Towards Object Detection in the Real World

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Chenchen Zhu

B.S., Electronic Information Science & Technology, Nanjing University
M.S., Electrical & Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

August 2021

Acknowledgements

My Ph.D. journey is one of the most important and memorable experiences in my life. Through this journey, I am fortunate to be guided by several individuals, each of whom has played an irreplaceable role towards the successful completion of this degree. It is my great pleasure to thank them for their contribution to my works.

First, I would like to thank my advisor and the chair of my doctoral committee, Prof. Marios Savvides, for providing me the opportunity to become a researcher and the invaluable trust in my abilities over the years. My relationship with him extends since Feb. 2014 when I joined the lab as a master intern. Under his guidance, I grow from a student who takes in knowledge to a researcher who contributes new knowledge to the community. This wouldn't be possible without his encouragement and support for research freedom. This greatly cultivates my abilities to dive deep, think critically, and find the underline pattern, which is essential for an independent researcher. Apart from the academic support, he also plays the role of a family member and friend. One thing he always says is "I take care of you." I will never forget the moments he took the lab members to several restaurants and ordered a full table of food and the birthday parties with massive cakes.

I would also like to thank my doctoral committee members, Prof. Vijayakumar Bhagavatula, Prof. John Dolan, and Dr. Saad Bedros. They have provided valuable suggestions for my thesis which have greatly improved my works and this dissertation. Their support, comments, and guidance are much appreciated. I am very fortunate to have such a distinguished thesis panel.

The CyLab Biometrics Center has always been my home stadium on campus, where I have the privilege of collaborating with some of the smartest people on several challenging research problems. I have learned a lot from them and got many inspirations from our discussions. The Ph.D. students and staff at the Center have been great collaborators and friends: Khoa Luu, Utsav Prabhu, Shreyas Venugopalan, Keshav Sheshadri, Felix-Juefei Xu, Thi Hoang Ngan (Nancy) Le, Chi Nhan Duong, Kha Gia Quach, Vishnu Boddeti, Chandrasekhar Bhagavatula, Raied Aljadany, Dipan Pal, Yutong Zheng, Uzair Ahmed, Ran Tao, Thanh Hai Phan, Yu-Kai Huang, Fangyi Chen, Han Zhang, Sreena Nallamothu, and Snow Zhang. There are also many talented master students I have the pleasure to work with: Tianhao Tang, Ligong Han, Wei Yu, Yihui He, Qi Wang, Chenxi Li, Adam Chiu, Magesh Kannan, etc. I wish you all the best in your future careers. Particularly, I would like to thank Khoa for giving me the first impression of research, Utsav for many sincere suggestions and support along my career path, Sekhar for the help with my paper and presentation as a native speaker, Yutong for giving me the sticky rubber animal toy as an effective debugging tool, and Fangyi for collaborating on several projects and League of Legends.

The CyLab and CMU staff have always been super supportive to help us with any problems and numerous last-minute tricky requests. Thank Chelsea, Tina, Ivan, Brittney, Kelley, Brigette, Nathan for being accommodating. They all made my life much easier here.

My life in Pittsburgh couldn't be extraordinarily fun without the awesome Pittsburgh United Soccer Team, a group of Chinese soccer enthusiasts consisting of students, postdocs, faculty, researchers, and engineers in Pittsburgh. Together, we have fought for many battles and won many championships and trophies, including the CMU IMLeague, the UPitt IMLeague, the Michigan Tournament, the Ohio Tournament, the PCA Spring Soccer Tournament, and the Steel City World Cup. We have been through many hard games, countless cold nights, matches in rain or snow, but we still move forward. Many thanks to Captain Dinghuan Zhu, Captain Xi Tan, Captain Xiaofeng Yu, Captain Xinji, Juli Liu, Ran Ji, Tianyao Chen, Wenda Xu, Yimu Wang, Song Yao, Wei Jin, Fangxin Li, Xiaoxi Zhao, Xiaonan Shao, Yilin Yang, Jihe Liu, Junshi Wang, Senjun Fan, Tianjun Fan, Xin Yong, Chi Zhang, Weikang Wang, Yunye Zhu, Chaosheng Dong, Yunyang Liu, Ruochen Xu, Yifan Zhu, Ng Chuen Yan, Fan Sun, Huanghao Zhang, Hanche Liu, Kuangjie Sheng, Jiayuan Li, Xiaochen Zhou, Chenxi Wang, Kai Wu, Chenge Yang, Shuowen Zhang, Rundong Lyu, etc. Thanks to them, I understand soccer and life better.

