107].

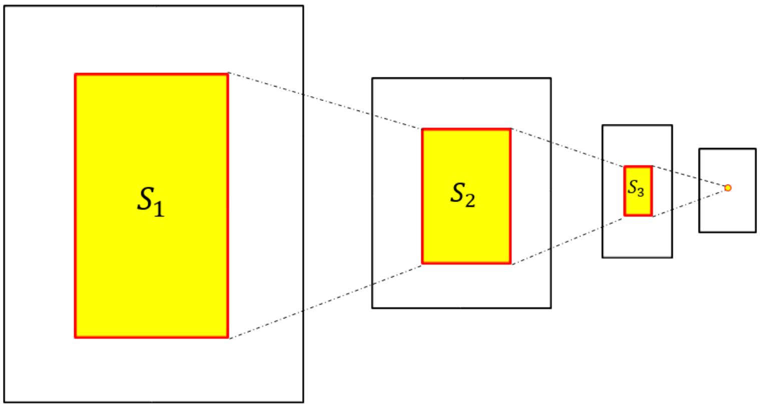


FIGURE 3.3: Illustration of fully convolutional network. The yellow pixel in the output map shows the classification confidence of the yellow window region S_1 in the input image and will not be affected by other regions in the input image.

3.3 Multiscale Fully Convolutional Networks

3.3.1 Fully Convolutional Networks

Convolutional Neural Network (CNN) can be regarded as an automatic hierarchical feature extractor. Such a learning-based deep feature extraction pipeline avoids hand-crafted feature designing which may not be suitable for a specific task. Recently, as an extension of the classic CNN for classification problems [6, 12, 13], fully convolutional networks can take an input of arbitrary size and output a map whose size corresponds to the input, which can be used for dense prediction problems (e.g., semantic segmentation [46, 108] and image restoration [109]).

A whole input image is fed into the fully convolutional network to obtain a pixel-wise vehicle confidence map. This feed-forward process can be seen as vehicle/non-vehicle binary classification for the densely sampled sliding windows in the input image. Each output pixel in the confidence map shows the classification confidence of one specific sliding window in the input image, as illustrated in Figure 3.3. To map back to the

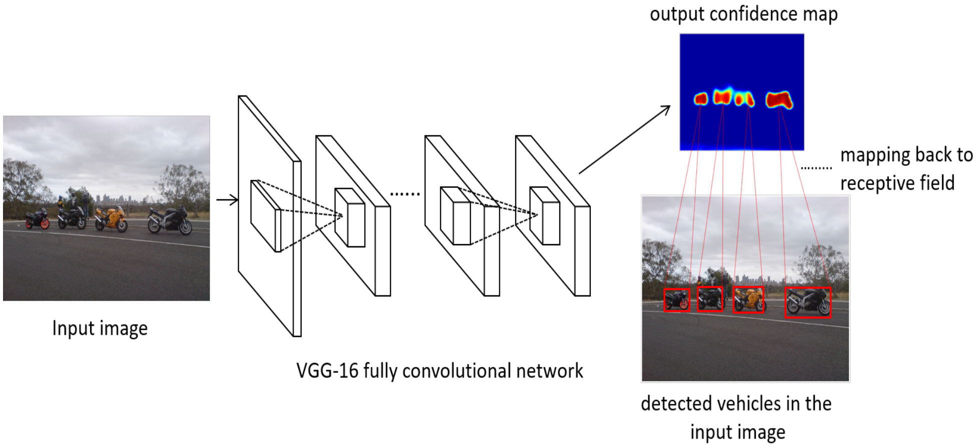

output confidence map
mapping back to receptive field
Input image
VGG-16 fully convolutional network
detected vehicles in the input image

FIGURE 3.4: The illustration of detection pipeline using FCN. An input image is fed into a fully convolutional network to generate the output vehicle confidence map. Then based on the receptive field computation, the vehicle bounding boxes can be obtained by mapping the pixels with high vehicle confidences back to the corresponding bounding boxes (receptive fields) in the input image.

input vehicle detection bounding boxes from the output vehicle confidence map, there is a need to decide how big of an area the output pixel can correspond to in the input image (receptive field size). Assuming the receptive field size of each layer is S_i (i=1, 2, ..., n) and S_1 is the receptive field size in the input image, the receptive field size of each layer can be computed using the recursive formula below:

$$S_{i-1} = up(S_i) = s_i(S_i - 1) + k_i \tag{3.1}$$

where s_i and k_i represent the stride and the convolution kernel size of the i^{th} convolutional or pooling layer. S_{i-1} and S_i denote the receptive field size of the $(i-1)^{th}$ and the i^{th} layer respectively. To accurately map an output pixel back to the window region it covers in the input image, sliding window sampling stride Str is also indispensable. Fully convolutional network has its inherent sampling stride Str, which is the product of the strides of all the layers, i.e.,

$$Str = \prod_{i=1}^{n} s_i \tag{3.2}$$

where s_i indicates the stride of the i^{th} convolution or pooling layer. Given S_1 and *Str*, the window region which corresponds to the output pixel (x_o, y_o) can be decided below:

$$\begin{aligned} x_{min} &= x_o Str \\ x_{max} &= x_o Str + S_1 \\ y_{min} &= y_o Str \\ y_{max} &= y_o Str + S_1. \end{aligned} \tag{3.3}$$

In contrast to other sliding window approaches that compute the entire pipeline for each window, fully convolutional networks are inherently efficient since they naturally share computation common to different overlapping regions. When applying a fully convolutional network to the input of an arbitrary large size in testing, convolutions are applied in a bottom-up manner so that the computation common to neighboring windows only needs to be done once. Therefore, we now are able to map the pixels with high vehicle confidences in the output map back to the corresponding bounding boxes in the input image which are highly possible to contain a vehicle. The whole pipeline of bounding box inference with FCN is shown in Figure 3.4.

3.3.2 Network Training

As VGG-16 layer network [12] shows outstanding discrimination power in the ILSVRC classification task, we adopt the publicly released VGG-16 layer model as our pre-trained model for further fine-tuning on the vehicle detection datasets. The two fully-connected layers in VGG-16 network are replaced by two convolution layers with 1×1 convolution kernels to meet the requirements of fully convolutional networks. The last fully-connected layer in VGG-16 has 1,000 neurons as there are 1,000 categories in ILSVRC classification task. Therefore, we replace the last layer of VGG-16 with a 1×1 convolution layer with 2 output neurons as we only have 2 categories (vehicle and non-vehicle). The loss function to be optimized during fine-tuning is the cross-entropy loss, i.e.,

$$E = t_k \ln(y_k) + (1 - t_k) \ln(1 - y_k) \tag{3.4}$$

where t_k denotes the k^{th} target value and y_k represents the k^{th} prediction value.

In terms of fine-tuning, we treat the network as an vehicle/non-vehicle binary classification network and use a patch-wise training strategy instead of using the whole images to train a dense structured prediction network. To this end, we crop the patches from the images with annotated objects and resize them to 256×256, the same as the input size of VGG-16 model. Among the cropped patches, those having intersection over union (IoU) ≥ 0.5 with a ground-truth box are treated as positive samples and the rest as negatives. To balance the number between the positives and the negatives, we crop multiple patches around each ground-truth box while sparsely sampling the patches in the background regions.

For the stochastic gradient descent (SGD) training process, the weights of all the pre-trained layers are initialized with the weights of the publicly released VGG-16 model. The last layer is initialized with a zero-mean Gaussian distribution with the standard deviation 0.01. The learning rate is set as 0.01 and 0.001 for the last layer and all the pre-trained layers, respectively. The main reasons for this setting are as follows: the first few convolution layers mainly extract low-level invariant features, such as texture, shape, thus the weights are more consistent from the pre-trained dataset to target dataset, whose learning rate are set as a relative low value (0.001); while for the last fully-connected layer, which is specifically adapted to the new target dataset, a higher learning rate is required to guarantee its fast convergence to the new optimum. The learning rates of all the layers are reduced by a scale of 10 after every 20 epochs.

3.3.3 Multiscale Inputs Inference

Using the aforementioned fully convolutional network, each pixel in the output map only covers a window region with a fixed size S_1. To enable the network to predict the vehicle bounding boxes with different sizes and aspect-ratios, we rescale the original image to different scales. By doing this, a window with the size equaling to the receptive field size S_1 in the rescaled inputs will correspond to windows of different scales and aspect-ratios in the original image.

Subsequently, the multiscale inputs after rescaling are fed into the network individually to obtain the multiscale vehicle confidence maps. Here we present the multiscale setting in detail to specify the scales needed in our approach. Generally, the more and the denser the scales are, the more a concentrated set of bounding boxes near the areas is likely to contain a vehicle. However, the downside is that noisy bounding boxes which may lower the precision will be produced as well. This issue introduces a trade-off in parameter selection for the multiscale setting.

Specifically, we define α as the stepsize indicating the IoU for neighboring boxes. In other words, the step sizes in scale and aspect ratio are determined such that one step results in neighboring boxes having an IoU of α. The scale values range from a minimum box area of 1,000 pixels to the full image. The aspect ratio changes from 1/3 to 3. The exact values of scales and aspect ratio can be computed with Eqn. (5) and Eqn. (6).

$$\text{scale} = \sqrt{1000}(\sqrt{1/\alpha})^{\text{s}}, \tag{3.5}$$

$$\text{aspect ratio} = (\frac{1+\alpha}{2\alpha})^{\text{r}}. \tag{3.6}$$

Here the index s can be any integer from 0 to $\lfloor \log(\text{image size}/\sqrt{1000})/\log(\sqrt{1/\alpha}) \rfloor$, and the index r can be any integer from $-\lfloor \log(3)/\log(\frac{1+\alpha}{2\alpha})^2 \rfloor$ to $\lfloor \log(3)/\log(\frac{1+\alpha}{2\alpha})^2 \rfloor$. For the multiscale detection bounding boxes, we first remove those with vehicle confidences lower than 0.2, reducing the total number of detection bounding boxes from several tens of thousand to less than 10,000. Next, we sort all the remained bounding boxes based on their vehicle confidences in a descending order. Finally, non-maximal suppression (NMS) is performed on the sorted bounding boxes. Specifically, we find the bounding boxes with the maximum vehicle confidences and remove all the bounding boxes with an IoU larger than an overlap threshold (we use 0.8 in all our experiments).

3.4 Bounding Box Refinement

Due to the fixed multiscale setting and box sampling strategy, the above obtained raw bounding boxes have their inherent weakness of being pre-defined both in scales and locations which may cause misdetection of ground-truth boxes. To overcome this, we adopt a greedy iterative search method to refine each raw bounding box.

Previous works show that objects of interest are stand-alone things with well-defined closed boundaries [110–112]. Based on this observation, gradient and edge information are widely used for implying the presence of objects in early works, e.g., BING [113] and Edge Boxes [114]. Considering this, we also rely on the gradient cues instead of 3-channel color information for the efficient implementation of our method. Specifically, we train a linear SVM vehicle/non-vehicle classifier on the gradient maps of the patches from the images with annotated vehicles. We use the ground-truth boxes of the annotated vehicles as positive samples, and crop the patches in the images and treat those with IoU < 0.3 for all the ground-truth boxes as negative samples. For all the chosen samples, we resize them to 16×16 before training the SVM. After training the 256-d SVM, to refine the bounding boxes, we maximize the SVM score of each box over the neighboring positions, scales and aspect ratios. After each iteration, the search step is reduced in half. The search is stopped once the translational step size is less than 2 pixels. The procedure is summarized in pseudo-code in Algorithm 1.

Algorithm 1 Refine the bounding boxes [B_1, B_2, ..., B_n]

```
Require: : A set of raw bounding boxes [$B_1$, $B_2$, ..., $B_n$]
  for $B = [B_1, ...B_n]$ do
    $Confidence \Leftarrow SVM(B)$
    $S_c \Leftarrow 0.2B_w$ ($B_w$ is the bounding box width)
    $S_r \Leftarrow 0.2B_h$ ($B_h$ is the bounding box height)
    while $S_c > 2$ and $S_r > 2$ do
      $[B_{c1}, B_{c2}, ..., B_{cn}] \Leftarrow ColomnNeighbors(B, S_c)$
      $B_{cmax} \Leftarrow argmax(SVM(B_{ci})),\ i = [1, 2, ..., n]$
      if $svm(B_{cmax}) > Confidence$ then
        $Confidence \Leftarrow SVM(B_{cmax})$
        $B \Leftarrow B_{cmax}$
      end if
      $[B_{r1}, B_{r2}, ..., B_{rn}] \Leftarrow RowNeighbors(B, S_r)$
      $B_{rmax} \Leftarrow argmax(SVM(B_{ri})),\ i = [1, 2, ..., n]$
      if $SVM(B_{rmax}) > Confidence$ then
        $Confidence \Leftarrow SVM(B_{rmax})$
        $B \Leftarrow B_{rmax}$
      end if
      $S_c \Leftarrow S_c/2$
      $S_r \Leftarrow S_r/2$
    end while
  end for
```

3.5 Experiments

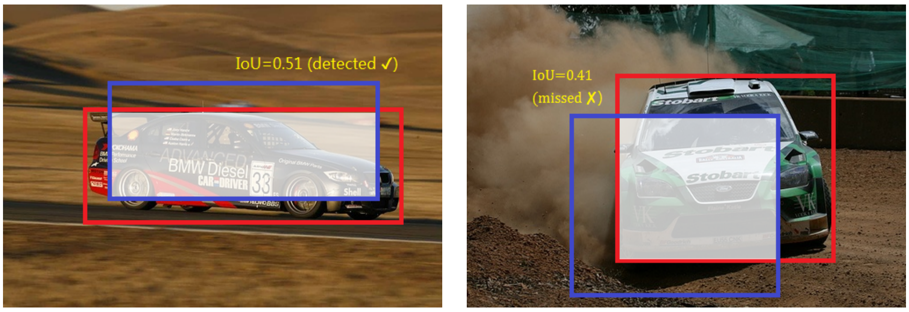
3.5.1 Experimental Datasets

To verify the effectiveness of the proposed approach for on-road vehicle and pedestrian detection, extensive experiments are conducted on two benchmark datasets, i.e., PASCAL VOC 2007 dataset [115], LISA-Q Front FOV dataset [116] and KITTI dataset.

PASCAL VOC 2007 dataset is a popular standard benchmark which contains 20 pre-defined categories for general object classification, detection and segmentation tasks in computer vision area. We manually select the images containing vehicles (including car, bus, motorbike and bicycle four categories) for both training and testing in the experiments. In this way, we obtain 1,292 images for training the model and 1,309 images for testing. For the four categories of vehicles in PASCAL VOC, various view-angles of the vehicles may be included as the images are obtained by the ways not only restricted in the front-view cameras in the cars. Thus the visual appearance may be even more diverse and brings higher difficulty for vehicle detection.

In LISA dataset, on-road data are captured daily by LISA-Q testbed [117], which has synchronized the capture of vehicle controller area network data, Global Positioning System, and video from six cameras [118]. The videos from the front-facing camera comprises the LISA-Q Front FOV data sets. It consists of three video sequences, consisting of 1,600, 300, and 300 consecutive frames, respectively. We randomly sample 70% frames from each of the three video sequences as training images and use the rest for testing.

The challenging KITTI dataset [119] consists of 7,481 training and 7,518 test images, which are captured from an autonomous driving platform. Evaluation is done at three levels of difficulty: easy, moderate and hard, where the difficulty is measured by the minimal scale of the pedestrians to be considered and the occlusion and truncation of the pedestrians. For the moderate setting which is used to rank the competing methods in the benchmark, the pedestrians over 25 pixels tall with no or low partial occlusion and truncation are considered. Since the annotations of the testing set are not available, we split the KITTI training set into train and validation subsets as suggested by [120]. The images are resized as 800 pixels on the shortest side during the training and testing time.



(A) detected vehicle (IoU>0.5) (B) missed vehicle (IoU<0.5)

FIGURE 3.5: Examples of detected ground-truth vehicle (left) and missed ground-truth vehicle (right). In both images, red bounding boxes are the ground-truth annotation bounding boxes, and blue bounding boxes are the detections. White regions are the intersections between the ground-truth vehicles and the detections.

3.5.2 Performance Metrics

We consider recall, precision and efficiency as the performance metrics when evaluating our detection method. In specific, recall is defined as the proportion of the truly detected vehicles in all the annotated ground-truth vehicles, see Eqn. (7); precision is defined as the proportion of detection that are true vehicles, see Eqn. (8). In all our experiments, a ground-truth vehicle is regarded as detected only when there is a detection bounding box has an intersection over union (IoU) larger than 0.5 with this ground-truth vehicle bounding box, see Figure 3.5. This criterion is widely used in the evaluations of several computer vision detection benchmarks, e.g., ImageNet Large Scale Visual Challenge Competition (ILSVRC) [121], PASCAL VOC challenge and Microsoft COCO [122]. For efficiency evaluation, we report the running speed using per image using our method.

$$\text{Recall} = \frac{\#\ \text{detected vehicles}}{\#\ \text{all ground-truth vehicles}} \tag{3.7}$$

$$\text{Precision} = \frac{\#\ \text{detected vehicles}}{\#\ \text{detected vehicles} + \#\ \text{false positives}} \tag{3.8}$$

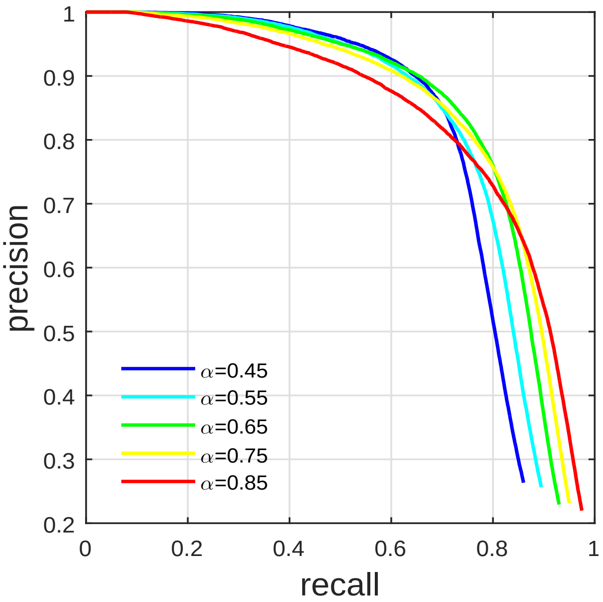


Figure 3.6: Precision-recall curves of our FCN method with different multiscale stepsize α on PASCAL VOC 2007 testing set.

3.5.3 Experimental Results

3.5.3.1 Variant Analysis

We begin the experiments by analyzing the effects of the granularity of the multi-scale search on the PASCAL VOC 2007 testing set. Specifically, we vary the stepsize parameter α and plot precision-recall curves for different α. From Figure 3.6, it is found that when α is between 0.45 to 0.65, as α increases, both recall and precision can be improved generally. This is natural as more scales provide more chances to have a detection close to the groundtruth bounding box. However, when α exceeds 0.65, as α increases, recall is enhanced while precision drops. The reason probably lies in that too many detections concentrated on a small area are introduced , resulting in a higher possibility of false positives and loss of the precision. From Figure 3.6, α should be set as 0.65 or 0.75.

Table 3.1: Runing speed of FCN with different α on PASCAL VOC 2007 testing set.

α	Runing time per image
0.45	0.55s
0.55	0.67s
0.65	0.87s
0.75	1.93s
0.85	4.52s

Table 3.2: Runing speed of FCN, R-CNN and DPM on PASCAL VOC 2007 testing set.

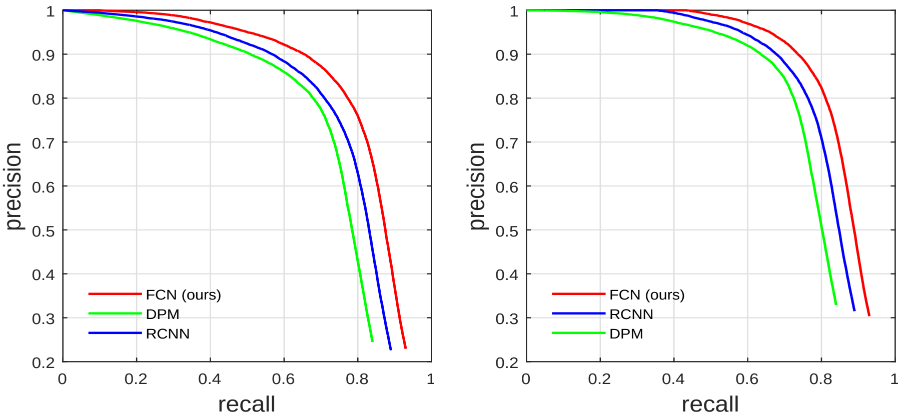
Method	Runing time per image
FCN (ours)	0.87s
R-CNN	9.03s
DPM	12.23s

We also conduct the running time comparison experiment for each search stepsize α on the PASCAL VOC 2007 testing set. Table 1 presents the detailed running time for various values of α. As can be seen, the running time grows exponentially with the increasing of α. Since the balance between recall and precision is similar good when α equals to 0.65 and 0.75, we thus fix α as 0.65 in all the later experiments considering that α as 0.75 has a quite high time cost.

3.5.3.2 Comparison to the State-of-the-art

To verify the effectiveness of the proposed method, we conduct the comparisons between our FCN method and two state-of-the-art object detection algorithms in computer vision, i.e., Regions with CNN features (R-CNN) [107] and Deformable Part Models (DPM) [123]. The comparisons are also focused on recall, precision and running speed.

We plot the precision-recall curves for our FCN method, R-CNN and DPM in Figure 3.7. It shows the comparisons on both PASCAL VOC 2007 testing set and LISA-Q testing set.



(A) Comparisons on PASCAL VOC 2007

(B) Comparisons on LISA-Q

FIGURE 3.7: Precision-recall curves of our FCN method and other state-of-the-art detectors on PASCAL VOC 2007 testing set (left) and LISA-Q testing set (right).

As can be seen, on both benchmark datasets, the proposed FCN method outperforms the two state-of-the-art detectors significantly. In specific, the proposed FCN method can achieve recall and precision both higher than 80% at the same time in LISA-Q dataset, which shows promising in realistic applications.

The running speed comparison is presented in Table 2. Please note that for the two CNN-based methods (i.e., FCN (ours) and R-CNN), the implementation is on GPU and based on the popular deep learning open source platform Caffe [124]. We adopt the standard setting of R-CNN and DPM. In specific, 2,000 object proposals generated by Selective Search [125] are used for post-classification in R-CNN. HOG [28] feature is utilized in DPM. From Table 2, it is found that the proposed FCN method is much faster than the two state-of-the-art detectors. Considering that both FCN and R-CNN are CNN-based and require the use of GPU, the big difference in speed comes from the difference in the number of feed-forward computation of CNN. In R-CNN, the number of feed-forward computation of CNN equals to the number of object proposals (2,000 by default). By

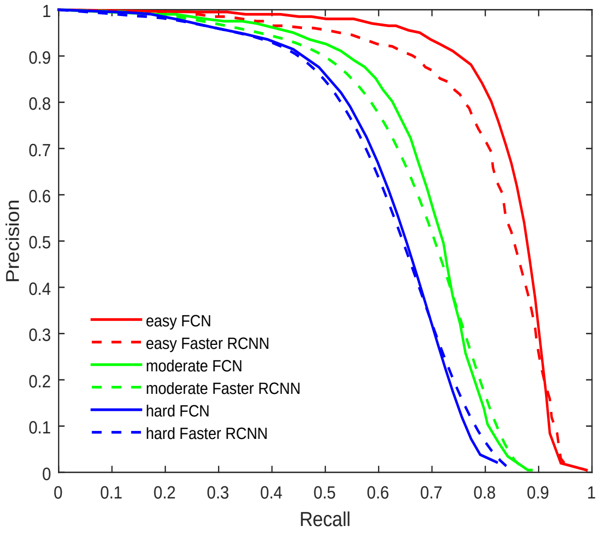


FIGURE 3.8: Precision-recall curves of our FCN method and Faster R-CNN for pedestrian detection in KITTI dataset.

contrast, our FCN method only require N times of feed-forward computation of CNN, which is the number of scales in the multi-scale inference (less than 100). As for DPM, the computation of hand-crafted feature–HOG, takes most of the time.

For pedestrian detection, we compared the our FCN method with a competitive baseline Faster R-CNN on KITTI dataset. The precision-recall curves of both methods are demonstrated in Figure 3.8. One can observe that for all the annotated pedestrians, including easy, moderate and hard ones, FCN consistently outperforms Faster R-CNN. The advantage of FCN over Faster R-CNN is the most significant in easy ones. This verifies the effectiveness of FCN method for pedestrian detection.

Although the proposed FCN method performs well for either only on-road vehicle detection or pedestrian detection, it does not guarantee that detection of both of them with a single trained FCN model still works well. Therefore, we trained a FCN for 5-category (i.e., car, bus, motorbike, bicycle and pedestrian) classification on PASCAL VOC 2007 trainval set and then use the FCN to detect objects belonging to these categories. The

TABLE 3.3: Vehicle and pedestrian detection average precision for all the 5 categories on the PASCAL VOC 2007 testing set comparison.

	bike	bus	car	mbike	person
R-CNN	65.8	59.7	60.0	69.0	58.1
Faster R-CNN	77.4	81.8	79.0	75.9	74.5
FCN	77.5	81.3	78.5	74.3	73.8

quantitative results for the 5 categories are shown in Table 3.3. It can be seen that FCN obtains much better than R-CNN and also comparable detection precisions with Faster R-CNN, which validates the efficacy of the FCN method.

3.5.3.3 Visualizations

Figure 3.9 shows examples of confidence maps and detections for given images. It is observed that the FCN produce reliable confidence maps for the different types of vehicles (car, motorbike, bus, etc.). Moreover, the detected vehicles are in great diversity of visual appearances (e.g., shape, size and color) and environmental conditions (e.g., weather, illumination and view angle), showing the robustness of the FCN method. After producing the confidence maps, detection bounding boxes can be directly obtained by mapping the pixels with high confidences back to their CNN receptive fields in the input images. This localization manner enables effective detection of occluded vehicles, as shown in the third column of Figure 3.9.