Many others contributed to this thesis indirectly, in person, over the phone, via emails across time zones. Big thanks to my friends, especially Yandong Wen, Yang Zou, Yan Xu, Chaoyang Wang, Jianxiao Ge, Shuguan Yang, Yuxiang Wang, Yun Zhou, Jian Shen, Yiming Lyu, Liyue Yu, Shi Zong, Haomin Yan, Jingxian Bao, Wen Wang, Zechun Liu, Yang Gao, Zihao Wang, Yao Li, Shumian Xin, Chao Liu, Qichen Pan, Wenbo Zhao, Zhiding Yu, Wenbo Liu, Tianlong Yu, Mengwen He, Miao Yu, Weijin Shi, Tianyi Zhang, Estelle Jiang, etc, for the wonderful memories.

One other person who has been an absolute pillar in my Ph.D. journey is Ruonan Zhang. We have been through many life adventures together, no matter high or low moments we share happiness, joy, excitement, sadness, and sorrow. Her love, support, companionship, cuteness, naughtiness, as well as surprising skills at video games have given life the color it deserves.

Finally, my parents, Sihai Zhu and Ningxian Chen are my source of inspiration in science and research. They are the reasons for who I am today. At every point in my life, they play an important role to give me ever-present support and insight. This thesis is indeed dedicated to them.

This work has been partly funded by Carnegie Mellon University Cylab, Bossa Nova Gift & Sponsored, Army Low Shot, and NAVMAR Applied Sciences Corporation grant no. NASC-0040-CMU.

Abstract

Object detection is one of the most fundamental tasks in the computer vision field, which aims at localizing and classifying instances of semantic objects of certain classes in digital images. Object detection serves as a crucial step for many downstream vision tasks such as action recognition, face analysis, instance segmentation, object re-identification, retail scene understanding, etc. Therefore, it has been carefully studied by the computer vision community for decades. Thanks to the advance of deep neural networks and well-annotated challenging datasets, object detection algorithms have been greatly improved. However, object detectors are still far from robust when deployed in real-world AI applications. The performance can drop dramatically due to the challenging conditions introduced by the varying nature of the real-world data. We summarize the majority of this varying nature as three aspects, i.e. appearance variation, scale variation, and availability variation. In some extreme cases where multiple variations co-exist, the failure of object detectors may even lead to the crash of the entire AI system.

The focus of this thesis is to construct the solutions addressing the mentioned three types of data variations. For the appearance variation, we study the effect of the context information on the detection of the human face, one of the most common objects. We propose an explicit contextual reasoning module for the detection network to capture the local information surrounding the face. For the scale variation challenge, we start with the anchor-based formulation of object detection where the anchor-object matching mechanism is theoretically investigated. This inspires us to propose several better designs of robust anchors. Then we discover the inherent limitations of anchor-based detection, leading to the reformulation of detection from an anchor-free perspective. Advanced techniques for dynamic feature selection are proposed to achieve the goal that less is more. For the availability variation, we address the inherent long-tail distribution of the real-world data by studying object detection in the few-shot setting in which there are some rare classes with only a few annotated objects available while other common classes dominate the dataset with abundant labeled samples. Given limited visual information of the rare classes, we propose semantic relation reasoning with prior knowledge from natural language to take advantage of the constant relationship between common classes and rare classes regardless of the data availability. We thoroughly analyze the effect of proposed techniques by conducting several experiments on challenging real-world datasets, such as WiderFace, VOC, COCO, etc. Comparisons with the previous state of the arts demonstrate the superiority of our methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Background

Object detection aims at understanding the scene by answering the question: what and where are the objects in the scene? The scene can be represented by 2D images or 3D data like videos or point clouds. In this thesis, we focus on the 2D RGB images, so the answers are given by a set of bounding boxes with the class labels from a close set and confidence scores. The class set is predefined which includes some objects with semantically meaningful names, such as humans, animals, vehicles, or furniture.

Object detection is an important task in the computer vision community. It serves as a prerequisite for various down-streaming vision applications such as instance segmentation [33], facial analysis [3, 110], action recognition [86], video analysis [61, 88], image captioning [42], object tracking [87], and the prediction and planning for autonomous driving cars [19, 54]. Without a proper object detector, the objects we care about in the scene cannot further enter the analysis pipeline of AI systems. Due to its importance, people have been studying it for decades. In recent years, there have been remarkable breakthroughs in the performance of object detectors thanks to the rapid development of deep learning techniques [45, 81, 34, 94] and well-annotated detection datasets [101, 22, 58] to simulate the real-world scenario.

1.1 Object Detection in the Wild - Difficulties & Challenges

Although object detection is a mature and well-defined problem, it is still challenging in the real world. It is not easy to answer the question "what are the difficulties and challenges in object detection?" because different detection tasks have different objectives and constraints. In our opinion, the challenges mainly come from the varying nature of the real-world data. We summarize the majority of this varying nature as three aspects, i.e. appearance variation, scale variation, and availability variation.

The appearance variation is about the change of appearance for a certain object class. Because we are capturing the object with 2D images, some information may be lost in the process of projecting a 3D object from the real world to 2D images. For example, the object can look very differently at given poses. And the object can be occluded by other objects in 2D images even it is far away from others in the 3D world, due to the viewpoint of the camera. When the image is taken with an out-of-focus camera, the object will have low resolution. Additionally, a given object class may include multiple subclasses so fine-grained visual differences between subclasses can exist. For the human face, its appearance in images can be largely affected by illumination, expression, makeup, etc. Object detectors are given only 2D images to learn and predict the objects, hence it is challenging to understand the appearance variation which is happening in the 3D world.

The scale variation is about the change of object size. The variation can be on the area of the object bounding box according to the distance between the object and the camera, as near big far small. It can also be the variation of the box aspect ratio. Some objects can be very flat, such as skateboards and surfboards, while others may be very thin, e.g. giraffes, forks, knives. Scale variation is a major challenge for object detection so that many important inventions and technical designs are specifically for this challenge. And the objects can appear with arbitrary sizes ranging from a few pixels to the entire image. Extensive searching in this wide range is a hard task for efficient object detection.

The availability variation is about the inherent long-tail distribution of the real-world data. In the real world, there are always some common categories that have a large number of samples. The data amount of these common categories can take up the majority of the entire dataset. On the other hand, some rare categories have very limited samples available, such as rare road conditions or rare animals. And we are unable to alleviate this situation of scarce cases by simply collecting more data, not only because the data collection of these rare categories is costly but also because we may find even rarer objects during data collection. If we directly train an object detector using this imbalanced data, the detector can be heavily biased to the common objects, i.e. it will miss the rare objects with high probability.

The above three types of variations can co-exist at the same time. In this extreme situation, a detector designed without considering those variations could easily fail the detection task, which may further lead to the failure of down-streaming vision tasks and even the crash of the entire AI system. One such example is that the self-driving car fails to detect an overturned truck on the highway because the overturned truck has an uncommon appearance due to overturned pose, small scale due to far distance, and belongs to rare road conditions that the detector never saw before. The failure of the detector caused this truck to bypass the prediction and planning modules of the self-driving car and ended up with a costly crash.

1.2 A Recent History of Object Detectors

In the past two decades, much effort has been devoted to the progress of object detection. The progress can be roughly categorized as two historical periods, i.e. pre-deep-learning period and deep-learning-based period. In this section, we briefly introduce some representative methods to highlight these periods.

In the pre-deep-learning period, most of the early object detection algorithms are built on handcrafted features. Sophisticated feature representations are designed and some speedup skills are proposed given limited computing resources. One of the most popular detection methods is the Viola-Jones (VJ) face detector [84]. It follows the sliding window principle to check all possible locations and scales in an image if any window contains a human face. To reduce the computation complexity, the VJ detector introduces several important techniques, e.g. integral image for fast Haar feature extraction, feature selection with Adaboost [26], and detection cascades for quick rejection of background windows. Later, Histogram of Oriented Gradients (HOG) [16] is proposed for pedestrian detection to extract invariant features over a dense grid of uniformly spaced cells. It is originally designed for pedestrian detection but its core ideas have been an important foundation of many computer vision applications for many years. The peak of the pre-deep-learning period is the Deformable Parts Model (DPM) [25]. The DPM solves object detection in a divide-and-conquer style where the training can be seen as learning a way to decompose an object, and the inference can be viewed as an ensemble of different detected object parts. DPM has provided many valuable insights for detector design such as mixture models, hard negative mining, bounding box regression, etc, which still have a deep influence on modern object detection approaches.

As the performance of handcrafted features became saturated, object detection also reached a plateau. Then in 2012, the renaissance of convolutional neural networks (CNNs) [45] completely changed the design pattern of computer vision algorithms. A deep convolutional neural network can learn semantically high-level and robust feature representations from large-scale datasets. Therefore, a natural idea is to replace the handcrafted features with more powerful features from CNN. This idea leads to the proposal of Regions with CNN features (RCNN) for object detection [30]. The RCNN starts with a set of object proposals by selective search [78]. And each proposal is resized to a fixed size and fed into a CNN model pretrained on the ImageNet dataset [17] to extract a feature representation. Finally, some SVM classifiers are trained to predict the classification scores of objects within each region. The RCNN represents the start of the deep learning era for object detection, and since then object detection begins to evolve at an unprecedented speed.