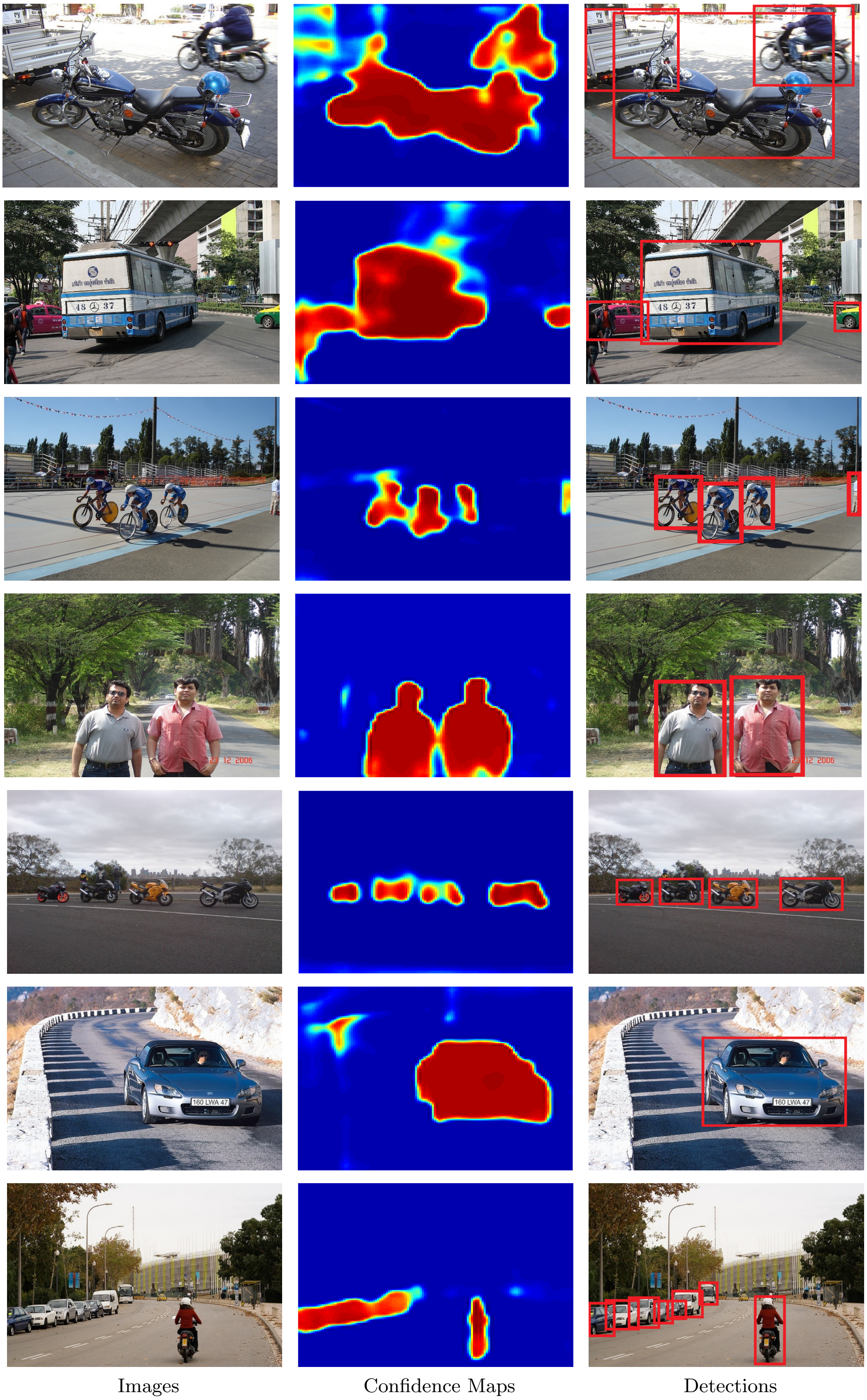


Figure 3.9: Examples of predicted vehicle confidence maps (second column) and detection bounding boxes (third column) for the input images (first column). In the confidence maps, red color indicates high confidence to be a vehicle while blue color represents low confidence.

3.6 Conclusions

In this chapter, we propose a data-driven learning framework which can directly process the raw visual data captured by cameras to perform vehicle and pedestrian detection. The proposed method is based on fully convolutional networks (FCN), which can accept input images with arbitrary sizes and produce output vehicle confidence maps with corresponding sizes. Each pixel in the vehicle confidence map shows the probability to contain a vehicle or a pedestrian for the receptive field of this pixel in the input image. In addition, based on the image gradients, a bounding box refinement step is utilized to refine the raw bounding boxes obtained by FCN. The extensive experiments on PASCAL VOC 2007 testing set, LISA-Q dataset and KITTI dataset validate the effectiveness of the proposed FCN method. Compared with two state-of-the-art detectors, i.e., R-CNN and DPM, FCN method shows better precision, recall and computation efficiency in vehicle detection, which shows promising in the realistic real-time on-road vehicle and pedestrian detection application.

Chapter 4

Object Proposal Based Object Detection

4.1 Object Proposal Generation with Fully Convolutional Networks

4.1.1 Overview

Object proposal generation has become crucial for object-based vision tasks, like class-specific object detection and semantic segmentation. Instead of dealing with 10^6 to 10^7 bounding boxes across all possible scales in a sliding window manner [106], object proposal generation aims to find all candidate regions that may contain objects in an image [126]. Compared with the sliding window scheme, object proposals benefit the object detection in two aspects: saving computation time spent on the tremendous number of sliding windows and improving the detection accuracy by enabling the use of more sophisticated detectors [11, 127, 128] due to the smaller number of inputs passed to the detector.

A generic object proposal generator should normally satisfy the following requirements: it should be able to capture objects of all scales, have small biases towards object class, achieve high recall with a manageable number of proposals (from several hundred to a few thousand per image) and be computationally efficient.

Current object proposal generators primarily rely on low-level image cues, such as saliency, gradient and edge information [113, 114]. The main rationale behind these methods is that all objects of interest share common visual properties that can easily distinguish them from the background. However, sometimes visual appearance variation of objects makes it difficult for low-level cues to distinguish them from background (e.g., a girl wearing green dress running on the grassland, or in the forest with messy background and strong texture). Therefore, "objectness" is more of a high-level semantic concept showing semantic information of a region, which implies the presence of objects better than the low-level cues. In addition, when faced with image perturbations (e.g., blurring, JPEG compression and "salt and pepper" noise) which may cause big low-level appearance variation, such a semantic definition of objectness also provides stronger robustness and stability.

In this chapter, we present a data-driven learning pipeline to produce a high-level semantic objectness score, which shows to what extent a specific region may contain an object. Briefly, we train an object/non-object binary classifier using a fully convolutional

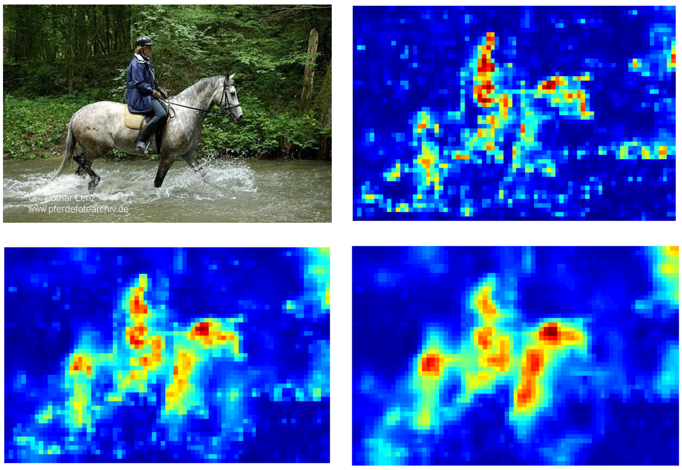


Figure 4.1: Illustration of original image (left top), objectness map of scale 32*32 (right top), objectness map of scale 64*64 (left bottom) and objectness map of scale 128*128 (right bottom). All the objectness maps have been scaled up to the same size as the original image. Each pixel in the objectness map shows the probability of a corresponding box region containing an object.

network (FCN) [46] on patches from images with annotated objects. The fully convolutional network can take an input image of arbitrary size and output a dense "objectness map" showing the probability of containing an object for each corresponding box region in the original image. An example is shown in Figure 4.1. To predict the objectness for boxes of different scales, we rescale the original image into multiscales and feed them to the network to obtain the objectness maps of different scales correspondingly. Then, non-maximal suppression (NMS) is performed to remove redundant low-quality proposals. Finally, we train an SVM on the image gradients to refine the proposals by finding the proposal with the highest objectness score among the neighboring boxes of each rough proposal obtained by FCN.

Extensive experiments on the PASCAL VOC 2007 [129] demonstrate the superiority of our approach both on the object recall rate and class-specific object detection mAP. The robustness is investigated by testing perturbed images from PASCAL VOC 2007 and the generalization ability of the approach is validated using ILSVRC 2013 [121].

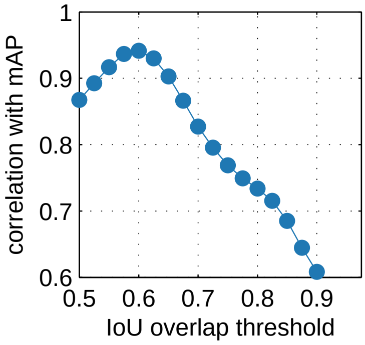


FIGURE 4.2: Correlation between detection mAP of Fast R-CNN on the PASCAL VOC 2007 and recall at different IoU thresholds.

The remainder of the chapter is organized as follows. First, we survey the related works on object proposal generation in Section 4.1.2. Then, we elaborate the multiscale fully convolutional networks for object proposal generation in Section 4.1.3. Subsequently, the proposal refinement with SVM is introduced in Section 4.1.4. After showing the experimental results and analysis in Section 4.1.5, we make some discussions and the conclusion in Section 4.1.6.

4.1.2 Related Work

The existing approaches for generating object proposals can be classified into two types: *Segment grouping methods* and *Window scoring methods* [130]. Apart from these, we also list the related approaches for object proposals/detection which are based on Convolutional Neural Networks (CNN).

Segment grouping methods aim to generate multiple segments that may contain objects. This type of methods typically depends on an initial oversegmentation (e.g., superpixels [131]). Then different merging strategies are adopted to group the similar segments into object proposals. Similarity measures usually rely on diverse low-level cues, e.g., shape, color and texture. For example, Selective Search [125] greedily merges superpixels to generate proposals in a hierarchical scheme without learning. Randomized Prim [132] learns a randomized merging strategy based on the superpixel connectivity graph. Rantalankila et al. [133] used superpixel merging combined with graph cuts to generate proposals. Multiscale Combinatorial Grouping (MCG) [134] utilizes multi-scale hierarchical segmentation and merges them based on edge strength to obtain proposals. Geodesic Object Proposal (GOP) [135] starts from over-segmentation, and then computes a geodesic distance transform and selects certain level sets of the distance transform as the object proposals.

Usually this type of methods achieves high recall when the intersection over union (IoU) threshold criterion is relatively large (>0.7), indicating the precise localization ability. However, when choosing a relatively loose IoU threshold criterion (<0.7), the recall may not be as good as *Window scoring methods*. In addition, high quality proposals of these methods are often obtained by multiple segmentations in different scales and colorspaces, thus they are computationally expensive and more time-consuming.

Window scoring methods are designed to show how likely a candidate window is to contain an object of interest. Generally, this type of methods first initializes a set of candidate bounding boxes across scales and positions in the image, and then sorts them with a scoring model and selects the top ranked boxes as object proposals. Objectness [110] selects some salient locations from an image, and then sorts them according to multiple low-level cues, e.g., color, edge, location and size. Zhang et al. [136] proposed a cascade of SVMs trained on gradient features to estimate the objectness. The SVMs are trained for different scales and the method outputs a pool of boxes at each scale, followed by another SVM to rank all these obtained boxes. BING [113] trains a simple linear SVM classifier over the gradient map and applies it in a sliding window manner when testing. Using binary approximation enables it to be finished within 10ms per image. Edge Boxes [114] is also performed in a sliding window manner and scores the windows based on the edge maps obtained by some edge detection techniques [137]. Then, box refinement is used to improve localization precision.

Compared to *segment grouping methods*, *window scoring methods* are usually computationally efficient as they do not output a segmentation mask. Another advantage of them is the high recall when setting a relatively low IoU threshold criterion ($<$0.7). The main drawback of this type is the poor localization accuracy due to the discrete sampling of the sliding windows, leading to a low recall given a high IoU threshold criterion. However, the recent findings [130] showed that object detection mean average precision (mAP) has the strongest correlation with the recall at IoU threshold around 0.6, and the correlation decreases with the increasing of the IoU threshold, as shown in Figure 4.2. This suggests that high recall at a relatively low IoU threshold is more important than precise localization of the proposals for achieving a good detection mAP.

CNN in object proposal/detection. CNN, as a popular deep learning model, is also utilized for object proposal/detection tasks. Overfeat [138] trains a deep CNN to simultaneously predict the box coordinates and category confidence for each object in a sliding window manner to solve the class-specific object detection problem. MultiBox [139, 140] trains a CNN to directly regress a fixed number of proposals without sliding the network over the image and then ranks the proposals by their CNN confidences of being the bounding box of an object. They achieve top results on the ImageNet detection task.

Karianakis [141] extracted the convolutional responses of an image from first layers of the CNN, and then fed them to a boosting model which differentiates object proposals from background. Pinheiro [142] trained a CNN to output a class-agnostic segmentation mask and the likelihood of the patch being centered at a full object for each patch in an image. Trained on the expensive pixel-level labeled images, they reported top recall on both PASCAL and Microsoft COCO benchmarks [122].

Another type of approaches does not output the proposals by themselves, and instead they re-rank the proposals generated by other methods. DeepBox [143] re-ranks the proposals of other methods based on their CNN output values which reflect the high-level objectness and improve the object recall. Each proposal is fed into the network to obtain an objectness value and thus a high time cost is required to pass all the proposals (usually several thousand) separately to the CNN network. Salient Object Subitizing [144] trains a CNN to identify the number of salient objects in an image and selectively reduces the number of retrieved proposals according to the predicted number of salient objects. The recall of other object proposal methods can be improved by allocating a proper number of proposals in this way.

Our method can also be categorized as a *Window scoring method*. The difference between our approach and the existing *Window scoring methods* is the window scoring scheme. We use fully convolutional networks to output high-level semantic objectness maps instead of judging from low-level cues. The most similar method to ours may be Region Proposal Networks (RPN), which is used in the Faster R-CNN detection pipeline for class-agnostic proposal generation. RPN predicts proposals for each image region in a sliding window manner based on a set of pre-defined anchors in the region. Compared

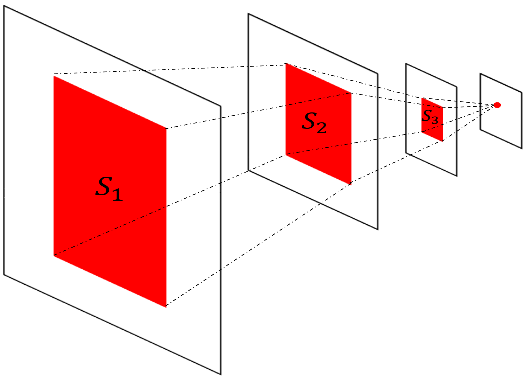


Figure 4.3: Illustration of fully convolutional network. Red pixels in the output map show the classification confidence of the red window region S_1 in the input image and will not be affected by other regions in the input image.

to other CNN-based object proposal methods, our FCN neither generates only a fixed number of proposals nor needs expensive pixel-level labeled training samples, and it can be end-to-end both in training and testing stages. Another difference is that we do not regress the box coordinates like OverFeat [138], MultiBox [139, 140] and RPN [145], and instead decide the window which the pixel in the output map corresponds to as a proposal. Combining such a mapping localization method with the multi-scale scheme obtains better precision than the box coordinates regression. To improve the localization precision, a learning based refinement method is utilized to iteratively search for a window with a higher objectness score.

4.1.3 Multiscale Fully Convolutional Networks

4.1.3.1 Fully Convolutional Networks for Dense Objectness Prediction

Convolutional Neural Network (CNN) can be seen as an automatic hierarchical feature extractor combined with a single classifier. Such a learning-based deep feature extraction pipeline avoids hand-crafted feature designing which may not be suitable for

Table 4.1: **Fully convolutional network architecture.** The spatial size of the feature map depends on the input image size, which varies during our inference step. Here we show training spatial sizes.

Layer	1	2	3	4	5	6	7	8	9	10	11
Type	conv	conv	conv+max pool	conv	conv	conv+max pool	conv	conv	conv+max pool	conv	conv
#channels	64	64	64	128	128	128	256	256	256	512	2
Conv. kernel size	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1	1×1	3×3	1×1
Conv. stride	1×1	1×1	1×1	1×1	1×1	1×1	1×1	1×1	1×1	1×1	1×1
Pooling size	-	-	2×2	-	-	2×2	-	-	2×2	-	-
Pooing stride	-	-	2×2	-	-	2×2	-	-	2×2	-	-
Zero-padding size	1×1	1×1	1×1	1×1	-	-	-	-	-	-	-
Spatial input size	40×40	40×40	40×40	20×20	20×20	18×18	8×8	6×6	6×6	3×3	1×1

a particular task, and meanwhile strengthens the discrimination power of the feature. Recently, as an extension of the classic CNN for classification problems [6, 12, 13], fully convolutional networks can take an input of arbitrary size and output a map whose size corresponds to the input, which can be used for dense prediction problems (e.g., semantic segmentation [46, 108], image restoration [109] and depth estimation [146]).

We feed the whole image into the fully convolutional network to obtain a dense objectness map. This feed-forward process can be seen as object/non-object binary classification for the densely sampled sliding windows in the input image. Each output pixel in the objectness map shows the classification confidence of one specific sliding window in the input image, as illustrated in Figure 4.3. To map back to the input object proposal boxes from the output objectness map, we have to decide how big the area is that the output pixel can correspond to in the input image (receptive field size). Assume the receptive field size of each layer is S_i (i=1, 2, ..., n) and S_1 is the receptive field size in the input image. The receptive field size of each layer can be computed using the recursive formula below:

$$S_{i-1} = up(S_i) = s_i(S_i - 1) + k_i \tag{4.1}$$

where s_i and k_i represent the stride and the convolution kernel size of the i^{th} convolutional or pooling layer. S_{i-1} and S_i denote the receptive field size of the $(i-1)^{th}$ and the i^{th} layer respectively. To accurately map an output pixel back to the window region it covers in the input image, apart from knowing the receptive field size in the input image, the sliding window sampling stride *Str* is also indispensable. A fully convolutional network has its inherent sampling stride *Str*, which is the product of the strides of all the layers, i.e.,

$$Str = \prod_{i=1}^{n} s_i \tag{4.2}$$

where s_i indicates the stride of the i^{th} convolutional or pooling layer. With the known S_1 and *Str*, the window region which corresponds to the output pixel $(x_o,\ y_o)$ can be decided as below:

$$\begin{aligned} x_{min} &= x_o Str \\ x_{max} &= x_o Str + S_1 \\ y_{min} &= y_o Str \\ y_{max} &= y_o Str + S_1. \end{aligned} \tag{4.3}$$

In contrast to other sliding window approaches that compute the entire pipeline for each window, fully convolutional networks are inherently efficient since they naturally share computation common to different overlapping regions. When applying a fully convolutional network to the input of an arbitrary large size in testing, convolutions

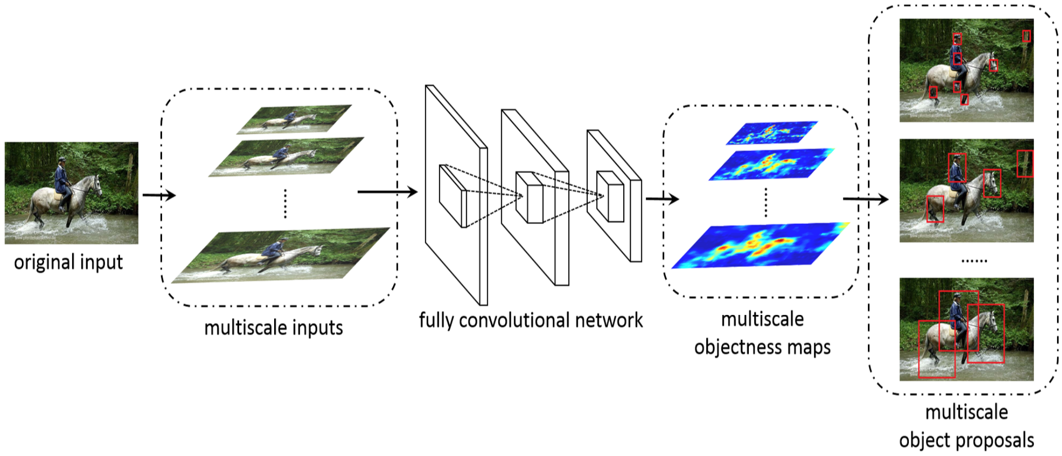

original input
multiscale inputs
fully convolutional network
multiscale objectness maps
......
multiscale object proposals

FIGURE 4.4: Pipeline of multiscale object proposal generation by a single fully convolutional network.

are applied in a bottom-up manner so that the computation common to neighboring windows only needs to be done once.

4.1.3.2 Network Architecture and Patch-wise Training

For the implementation of our idea, a new fully convolutional network architecture for objectness prediction is designed and trained from scratch. The detailed architecture of the network is shown in Table 4.1. The network architecture is similar to VGG [12]: the first two pooling layers follow three convolutional layers with kernel size 3. The last two 1×1 convolutional layers follow the idea of Network in Network (NIN) [147], and they can be seen as the cascaded cross channel pooling structure allowing complex and learnable interactions of cross channel information. All the convolutional layers are followed by a ReLU non-linear activation layer. On the top of the network, a softmax normalization layer is used to ensure the output confidence within the range (0, 1). The loss function to be optimized is the cross-entropy loss, i.e.,

$$E = t_k \ln(y_k) + (1 - t_k) \ln(1 - y_k) \tag{4.4}$$

where t_k denotes the k^{th} target value and y_k represents the k^{th} prediction value. According to Eqn. 4.1 and Eqn. 4.2, the receptive field size of the input S_1 for this network is 40 and the sampling stride for the network is 8. It is worth mentioning that the sampling stride is 0.2 timing the receptive field size (also the window size), which is close to the empirically optimal sliding window sampling stride ratio recommended by [114]. This is the main factor we consider in designing the network.

In terms of training, we treat the network as an object/non-object binary classification network and use a patch-wise training strategy instead of using the whole images to train a dense structured prediction network. To this end, we crop the patches from the images with annotated objects and resize them to 40×40, the same as S_1. Among the cropped patches, those with IoU ≥ 0.5 with a ground-truth box are treated as positive samples and the rest as negatives. To balance the number between the positives and the negatives, we crop multiple patches around each ground-truth box while sparsely sampling the patches in the background regions.

For the stochastic gradient descent (SGD) training process, the weights of all the layers are initialized with a zero-mean Gaussian distribution with the standard deviation 0.01 and the biases are initialized with 0. The learning rate is 0.01, which will be reduced by a scale of 10 after every 20 epochs. The minibatch size is set as 256.

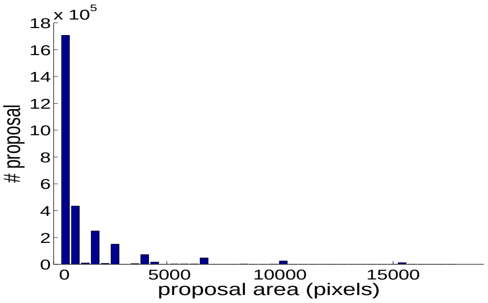


Figure 4.5: The distribution of the proposal area.

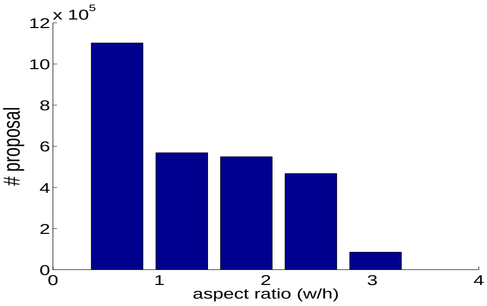


Figure 4.6: The distribution of the aspect ratio of the proposals.

4.1.3.3 Multiscale Inputs Inference

Using the above mentioned fully convolutional network, each pixel in the output map only covers a window region with a fixed size 40. To enable the network to predict the object proposals with different sizes and aspect-ratios, we rescale the original image to different scales. By doing this, a window with the size equal to the receptive field size 40 in the rescaled inputs will correspond to windows of different scales in the original image. The rescaled input size S_r can be computed according to the original input size S_o and the size of a window region which is needed to be detected using Eqn. 4.5 denoted as S_w.

$$\frac{S_r}{40} = \frac{S_o}{S_w}. \tag{4.5}$$

Subsequently, the multiscale inputs after rescaling are fed into the network individually to obtain the multiscale objectness maps (see Figure 4.1). It can be seen that the map corresponding to a small scale (32×32) characterizes the boundaries better but can hardly capture the internal regions of the objects. In contrast, the map corresponding to a large scale (128×128) focuses more on the localization of the whole big objects but is unable to depict the boundary details well. Therefore, we utilize the multiscale strategy to generate the object proposals in all kinds of scales. The pipeline of our method is illustrated in Figure 4.4.

Here we present the multiscale setting in detail to specify the scales needed in our approach. Generally, the more and the denser the scales are, the more a concentrated set of bounding boxes near the areas is likely to contain an object. However, the downside is that noisy bounding boxes which may lower the recall of the top candidate boxes will be produced as well. This issue introduces a trade-off in parameter selection for the multiscale setting.

Specifically, we define α as the stepsize indicating the IoU for neighboring boxes. In other words, the step sizes in scale and aspect ratio are determined such that one step results in neighboring boxes having an IoU of α. The scale values range from a minimum box area of 1000 pixels to the full image. The aspect ratio changes from 1/3 to 3. The exact values of the scale and the aspect ratio can be computed with Eqn. 4.6 and Eqn. 4.7.

$$\mathrm{scale} = \sqrt{1000}(\sqrt{1/\alpha})^{\mathrm{s}}, \tag{4.6}$$

$$\mathrm{aspect\ ratio} = (\frac{1+\alpha}{2\alpha})^{\mathrm{r}}. \tag{4.7}$$

Here the index s can be any integer from 0 to $\lfloor \log(\mathrm{image\ size}/\sqrt{1000})/\log(\sqrt{1/\alpha})\rfloor$, and the index r can be any integer from $-\lfloor \log(3)/\log(\frac{1+\alpha}{2\alpha})^2\rfloor$ to $\lfloor \log(3)/\log(\frac{1+\alpha}{2\alpha})^2\rfloor$. A value of $\alpha = 0.65$ is ideal for most of the cases [114] so we fix α as 0.65 in the experiments. The distribution of the proposals in terms of their areas and aspect ratios are shown in Figure 4.5 and Figure 4.6 respectively, from 100 images which are randomly selected from

PASCAL VOC 07 test set when setting α as 0.65. For the multiscale proposals, we first remove those with objectness lower than 0.2, reducing the total proposal number from several tens of thousands to less than 10000. Next, we sort all the remained proposals based on their objectness in a descending order. Finally, non-maximal suppression (NMS) is performed on the sorted proposals. Specifically, we find the proposal with the maximum objectness score and remove all the proposals with an IoU larger than an overlap threshold (we use 0.8 in all our experiments).

Algorithm 2 Refine the proposals [P_1, P_2, ..., P_n]

```
Require: : A set of raw proposals [$P_1$, $P_2$, ..., $P_n$]
  for $P = [P_1, ...P_n]$ do
    $Obj \Leftarrow svm(P)$
    $S_c \Leftarrow 0.2P_w$ ($P_w$ is the proposal box width)
    $S_r \Leftarrow 0.2P_h$ ($P_h$ is the proposal box height)
    while $S_c > 2$ and $S_r > 2$ do
      $[P_{c1}, P_{c2}, ..., P_{cn}] \Leftarrow ColomnNeighbors(P, S_c)$
      $P_{cmax} \Leftarrow argmax(svm(P_{ci})),\ i = [1, 2, ..., n]$
      if $svm(P_{cmax}) > Obj$ then
        $Obj \Leftarrow svm(P_{cmax})$
        $P \Leftarrow P_{cmax}$
      end if
      $[P_{r1}, P_{r2}, ..., P_{rn}] \Leftarrow RowNeighbors(P, S_r)$
      $P_{rmax} \Leftarrow argmax(svm(P_{ri})),\ i = [1, 2, ..., n]$
      if $svm(P_{rmax}) > Obj$ then
        $Obj \Leftarrow svm(P_{rmax})$
        $P \Leftarrow P_{rmax}$
      end if
      $S_c \Leftarrow S_c/2$
      $S_r \Leftarrow S_r/2$
    end while
  end for
```

4.1.4 Box Refinement with Gradients Cues

Due to the fixed multiscale setting and box sampling strategy, the above obtained raw proposals have their inherent weakness of being pre-defined both in scales and locations

Table 4.2: Runing speed of FCN with different α and the refinement step.

	Runing time per image
$\alpha = 0.45$ no refine	0.53s
$\alpha = 0.45$ refine	0.62s
$\alpha = 0.55$ no refine	0.66s
$\alpha = 0.55$ refine	0.77s
$\alpha = 0.65$ no refine	0.95s
$\alpha = 0.65$ refine	1.10s
$\alpha = 0.75$ no refine	2.12s
$\alpha = 0.75$ refine	2.39s
$\alpha = 0.85$ no refine	5.23s
$\alpha = 0.85$ refine	5.83s

which may cause misdetection of ground-truth boxes. To overcome this, we adopt a greedy iterative search method to refine each raw proposal.

Previous works show that objects are stand-alone things with well-defined closed boundaries and centers [110–112]. Based on this observation, gradient and edge information are widely used for implying the presence of objects in early works, e.g., BING [113] and Edge Boxes [114]. Considering this, we rely on the low-level gradients cues instead of 3-channel RGB information for the efficient implementation of our method. Specifically, we train a linear SVM object/non-object classifier on the gradient maps of the patches from the images with annotated objects. We use the ground-truth boxes of the annotated objects as positive samples, and crop the patches in the images and treat those with IoU < 0.3 for all the ground-truth boxes as negative samples. For all the chosen samples, we resize them to 16×16 before training the SVM. Having trained the 256-d SVM, to refine the proposals, we maximize the SVM score of each box over the neighboring positions, scales and aspect ratios. After each iteration, the search step is reduced in half. The search is stopped once the translational step size is less than 2 pixels. The procedure is summarized in pseudo-code in Algorithm 2.

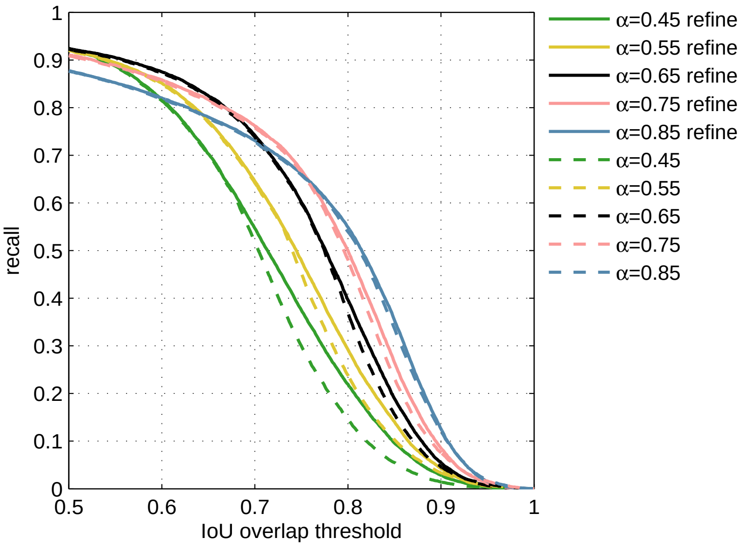
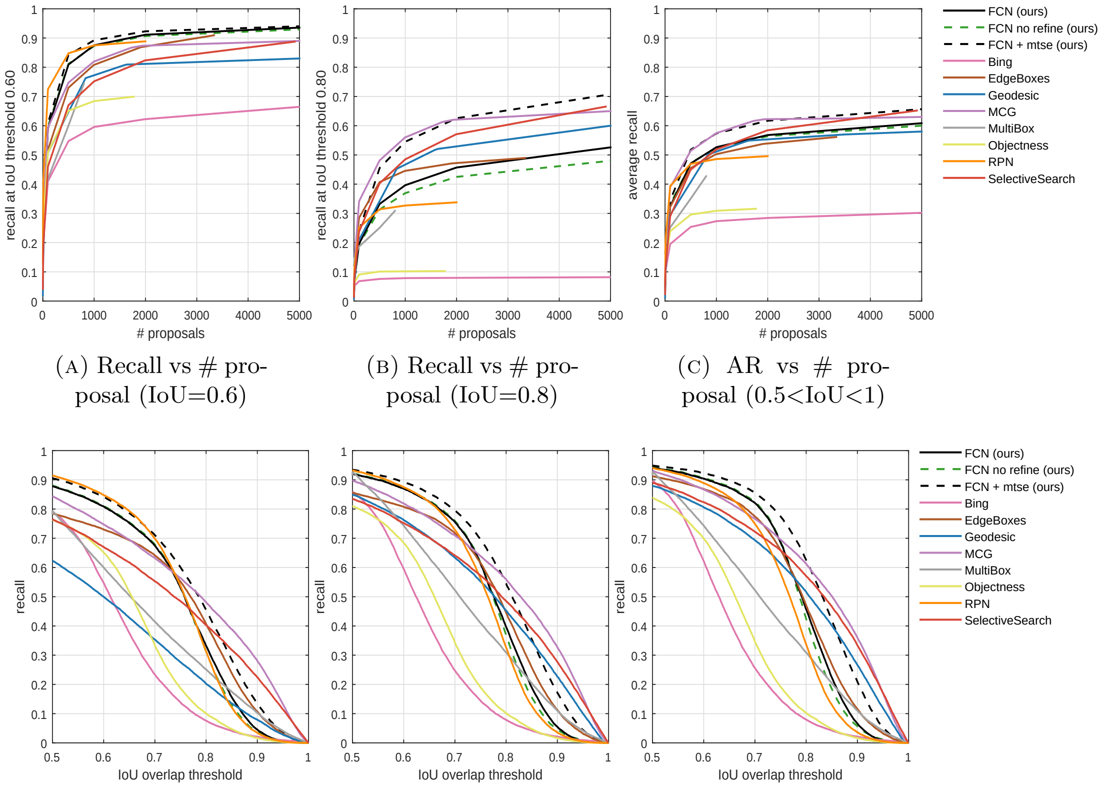


Figure 4.7: Recall versus IoU threshold for various search stepsizes α (1000 proposals per image) on the PASCAL VOC 2007 test set.



(A) Recall vs # proposal (IoU=0.6)

(B) Recall vs # proposal (IoU=0.8)

(C) AR vs # proposal (0.5<IoU<1)

(E) Recall vs IoU (500 proposals)

(F) Recall vs IoU (1000 proposals)

(G) Recall vs IoU (2000 proposals)

Figure 4.8: Recall comparison between the FCN method and other state-of-the-art methods on PASCAL VOC 2007 test set.

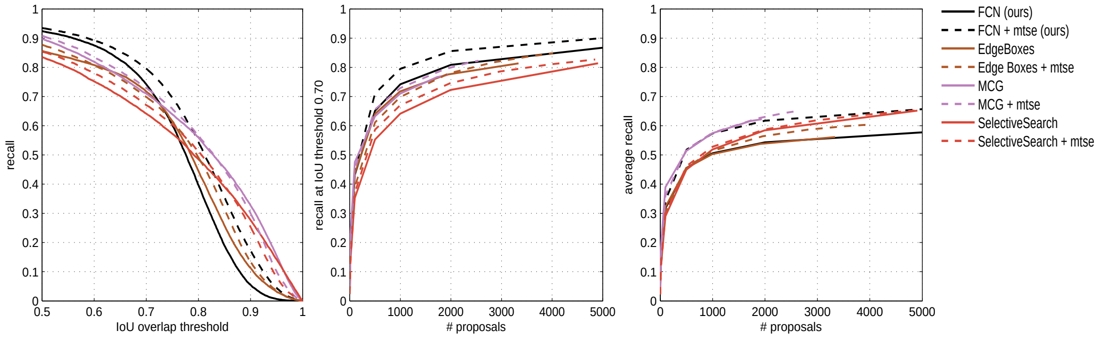
4.1.5 Experiments and Discussion

In this section, we evaluate the performance of our method on PASCAL VOC 2007 test set, PASCAL VOC 2012 validation set, ILSVRC 2013 validation set and MS COCO 2014 validation set. To be fair, similar to other supervised learning based methods, we train the fully convolutional network on the PASCAL VOC 2007 trainval set, which contains 5011 images and around 15000 annotated objects. Our method will be compared with the state-of-the-art in terms of the following four parts: object recall, detection mAP, robustness to image perturbation, generalization to unseen categories.

4.1.5.1 Approach Variants

We begin the experiments by testing different variants of the approach with various parameter settings. First, we analyze the effects of the granularity of the multi-scale search as well as the box refinement step. Figure 4.7 shows the algorithm's behavior based on the search stepsize parameter α and the refinement step, when generating 1000 proposals per image.

As the stepsize α increases, the scales to be computed are increased, leading to more CNN feed-forward passing times. From Figure 4.7, when α is between 0.45 to 0.65, as α increases, recall increases for all the IoU thresholds between 0.5 to 1. This is natural as more scales provide more chances to have a proposal close to the groundtruth bounding box. However, when α exceeds 0.65, as α increases, recall at high IoU thresholds (>0.7) increases while recall at low IoU thresholds (<0.7) decreases. The reason probably lies in that too many boxes concentrated on a small area are introduced, resulting in a loss



(A) 1000 proposals per image

(B) Recall at 0.7 IoU

(C) AR vs # proposal (0.5<IoU<1)

FIGURE 4.9: Recall comparison between methods with MTSE refinement w.r.t different IoU thresholds on PASCAL VOC 2007 test set.

of the recall for top-selected candidate proposals. From Figure 4.7, α should be set as 0.65 or 0.75.

Another critical component to be evaluated is the box refinement step. Figure 4.7 also shows the effect of the refinement step for different search stepsizes α. As can be seen from Figure 4.7, the refinement step indeed improves the recall for all the stepsizes α. However, the smaller the stepsize α is, the more recall improvement is brought by the refinement step. Another finding is that the refinement step only improves the recall at high IoU thresholds and has little effect on the recall at low IoU thresholds. This suggests that the refinement step mainly refines the coarsely localized proposals to fine localized ones, which means improving the IoU of the coarsely localized proposals from > 0.5 to even higher values (e.g., > 0.7).

We also conduct the running time comparison experiment for each search stepsize α and the refinement step on the PASCAL VOC 2007 test set. Table 4.2 presents the detailed running time for various values of α and the refinement step. It is found that for a certain value of α, the time spent on the refinement step is relatively much less than the multi-scale FCN feed-forward computation, e.g, 0.09s for α=0.45, 0.11s for α=0.55 and 0.15s for α=0.65. The major time cost is on the multi-scale FCN computation and when α=0.75, the running time can reach a rather 2.39s with the refinement step. Although setting α as 0.75 achieves a higher recall at high IoU thresholds than 0.65 according to Figure 4.7, we still fix α as 0.65 in all the later experiments for the trade-off between the recall and the running speed.

4.1.5.2 Object Recall

When using object proposals for detection, it is crucial to have a good coverage of all the objects of interest in the testing image, because the missed objects can never be recovered in the subsequent classification stage. Therefore it is a common practice to evaluate the proposal quality based on the object recall. We compare our method with many state-of-the-art methods, including BING [113], CPMC [148], Edge Boxes [114], Geodesic Object Proposal [135], MCG [134], Objectness [110], and Selective Search [125].

Metrics In class-independent object proposals, one of the primary metrics is the object recall, for a fixed IoU threshold, as the number of proposals is changed. Another widely used metric is, for a fixed number of proposals, the object recall as the IoU threshold is varied.

Results We first evaluate recall on the PASCAL VOC 2007 test set, which contains 4952 images with about 15000 annotated objects (including the objects labeled as "d-ifficult") in 20 categories. For the recall computation, the same as [130], we compute the matching between the proposals and the ground-truths so that one proposal cannot cover two ground-truth objects. Figure 4.8(a)-(c) present the recall when varying the number of proposals for different IoU thresholds. We choose two commonly used IoU thresholds, i.e., 0.6 and 0.8 for evaluation, around which the recall shows the strongest correlation with detection mAP (see Figure 4.2). In addition, we plot the average recall (AR) versus the number of proposals curve for the methods. This is because AR summarizes proposal performance across IoU thresholds and correlates well with detection performance. It can be seen that our approach performs better than most of the existing methods at IoU threshold 0.6 for both small and large numbers of proposals. The advantage of our approach reaches the maximum for a small number of proposals (e.g., < 1000), suggesting that our approach can roughly localize the positions of objects with a small number of proposals. For IoU threshold 0.8, our method does not work well, even though the box refinement step boosts the recall by about 5%. This implies that our method does not perform well in localizing objects with very high precision (with IoU < 0.8). As for average recall, our method is only slightly lower than MCG which is the best one in terms of AR. Figure 4.8(e)-(g) demonstrate the recall when the IoU threshold changes within the range [0.5, 1]. It can be seen that no single method can take the dominant place across all IoU thresholds. However, our approach takes the lead by a wide margin when IoU ranges from 0.5 to about 0.75. Please note that we directly employ the publicly available MultiBox model trained on ILSVRC benchmark to extract the proposals. It is surprising that MultiBox does not work well compared to other state-of-the-art methods. We attribute its inferior performance to the poor generalization from ILSVRC benchmark to PASCAL VOC benchmark. For RPN, we directly use the publicly released model which is trained on PASCAL VOC 2012 dataset. It is found that RPN performs slightly better than ours for low IoU thresholds (e.g., 0.6) with a small number of proposals (e.g., <1000), but suffers from poor localization accuracy for high IoU thresholds. This is probably because that RPN does not utilize the multi-scale prediction strategy since multi-scale inference generates much more proposals which brings better results for a large number of proposals and high IoU thresholds but worse recall for a small number of proposals and low IoU thresholds.

Another finding is that the recall of our approach decreases sharply with the increasing of IoU threshold when it is above 0.75. The phenomenon that *window scoring methods* usually outperform *segment grouping methods* for low IoU thresholds while fall behind for high IoU thresholds is also found for other *window scoring methods*, e.g., BING and Edge Boxes. A possible explanation lies in the inherent drawback of *window scoring methods* that discretely sample windows over pre-defined positions and scales.

To improve the poor recall of our method for high IoU thresholds (>0.8), Multi-Thresholding Straddling Expansion (MTSE) [149] can be introduced to adjust our proposals to be better aligned with the boundaries of the superpixels. From Figure 4.8, it can be seen that FCN+MTSE almost takes the first place in all the evaluation cases. To be fair, we also conduct experiments to compare the recall of other state-of-the-art methods with MTSE refinement with ours. Figure 4.9 demonstrates that compared with other methods with MTSE refinement, the FCN method with MTSE achieves better recall for low IoU thresholds (i.e., < 0.8). When looking at high IoU thresholds and average recall between IoU 0.5 to 1, MCG with MTSE refinement performs the best.

For better visualization of our proposals, we show the distribution of the proposals of our method as well as Edge Boxes and MCG for comparison in Figure 4.10. The distribution figures are obtained by assigning red color to the proposal regions according to the

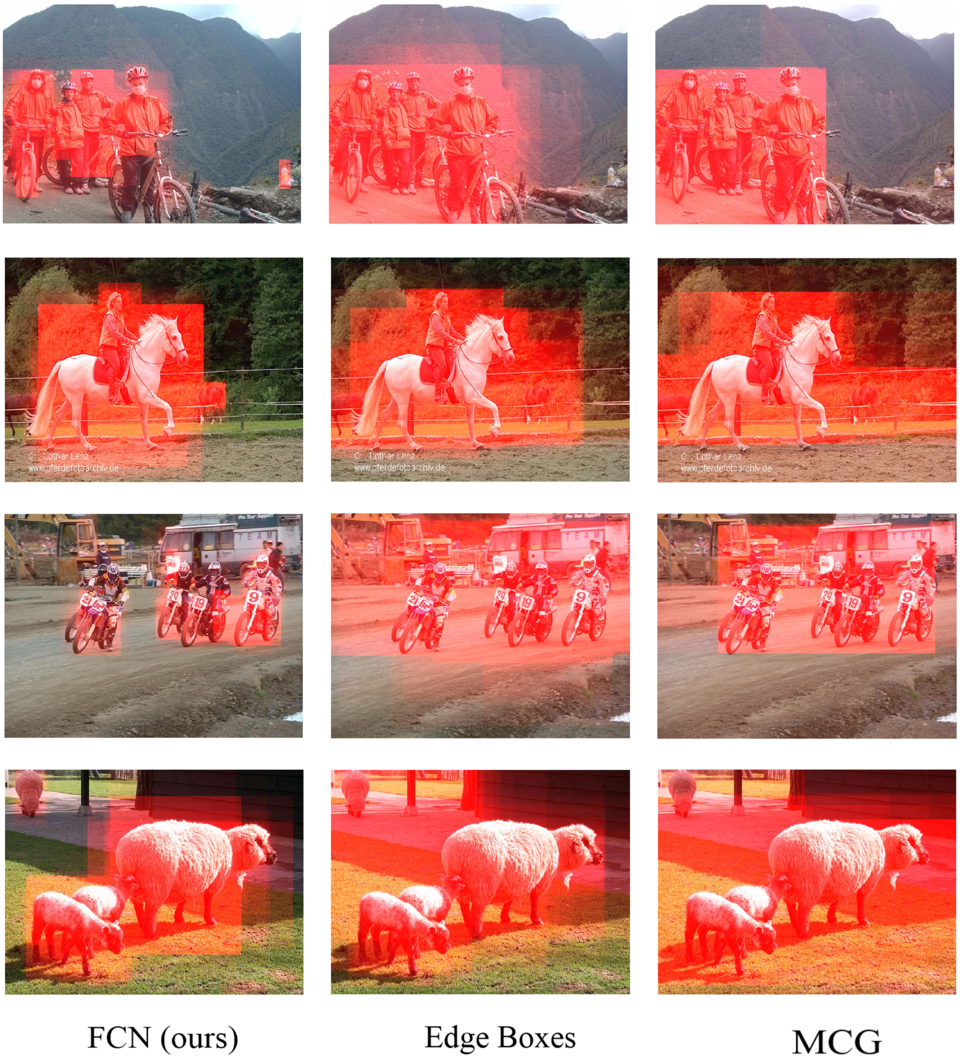


Figure 4.10: Examples of the proposal distribution of Edge Boxes, MCG and our method. Top 2000 proposals are illustrated for each image. For each row, our FCN is on the left, Edge Boxes is in the middle and MCG is on the right.

density of the proposals in that region. It is clear that the proposals of our method are more tightly concentrated on the objects. In contrast, the proposals of Edge Boxes and MCG often spread evenly across a much larger region rather than the objects of interest.

Speed The detailed running speed of our FCN method as well as other state-of-the-art methods is presented in Table 4.3. The detailed setting of parameters for each method

Table 4.3: Runing speed of the state-of-the-arts and our method.

	Runing time per image
BING	0.01s
CPMC	250s
Edge Boxes	0.3s
Geodesic	1s
MCG	30s
Objectness	3s
Selective Search	10s
Our method (no refine)	0.95s
Our method	1.1s

is as follows. We choose the single color space (i.e., RGB) proposal computation for BING, and the "Fast" version for selective search. For the rest methods, we directly run their default codes. Inference for an image of PASCAL VOC size takes 1.1s for our FCN method. Although it is not one of the fastest object proposal methods (compared to BING and Edge Boxes), our approach is still competitive in speed among the proposal generators. We do, however, require use of the library Caffe [124] which is based on GPU computation for efficient inference like all CNN based methods. To further reduce the running time, some CNN speedup methods such as FFT, batch parallelization, or truncated SVD could be used in the future.

4.1.5.3 Object Detection Performance

In this subsection we analyze object proposals for use with object detectors to evaluate the effect of proposals on the detection quality. We utilize the recently released Fast R-CNN [21] detector as the benchmark. For fast evaluation, we adopt the AlexNet [6] instead of the VGG net [12] as the model. The proposals obtained by our approach and another three state-of-the-art object proposal generators, i.e., Edge Boxes, Selective

Search and MCG are used as training samples for fine-tuning the Fast R-CNN detector. Object proposals having IoU ≥ 0.5 with a ground-truth bounding box are positive samples and the rest are negatives. For each method, only the top 2000 proposals are chosen to fine-tune the Fast R-CNN detector.

The detection mean average precision (mAP) and the average precisions of all the 20 categories are presented in Table 4.4. It can be seen that our approach wins on 8 categories among the 20 categories of PASCAL VOC 2007 in terms of detection average precision and also achieves the best mAP 57.3%. Considering that our approach cannot obtain as good recall as the rest three methods when IoU threshold is greater than 0.8, the good detection performance of our approach supports the finding that recall at a very high IoU threshold is not a good predictor for detection mAP compared with the recall at around 0.6 [130], which is shown in Figure 4.2.

4.1.5.4 Robustness

The distribution of the object proposals is quite different from that of sliding windows both on the positive and negative samples used for training a class-specific detector. This requires the proposal generators to be able to consistently propose stable object proposals on the slightly different images with the same image content. This property is associated with the object proposal robustness (called “repeatability” in [130]) when faced with image perturbation. To investigate the proposal robustness, we generate the perturbed versions of the images in the PASCAL VOC 2007 test set and evaluate the robustness faced with three kinds of perturbation, i.e., JPEG artifacts, blurring and “salt and pepper” noise (see Figure 4.11).

Table 4.4: Object detection average precision for all the 20 categories as well as the mean average precision (mAP) on the PASCAL VOC 2007 test set using Fast-RCNN trained on several different proposals.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Sel search	64.8	70.5	55.8	40.2	22.7	67.0	69.2	70.9	30.1	62.1	60.7	62.7	72.1	67.3	56.3	26.0	49.2	57.5	69.2	56.2	56.5
Edge Box	62.2	65.0	50.9	41.8	29.2	70.5	71.4	70.1	30.2	63.7	56.2	61.2	72.8	66.6	60.9	28.5	53.0	54.3	68.2	56.1	56.6
MCG	61.8	64.1	49.9	38.2	21.4	63.4	61.1	67.6	27.1	53.0	63.0	58.7	67.9	59.4	49.4	22.4	46.0	59.8	64.9	57.7	52.8
Ours (no refine)	63.1	67.4	54.2	42.0	33.1	68.9	71.2	69.7	29.1	58.7	51.6	63.7	74.4	66.4	62.8	30.8	51.4	58.1	65.3	59.3	57.0
Ours	63.5	69.9	52.9	44.6	31.9	68.5	71.2	71.1	29.9	62.4	50.8	62.8	75.0	68.0	62.1	29.2	52.0	56.6	65.7	57.5	57.3

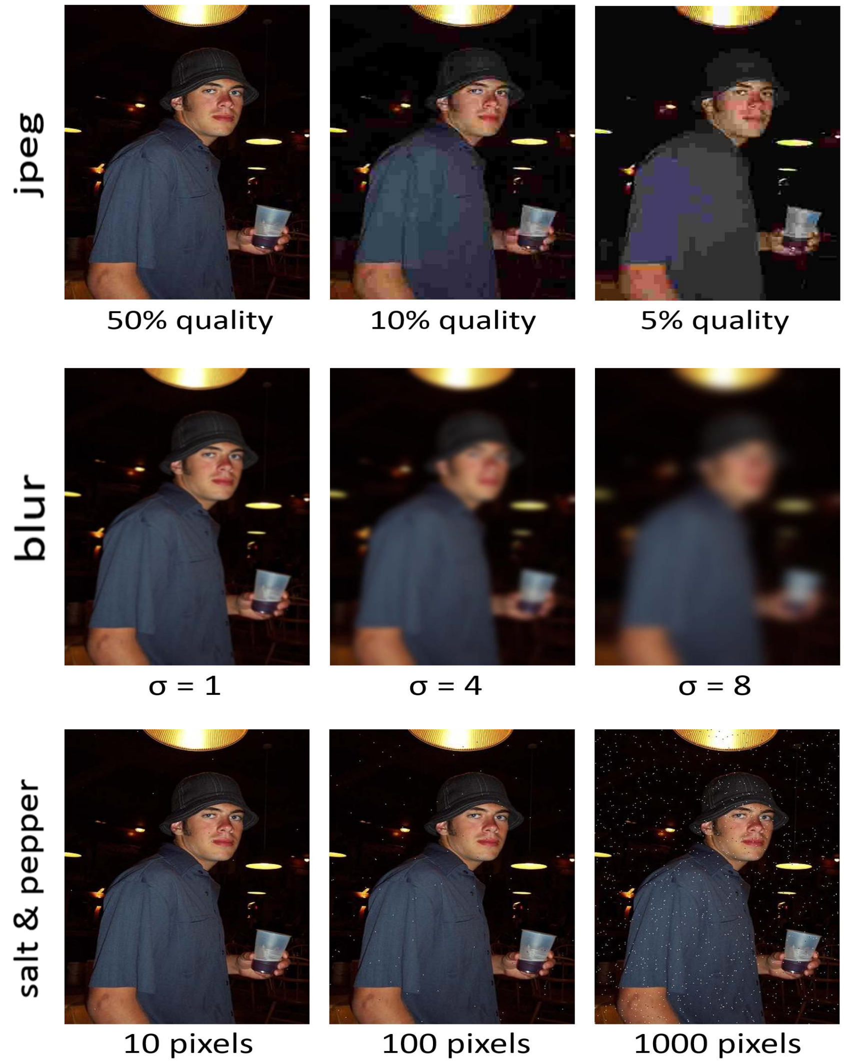
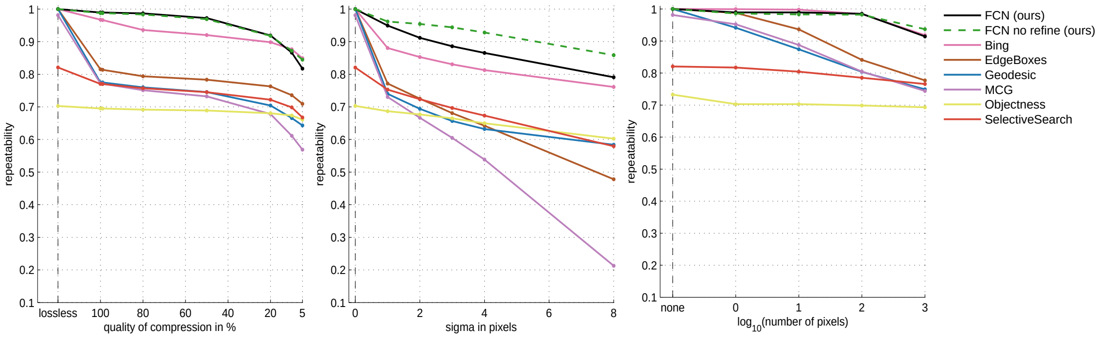


Figure 4.11: Illustration of the perturbation images for the robustness experiments.

Metrics For each pair of the original image and the perturbed image we generate the proposals (top 1000 proposals) for each method. The proposals of the perturbed image are mapped back to the original image and matched to the proposals of the original image. Matching is performed at different IoU thresholds. Next, we plot the recall for every IoU threshold and define the robustness as the area under this "recall versus IoU threshold" curve between IoU 0 and 1. By doing this, the methods which generate proposals at similar locations for the original image and the perturbed image will obtain higher robustness.

Results Figure 4.12(a) shows the robustness of the methods faced with JPEG artifacts. The perturbed images are obtained by writing the images with Matlab "imwrite" function with different compression quality settings from 5% to 100% (see Figure 4.11). Because even 100% quality setting is still lossy in the image quality, we include a lossless setting. It can be seen that except for 5% quality, our methods (both the refined one or the non-refined one) lead across all the compression qualities by a wide margin. Figure 4.12(b) demonstrates the robustness after blurring with different degrees. The blurred images are obtained by smoothing the original images using a Gaussian kernel with s-tandard deviations $0 \leq \sigma \leq 8$ (see Figure 4.11). Similarly, our methods outperform the others significantly in all the cases. It is worth mentioning that the non-refined version of FCN outperforms the refined version, mainly because the refined version relies on the image gradients which are heavily affected by the blurring. Figure 4.12(c) presents the robustness faced with salt and pepper noise. The noise is produced by adding the noise to the image in between 1 to 1000 random locations. Our methods (both the refined



(A) JPEG artifacts (B) Blur (C) Salt and pepper noise

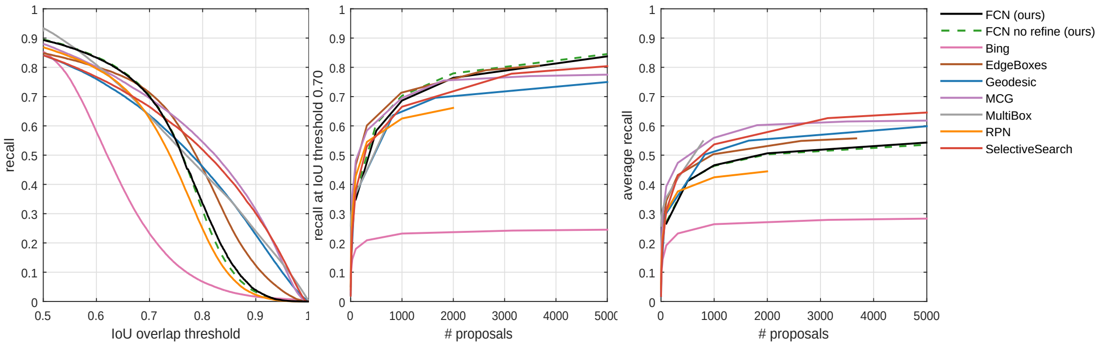
Figure 4.12: Robustness results under various perturbation.

one and the non-refined one) almost achieve the same robustness with BING, which is the best among the state-of-the-art ones.

In general, we find that the *segment grouping methods* which are based on superpixels are more prone to small perturbation and have worse robustness compared to the *window scoring methods* (e.g., our method, BING and Edge Boxes). This may be due to the fact that superpixels strongly depend on the low-level cues which are more sensitive to small image perturbation. In contrast, our method keeps the best robustness in most of the perturbation cases. The superiority reflects that the high-level semantic learning based objectness not only helps to achieve good recall but also provides more stable proposals in the perturbed images.

4.1.5.5 Generalization to Unseen Categories

The good recall our approach achieves on the PASCAL VOC 2007 test set does not guarantee it to have learned the generic objectness notion or be able to predict the object proposals for the images containing novel objects in unseen categories. Because



(A) 1000 proposals per image

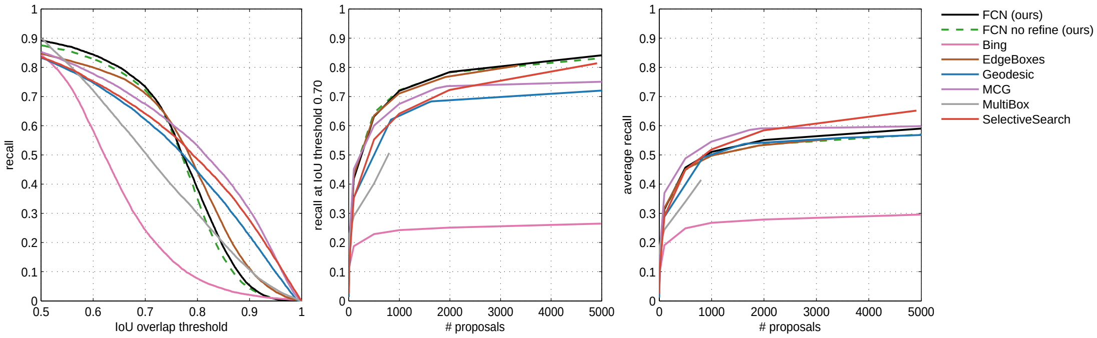
(B) Recall at 0.7 IoU

(C) Average recall (IoU between (0.5, 1))

FIGURE 4.13: Recall versus IoU threshold on ImageNet ILSVRC 2013 validation set.

it is possible that the model is highly tuned to the 20 categories of PASCAL VOC. To investigate whether it is capable of predicting the proposals for the unseen categories beyond training, we evaluate our approach on the ImageNet ILSVRC 2013 validation set which contains more than 20k images with around 50k annotated objects in 200 categories. Note that the 200 categories are not fine grained versions of the 20 categories of PASCAL VOC. Many of them are totally different from the PASCAL VOC categories, such as food (e.g., bananas) or sports equipment (e.g., rackets). We also conduct the generalization test on PASCAL VOC 2012 validation set which is more difficult to overfit on. In addition, MS COCO validation set which contains lots of small challenging annotated objects is also used for this evaluation.

For ILSVRC 2013 evaluation, we plot several recall curves in Figure 4.13. Here we include the MultiBox method from google to compare our FCN method with other CNN-based object proposal methods. Since MultiBox only produces 800 proposals for each image, we set the number of proposals for MultiBox as 800 in Figure 4.13(a). From Figure 4.13(a), we find that MultiBox achieves high recall at low IoU thresholds (i.e.,

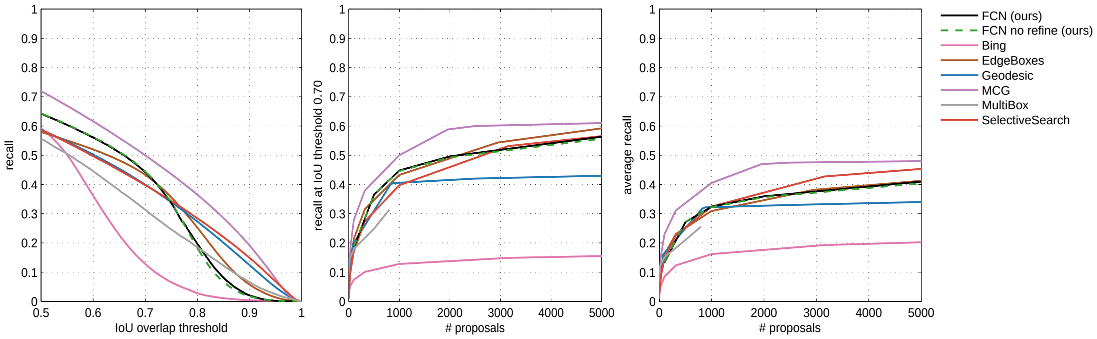


(A) 1000 proposals per image

(B) Recall at 0.7 IoU

(C) Average recall (IoU between (0.5, 1))

Figure 4.14: Recall versus IoU threshold on PASCAL VOC 2012 validation set.



(A) 1000 proposals per image

(B) Recall at 0.7 IoU

(C) Average recall (IoU between (0.5, 1))

Figure 4.15: Recall versus IoU threshold on MS COCO 2014 validation set.

0.5<IoU<0.55) but also decreases fast with the increasing of IoU threshold. From Figure 4.13(b), it is seen that MultiBox almost keeps performance as good as the state-of-the-art at 0.7 IoU threshold with a very limited number of proposals. In terms of average recall, MultiBox also shows its superiority when generating very few proposals (less than 800 per image). However, due to the limitation of its maximum number (i.e., 800 per image) of proposals, MultiBox cannot boost its recall further by generating more proposals. To summarize, MultiBox is able to roughly localize the objects with a small number of proposals. As for our FCN method, the overall trend of the recall remains consistent with that on the PASCAL VOC 2007. Specifically, our approach almost keeps the same recall as the best method, i.e., Edge Boxes across a broad range of proposal numbers (see Figure 4.13(b)). Figure 4.13(a) demonstrates that the recall of our method is still competitive across a wide range of IoU thresholds (from 0.5 to 0.7). In terms of AR, our approach is slightly worse than selective search, which achieves the highest AR. Please note that we also directly use the publicly released RPN model trained on PASCAL VOC in the generalization evaluation here. It is observed that RPN does not perform as well as on PASCAL VOC 2007. This may be because object class information is utilized when training the layers of RPN which are shared with class-specific detectors on PASCAL VOC. Therefore, the generalization to ILSVRC may be influenced by the class-aware training of RPN on PASCAL VOC.

Figure 4.14 shows the results of all the methods on PASCAL VOC 2012 validation set. The overall trends of all the methods are consistent with those on PASCAL VOC 2007 test set. Benefited from similar visual appearance and the same categories of PASCAL VOC 2007 and PASCAL VOC 2012, the proposed FCN method keeps similar good performance with that on PASCAL VOC 2007 test set, which is better than on ILSVRC 2013 validation set. The poor generalization ability from ILSVRC to PASCAL VOC of MultiBox results in similar inferior results to those on PASCAL VOC 2007 test set.

As for MS COCO 2014, a similar set of recall figures are shown in Figure 4.15. Different from the PASCAL VOC and ILSVRC 2013, it is found that MCG is the best one among all the evaluation cases. Our method shows a similar trend with the previous two benchmarks while all the recalls are lower than the best method. We attribute the difference to different statistics of the datasets, especially the different size distributions

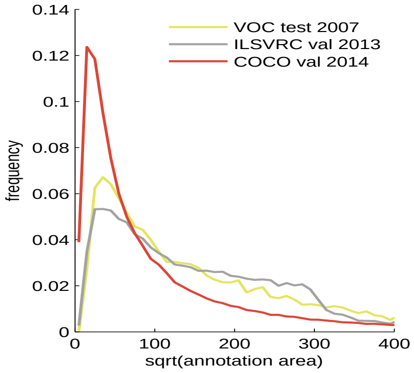


FIGURE 4.16: Comparison of the distribution of the sizes of the groundtruth objects among all considered datasets: PASCAL VOC 2007 test set, ILSVRC 2013 validation set and MS COCO 2014 validation set.

of objects (see Figure 4.16). As can be seen, MS COCO 2014 contains a large proportion of small objects. This is challenging for *Window scoring methods* as they need add more small scales to avoid missing the small groundtruth objects, which will lead to a higher chance of false positives and much higher computation cost.

Considering the significantly different statistics of MS COCO 2014, based on the above results on ILSVRC 2013, PASCAL VOC 2012 and MS COCO 2014, no significant overfitting towards the PASCAL VOC categories is found in our approach. In other words, the proposed approach has learned a generic notion of objectness and can generalize well to the unseen categories on the whole.

4.1.6 Summary

In this chapter, we utilize fully convolutional network (FCN) to generate object proposals in images. The novel high-level semantic objectness concept produced by FCN enables more accurate judgement on whether a patch contains an object or not. Moreover, proposals produced according to their high-level objectness scores are more stable when faced with image perturbation compared to low-level based methods. Both advantages of our proposals benefit the object recall and detection mean avearge precision. In addition, the novel localization way which directly maps the output neuron in the objectness map to its receptive field in the image does not involve any coordinates regression and shows to be more effective. Apart from this, a proper setting of the multiscale scheme is also critical. Although crossing many scales means a higher chance to localize the objects precisely, it may also bring more false positive objects and higher computation cost. We finalize the setting by fixing α as 0.65 after the tradeoff between recall and speed. Finally, the generalization of our model to unseen categories is also evaluated and validated to ensure that the network can be used to locate generic objects in images, which should be meaningful in real-world applications.

It should be mentioned that the proposed FCN method does not perform well in finding very small objects (i.e., containing less than 500 pixels). This is because of the inherent weakness of *Window scoring methods* that they need smaller scales to find s-mall groundtruth objects, but have a higher chance to have false positives and higher computation cost at the same time.

Although our FCN method focuses on dealing with the static image object proposal problem, it can also be extended to dynamic video sequences. As is well known, optical flow is the most widely used feature to describe the motion information in videos. It is possible to combine the optical flow map with the 3-channel RGB information together as the FCN input for each frame in the videos. Trained on the samples with 4-channel input, the FCN is expected to gain stronger power in finding moving objects in the videos. In the future, we will do research considering the motion information as input for video sequences.

In future work, we will also continue using the semantic objectness obtained by CNN to solve segment proposal problems as the segment proposal provides more precise information about the locations and the shape of the objects of interest.

4.2 Scale-aware Pixel-wise Object Proposal Networks

4.2.1 Overview

In recent years, object proposal has become crucial for modern object detection methods as an important pre-processing step [11, 21, 127]. It aims to identify a small number (usually at the order of hundreds or thousands) of candidate regions that possibly contain class-agnostic objects of interest in an image. Compared with the exhaustive search scheme such as sliding windows [106], object proposal methods can significantly reduce the number of candidates to be examined and benefit object detection in following two aspects: they can reduce computation time and allow for applying more sophisticated classifiers.

Most of existing object proposal methods can be roughly divided into two categories: the classic low-level cues based ones and the modern convolutional neural network (CNN) based ones. The former category of methods mainly exploit low-level image features, including edge, gradient and saliency [113, 114, 125, 126, 132, 136] to localize regions possibly containing objects. Typically they either follow a bottom-up paradigm *e.g.*, hierarchical image segmentation [125, 134] or examine densely distributed windows [113, 114]. However, it is difficult for them to balance well between localization quality and computation efficiency – they cannot provide object proposals of high quality without incurring expensive computational cost. On the other hand, CNN-based methods either directly predict the coordinates of all the objects in an image [139] or scan the image with a fully convolutional network (FCN) [145] to find the regions of high objectness[1].

[1] "Objectness" measures membership to foreground objects *vs.* background

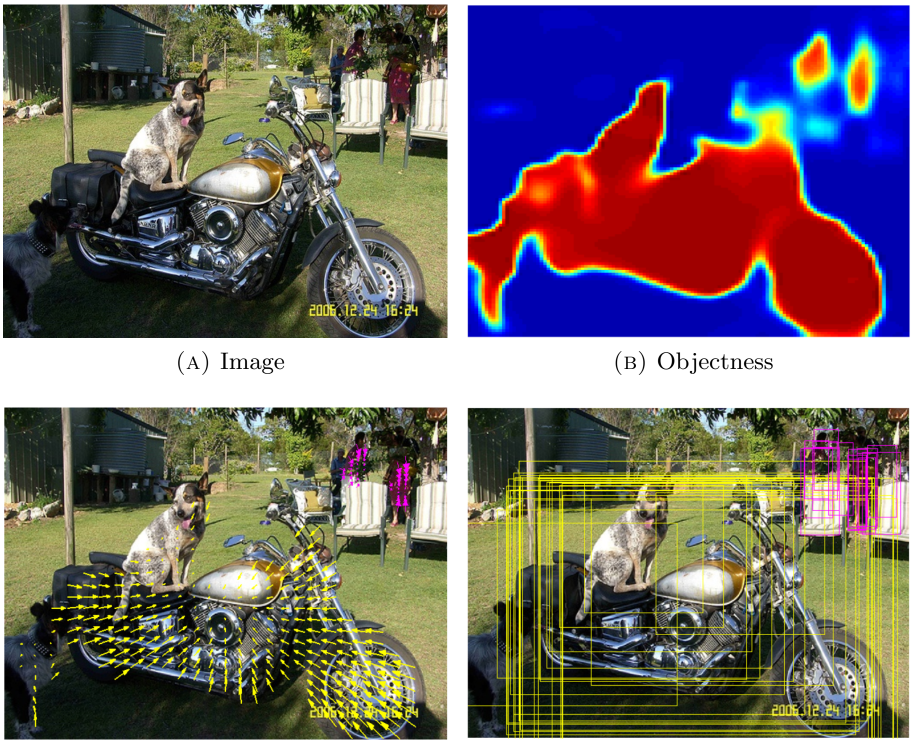
Although they can achieve high recall rate w.r.t. relatively loose overlap criteria, *e.g.* intersection over union (IoU) with a threshold value of 0.5, this type of methods usually fails to provide high recall rate under more strict criteria (*e.g.* IoU > 0.7), suggesting their poor localization quality.

Ideally, a generic object proposal generator should offer the following desired features: high recall rate on objects of various categories with only a few proposals, good localization quality for each specific object instance and high computation efficiency. In this work, we make an effort to develop the object proposal method toward these targets.

Our method is motivated by a statistical study on the scale of objects in a collection of natural images. As shown in Figure 4.18, we plot the distribution of objects with varying scales (measured by number of pixels) from the training and validation sets of the PASCAL VOC detection benchmark [129]. From the figure, one can observe that the objects of small scales (less than 2,000 pixels) actually dominate the distribution. Similar observations also hold in the ILSVRC 2013 and 2014 benchmark [121]. Unfortunately, most of existing methods perform poorly in localizing objects of such small sizes, in terms of the best overlap[2]. Based on these empirical observations, we argue that the quality of small objects localization is one main bottleneck for further improving the recall rate and average best overlap (ABO) for object proposal methods. Therefore, we focus on tackling such a challenging problem in this work.

In particular, we develop a novel CNN based object proposal method which contains a pixel-wise object proposal network, sharing the similar spirit with object segmentation

[2] Best overlap of a particular ground-truth object is defined as the maximal intersection over union (IoU) among all the given proposals w.r.t. this object. Throughout the chapter, Average Best Overlap (ABO) is obtained by averaging the best overlap of all the ground-truth objects



(A) Image

(B) Objectness

(C) Offsets to object center

(D) Object proposals

Figure 4.17: Examples of predicted "objectness map" in (b), "offsets to object center" after weighted combination in (c) and "object proposals" in (d). "Offsets to object center" is indicated by the arrows pointing to $((x^i_{\text{min}}+x^i_{\text{max}})/2-x_i, (y^i_{\text{min}}+y^i_{\text{max}})/2-y_i)$ for each pixel i. Yellow and magenta colors in "offsets to object center" and "object proposals" indicate that the prediction is from a pixel with large-size confidence higher than 0.5 or less than 0.5. In the figure, only the predictions for the pixels with objectness higher than 0.5 are shown.

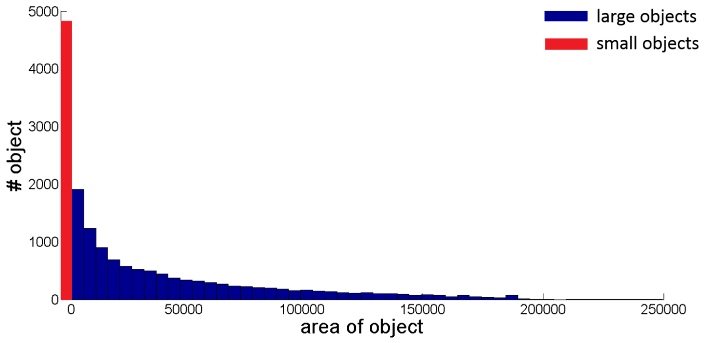


Figure 4.18: Distribution of objects w.r.t. their areas (measured by number of contained pixels) on the PASCAL VOC 2012 benchmark. It can be seen that small objects occupy a large proportion of the collection.

networks [46, 150]. Here the "pixel-wise" refers to: for *every* pixel in an image, our proposed network model will predict a bounding box of the object containing this pixel. Such a pixel-level comprehensive object proposal strategy fully exploits the available annotations for object segmentation[3] and substantially improves the quality of object proposals through enhancing the opportunities of accurately hitting the ground-truth object. As the receptive field of each pixel in CNN is a local region around the pixel, directly predicting the coordinates of the bounding box is challenging due to the various spatial displacements of objects. We thus propose to predict the *offset* of the bounding box w.r.t this pixel, for each pixel.

We then take a further step to focus on enhancing the localization precision for small-scale objects. We propose a new *scale-aware* strategy for object proposal, which is inspired by the divide-and-conquer philosophy. Specifically, we train two independent networks, each of which predicts bounding box coordinates for objects at different scales (small or large). Then for each pixel, we will obtain two object proposals for choice. To adaptively fuse them, we introduce another object confidence network. The network consists of two branches – one for predicting objectness confidence and the other one for weighting the large-/small-size[4] object localization networks. The objectness branch predicts the likelihood of each pixel coming from an object of interest, and the large-/small-size weighting branch trade-offs the contribution of the large-size and small-size networks to final prediction, by predicting the probability of the pixel belonging to an object of a large size. In the training phase, the size of an object can be easily inferred from its annotated segmentation mask, which is used for training the proposed

[3] The segmentation annotations can be readily collected from many public benchmark datasets.

[4] Throughout the chapter, we use "large-size network"/"small-size network" to refer to a localization network trained specifically for localizing objects of large/small sizes.

network. For a new image without annotation, both the large-size and small-size object localization networks will predict the bounding box coordinates which are combined according to the weights from the confidence network. An overview of the proposed network model is presented in Figure 4.17.

Therefore, the scale-aware coordinates prediction can achieve outperforming localization quality for a wide range of object sizes as for various object sizes, the final result can always considers and fuses the bounding boxes predicted by two localization networks robustly based on a reliable large-/small-size weighting mechanism.

To further improve the performance of localizing small objects, we employ a multi-scale strategy for object proposal on a new image. This is inspired by the observation that by enlarging the challenging small object into a larger one, the coordinates prediction error of the small object will be scaled down, as in the case of zooming in on a small object to obtain a clearer view for humans or cameras. Finally, a superpixel based bounding box refinement operation is applied to fine tune the proposals.

In short, we make the following contributions to object proposal generation. Firstly, we introduce a segmentation-like pixel-wise localization network to densely predict the object coordinates for each pixel. Secondly, we develop a scale-aware object localization strategy which combines the predictions from a large-size and a small-size network with a weighting mechanism to boost the coordinates prediction accuracy for a wide range of object sizes. Thirdly, we conduct extensive experiments on the PASCAL VOC 2007 and ILSVRC 2013 datasets. The results demonstrate that our proposed approach outperforms the state-of-the-art methods by a significant margin, verifying the superiority of the proposed scale-aware pixel-wise object proposal network.

The remainder of this chapter is organized as follows. In Section 4.2.2, we review the related works on object proposal generation. In Section 4.2.3, we describe our scale-aware pixel-wise localization network. After showing the experimental results in Section 4.2.4, we draw the conclusion in Section 4.2.5.

4.2.2 Related Work

The existing object proposal generation methods can be classified into three types: *window scoring methods*, *segment grouping methods* and *CNN-based methods*.

Window scoring methods design different scoring strategies to predict the confidence of containing an object of interest for each candidate window. Generally, this type of methods first initializes a set of candidate window regions across scales and positions in an image, and then sorts them with a scoring model and selects the top ranked windows as proposals. Objectness [110] selects the initial proposals from the salient regions in an image and sorts them based on multiple low-level cues, such as color, edges, location size, etc. [136] proposed a cascade of SVMs trained on gradient features to estimate the objectness. BING [113] trains a simple linear SVM on image gradients and applies it in a sliding window scheme to find the highest scored windows as object proposals. Edge Boxes [114] is also performed in a sliding window manner, but relies on a carefully hand-designed scoring model which sums the edge strengths fully inside the window. Window scoring methods are usually computationally efficient as they do not output segmentation masks for the proposals. However, it seems difficult for them to achieve high recall rate under high overlap criteria (*e.g.* IoU > 0.7), which suggests the poor localization quality. This can probably be attributed to the discrete sampling of the sliding windows which are all in the pre-defined scales and positions.

Table 4.5: Details of DeepLab-LargeFOV network architecture.

layer	#channel	kernel size	stride	zero-padding size	hole size	training map size	receptive field size	#weight
input image	3	-	-	-	-	513*513	435*435	-
conv1_1	64	3*3	1*1	1*1	-	513*513	433*433	1.75K
conv1_2	64	3*3	1*1	1*1	-	513*513	431*431	36K
pool1	64	3*3	2*2	1*1	-	257*257	215*215	-
conv2_1	128	3*3	1*1	1*1	-	257*257	213*213	72K
conv2_2	128	3*3	1*1	1*1	-	257*257	211*211	144K
pool2	128	3*3	2*2	1*1	-	129*129	105*105	-
conv3_1	256	3*3	1*1	1*1	-	129*129	103*103	288K
conv3_2	256	3*3	1*1	1*1	-	129*129	101*101	576K
conv3_3	256	3*3	1*1	1*1	-	129*129	99*99	576K
pool3	256	3*3	2*2	1*1	-	65*65	49*49	-
conv4_1	512	3*3	1*1	1*1	-	65*65	47*47	1.13M
conv4_2	512	3*3	1*1	1*1	-	65*65	45*45	2.25M
conv4_3	512	3*3	1*1	1*1	-	65*65	43*43	2.25M
pool4	512	3*3	1*1	1*1	-	65*65	41*41	-
conv5_1	512	3*3	1*1	2*2	2*2	65*65	37*37	2.25M
conv5_2	512	3*3	1*1	2*2	2*2	65*65	33*33	2.25M
conv5_3	512	3*3	1*1	2*2	2*2	65*65	29*29	2.25M
pool5	512	3*3	1*1	1*1	-	65*65	27*27	-
pool5a	512	3*3	1*1	1*1	-	65*65	25*25	-
fc6	1024	3*3	1*1	12*12	12*12	65*65	1*1	4.5M
fc7	1024	1*1	1*1	-	-	65*65	1*1	1M

Segment grouping methods are usually initialized with an oversegmentation to obtain superpixels for an image. Then different merging strategies are adopted to group the similar segments hierarchically to generate the object proposals of all scales. Generally, they follow a bottom-up scheme which relies on diverse low-level image cues including color, shape and texture. For example, Selective Search [125] iteratively merges the most similar segments to form proposals based on several low-level cues. Randomized Prim [132] learns a randomized merging strategy based on the superpixel connectivity graph. Multiscale Combinatorial Grouping (MCG) [134] utilizes multi-scale hierarchical segmentations based on the edge strength and the obtained proposals are then ranked using features including size, location, shape and contour. Geodesic object proposal [135] also depends on superpixels as initialization, and then computes a geodesic distance transform and selects certain level sets of the distance transform as object proposals. [151] proposes learning conditional random field (CRF) in multiscales to classify the superpixels into objects or background. Generally, compared with *window scoring methods*, *segment grouping methods* achieve more consistent and acceptable recall under both loose and strict overlap criteria, indicating a better localization ability. Nevertheless, these methods produce high quality proposals often by multiple segmentations in different scales and color spaces, thus are quite computationally expensive and time-consuming.

CNN-based methods follow the great success of Convolutional Neural Network in other vision tasks [6, 13, 152]. They leverage the powerful discrimination ability of Convolutional Neural Network (CNN) to extract visual features as inputs of other techniques to produce proposals or directly regress the coordinates of all the object bounding boxes in an image. MultiBox [139] trains a network to directly predict a fixed number of proposals and their confidences in an image and ranks them with the obtained confidences. RPN [145] uses a Fully Convolutional Network (FCN) to densely generate the proposals in each local patch based on several pre-defined "anchors" in the patch. DeepProposal [153] hunts for the proposals in a sliding window manner by using the CNN features from the final to the beginning layers and training a cascade of linear classifiers to obtain the highest scored windows. Current CNN-based methods typically achieve high recall with only a small number (usually $< 1{,}000$) of proposals, under loose overlap criteria (*e.g.* 0.5<IoU<0.6). But similar to window scoring methods, they can hardly achieve high recall rate under more strict overlap criteria (*e.g.* IoU > 0.7). To improve the object proposal localization quality, different from them, our approach predicts the object locations in a pixel-wise manner so that we have much more chances to localize each object with high precision. This also takes the full advantage of the publicly available segmentation masks annotations. This is similar to [154] which deals with object detection task in the object coordinates prediction part. In addition, our scale-aware prediction strategy provides adaptive accurate prediction for both large-size and small-size objects, which also distinguishes our method from others.

4.2.3 Scale-aware Pixel-wise Proposal Network

The proposed Scale-aware Pixel-wise Object Proposal Network (SPOP-net) is based on a pixel-wise segmentation-like object coordinates prediction network, and includes a scale-aware localization mechanism for predicting the coordinates of objects of different sizes. In addition, a multi-scale prediction strategy is employed during testing to boost the small objects localization. Finally, a superpixel boundary based proposal refinement is introduced to further improve the proposal precision. We will elaborate all the components of SPOP-net in this section.

4.2.3.1 Pixel-wise Localization Network

The proposed Scale-aware Pixel-wise Object Proposal Network (SPOP-net) takes an image of *any* size as input and predicts the location of the object w.r.t. each pixel in the image. More concretely, for each pixel, SPOP-net predicts the normalized coordinates of the bounding box of the object that contains the pixel. The predictions from the background pixels make no sense and will be ranked behind due to low objectness scores they obtain, thus making no difference to the recall performance of top-ranked proposals, which will be detailed later. In this subsection, we first explain the architecture of SPOP-net and then elaborate on how to train and apply the SPOP-net.

Architecture Our SPOP-net is built upon a pre-trained DeepLab-LargeFOV segmentation network [150]. Its architecture is shown in Table 4.5. The receptive field of our localization network in the last layer is 435 × 435. This large receptive field enables SPOP-net to "see" a large region of the image in its last layer and predict the object bounding boxes effectively.

Training For each pixel, the pixel-wise localization network aims to predict the bounding box coordinates $\mathbf{t} = (x_{\min}/w, y_{\min}/h, x_{\max}/w, y_{\max}/h)$ of the object that contains this pixel. Here $(x_{\min},\ y_{\min})$ and $(x_{\max},\ y_{\max})$ denote the coordinates of the top-left and bottom-right corners of the object bounding box containing the pixel; h and w represent the height and the width of the image plane respectively. Therefore, for a single object, all the pixels inside this object are given the same ground-truth values $(x_{\min}/w, y_{\min}/h, x_{\max}/w, y_{\max}/h)$. We train the pixel-wise localization network to minimize the following localization error $\mathcal{L}$ that is proportional to the Euclidean distance between the predicted coordinate vector $\mathbf{t}_i$ and the ground-truth coordinate vector $\mathbf{t}_i^*$ for all the foreground pixels. The loss function $\mathcal{L}$ is defined as

$$\mathcal{L} = \sum_i p_i^* \|\mathbf{t}_i - \mathbf{t}_i^*\|^2, \tag{4.8}$$

where $\mathbf{t}_i$ is the predicted 4-d object coordinate vector, and p_i^* is a binary variable indicating whether the pixel i is a foreground one: it takes 1 if the pixel i is from a foreground object and 0 otherwise. Such a filtered loss (through p_i^*) enables the localization network to concentrate on localizing foreground objects without being distracted by background pixels in the training phase. In the practical implementation, as the final layer has smaller size than the input image, we resize the ground-truth coordinate map to the same small size as the final layer.

However, due to the possible spatial displacement (*e.g.* two exactly the same objects could appear at different locations in an image), accurately predicting the absolute object bounding box coordinates is difficult. It is because these two objects have the same visual input for the model, but their locations the model needs to learn to predict are totally different. To solve this issue, for each pixel, we change its learning targets from the absolute object bounding box coordinates to the offsets from the pixel to the object bounding box. E.g. for object bounding box coordinate x_{min}/w, we change the target from x_{min}/w to $(x_{\text{min}} - x_{\text{self}})/w$, here x_{self} is the x coordinate of the pixel itself. Changing the coordinates to offsets can be conveniently achieved by element-wisely summing the output of the 2nd last layer and the spatial coordinate map (x or y values of all the pixels themselves). Then the absolute object bounding box coordinates can be used as learning targets for the final layer. In this way, applying the absolute coordinates learning targets to the final layer is equivalent to applying the following object coordinate offsets to the 2nd last layer.

$$\left(\frac{x_{\text{min}} - x_{\text{self}}}{w}, \frac{y_{\text{min}} - y_{\text{self}}}{h}, \frac{x_{\text{max}} - x_{\text{self}}}{w}, \frac{y_{\text{max}} - y_{\text{self}}}{h}\right)$$

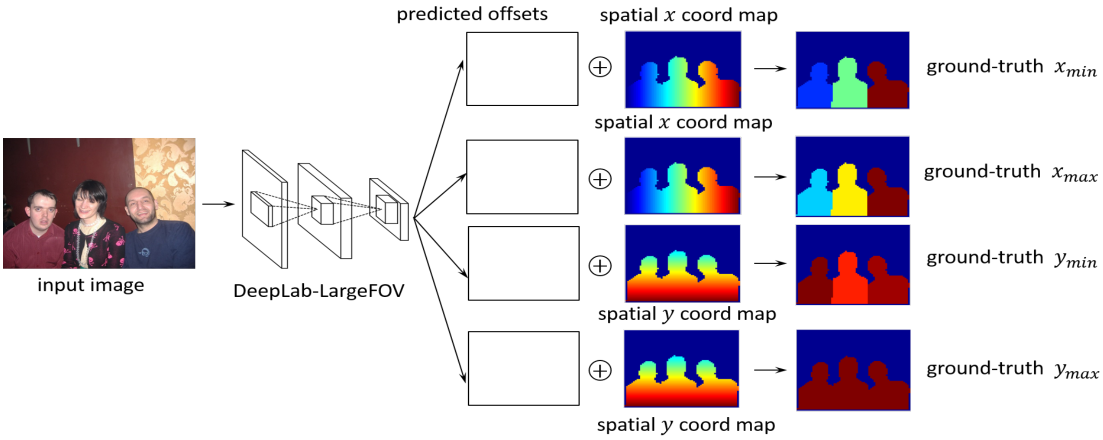


Figure 4.19: An image passes through several layers to obtain four maps in the second last layer. In the second last layer, two maps are element-wise summed with spatial x coord map to produce the final prediction for the $x_{\min}$ and $x_{\max}$ of the corresponding objects for all the pixels, and the other two maps are element-wise summed with spatial y coord map to produce the final prediction for the $y_{\min}$ and $y_{\max}$ of the corresponding objects for all the pixels. In this way, the four maps in the second last layer in our fully trained network actually predict the four offsets between each pixel position and its corresponding object location, which makes it easier for the network to predict the object coordinates in the final layer. Different colors in the ground-truth maps and spatial coord maps represent different values. Note that we only show the foreground regions of spatial x and y maps for better view.

Then we can directly obtain the absolute object proposal coordinates from the predictions of the final layer. After obtaining the output map from the final layer having a smaller size than the input image, all the subsequent procedures (*e.g.* refinement, ranking and NMS) are only based on the output map of smaller size. Because we just leverage pixel-level prediction of proposals for having higher chance to hit the ground-truth objects accurately instead of doing pixel-level classification as DeepLab. If resizing the smaller output map back into the original size, the subsequent refinement, ranking and NMS steps will bring much higher computation burden but not significant performance improvement.

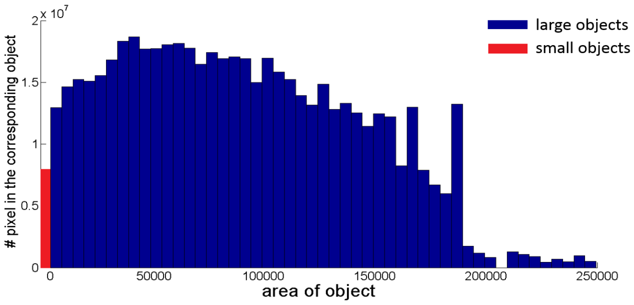


FIGURE 4.20: The distribution of all the pixels w.r.t. the area of the object each pixel belongs to. It is shown that although the number of small objects is large according to Figure 4.18, the number of pixels belonging to small objects is still small, leading to the unbalanced pixel-level training samples.

4.2.3.2 Scale-aware Localization

A fully trained pixel-wise localization network can predict the coordinates of object bounding boxes w.r.t. each pixel from an image. However, a single network model may not be able to well handle all the annotated objects that have quite diverse sizes and only offers inferior localization performance for objects of small sizes. To verify this point, we conduct the following preliminary experiments to evaluate the errors of bounding box prediction for large and small objects, using a single pixel-wise localization network trained on the annotated objects of all sizes. The evaluation results are shown in Table 4.6.

From Table 4.6, one can observe that the network trained on all the objects of different sizes produces an L_2 error for small objects that is about 3 to 6 times larger than the error for large objects. This demonstrates the poor localization ability of a single network model for small objects.

The difficulty of accurately localizing both large and small objects using a single network arguably lies in handling the highly diverse offsets of large and small objects. Apart from

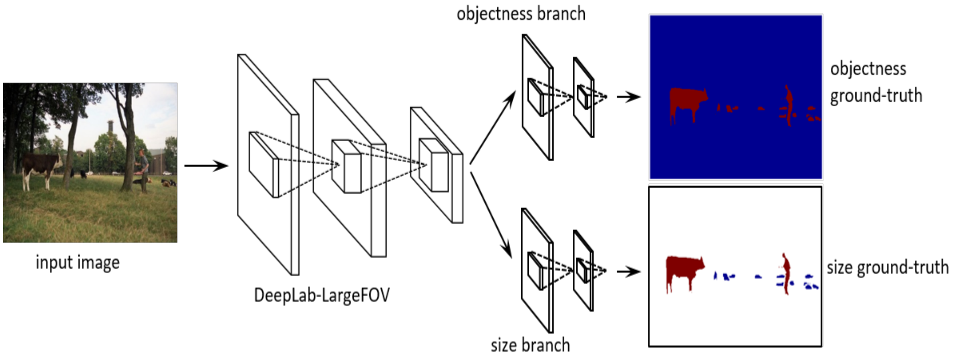


FIGURE 4.21: Illustration of the "confidence network" which bifurcates into two branches to perform foreground/background classification and large/small object classification after all the layers of "DeepLab-LargeFOV" network. Both the sub-networks contain two convolution layers with 3 × 3 kernel size. The first layer outputs 1,024 feature maps while the second (also the last) layer produces two maps showing the final confidence of their own task. In the ground-truth map of the foreground/background classification branch, red pixels are in foreground objects while blue pixels are in background. In the ground-truth map of the large/small object classification branch, red pixels are in large objects, blue pixels are in small objects and white pixels are background pixels thus are not considered during training.

TABLE 4.6: L_2 errors of normalized coordinates prediction for both large ($\geq$ 2,000 pixels) and small objects ($<$ 2,000 pixels) in VOC 2007 testing set, based on the network trained on the annotated objects of all sizes.

	large objects	small objects
$x_{\min}^{err}/w$	0.0090	0.0270
$y_{\min}^{err}/h$	0.0064	0.0160
$x_{\max}^{err}/w$	0.0080	0.0412
$y_{\max}^{err}/h$	0.0088	0.0476

this, another difficulty comes from the extremely unbalanced training samples between the pixels from large and small objects. Such imbalance leads to the fact that training error of large objects dominates the training loss to minimize.

Also, we empirically verify the sample imbalance through statistics on the pixel-level distribution of the annotations in terms of the area of the object (see Figure 4.20) since our pixel-wise localization network is trained on pixel-level annotations.

To improve the localization accuracy for small objects, we propose a *scale-aware* localization strategy. Roughly, in the scale-aware strategy, two localization networks are trained – which share the same architecture – with two non-overlapped subsets of the objects. The large-size network is only trained on the pixels belonging to large object-s and the small-size network is only trained on the pixels belonging to small objects. The loss function to be optimized for the large-size and small-size network are shown in Eqn. (2) and Eqn. (3) below respectively:

$$L_l = \sum_i l_i^* \|\mathbf{t}_i - \mathbf{t}_i^*\|^2 \tag{4.9}$$

$$L_s = \sum_i s_i^* \|\mathbf{t}_i - \mathbf{t}_i^*\|^2 \tag{4.10}$$

where l_i^* and s_i^* are binary indicators showing whether the pixel i belongs to a large object or a small object. The effectiveness of training such scale-aware networks is validated by evaluating the L_2 errors of small objects location prediction with the small-size network. See Table 4.7. During the testing phase, the two networks work simultaneously to output their own prediction for an image. Then, the predictions from two networks are combined with an adaptive weighting scheme.

The weight is output by a network trained for classifying large and small objects pixel-wisely and the weight is equal to the confidence of the pixel belonging to a large object obtained in the last layer of the network. Such a classification network is termed as “confidence network”.

The structure of the confidence network is illustrated in Figure 4.21. Apart from the large/small classification branch, the confidence network also outputs the objectness

Table 4.7: L_2 errors of normalized coordinates prediction for small objects ($<$ 2,000 pixels) in VOC 2007 testing set, based on the network trained only on small objects.

	small objects
$x_{\min}^{err}/w$	0.00058
$y_{\min}^{err}/h$	0.00040
$x_{\max}^{err}/w$	0.00068
$y_{\max}^{err}/h$	0.00086

confidence in another branch aiming to classify all the pixels into two categories, *i.e.*, foreground pixels and background pixels.

In the confidence network, the two branches share the convolutional features in the lower layers. The last feature maps shared are then fed into the two branches. The intuition for dividing the confidence network into two branches at the higher layer is that for different tasks, the low-level features are usually common and can be shared [155], while the semantically high-level features extracted by the higher layers may be totally different for different tasks. For example, the foreground/background classification task prefers the common features that are insensitive to different sizes of objects, but the large/small classification task aims to extract the discriminative features between large and small objects. The large receptive field (*i.e.* 435 × 435) in the last layer of the "confidence network" provides a sufficient large view enabling the prediction of both foreground/background and large/small classifications.

The objective function to be optimized during training the confidence network is a multi-task cross-entropy loss:

$$\begin{aligned} L = \sum_i p_i^* \log(p_i) + (1 - p_i^*) \log(1 - p_i) + \\ \sum_i p_i^* (z_i^* \log(z_i) + (1 - z_i^*) \log(1 - z_i)). \end{aligned} \tag{4.11}$$

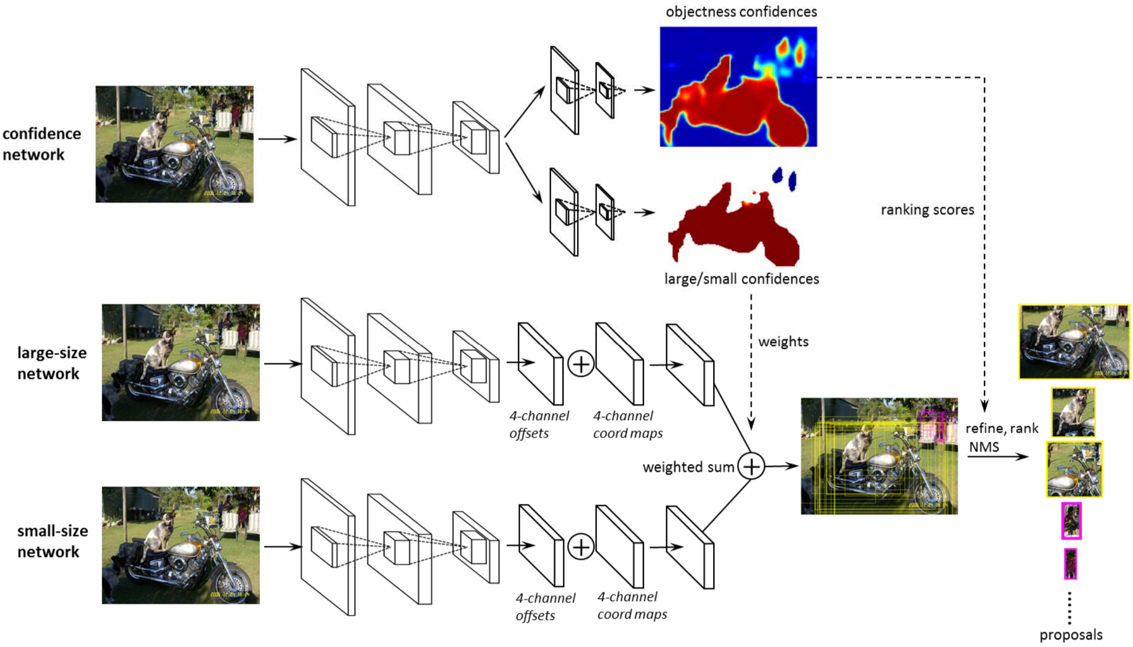


Figure 4.22: Overview of our approach. An image passes the confidence network to obtain the pixel-wise objectness confidences and large/small size confidences (red color represents higher values, *e.g.*, high objectness and high large size confidences). The image also passes two localization networks to obtain the predicted pixel-wise large and small object coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, respectively. The feed-forward computation of the three networks are independent and can be run in parallel. Then the final predicted object coordinates are the sum of the predictions by large-/small-size networks weighted by the large/small size confidences. Using the objectness confidences as ranking scores, the final proposals are produced after refinement, ranking and NMS. For multi-scale inference, all the above procedures are run for the enlarged input image as well. Then the proposals obtained by both the original and enlarged scales are mixed in the ranking and NMS.

Here p_i^* and p_i are the ground-truth label of the foreground/background classification and the predicted confidence of being a foreground pixel for pixel i, respectively. z_i^* and z_i are the ground-truth label of the large/small object classification and the predicted confidence of being contained in a large object for pixel i, respectively. Note that the second term is only activated when p_i^* equals 1, indicating that the pixel belongs to a foreground object. After the large object confidence z_i for the pixel i is obtained, the final predicted coordinates of the object it belongs to are the weighted sum of the predictions by the large-size and small-size networks as follows.

$$\mathbf{t}_i = z_i \mathbf{t}_{l,i} + (1 - z_i)\mathbf{t}_{s,i} \tag{4.12}$$

where $\mathbf{t}_{i,l}$ and $\mathbf{t}_{i,s}$ are the predictions by the large-size and the small-size network respectively. Then we treat the predicted object coordinates by each pixel as an initial proposal to be passed to the later proposal refinement and non-maximum suppression (NMS) steps to obtain the final object proposals.

4.2.3.3 Multi-scale Inference

To further enhance the accuracy of small objects localization, we propose to employ a multi-scale prediction strategy in the testing phase. The motivation is quite straightforward: by enlarging the challenging small object into a larger one, the coordinates prediction error of the small object will be scaled down, which is similar to zooming in on a small object to improve the localization accuracy. At the enlarged scale, all the proposals in the enlarged image will be mapped back to their corresponding positions at the original scale.

Therefore, given a testing image, in addition to its original scale, we resize it into a larger scale and run the prediction process as well. Specifically, both on the original scale and the enlarged scale, we simultaneously run the two localization networks (i.e. large-size and small-size) and the confidence network, and combine the both location

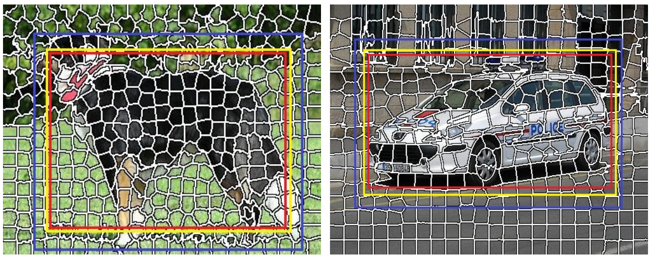


Figure 4.23: Illustration of proposal refinement using superpixel boundary based expansion and shrinkage. Yellow boxes represent initial proposals; red boxes and blue boxes are the corresponding refined proposals after shrinkage and expansion respectively. In the left example, expansion finds a closer box to the ground-truth, but in the right example, shrinkage helps the proposal get close to the ground-truth.

predictions weighted by the large object confidence z_i of its own scale. As all the feed-forward computation of the networks is independent and can be performed in parallel, the computation time cost can remain relatively low.

4.2.3.4 Proposal Refinement

We then refine the two sets of proposals obtained in both original and enlarged scales. An inherent weakness for object localization by regressing the four coordinates with CNN is that the objectness and coordinates ground-truths only permit determining the most discriminative foreground windows. Therefore, even though the windows decided by the localization networks are likely to overlap with target objects, it cannot be ensured that they are able to delineate object boundaries well.

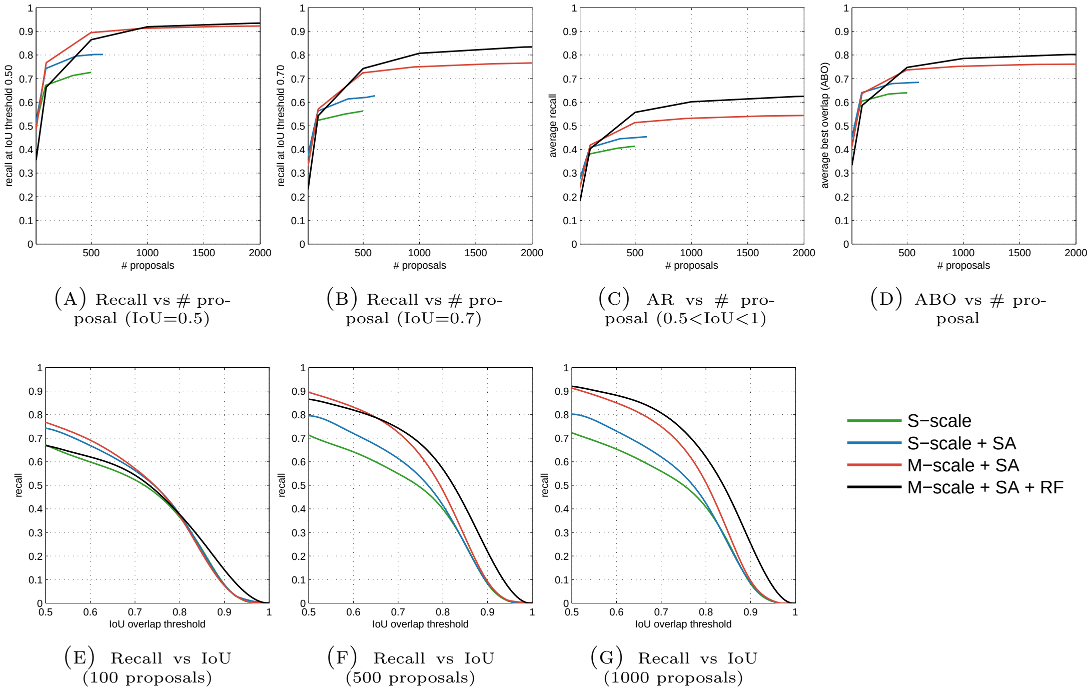
To take object boundaries into consideration, we utilize a superpixel boundary based window refinement method, similar to [149]. The main idea is to expand or shrink the proposals to align the four sides of the proposals with the boundaries of the superpixels better. The reason for using superpixels is that the boundaries of superpixels are informative indicators of object boundaries and superpixels can be generated efficiently with off-the-shelf algorithms (*e.g.* SLIC [131]). Specifically, for each proposal, we generate two versions of refined proposals, *i.e.* the minimum bounding rectangle of all the superpixels entirely inside this proposal and the minimum bounding rectangle of all the superpixels entirely inside this proposal or straddling this proposal (see Figure 4.23). As illustrated in Figure 4.23, expansion and shrinkage offer two possible ways of getting close to the ground-truth box for the proposals with different location biases to the ground-truth. Therefore, we pass all the two versions of refined proposals as well as the initial proposals to the later proposal ranking and NMS processing.

In the stage of proposal ranking , we sort all the proposals (including the initial and the two refined ones in both original and enlarged scale) by their objectness confidence p_i. Recall p_i is the output from foreground/background classification branch of the confidence network. For each initial proposal, its two versions of refined proposals are assigned with the same objectness confidence p_i as itself. Finally, the standard non-maximum suppression (NMS) is employed to remove the highly overlapped redundant proposals.

4.2.4 Experiments and Discussion

4.2.4.1 Experimental Setups

The proposed Scale-aware Pixel-wise Object Proposal Network (SPOP-net) is trained on the SBD annotations [156] of PASCAL VOC 2012 trainval set, which provides 11,355 images with fine segmentation masks annotations. We manually label the objects containing more than 2,000 pixels as large objects and those containing less than 2,000



(A) Recall vs # proposal (IoU=0.5)

(B) Recall vs # proposal (IoU=0.7)

(C) AR vs # proposal (0.5<IoU<1)

(D) ABO vs # proposal

(E) Recall vs IoU (100 proposals)

(F) Recall vs IoU (500 proposals)

(G) Recall vs IoU (1000 proposals)

FIGURE 4.24: Recall and average best overlap (ABO) comparison between different variants. S-scale, S-scale+SA, M-scale+SA, M-scale+SA+RF denote single-scale, single scale with scale-awareness, multi-scales with scale-awareness, multi-scales with scale-aware and refinement, respectively. “SA” and “RF” denote “scale-awareness” and “refinement”, respectively.

pixels as small ones. Considering the unbalanced pixel samples when training the large-/small-size weighting branch, for each large object, we randomly sample 100 pixels in it for training to balance the number of training pixels belonging to large and small objects. Both the “confidence network” and the two localization networks are trained using the published DeepLab code [150], which is based on the publicly available Deep Learning platform Caffe [124]. The weights in the newly added layers are all initialized with a zero-mean Gaussian distribution with the standard deviation 0.01 and the biases are initialized with 0. The initial learning rate is 0.001 for the pre-trained layers in the DeepLab-LargeFOV network and 0.01 for the newly-added layers. All of them are reduced by a scale of 10 after every 20 epochs. The mini-batch size is set as 8. We train the network for about 60 epochs. The overlap threshold for NMS in our experiments is set to 0.8 for a good trade-off between the recall at low IoU thresholds (*e.g.* 0.5) and high IoU thresholds (*e.g.* 0.8). The training images are all resized to 513*513. During testing, for original scale, all the images are directly fed into the networks without any scaling; for enlarged scale, all the images are enlarged by a factor of 2.

The proposed SPOP-net is then extensively evaluated on PASCAL VOC 2007 testing set which is the most widely used in comparison of object proposal algorithms. It contains 4,952 images with annotated objects in bounding boxes. We are not able to evaluate on PASCAL VOC 2012 testing set because the ground-truths are not publicly released. Since the missed objects can never be recovered in the post-classification stage in a proposal-based object detection pipeline, object recall rate is naturally regarded as the standard evaluation metric for object proposal algorithms. Also, we evaluate the localization quality measured by Average Best Overlap (ABO). In addition, the object detection performance using our proposals in Fast-RCNN [21] detection pipeline is evaluated to validate the effectiveness of our proposals in the object detection task. Finally, we conduct the generalization ability evaluation by testing the recall rate on ILSVRC 2013 validation set using our network which is trained on PASCAL VOC 2012.

4.2.4.2 Ablation Studies

We first study the effectiveness of the four components in our method: pixel-wise localization network (basic setting), scale-aware localization, multi-scale inference and proposal refinement. Several simplified variants of the SPOP-net are tested in terms of the object recall rate on PASCAL VOC 2007 testing set. Specifically, we use the prediction only at the original scale without scale-awareness and proposal refinement as our baseline, which is referred to as single scale. Without scale-awareness, only one localization network is trained on all of the foreground pixels including both large-size and small-size ones. Then, we accumulatively add scale-awareness, multi-scale inference, proposal refinement to the baseline to see the benefits of each component. Please note that multi-scale inference here indicates the prediction at two scales, namely the original image scale and the 2-time enlarged scale.

Figure 4.24 shows the recall and average best overlap (ABO) comparisons under different scenarios between the four variants, *i.e.* single scale, single scale with scale-awareness, multi-scales with scale-awareness, multi-scales with scale-awareness and refinement. The number of proposals of S-scale and S-scale+SA are around 500 due to that most proposals can be filtered after NMS as pixel-wise localization networks generate highly overlapped proposals (see Figure 4.30). From Figure 4.24(a), 4.24(b) and 4.24(c), 4.24(e), 4.24(f) and 4.24(g), we find that both scale-awareness and multi-scale inference improve the recall under both low IoU threshold (*e.g.* 0.5) and high IoU threshold (*e.g.* 0.7). As for proposal refinement, it is found that it harms the recall under low IoU thresholds (*e.g.* 0.5) when the number of proposals is less than 500. The reason probably lies in the large number of proposals after refinement, which is 3 times as big as that before refinement. Although this increases the opportunities of getting close to the ground-truths which can boost the recall for a large number of proposals, this also causes too many duplicate proposals to concentrate on a small area, which lowers down the recall under loose IoU criteria when only requiring a small number of proposals. For average best overlap, it shows a similar trend to the recall from Figure 4.24(d), suggesting the benefits of all three components in terms of localization quality.

We then study the contributions of all the components for different object areas. Figure 4.25 presents the distributions of the detected objects of both the four variants of SPOP-net and the ground-truths w.r.t the object areas. It is found that the baseline variant, *i.e.* single scale without scare-awareness and refinement, can hit most of big objects but performs poor for small objects. Scare-aware weighted combination mechanism and multi-scale inference help improve the recall for small objects significantly, which shows the effectiveness of both the proposed scare-aware localization strategy and multi-scale inference.

To further verify the effectiveness of scale-awareness and multi-scale inference in small objects localization, we break up the SPOP-net into four building blocks, *i.e.* large-size network and small-size network in original scale, and large-size network and small-size network in enlarged scale, in order to investigate their respective contributions to the final localization. We evaluate the average best overlap (ABO) of the four building blocks for the ground-truth objects with different areas. Figure 4.26 shows the ABO versus object area curves of the four building blocks. It can be seen that when the object becomes larger, the large-size network in original scale predicts more accurate localization results. The small-size network in original scale achieves the highest ABO when the object area is around 2,000, but it also performs poorly for those too small objects. Fortunately, the small-size network in enlarged scale covers this shortage, and gives the best performance for very small objects due to the enlarged view of small objects. As for the large-size network in enlarged scale, it performs the best for those middle-size objects containing 2,000 to 20,000 pixels, serving as the bridge between the

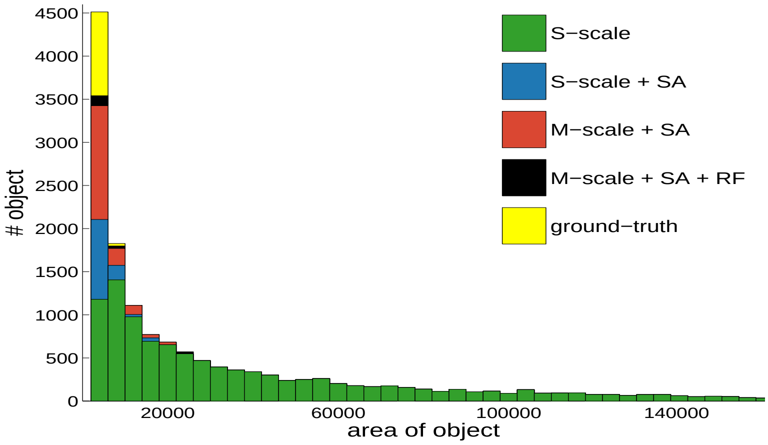


Figure 4.25: Distribution of the detected objects w.r.t. the object areas (measured by number of contained pixels) on the PASCAL VOC 2007 testing set of the four variants of the SPOP-net. The IoU threshold is 0.5. 2,000 proposals are generated for each image.

large-size network in original scale and the small-size networks in both scales. The reason for the behavior of the large-size network in enlarged scale is probably that when the small objects are enlarged, they become “large objects” such that it becomes easier for the large-size network to predict, but original large objects become even larger which cannot be covered by the receptive field, making it difficult to precisely localize them. In both original scale and enlarged scale, the result after *scale-aware* fusion can achieve the maximal ABO among the two ABOs obtained by large-size and small-size networks, validating the effectiveness of the adaptive *scale-aware* fusion strategy.

By investigating the building blocks of the proposed SPOP-net, it is found that they can complement each other in localizing the objects with different areas and ensures the SPOP-net to perform well for a wide range of object sizes.

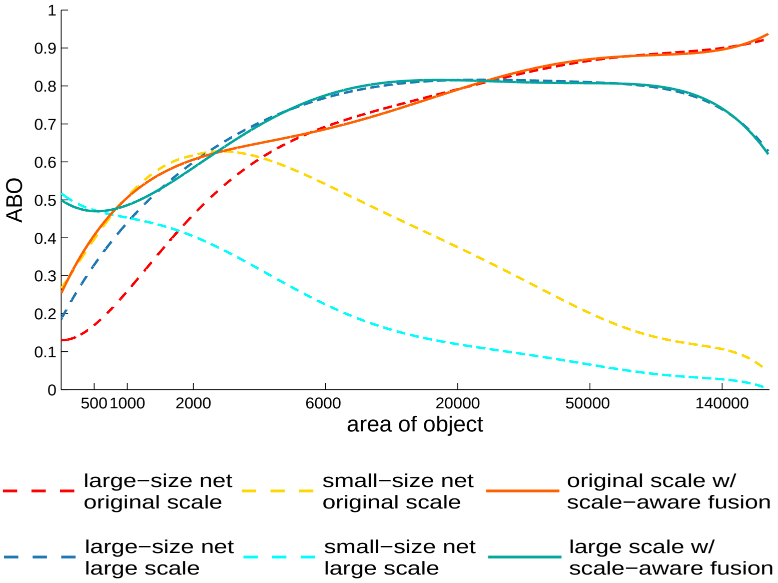


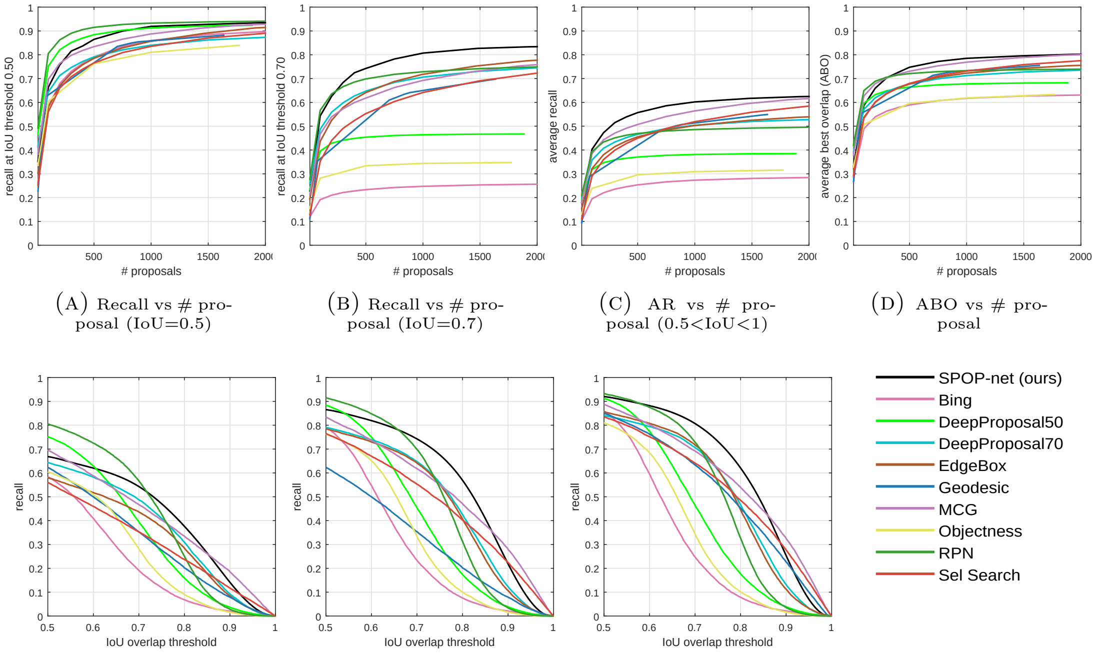
FIGURE 4.26: Average best overlap (ABO) versus ground-truth object area for the four building blocks localization results: large-size network in original scale, small-size network in original scale, large-size network in enlarged scale and small-size network in enlarged scale. All the ABOs are computed given the top 1,000 proposals per image.

4.2.4.3 Comparisons on Object Recall

We compare our SPOP-net with the following state-of-the-art object proposal methods: BING [113], Edge Boxes [114], Geodesic Object Proposal [135], MCG [134], Objectness [110], Selective Search [125] and Region Proposal Network (RPN) [145]. We first evaluate object recall on PASCAL VOC 2007 testing set, which contains 4,952 images with about 15,000 annotated objects. Proposals of most state-of-the-art methods were provided by Hosang et al. [130] in a standard format.As for DeepProposal approach, we directly downloaded the pre-computed proposals from the official website[5].

Figure 4.27(a) and 4.27(b) show the recall when varying the number of proposals for different IoU thresholds. As can be seen, under a loose 0.5 IoU threshold, RPN takes the lead all the time for both a small and a large number of proposals.DeepProposal 50 also

[5] https://github.com/aghodrati/deepproposal



(A) Recall vs # proposal (IoU=0.5)

(B) Recall vs # proposal (IoU=0.7)

(C) AR vs # proposal (0.5<IoU<1)

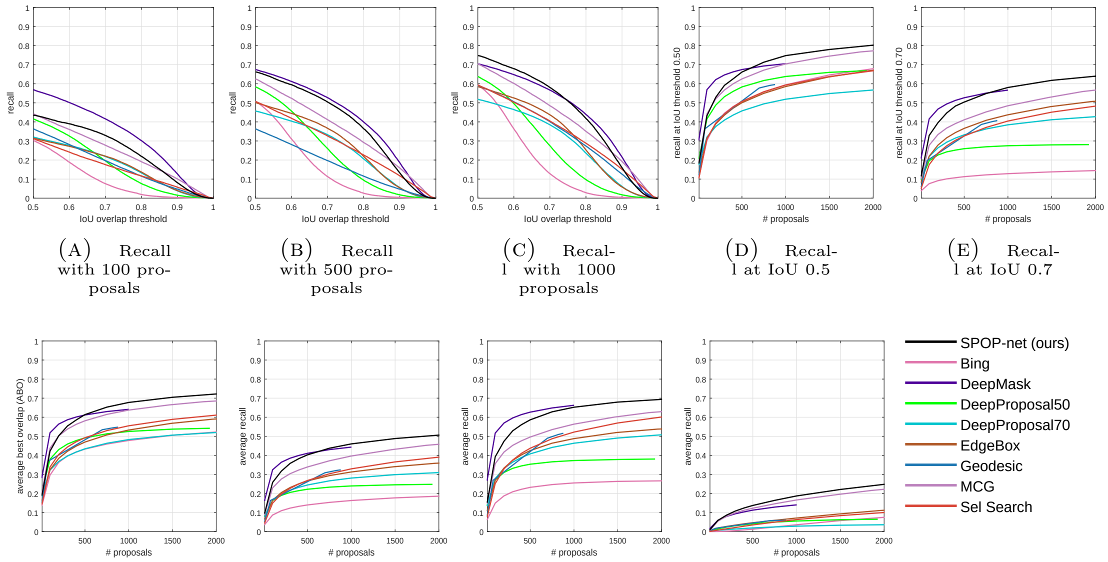
(D) ABO vs # proposal

(E) Recall vs IoU (100 proposals)

(F) Recall vs IoU (500 proposals)

(G) Recall vs IoU (1000 proposals)

Figure 4.27: Recall and average best overlap (ABO) comparison between our SPOP-net and other state-of-the-arts on PASCAL VOC 2007 testing set.



(A) Recall with 100 proposals

(B) Recall with 500 proposals

(C) Recall with 1000 proposals

(D) Recall at IoU 0.5

(E) Recall at IoU 0.7

(F) ABO vs # proposal

(G) AR vs # proposal

(H) AR vs # proposal (large)

(I) AR vs # proposal (small)

Figure 4.28: Recall and average best overlap (ABO) comparison between our SPOP-net and other state-of-the-arts on MS COCO 2014 validation set.

performs well under low IoU thresholds (*e.g.* 0.5). Given a more strict IoU threshold 0.7, our SPOP-net almost keeps the best consistently. We also plot the average recall (AR) versus the number of proposals curves for all the methods in Figure 4.27(c). This is because AR summarizes proposal performance across IoU thresholds and correlates well with object detection performance [130]. The proposed SPOP-net also takes the first place all the time regarding the number of proposals. Figure 4.27(d) shows the average best overlap (ABO) when changing the number of proposals. The proposed SPOP-net shows good localization quality, especially when the number of proposals is more than 500. Figure 4.27(e), 4.27(f) and 4.27(g) demonstrate the recall when the IoU threshold changes within the range [0.5, 1] for different numbers of proposals. It is found that RPN performs well with a small number of proposals when setting a low IoU threshold (< 0.7). When increasing the number of proposals from 100 to 1,000, our SPOP-net gradually shows its advantage. Especially for the top 1,000 proposals, the SPOP-net performs superiorly across a wide range of IoU thresholds from 0.5 to 0.85, which have the strongest correlation to object detection performance thus are typically desired in practice [130].

Figure 4.29 shows the average best overlap (ABO) of the proposed SPOP-net as well as several state-of-the-art methods for the ground-truth objects with different areas. For most object sizes, the SPOP-net shows outstanding performance. Especially for small objects whose area is less than about 1,000, the SPOP-net takes the first place, achieving an ABO higher than 0.5. RPN can achieve a good ABO around 0.7 for the objects whose areas are more than 2,000 pixels, but can hardly reach a higher ABO even if the object is large. This may explain why the recall of RPN is very high when setting a loose IoU threshold (*e.g.* 0.5) but decreases sharply with the increasing of IoU threshold when it exceeds 0.7. The classic low-level cues based methods (*e.g.* Selective Search, MCG, GOP) perform very well for large objects but have inferior performance for small ones compared with two CNN-based methods (*i.e.* SPOP-net, RPN).

For better understanding of the keys of enabling the SPOP-net to work well, we show the intermediate output maps of both the localization and confidence networks for visualization in Figure 4.30. For each image, we show its "objectness confidence map", "offsets map" pointing to the object center, and its proposals. We argue that the first key is the reliable objectness prediction as the proposals predicted by the pixels obtaining low objectness confidence will be ranked behind. Based on an accurate objectness confidence, for each ground-truth object, each pixel inside it predicts its own location of this object, as shown in the "offsets maps", thus greatly increasing the chances of precise localization. Another advantage of this pixel-wise prediction is that most of predicted bounding box locations from the pixels within the same object are heavily overlapping, which can be easily filtered by NMS. Normally only a few proposals are remained after NMS, thus improving the recall of the top-ranked proposals. For small objects, to overcome the inherent weakness that less chances are available to propose the correct locations, a scale-aware prediction is adopted by relying on an accurate estimation of the object size (*i.e.* large or small) and combining the predictions of two networks.

The detailed running speed of the SPOP-net as well as other state-of-the-art methods is presented in Table 4.8. The detailed setting of parameters for each method is as follows. We choose the single color space (*i.e.* RGB) proposal computation for BING, and the "Fast" version for selective search. For the rest methods, we directly run their

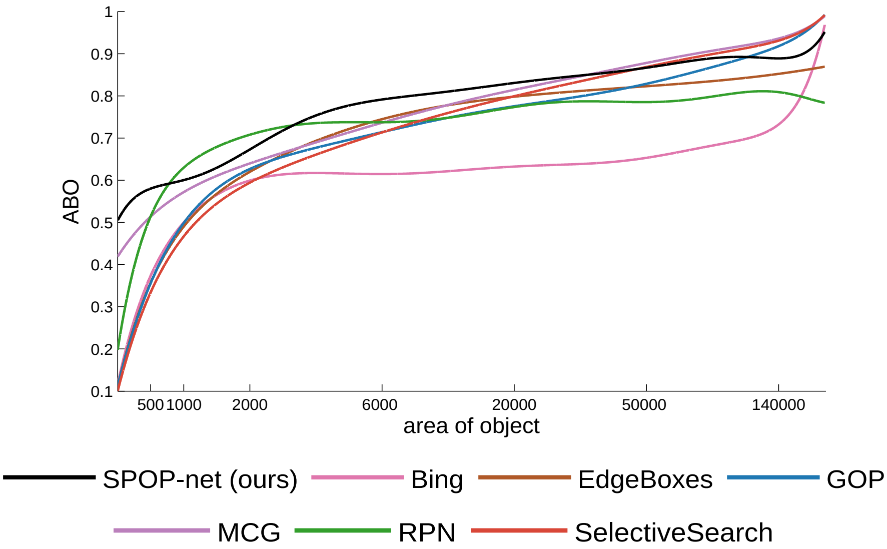


Figure 4.29: Average best overlap (ABO) versus ground-truth object area for the SPOP-net and other state-of-the-art methods. All the ABO are computed given the top 1,000 proposals per image.

Table 4.8: Time cost of the state-of-the-arts and our method.

	Time cost per image
BING	0.01s
Edge Boxes	0.3s
Geodesic	1s
MCG	30s
Objectness	3s
Selective Search	10s
RPN	0.15s
SPOP-net (ours)	1.03s

default codes. As can be seen, *window scoring methods* and *CNN-based methods* are much faster than *segment grouping methods*. Inference for an image of PASCAL VOC size (*e.g.* 300*500) takes 1.03s for our SPOP-net on a single TITAN X CPU. Specifically, testing one network of the original scale and the enlarged scale takes 0.11s and 0.23s on a single TITAN X GPU, respectively. However, as the computation within different CNNs of SPOP-net are independent of each other, training and testing SPOP-net can be accelerated by parallel computation over multiple GPUs. Although it is not one of the fastest object proposal methods (compared to BING, RPN and Edge Boxes), our

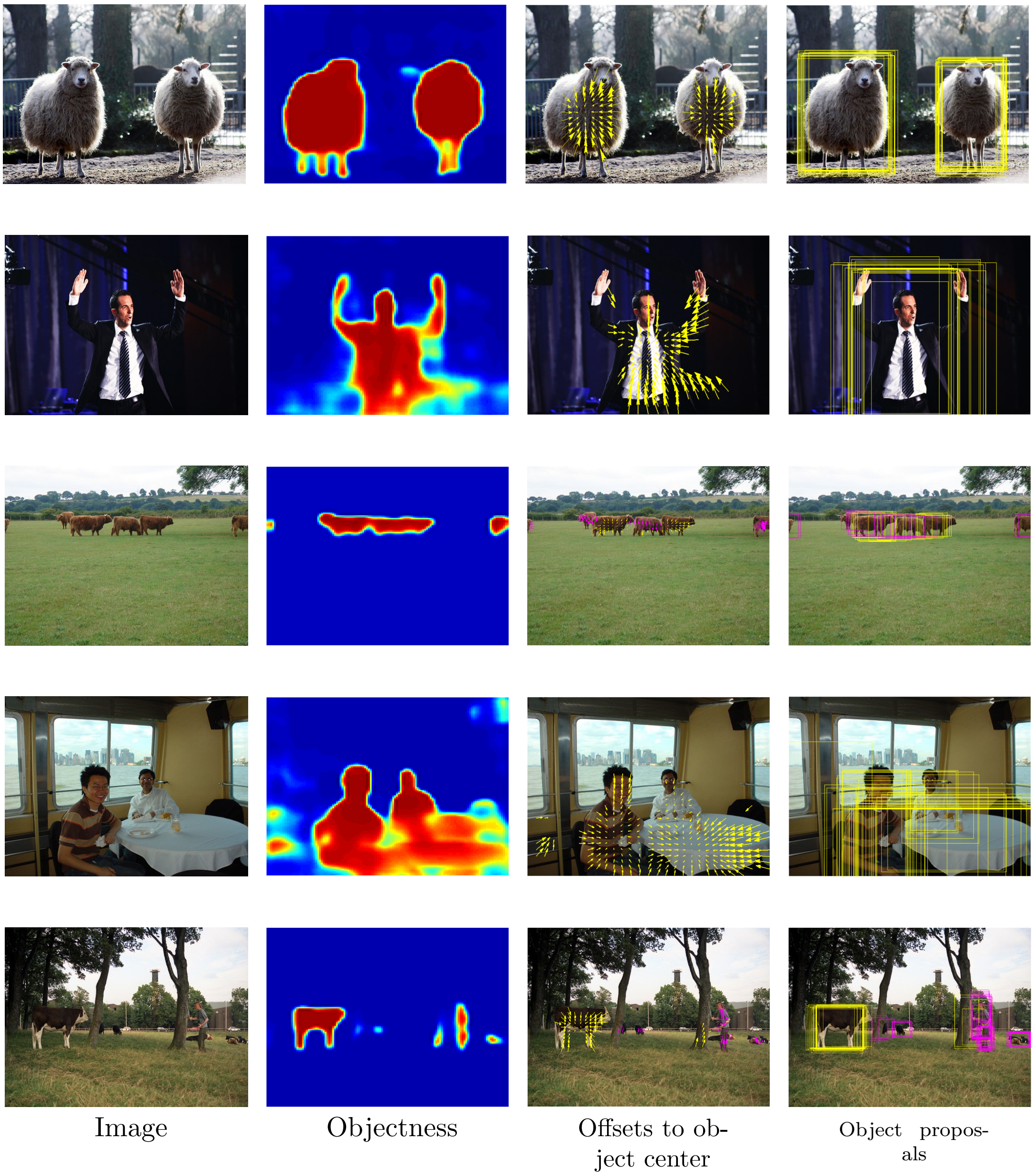


FIGURE 4.30: Example results of predicted "objectness map" (second column), "offsets to object center" after weighted combination (third column) and "object proposals" (fourth column) for the input images (first column).

approach is still competitive in speed among the proposal generators. We do, however, require use of the library Caffe [124] which is based on GPU computation for efficient inference like all CNN based methods. To further reduce the running time, some CNN speedup methods such as FFT, batch parallelization, or truncated SVD could be used in the future.

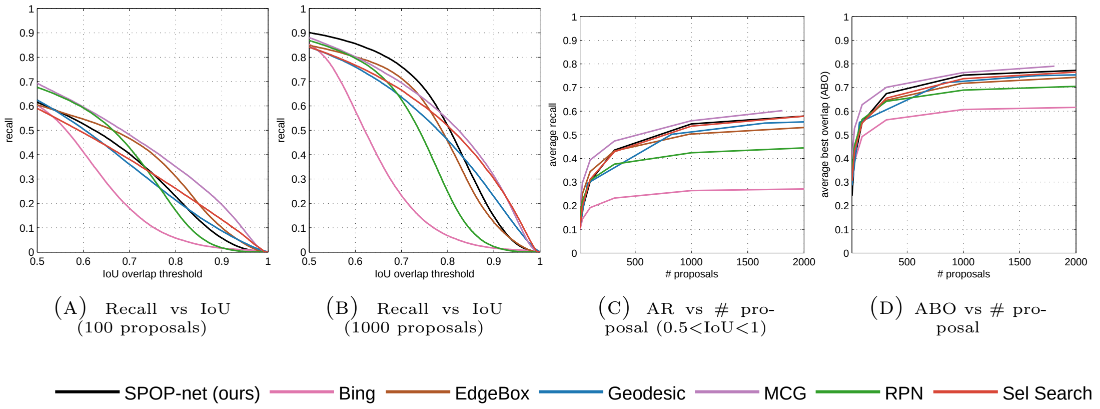
We also evaluate the proposed SPOP-net on MS COCO [122] 2014 validation set and the results are shown in Figure 4.28. The SPOP-net model here is trained on MS COCO training set which contains more than $80,000$ pixel-level annotated images. To conduct fair comparisons with the state-of-the-art segmentation annotations based approach, *i.e.,* DeepMask, we only evaluate on the first $5,000$ images. Note that we directly used the public DeepProposal model to extract proposals on MS COCO images. It is observed that DeepMask performs well, especially for the cases with low IoU thresholds (*e.g.* 0.5) and a few proposals (*e.g.* 100 proposals). The performance of the proposed SPOP-net gradually increases and SPOP-net demonstrates its superiority as the number of proposals increases. Specifically, SPOP-net outperforms DeepMask in terms of recall at IoU 0.5 (Figure 4.28(d)), recall at IoU 0.7 (Figure 4.28(e)), ABO (Figure 4.28(f)) and average recall (Figure 4.28(g)) when the number of proposals is more than 500. Figure 4.28(h) and Figure 4.28(i) shows the average recall of all the methods on large and small objects, respectively. On can observe that SPOP-net performs best on detecting small objects in terms of AR, which clearly validates the superiority of SPOP-net in small objects localization.

Table 4.9: Object detection average precision for all the 20 categories as well as the mean average precision (mAP) on the PASCAL VOC 2007 testing set using the publicly available Fast-RCNN detector trained on VOC 2007 trainval set.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Selective Search	**76.1**	77.3	65.3	53.9	37.8	76.9	78.2	80.9	40.6	74.0	67.2	79.4	82.4	74.9	66.1	33.3	66.0	67.3	73.3	67.3	66.9
Edge Boxes	62.8	77.3	66.2	53.6	42.9	80.6	77.7	81.5	41.4	73.5	65.3	78.1	79.5	**76.2**	67.8	**36.7**	64.5	62.4	70.3	67.9	66.3
MCG	69.3	72.3	62.4	54.4	39.2	77.8	70.1	80.4	40.1	67.2	**68.7**	77.3	75.0	68.8	60.7	34.1	59.5	64.7	70.6	68.2	64.0
RPN (1,000 props)	70.0	76.6	67.2	59.1	44.6	80.0	78.6	86.2	44.1	75.5	60.7	81.3	80.4	75.8	74.1	30.5	72.9	67.2	79.4	69.1	68.7
RPN (300 props)	71.8	77.4	68.0	58.9	**46.3**	81.8	79.0	86.6	45.6	**79.4**	60.2	81.7	81.1	75.9	**74.5**	31.7	73.6	67.2	79.5	**70.6**	69.5
SPOP-net (ours)	70.6	**78.5**	**69.3**	**62.5**	41.1	**82.8**	**79.1**	**88.6**	**47.7**	76.6	66.5	**83.7**	**83.6**	73.1	69.6	36.1	67.8	**72.1**	**85.4**	69.6	**70.2**

4.2.4.4 Object Detection Performance

We conduct experiments analyzing object proposals for use with object detectors to evaluate the effects of proposals on the detection quality. We utilize the standard Fast-RCNN [21] framework as the benchmark. We choose the publicly released VGG 16-layer [12] detector trained on VOC 2007 trainval set in all the evaluation experiments. The proposals of the proposed SPOP-net, Selective Search, Edge Boxes, MCG and RPN are evaluated. Please note that RPN itself integrates proposal generation and detection in a unified framework, called Faster-RCNN. To be fair, we do not adopt this unified detector for object detection with RPN proposals in our evaluation. This is because this unified detector has a weights sharing mechanism in 13 layers which are used for both proposal generation and object detection. These layers are trained on the class-specific annotations with object category information that is not employed in training other methods. For SPOP-net, Selective Search, Edge Boxes and MCG, we select the top 1,000 proposals to pass through the object detectors for post-classification. For the RPN method, considering that it only needs a small number of proposals to achieve high recall, and more proposals do not bring too many improvements to the recall but introduce more false positives, we conduct an extra setting which uses the top 300



(A) Recall vs IoU (100 proposals)

(B) Recall vs IoU (1000 proposals)

(C) AR vs # proposal (0.5<IoU<1)

(D) ABO vs # proposal

FIGURE 4.31: Recall and average best overlap (ABO) comparison between our SPOP-net and other state-of-the-art methods on ILSVRC 2013 validation set.

proposals for detection, which is also claimed by [145].

The detection mean average precision (mAP) as well as the average precision of 20 categories is presented in Table 4.9. It can be seen that the proposed SPOP-net wins on 11 categories among the 20 categories of PASCAL VOC 2007 and also achieves the best mAP 70.2%. Using 1,000 RPN proposals for detection, 68.7% mAP can be obtained. With only 300 proposals, RPN achieves a better mAP 69.5% than 1,000 proposals. This verifies the good performance of RPN when generating a small number of proposals.

4.2.4.5 Generalization to Unseen Categories

The high recall rate which our approach achieves on the PASCAL VOC 2007 testing set does not guarantee it to have learned the generic objectness notion or be able to predict the object proposals for the images containing novel objects in unseen categories. This is because it is possible that the model is highly tuned to the 20 categories of PASCAL VOC. To investigate whether it is capable of predicting the proposals for the unseen categories beyond training, we evaluate our approach on the ImageNet ILSVRC 2013 validation set which contains more than 20,000 images with around 50,000 annotated objects in 200 categories.

From Figure 4.31, the overall trend of the SPOP-net remains consistent with that on the PASCAL VOC 2007. Specifically, with a small number of proposals (*e.g.* 100 proposals), the SPOP-net does not perform as well as MCG, RPN and Edge Boxes, but shows its superiority when the number of proposals reaches 1,000. See Figure 4.31(b). As for average recall (AR) and average best overlap (ABO), the SPOP-net is also one of the best methods across a broad range of proposal numbers. It is worth mentioning that RPN does not perform as well as on PASCAL VOC 2007. An obvious drop is seen under all the evaluation scenarios from Figure 4.31 compared to Figure 4.27. This may result from the category information employed when training the layers in the RPN network shared with class-specific detectors. Such class-awareness enables RPN to fit the 20 categories of PASCAL VOC 2007 better but affects its generalization ability to unseen categories.

Based on the high recall rate the SPOP-net remains when evaluated on ILSVRC 2013, no significant overfitting towards the PASCAL VOC categories is observed. In other words, the proposed approach has learned a generic notion of objectness and can generalize well to the unseen categories.

4.2.5 Summary

In this chapter, we developed an effective scale-aware pixel-wise localization network for object proposal generation. The network fully exploits the available pixel-wise segmentation annotations and predicts the proposals pixel-wisely. Each proposal combines two proposals predicted by two networks specialized for different sizes respectively. The combination follows a weighting mechanism utilizing the weighting confidence produced by a large-/small-size object classification model. This strategy is shown to enhance the accuracy of localization on small objects. Significant improvements over the state-of-the-art methods were achieved by the proposed SPOP-net on the PASCAL VOC 2007 testing set. The proposals of the SPOP-net used in Fast-RCNN detector also provide the highest mAP, benefiting from the high recall rate of the proposed model. In the future, we plan to extend our method to deal with both object proposal generation and bounding box regression step to achieve better localization performance.

Chapter 5

Tree-structured Reinforcement Learning for Sequential Detection

5.1 Overview

Modern state-of-the-art object detection systems [21, 127] usually adopt a two-step pipeline: extract a set of class-independent object proposals at first and then classify these object proposals with a pre-trained classifier. Existing object proposal algorithms usually search for possible object regions over dense locations and scales separately [113, 114, 145]. However, the critical correlation cues among different proposals (*e.g.*, relative spatial layouts or semantic correlations) are often ignored. This in fact deviates from the human perception process — as claimed in [157], humans do not search for objects within each local image patch separately, but start with perceiving the whole scene and successively explore a small number of regions of interest via sequential attention

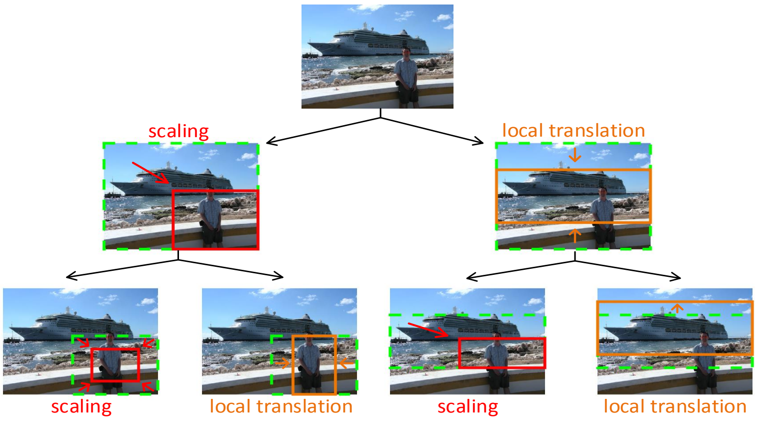


Figure 5.1: Illustration of Tree-RL. Starting from the whole image, the agent recursively selects the best actions from both action groups to obtain two next windows for each window. Red and orange solid windows are obtained by taking scaling and local translation actions, respectively. For each state, green dashed windows are the initial windows before taking actions, which are result windows from the last level.

patterns. Inspired by this observation, extracting one object proposal should incorporate the global dependencies of proposals by considering the cues from the previous predicted proposals and future possible proposals jointly.

In this chapter, in order to fully exploit global interdependency among objects, we propose a novel Tree-structured Reinforcement Learning (Tree-RL) approach that learns to localize multiple objects sequentially based on both the current observation and historical search paths. Starting from the entire image, the Tree-RL approach sequentially acts on the current search window either to refine the object location prediction or discover new objects by following a learned policy. In particular, the localization agent is trained by deep RL to learn the policy that maximizes a long-term reward for localizing all the objects, providing better global reasoning. For better training the agent, we propose a novel reward stimulation that well balances the exploration of uncovered new objects and refinement of the current one for quantifying the localization accuracy improvements.

The Tree-RL adopts a tree-structured search scheme that enables the agent to more accurately find objects with large variation in scales. The tree search scheme consists of two branches of pre-defined actions for each state, one for locally translating the current window and the other one for scaling the window to a smaller one. Starting from the whole image, the agent recursively selects the best action from each of the two branches according to the current observation (see Figure 5.1). The proposed tree search scheme enables the agent to learn multiple near-optimal policies in searching multiple objects. By providing a set of diverse near-optimal policies, Tree-RL can better cover objects in a wide range of scales and locations.

Extensive experiments on PASCAL VOC 2007 and 2012 [115] demonstrate that the proposed model can achieve a similar recall rate as the state-of-the-art object proposal algorithm RPN [145] yet using a significantly smaller number of candidate windows. Moreover, the proposed approach also provides more accurate localizations than RPN. Combined with the Fast R-CNN detector [21], the proposed approach also achieves higher detection mAP than RPN.

The rest of this chapter is organized as follows. Section 6.2 briefly reviews the related work to ours, including object proposal methods, active object search methods and visual attention models. Section 6.3 describes the details of our tree-structured reinforcement learning. Section 6.4 presents experimental results of using the proposed method to generate object proposals and solve generic object detection, followed by the summary drawn in Section 6.5.

5.2 Related Work

Our work is related to the works which utilize different object localization strategies instead of sliding window search in object detection. Existing works trying to reduce the number of windows to be evaluated in the post-classification can be roughly categorized into two types, *i.e.*, object proposal algorithms and active object search with visual attention.

Early object proposal algorithms typically rely on low-level image cues, *e.g.*, edge, gradient and saliency [113, 114, 126]. For example, Selective Search [125] hierarchically merges the most similar segments to form proposals based on several low-level cues including color and texture; Edge Boxes [114] scores a set of densely distributed windows based on edge strengths fully inside the window and outputs the high scored ones as proposals. Recently, RPN [145] utilizes a Fully Convolutional Network (FCN) [158] to densely generate the proposals in each local patch based on several pre-defined "anchors" in the patch, and achieves state-of-the-art performance in object recall rate. Nevertheless, object proposal algorithms assume that the proposals are independent and usually perform window-based classification on a set of reduced windows individually, which may still be wasteful for images containing only a few objects.

Another type of works attempts to reduce the number of windows with an active object detection strategy. Lampert *et al.* [159] proposed a branch-and-bound approach to find the highest scored windows while only evaluating a few locations. Alexe *et al.* [160] proposed a context driven active object searching method, which involves a nearest-neighbor search over all the training images. Gonzeles-Garcia *et al.* [161] proposed an active search scheme to sequentially evaluate selective search object proposals based on spatial context information.

Visual attention models are also related to our work. These models are often leveraged to facilitate the decision by gathering information from previous steps in the sequential decision making vision tasks. Xu *et al.* [162] proposed an attention model embedded in recurrent neural networks (RNN) to generate captions for images by focusing on different regions in the sequential word prediction process. Minh *et al.* [163] and Ba *et al.* [164] also relied on RNN to gradually refine the focus regions to better recognize characters.

Perhaps [165] and [166] are the closest works to ours. [165] learned an optimal policy to localize a single object through deep Q-learning. To handle multiple objects cases, it runs the whole process starting from the whole image multiple times and uses an inhibition-of-return mechanism to manually mark the objects already found. [166] proposed a top-down search strategy to recursively divide a window into sub-windows. Then similar to RPN, all the visited windows serve as “anchors” to regress the locations of object bounding boxes. Compared to them, our model can localize multiple objects in a single run starting from the whole image. The agent learns to balance the exploration of uncovered new objects and the refinement of covered ones with deep Q-learning. Moreover, our top-down tree search does not produce “anchors” to regress the object locations, but provides multiple near-optimal search paths and thus requires less computation.

5.3 Tree-Structured Reinforcement Learning for Object Localization

5.3.1 Multi-Object Localization as a Markov Decision Process

The Tree-RL is based on a Markov decision process (MDP) which is well suitable for modeling the discrete time sequential decision making process. The localization agent sequentially transforms image windows within the whole image by performing one of pre-defined actions. The agent aims to maximize the total discounted reward which reflects the localization accuracy of all the objects during the whole running episode. The design of the reward function enables the agent to consider the trade-off between further refinement of the covered objects and searching for uncovered new objects. The actions, state and reward of our proposed MDP model are detailed as follows.

Actions: The available actions of the agent consist of two groups, one for scaling the current window to a sub-window, and the other one for translating the current window locally. Specifically, the scaling group contains five actions, each corresponding to a certain sub-window with the size 0.55 times as the current window (see Figure 5.2). The local translation group is composed of eight actions, with each one changing the current window in one of the following ways: horizontal moving to left/right, vertical moving to up/down, becoming shorter/longer horizontally and becoming shorter/longer vertically, as shown in Figure 5.2, which are similar to [165]. Each local translation action moves the window by 0.25 times of the current window size. The next state is then deterministically obtained after taking the last action. The scaling actions are

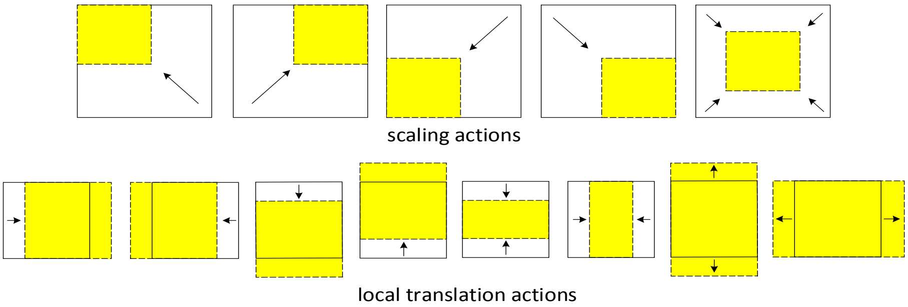


Figure 5.2: Illustration of the five scaling actions and eight local translation actions. Each yellow window with dashed lines represents the next window after taking the corresponding action.

designed to facilitate the search of objects in various scales, which cooperate well with the later discussed tree search scheme in localizing objects in a wide range of scales. The translation actions aim to perform successive changes of visual focus, playing an important role in both refining the current attended object and searching for uncovered new objects.

States: At each step, the state of MDP is the concatenation of three components: the feature vector of the current window, the feature vector of the whole image and the history of taken actions. The features of both the current window and the whole image are extracted using a VGG-16 [12] layer CNN model pre-trained on ImageNet. We use the feature vector of layer “fc6” in our problem. To accelerate the feature extraction, all the feature vectors are computed on top of pre-computed feature maps of the layer “conv5_3” after using ROI Pooling operation to obtain a fixed-length feature representation of the specific windows, which shares the spirit of Fast R-CNN. It is worth mentioning that the global feature here not only provides context cues to facilitate the refinement of the currently attended object, but also allows the agent to be aware of the existence of other uncovered new objects and thus make a trade-off between further refining the attended object and exploring the uncovered ones. The history of the taken actions is a binary vector that tells which actions have been taken in the past. Therefore, it implies the search paths that have already been gone through and the objects already attended by the agent. Each action is represented by a 13-d binary vector where all values are zeros except for the one corresponding to the taken action. 50 past actions are encoded in the state to save a full memory of the paths from the start.

Rewards: The reward function $r(s, a)$ reflects the localization accuracy improvements of all the objects by taking the action a under the state s. We adopt the simple yet indicative localization quality measurement, Intersection-over-Union (IoU) between the current window and the ground-truth object bounding boxes. Given the current window w and a ground-truth object bounding box g, IoU between w and g is defined as $\text{IoU}(w, g) \triangleq \text{area}(w \cap g)/\text{area}(w \cup g)$. Assuming that the agent moves from state s to state s' after taking the action a, each state s has an associated window w, and there are n ground-truth objects $g_1 \ldots g_n$, then the reward $r(s, a)$ is defined as follows:

$$r(s, a) = \max_{1 \leq i \leq n} \text{sign}(\text{IoU}(w', g_i) - \text{IoU}(w, g_i)). \tag{5.1}$$

This reward function returns $+1$ or -1. Basically, if any ground-truth object bounding box has a higher IoU with the next window than the current one, the reward of the action moving from the current window to the next one is $+1$, and -1 otherwise. Such binary rewards reflect more clearly which actions can drive the window towards the ground-truths and thus facilitate the agent's learning. This reward function encourages the agent to localize any objects freely, without any limitation or guidance on which object should be localized at that step. Such a free localization strategy is especially important in a multi-object localization system for covering multiple objects by running only a single episode starting from the whole image.

Another key reward stimulation +5 is given to those actions which cover any ground-truth objects with an IoU greater than 0.5 for the first time. For ease of explanation, we define $f_{i,t}$ as the hit flag of the ground-truth object g_i at the t^{th} step which indicates whether the maximal IoU between g_i and all the previously attended windows $\{w^j\}_{j=1}^{t}$ is greater than 0.5, and assign +1 to $f_{i,t}$ if $\max_{1 \leq j \leq t} \mathrm{IoU}(w_j, g_i)$ is greater than 0.5 and -1 otherwise. Then supposing the action a is taken at the t^{th} step under state s, the reward function integrating the first-time hit reward can be written as follows:

$$r(s,a) = \begin{cases} +5, & \text{if } \max\limits_{1 \leq i \leq n} (f_{i,t+1} - f_{i,t}) > 0 \\ \max\limits_{1 \leq i \leq n} \ \mathrm{sign}(\mathrm{IoU}(w', g_i) - \mathrm{IoU}(w, g_i)), & \text{otherwise.} \end{cases} \tag{5.2}$$

The high reward given to the actions which hit the objects with an IoU > 0.5 for the first time avoids the agent being trapped in the endless refinement of a single object and promotes the search for uncovered new objects.

5.3.2 Tree-Structured Search

The Tree-RL relies on a tree structured search strategy to better handle objects in a wide range of scales. For each window, the actions with the highest predicted value in both the scaling action group and the local translation action group are selected respectively. The two best actions are both taken to obtain two next windows: one is a sub-window of the current one and the other is a nearby window to the current

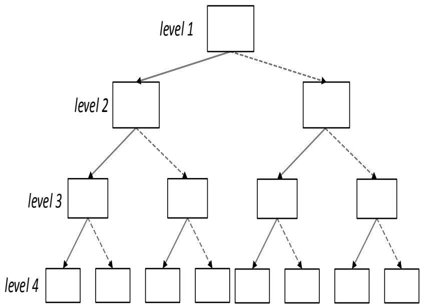


Figure 5.3: Illustration of the top-down tree search. Starting from the whole image, each window recursively takes the best actions from both action groups. Solid arrows and dashed arrows represent scaling actions and local translation actions, respectively.

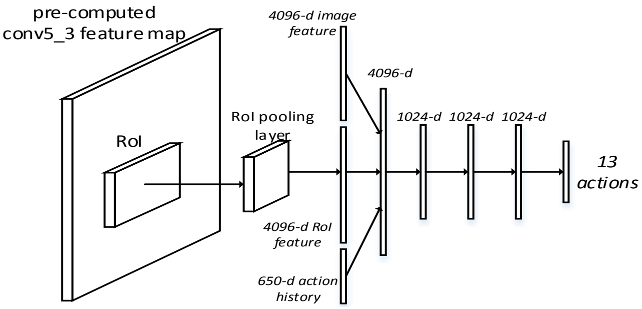


Figure 5.4: Illustration of our Q-network. The regional feature is computed on top of the pre-computed “conv5_3” feature maps extracted by VGG-16 pre-trained model. It is concatenated with the whole image feature and the history of past actions to be fed into an MLP. The MLP predicts the estimated values of the 13 actions.

one after local translation. Such bifurcation is performed recursively by each window starting from the whole image in a top-down fashion, as illustrated in Figure 5.3. With tree search, the agent is enforced to take both scaling action and local translation action simultaneously at each state, and thus travels along multiple near-optimal search paths instead of a single optimal path. This is crucial for improving the localization accuracy for objects in different scales. Because only the scaling actions significantly change the scale of the attended window while the local translation actions almost keep the scale the same as the previous one. However there is no guarantee that the scaling actions are often taken as the agent may tend to go for large objects which are easier to be covered with an IoU larger than 0.5, compared to scaling the window to find small objects.

5.3.3 Deep Q-learning

The optimal policy of maximizing the sum of the discounted rewards of running an episode starting from the whole image is learned with reinforcement learning. However, due to the high-dimensional continuous image input data and the model-free environment, we resort to the Q-learning algorithm combined with the function approximator technique to learn the optimal value for each state-action pair which generalizes well to unseen inputs. Specifically, we use the deep Q-network proposed by [167, 168] to estimate the value for each state-action pair using a deep neural network. The detailed architecture of our Q-network is illustrated in Figure 5.4. Please note that similar to [168], we also use the pre-trained CNN as the regional feature extractor instead of training the whole hierarchy of CNN, considering the good generalization of the CNN trained on ImageNet [9].

During training, the agent runs sequential episodes which are paths from the root of the tree to its leafs. More specifically, starting from the whole image, the agent takes one action from the whole action set at each step to obtain the next state. The agent's behavior during training is ϵ-greedy. Specifically, the agent selects a random action from the whole action set with probability ϵ, and selects a random action from the two best actions in the two action groups (*i.e.* scaling group and local translation group) with probability $1 - \epsilon$, which differs from the usual exploitation behavior that the single best action with the highest estimated value is taken. Such exploitation is more consistent with the proposed tree search scheme that requires the agent to take the best actions from both action groups. We also incorporate a replay memory following [168] to store the experiences of the past episodes, which allows one transition to be used in multiple model updates and breaks the short-time strong correlations between training samples. Each time Q-learning update is applied, a mini batch randomly sampled from the replay memory is used as the training samples. The update for the network weights at the i^{th} iteration θ_i given transition samples (s, a, r, s') is as follows:

$$\theta_{i+1} = \theta_i + \alpha(r + \gamma \max_{a'} Q(s', a'; \theta_i) - Q(s, a; \theta_i))\nabla_{\theta_i} Q(s, a; \theta_i), \tag{5.3}$$

where a' represents the actions that can be taken at state s', α is the learning rate and γ is the discount factor.

5.3.4 Implementation Details

We train a deep Q-network on VOC 2007+2012 trainval set [115] for 25 epochs. The total number of training images is around 16,000. Each epoch is ended after performing an episode in each training image. During ϵ-greedy training, ϵ is annealed linearly from 1 to 0.1 over the first 10 epochs. Then ϵ is fixed to 0.1 in the last 15 epochs. The discount factor γ is set to 0.9. We run each episode with maximal 50 steps during training. During testing, using the tree search, one can set the number of levels of the search tree to obtain the desired number of proposals. The replay memory size is set to 800,000, which contains about 1 epoch of transitions. The mini batch size in training is set to 64. The implementations are based on the publicly available Torch7 [169] platform on a single NVIDIA GeForce Titan X GPU with 12GB memory.

Table 5.1: Recall rates (in %) of single optimal search path RL with different numbers of search steps and under different IoU thresholds on VOC 07 testing set. We only report 50 steps instead of 63 steps as the maximal number of steps is 50.

# steps	large/small	IoU=0.5	IoU=0.6	IoU=0.7
31	large	62.2	53.1	40.2
31	small	18.9	15.6	11.2
31	all	53.8	45.8	34.5
50	large	62.3	53.2	40.4
50	small	19.0	15.8	11.3
50	all	53.9	45.9	34.8

Table 5.2: Recall rates (in %) of Tree-RL with different numbers of search steps and under different IoU thresholds on VOC 07 testing set. 31 and 63 steps are obtained by setting the number of levels in Tree-RL to 5 and 6, respectively.

# steps	large/small	IoU=0.5	IoU=0.6	IoU=0.7
31	large	78.9	69.8	53.3
31	small	23.2	12.5	4.5
31	all	68.1	58.7	43.8
63	large	83.3	76.3	61.9
63	small	39.5	28.9	15.1
63	all	74.8	67.0	52.8

5.4 Experimental Results

We conduct comprehensive experiments on PASCAL VOC 2007 and 2012 testing sets of detection benchmarks to evaluate the proposed method. The recall rate comparisons are conducted on VOC 2007 testing set because VOC 2012 does not release the ground-truth annotations publicly and can only return a detection mAP (mean average precision) of the whole VOC 2012 testing set from the online evaluation server.

5.4.1 Tree-RL vs Single Optimal Search Path RL:

We first compare the performance in recall rate between the proposed Tree-RL and a single optimal search path RL on PASCAL VOC 2007 testing set. For the single optimal search path RL, it only selects the best action with the highest estimated value by the deep Q-network to obtain one next window during testing, instead of taking two best actions from the two action groups. As for the exploitation in the ϵ-greedy behavior during training, the agent in the single optimal path RL always takes the action with the highest estimated value in the whole action set with probability $1 - \epsilon$. Apart from the different search strategy in testing and exploitation behavior during training, all the actions, state and reward settings are the same as Tree-RL. Please note that for Tree-RL, we rank the proposals in the order of the tree depth levels. For example, when setting the number of levels to 5, we have 1+2+4+8+16=31 proposals. The recall rates of the single optimal search path RL and Tree-RL are shown in Table **??** and Table **??**, respectively. It is found that the single optimal search path RL achieves an acceptable recall with a small number of search steps. This verifies the effectiveness of the proposed MDP model (including reward, state and actions setting) in discovering multiple objects. It does not rely on running multiple episodes starting from the whole image like [165] to find multiple objects. It is also observed that Tree-RL outperforms the single optimal search path RL in almost all the evaluation scenarios, especially for large objects[1]. The only case where Tree-RL is worse than the single optimal search path RL is the recall of small objects within 31 steps at IoU threshold 0.6 and 0.7. This may be because the agent performs a breadth-first-search from the whole image, and successively narrows down to a small region. Therefore, the search tree is still too shallow (*i.e.* 5 levels) to accurately cover all the small objects using 31 windows. Moreover, we also find that recalls of the single optimal search path RL become stable with a few steps and hardly increase with the increasing of steps. In contrast, the recalls of Tree-RL keep increasing as the levels of the search tree increase. Thanks to the multiple diverse near-optimal search paths, a better coverage of the whole image in both locations and scales is achieved by Tree-RL.

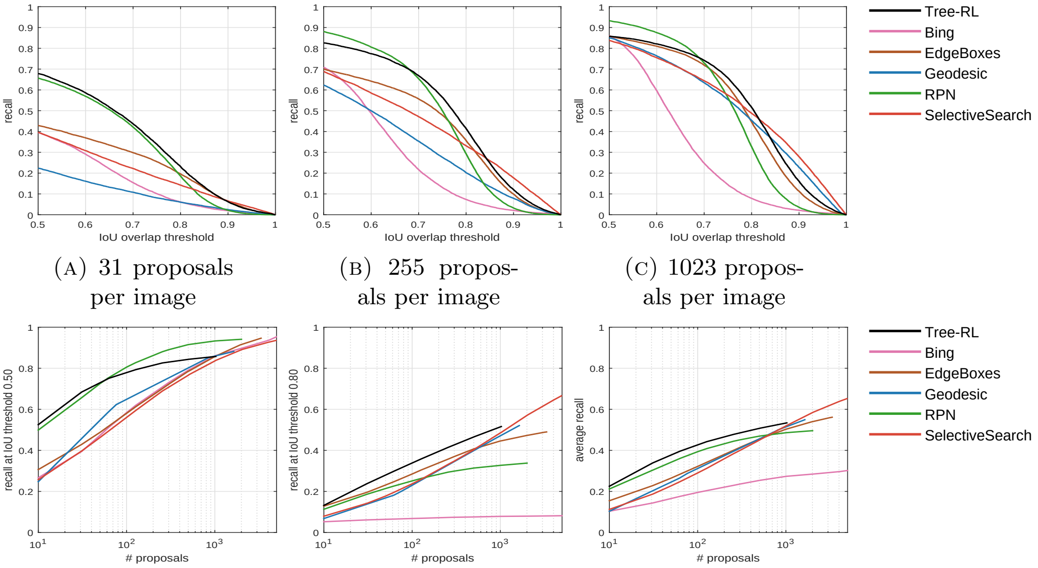
[1] Throughout the paper, large objects are defined as those containing more than 2,000 pixels. The rest are small objects.

5.4.2 Recall Comparison to Other Object Proposal Algorithms

We then compare the recall rates of the proposed Tree-RL and the following object proposal algorithms: BING [113], Edge Boxes [114], Geodesic Object Proposal [135], Selective Search [125] and Region Proposal Network (RPN) [145] (VGG-16 network trained on VOC 07+12 trainval) on VOC 2007 testing set. All the proposals of other methods are provided by [170]. Figure 5.5 (a)-(c) show the recall when varying the IoU threshold within the range [0.5,1] for different numbers of proposals. We set the number of levels in Tree-RL to 5, 8 and 10 respectively to obtain the desired numbers of proposals. Figure 5.5 (e)-(g) demonstrate the recall when changing the number of proposals for different IoU thresholds. It can be seen that Tree-RL outperforms other methods including RPN significantly with a small number of proposals (*e.g.* 31). When increasing the number of proposals, the advantage of Tree-RL over other methods becomes smaller, especially at a low IoU threshold (*e.g.* 0.5). For high IoU thresholds (*e.g.* 0.8), Tree-RL stills performs the best among all the methods. Tree-RL also behaves well on the average recall between IoU 0.5 to 1 which is shown to correlate extremely well with detector performance [170].

5.4.3 Detection mAP Comparison to Faster R-CNN

We conduct experiments to evaluate the effects on object detection of the proposals generated by the proposed Tree-RL. The two baseline methods are RPN (VGG-16) + Fast R-CNN (ResNet-101) and Faster R-CNN (ResNet-101). The former one trains a Fast R-CNN detector (ResNet-101 network) on the proposals generated by a VGG-16 based RPN to make fair comparisons with the proposed Tree-RL which is also based on



(a) 31 proposals per image

(b) 255 proposals per image

(c) 1023 proposals per image

(e) Recall at 0.5 IoU

(f) Recall at 0.8 IoU

(g) Average recall (0.5<IoU <1)

Figure 5.5: Recall comparisons between Tree-RL and other state-of-the-art methods on PASCAL VOC 2007 testing set.

VGG-16 network. The latter one, *i.e.* Faster-RCNN (ResNet-101), is a state-of-the-art detection framework integrating both proposal generation and object detector in an end-to-end trainable system which is based on ResNet-101 network. Our method, Tree-RL (VGG-16) + Fast R-CNN (ResNet-101) trains a Fast R-CNN detector (ResNet-101 network) on the proposals generated by the VGG-16 based Tree-RL. All the Fast R-CNN detectors are fine-tuned from the publicly released ResNet-101 model pre-trained on ImageNet. The final average pooling layer and the 1000-d fc layer of ResNet-101 are replaced by a new fc layer directly connecting the last convolution layer to the output (classification and bounding box regression) during fine-tuning. For Faster-RCNN (ResNet-101), we directly use the reported results in [171]. For the other two methods, we train and test the Fast R-CNN using the top 255 proposals. Table **??** and Table **??** show the average precision of 20 categories and mAP on PASCAL VOC 2007 and 2012 testing

Table 5.3: Detection results comparison on PASCAL VOC 2007 testing set.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
RPN (VGG-16)+ Fast R-CNN (ResNet-101)	77.7	82.7	77.4	68.5	54.7	85.5	80.0	87.6	60.7	83.2	71.8	84.8	85.1	75.6	76.9	52.0	76.8	79.1	81.1	73.9	75.8
Faster R-CNN (ResNet-101) [171]	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0	76.4
Tree-RL (VGG-16)+ Fast R-CNN (ResNet-101)	78.2	82.4	78.0	69.3	55.4	86.0	79.3	88.4	60.8	85.3	74.0	85.7	86.3	78.2	77.2	51.4	76.4	80.5	82.2	74.5	76.6

Table 5.4: Detection results comparison on PASCAL VOC 2012 testing set.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
RPN (VGG-16)+ Fast R-CNN (ResNet-101)	86.9	83.3	75.6	55.4	50.8	79.2	76.9	92.8	48.8	79.0	57.2	90.2	85.4	82.1	79.4	46.0	77.0	66.4	83.3	66.0	73.1
Faster R-CNN (ResNet-101) [171]	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6	73.8
Tree-RL (VGG-16)+ Fast R-CNN (ResNet-101)	85.9	79.3	77.1	62.1	53.4	77.8	77.4	90.1	52.3	79.2	56.2	88.9	84.5	80.8	81.1	51.7	77.3	66.9	82.6	68.5	73.7

set, respectively. It can be seen that the proposed Tree-RL combined with Fast R-CNN outperforms two baselines, especially the recent reported Faster R-CNN (ResNet-101) on the detection mAP. Considering the fact that the proposed Tree-RL relies on only VGG-16 network which is much shallower than ResNet-101 utilized by Faster R-CNN in proposal generation, the proposed Tree-RL is able to generate high-quality object proposals which are effective when used in object detection.

5.4.4 Visualizations

We show the visualization examples of the proposals generated by Tree-RL in Figure 5.6. As can be seen, within only 15 proposals (the sum of level 1 to level 4), Tree-RL is able to localize the majority of objects with large or middle sizes. This validates the effectiveness of Tree-RL again in its ability to find multiple objects with a small number of windows.

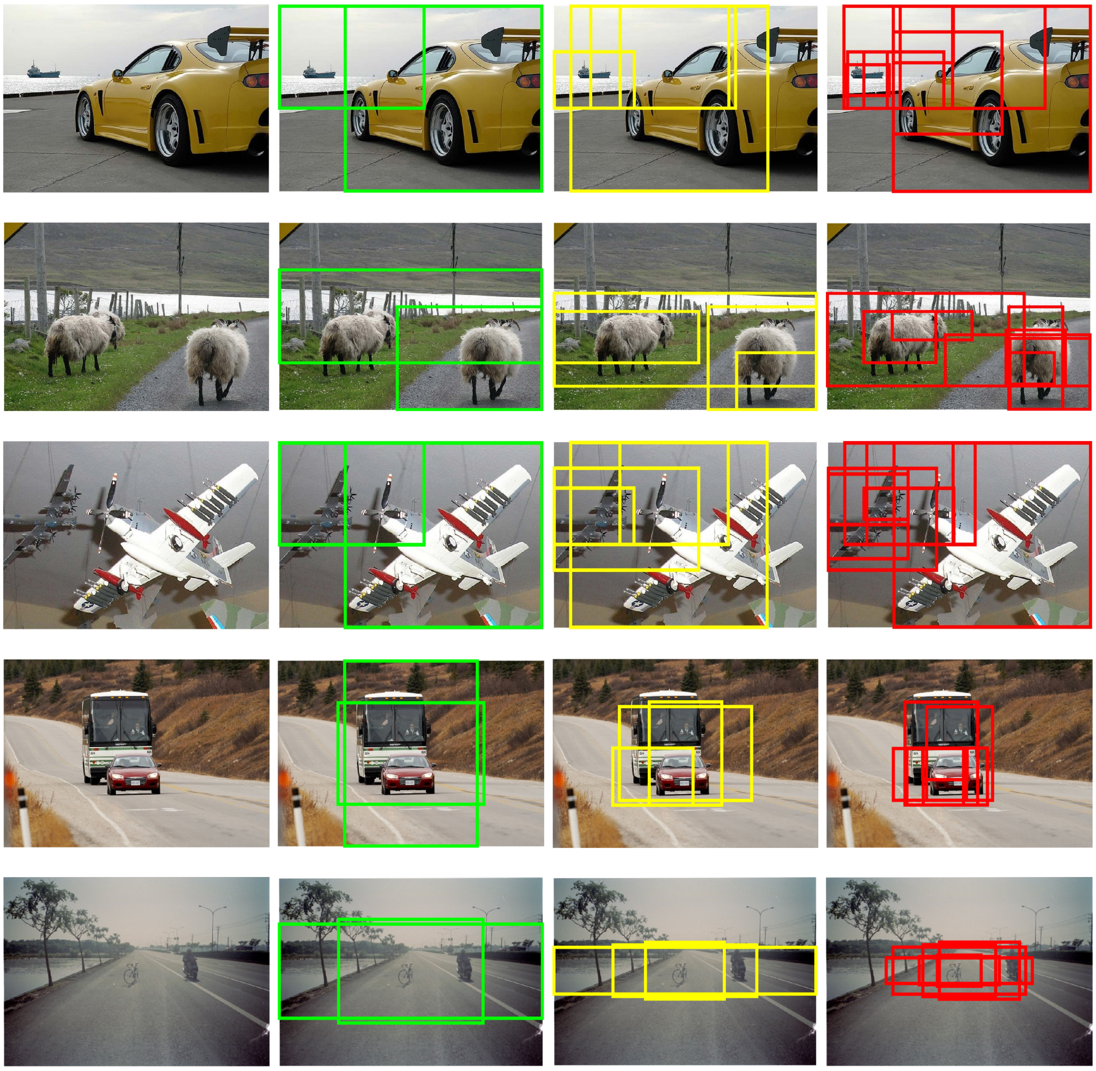


FIGURE 5.6: Examples of the proposals generated by Tree-RL. We only show the proposals of level 2 to level 4. Green, yellow and red windows are generated by the 2nd, 3rd and 4th level respectively. The 1st level is the whole image.

5.5 Summary

In this chapter, we proposed a novel Tree-structured Reinforcement Learning (Tree-RL) approach to sequentially search for objects with the consideration of global interdependency between objects. It follows a top-down tree search scheme to allow the agent to travel along multiple near-optimal paths to discovery multiple objects. The experiments on PASCAL VOC 2007 and 2012 validate the effectiveness of the proposed Tree-RL. Briefly, Tree-RL is able to achieve a comparable recall to RPN with fewer proposals and has higher localization accuracy. Combined with Fast R-CNN detector, Tree-RL outperforms the state-of-the-art detection system Faster R-CNN (ResNet-101).

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we conducted a thorough study on scale-robustness deep learning for visual recognition. Specifically, we focused on the study of improving both image-level and object-level scale-robustness in visual recognition, including scene classification, object proposal generation, object localization and object detection.

We first proposed a framework integrating multi-scale CNN features and cross-level LLC coding to enhance the scale invariance of scene recognition, which is especially important for running vehicles because the captured images can be in any scales. We experimentally verified that the LLC responses on the universal codebook outperform the CNN features and achieve the state-of-the-art performance on the two currently largest scene classification benchmarks, MIT Indoor Scenes and SUN 397.

Second, we proposed an end-to-end framework based on fully convolutional networks (FCN) to detect vehicles and pedestrians. Benefited from FCN, time cost is significantly reduced compared to the one-by-one CNN pass strategy in other deep learning methods. Experiments on the PASCAL VOC 2007 and LISA-Q benchmarks show that using high-level semantic vehicle confidence obtained by FCN, higher precision and recall are achieved. Additionally, FCN enables whole image inference, which makes the proposed method much faster than the object proposal or hand-crafted feature based detectors.

Next, we proposed a novel scale-aware pixel-wise object proposal network to tackle the poor localization performance for small objects. A fully convolutional network is employed to predict the location of object proposal for each pixel. The produced ensemble of pixel-wise object proposals enhances the chance of finding the object significantly without incurring heavy computational cost. To solve the challenge of localizing objects at small scale, two localization networks which are specialized for localizing objects with different scales are introduced. Extensive evaluations on PASCAL VOC 2007 show the SPOP network is superior over the state-of-the-art models. The high-quality proposals from SPOP network also significantly improve the mean average precision (mAP) of object detection with Fast-RCNN framework. Finally, the SPOP network (trained on PASCAL VOC) shows great generalization performance when testing it on ILSVRC 2013 validation set.

In object detection, it is common that multiple vehicles (or pedestrians) are showin in one captured image. Existing localization algorithms usually search for possible object regions over multiple locations and scales separately, which ignore the interdependency among different objects. To incorporate global interdependency between objects into localization, we propose an effective Tree-structured Reinforcement Learning (Tree-RL) approach to learn multiple searching policies through maximizing the long-term reward that reflects localization accuracy over all the objects. Tree-RL is able to find multiple objects with a single feed-forward pass and better cover different objects with various scales, which is quite appealing in vehicle and pedestrian detection. Experiments on PASCAL VOC 2007 and 2012 validate the effectiveness of the Tree-RL, which can achieve comparable recalls with current object proposal algorithms via much fewer candidate windows as well as better detection mAP than Faster R-CNN (ResNet-101).

In summary, the proposed novel methods are validated to perform better than the existing methods in terms of scale invariance in scene recognition, object proposal generation, object localization and object detection tasks.

6.2 Future Works

Based on the work conducted and the outcome of this research, there are two aspects of future work recommended as follows.

First, although the FCN method focuses on dealing with the static image vehicle and pedestrian detection problem, it can also be extended to dynamic video sequences. As is well known, optical flow is the most widely used feature to describe the motion information in videos. It is possible to combine the optical flow map with the 3-channel RGB information together as the FCN input for each frame in the videos. Trained on the samples with 4-channel input, the FCN is expected to gain stronger power in finding moving vehicles and pedestrians in the videos. In the future research, the motion information as input for video sequences will be considered.

Second, we will continue using the semantic confidence obtained by FCN to solve vehicle and pedestrian parsing problems as the segmentation masks of vehicles and pedestrians provide more precise information about the locations and the shape of them. Also, we plan to extend the scale-aware pixel-wise object proposal networks to deal with both object proposal generation and bounding box regression step to achieve better localization performance.

Bibliography

[1] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187: 27–48, 2016.

[2] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTAT-S*, volume 351, pages 423–424, 2012.

[3] Dan C Cireşan, Ueli Meier, and Jürgen Schmidhuber. Transfer learning for latin and chinese characters with deep neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.

[4] Jimmy SJ Ren and Li Xu. On vectorization of deep convolutional neural networks for vision tasks. *arXiv preprint arXiv:1501.07338*, 2015.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[7] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3:e2, 2014.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009.

[10] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1–9. IEEE, 2015.

[14] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. *arXiv preprint arXiv:1403.1840*, 2014.

[15] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361, 1995.

[16] Saeed Reza Kheradpisheh, Masoud Ghodrati, Mohammad Ganjtabesh, and Timothée Masquelier. Deep networks resemble human feed-forward vision in invariant object recognition. *arXiv preprint arXiv:1508.03929*, 2015.

[17] Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014.

[18] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[19] Joshua Gluckman. Scale variant image pyramids. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1069–1075. IEEE, 2006.

[20] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In *European conference on computer vision*, pages 241–254. Springer, 2010.

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

[22] Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang W Koh, Quoc V Le, and Andrew Y Ng. Tiled convolutional neural networks. In *Advances in neural information processing systems*, pages 1279–1287, 2010.

[23] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

[24] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *IEEE CVPR*, pages 923–930, 2013.

[25] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010.

[26] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: S-patial pyramid matching for recognizing natural scene categories. In *IEEE CVPR*, pages 2169–2178, 2006.

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[28] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, pages 886–893, 2005.

[29] Maxime Oquab, Leon Bottou, Ivan Laptev, Josef Sivic, et al. Learning and transferring mid-level image representations using convolutional neural networks. *arXiv preprint*, 2013.

[30] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[31] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE CVPR*, pages 3485–3492, 2010.

[32] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IEEE CVPR*, pages 3360–3367, 2010.

[33] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. *IEEE CVPR*, 2009.

[34] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification via plsa. In *ECCV*, pages 517–530. 2006.

[35] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE CVPR*, pages 1794–1801, 2009.

[36] Kai Yu, Yuanqing Lin, and John Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1713–1720. IEEE, 2011.

[37] Lingxi Xie, Jingdong Wang, Baining Guo, Bo Zhang, and Qi Tian. Orientational pyramid matching for recognizing indoor scenes. In *IEEE CVPR*, 2014.

[38] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

[39] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE CVPR*, pages 3304–3311, 2010.

[40] Aymen Shabou and Hervé LeBorgne. Locality-constrained and spatially regularized coding for scene categorization. In *IEEE CVPR*, pages 3618–3625, 2012.

[41] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[42] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, pages 494–502, 2013.

[43] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[44] Umesh Shankar. Pedestrian roadway fatalities. Technical report, 2003.

[45] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3431–3440, 2015.

[47] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE transactions on pattern analysis and machine intelligence*, 26(11):1475–1490, 2004.

[48] Xiaoxu Ma and W Eric L Grimson. Edge-based rich representation for vehicle classification. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1185–1192. IEEE, 2005.

[49] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[50] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[51] Tao Gao, Zheng-guang Liu, Wen-chun Gao, and Jun Zhang. Moving vehicle tracking based on sift active particle choosing. In *International Conference on Neural Information Processing*, pages 695–702. Springer, 2008.

[52] Khalil M Ahmad Yousef, Maha Al-Tabanjah, Esraa Hudaib, and Maymona Ikrai. Sift based automatic number plate recognition. In *Information and Communication Systems (ICICS), 2015 6th International Conference on*, pages 124–129. IEEE, 2015.

[53] Xiyan Chen and Qinggang Meng. Vehicle detection from uavs by using sift with implicit shape model. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3139–3144. IEEE, 2013.

[54] Liang Wei, Xie Xudong, Wang Jianhua, Zhang Yi, and Hu Jianming. A sift-based mean shift algorithm for moving vehicle tracking. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 762–767. IEEE, 2014.

[55] Wei Zhang, Bing Yu, Gregory J Zelinsky, and Dimitris Samaras. Object class recognition using multiple layer boosting with heterogeneous features. In *2005 IEEE Computer Society Conference on Computer Vision And Pattern Recognition (CVPR'05)*, volume 2, pages 323–330. IEEE, 2005.

[56] Apostolos P Psyllos, Christos-Nikolaos E Anagnostopoulos, and Eleftherios Kayafas. Vehicle logo recognition using a sift-based enhanced matching scheme. *IEEE transactions on intelligent transportation systems*, 11(2):322–328, 2010.

[57] Zhiming Qian, Jiakuan Yang, and Lianxin Duan. Multiclass vehicle tracking based on local feature. In *Proceedings of 2013 Chinese Intelligent Automation Conference*, pages 137–144. Springer, 2013.

[58] Zhenhai Wang and Kicheon Hong. A new method for robust object tracking system based on scale invariant feature transform and camshift. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, pages 132–136. ACM, 2012.

[59] Jun-Wei Hsieh, Li-Chih Chen, and Duan-Yu Chen. Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):6–20, 2014.

[60] Li-Chih Chen, Jun-Wei Hsieh, Hui-Fen Chiang, and Tsung-Hsien Tsai. Real-time vehicle color identification using symmetrical surfs and chromatic strength. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2804–2807. IEEE, 2015.

[61] BF Momin and SM Kumbhare. Vehicle detection in video surveillance system using symmetrical surf. In *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*, pages 1–4. IEEE, 2015.

[62] Tharindu D Gamage, Jayathu G Samarawickrama, and AA Pasqual. Gpu based non-overlapping multi-camera vehicle tracking. In *7th International Conference on Information and Automation for Sustainability*, pages 1–6. IEEE, 2014.

[63] Sun Shujuan, Xu Zhize, Wang Xingang, Huang Guan, Wu Wenqi, and Xu De. Real-time vehicle detection using haar-surf mixed features and gentle adaboost classifier. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 1888–1894. IEEE, 2015.

[64] Minkyu Cheon, Wonju Lee, Changyong Yoon, and Mignon Park. Vision-based vehicle detection system with consideration of the detecting location. *IEEE transactions on intelligent transportation systems*, 13(3):1243–1252, 2012.

[65] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, and David McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):748–758, 2012.

[66] Sebastian Tuermer, Franz Kurz, Peter Reinartz, and Uwe Stilla. Airborne vehicle detection in dense urban areas using hog features and disparity maps. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6 (6):2327–2337, 2013.

[67] Bing-Fei Wu, Chih-Chung Kao, Cheng-Lung Jen, Yen-Feng Li, Ying-Han Chen, and Jhy-Hong Juang. A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking. *IEEE Transactions on Industrial Electronics*, 61(8):4228–4237, 2014.

[68] Hu Huijie et al. The moving vehicle detection and tracking system based on video image. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2013 Third International Conference on*, pages 1277–1280. IEEE, 2013.

[69] Bin Tian, Ye Li, Bo Li, and Ding Wen. Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):597–606, 2014.

[70] Liang Lin, Tianfu Wu, Jake Porway, and Zijian Xu. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, 42 (7):1297–1307, 2009.

[71] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 37–44. IEEE, 2006.

[72] Derek Hoiem, Carsten Rother, and John Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[73] Sayanan Sivaraman and Mohan Manubhai Trivedi. Vehicle detection by independent parts for urban driver assistance. *IEEE transactions on intelligent transportation systems*, 14(4):1597–1608, 2013.

[74] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE, 2006.

[75] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 829–836. IEEE, 2005.

[76] Amnon Shashua, Yoram Gdalyahu, and Gaby Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 1–6. IEEE, 2004.

[77] Dariu M Gavrila and Vasanth Philomin. Real-time object detection for smart vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87–93. IEEE, 1999.

[78] Dariu M Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (8):1408–1421, 2007.

[79] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[80] Bo Wu and Ram Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[81] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[82] Yazhou Liu, Shiguang Shan, Wenchao Zhang, Xilin Chen, and Wen Gao. Granularity-tunable gradients partition (ggp) descriptors for human detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1255–1262. IEEE, 2009.

[83] Yazhou Liu, Shiguang Shan, Xilin Chen, Janne Heikkila, Wen Gao, and Matti Pietikainen. Spatial-temporal granularity-tunable gradients partition (stggp) descriptors for human detection. In *European Conference on Computer Vision*, pages 327–340. Springer, 2010.

[84] Christian Wojek and Bernt Schiele. A performance evaluation of single and multi-feature people detection. In *Joint Pattern Recognition Symposium*, pages 82–91. Springer, 2008.

[85] Greg Mori, Serge Belongie, and Jitendra Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[86] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1030–1037. IEEE, 2010.

[87] Bo Wu and Ram Nevatia. Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In *Computer vision and pattern recognition, 2008. cvpr 2008. IEEE conference on*, pages 1–8. IEEE, 2008.

[88] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39. IEEE, 2009.

[89] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[90] Sibt Ul Hussain and William Triggs. Feature sets and dimensionality reduction for visual object detection. In *BMVC 2010-British Machine Vision Conference*, pages 112–1. BMVA Press, 2010.

[91] Patrick Ott and Mark Everingham. Implicit color segmentation features for pedestrian and object detection. In *ICCV*, pages 723–730, 2009.

[92] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1713–1727, 2008.

[93] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[94] Boris Babenko, Piotr Dollár, Zhuowen Tu, and Serge Belongie. Simultaneous learning and alignment: Multi-instance and multi-pose learning. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[95] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 794–801. IEEE, 2009.

[96] Stefan Walk, Konrad Schindler, and Bernt Schiele. Disparity statistics for pedestrian detection: Combining appearance, motion and stereo. In *European Conference on Computer Vision*, pages 182–195. Springer, 2010.

[97] Piotr Dollár, Zhuowen Tu, Hai Tao, and Serge Belongie. Feature mining for image classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[98] Aharon Bar-Hillel, Dan Levi, Eyal Krupka, and Chen Goldberg. Part-based feature synthesis for human detection. In *European Conference on Computer Vision*, pages 127–142. Springer, 2010.

[99] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S Davis. Human detection using partial least squares analysis. In *2009 IEEE 12th international conference on computer vision*, pages 24–31. IEEE, 2009.

[100] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *European conference on computer vision*, pages 18–32. Springer, 2000.

[101] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[102] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *European conference on computer vision*, pages 113–127. Springer, 2002.

[103] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 878–885. IEEE, 2005.

[104] Piotr Dollár, Boris Babenko, Serge Belongie, Pietro Perona, and Zhuowen Tu. Multiple component learning for object detection. In *European conference on computer vision*, pages 211–224. Springer, 2008.

[105] Zhe Lin, Gang Hua, and Larry S Davis. Multiple instance ffeature for robust part-based object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 405–412. IEEE, 2009.

[106] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[107] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[108] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013.

[109] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 633–640. IEEE, 2013.

[110] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2189–2202, 2012.

[111] David A Forsyth, Jitendra Malik, Margaret M Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. *Finding pictures of objects in large collections of images*. Springer, 1996.

[112] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Computer Vision–ECCV 2008*, pages 30–43. Springer, 2008.

[113] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3286–3293. IEEE, 2014.

[114] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.

[115] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2): 303–338, 2010.

[116] Sayanan Sivaraman and Mohan Manubhai Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, 2010.

[117] Joel C McCall, Ofer Achler, and Mohan M Trivedi. Design of an instrumented vehicle test bed for developing a human centered driver support system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 483–488. IEEE, 2004.

[118] Mohan Manubhai Trivedi, Tarak Gandhi, and Joel McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):108–120, 2007.

[119] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[120] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[121] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[122] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollr. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[123] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Discriminatively trained deformable part models, release 4, 2010.

[124] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[125] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[126] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[127] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[128] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[129] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2014.

[130] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(4):814–830, 2016.

[131] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34 (11):2274–2282, 2012.

[132] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2536–2543. IEEE, 2013.

[133] Pekka Rantalankila, Juho Kannala, and Esa Rahtu. Generating object segmentation proposals using global and local search. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2417–2424. IEEE, 2014.

[134] Pablo Arbelaez, Jordi Pont-Tuset, Jonathan Barron, Ferran Marques, and Jagannath Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 328–335. IEEE, 2014.

[135] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Computer Vision–ECCV 2014*, pages 725–739. Springer, 2014.

[136] Ziming Zhang, Jonathan Warrell, and Philip HS Torr. Proposal generation for object detection using cascaded ranking svms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1497–1504. IEEE, 2011.

[137] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013.

[138] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations*, 2014.

[139] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2147–2154. IEEE, 2014.

[140] Christian Szegedy, Scott Reed, Dumitru Erhan, and Gomir Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[141] Nikolaos Karianakis, Thomas Fuchs, and Stefano Soatto. Boosting convolutional features for robust object proposals. *arXiv preprint arXiv:1503.06350*, 2015.

[142] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollr. Learning to segment object candidates. *arXiv preprint arXiv:1506.06204*, 2015.

[143] Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. Deepbox:learning objectness with convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on.* IEEE, 2015.

[144] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on.*

[145] 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.

[146] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[147] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representations*, 2014.

[148] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012.

[149] Xiaozhi Chen, Huimin Ma, Xiang Wang, and Zhichen Zhao. Improving object proposals with multi-thresholding straddling expansion. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2587–2595. IEEE, 2015.

[150] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015.

[151] Philipp Krähenbühl and Vladlen Koltun. Learning to propose objects. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1574–1582. IEEE, 2015.

[152] Sachin Sudhakar Farfade, Mohammad Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *ACM on International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015.

[153] Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2578–2586. IEEE, 2015.

[154] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[155] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[156] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011.

[157] Jiri Najemnik and Wilson S Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, 2005.

[158] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[159] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *TPAMI*, 31(12):2129–2142, 2009.

[160] Bogdan Alexe, Nicolas Heess, Yee W Teh, and Vittorio Ferrari. Searching for objects driven by context. In *NIPS*, 2012.

[161] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An active search strategy for efficient object class detection. In *CVPR*, 2015.

[162] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[163] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014.

[164] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[165] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*, 2015.

[166] Yongxi Lu, Tara Javidi, and Svetlana Lazebnik. Adaptive object detection using adjacency and zoom prediction. *arXiv preprint arXiv:1512.07711*, 2015.

[167] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[168] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[169] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *NIPS Workshop*, 2011.

[170] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *TPAMI*, 38(4):814–830, 2016.

[171] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.