In the deep learning era, object detectors can be roughly divided into two categories, i.e. two-stage and single-stage. The two-stage detectors have a class-agnostic region proposal stage which outputs the potential regions that may contain objects. Then the second stage is responsible for classifying the region and refining its location. Some examples of two-stage detectors include Fast R-CNN [29], Faster R-CNN [77], R-FCN [14], FPN [56], Cascade R-CNN [5], and Libra R-CNN [68]. The single-stage detectors, on the other hand, turn the class-agnostic proposal into the class-specific one and directly output the class confidence scores and the bounding box coordinates over a dense grid. Some representatives are YOLO [75], SSD [59], RetinaNet [57], and FreeAnchor [109].

Recently, there have been works focusing on solving the detection problem in the limited data scenario. LSTD [8] proposes the transfer knowledge regularization and background depression regularization to promote the knowledge transfer from the source domain to the target domain. [20] proposes to iterate between model training and high-confidence sample selection. RepMet [41] adopts a distance metric learning classifier into the RoI classification head. FSRW [40] and Meta R-CNN [96] predict per-class attentive vectors to reweight the feature maps of the corresponding classes. MetaDet [91] leverages meta-level knowledge about model parameter generation for category-specific components of novel classes. In [23], the similarity between the few shot support set and query set is explored to detect novel objects. Context-Transformer [103] relies on discriminative context clues to reduce object confusion. TFA [90] only fine-tunes the last few layers of the detector. Two very recent papers are MPSR [92] and FSDetView [93]. MPSR develops an auxiliary branch to generate multi-scale positive samples as object pyramids and to refine the prediction at various scales. FSDetView proposes a joint feature embedding module to share the feature from base classes. However, all these methods *depend purely on visual information* and suffer from availability variation.

To provide a context for this dissertation, we show some technical details of Faster R-CNN, FPN, and RetinaNet as follows.

1.2.1 Faster R-CNN

Faster R-CNN [77] is the first anchor-based detector that proposes the concept of anchor boxes. Anchor boxes are designed for discretizing the continuous space of all possible instance boxes into a finite number of boxes with predefined locations, scales, and aspect ratios. The detection problem is then formulated as anchor-object matching in the framework of deep convolutional neural networks. Instance boxes are matched to anchor boxes based on the Intersection-over-Union (IoU) overlap. During training, each object instance is matched with one or several highly overlapped anchors. Matched anchors are trained to output high confidence scores and then regress to ground-truth boxes. During inference, objects in a testing image are detected by classifying and regressing anchors. Faster R-CNN serves as a prototype for many modern

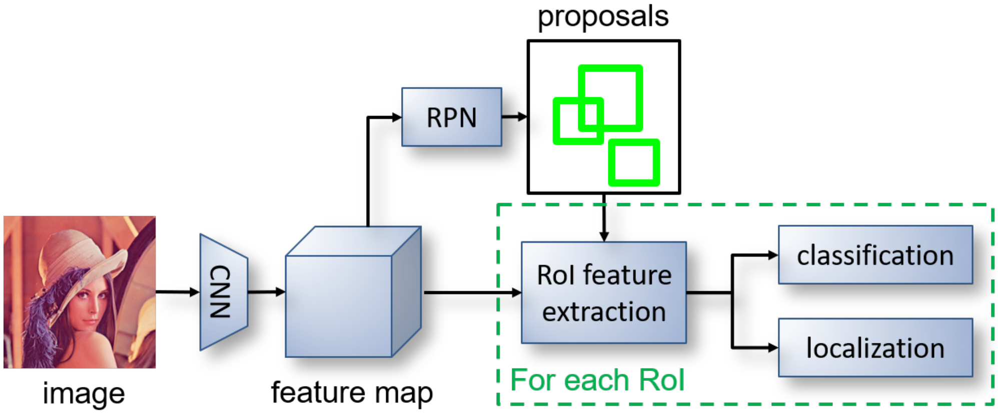


Figure 1.1: Overview of the Faster R-CNN prototype, the first anchor-based object detector.

deep-learning-based object detectors. Its frameworks have also been adopted and generalized to other methods, such as 3D object detection, instance segmentation and, image captioning.

Faster R-CNN is a single unified network for object detection, which consists of two modules as shown in Figure 1.1. The first one is a fully convolutional network termed Region Proposal Network (RPN) to propose potential regions of interest. The second module is a per-region prediction network that uses the proposed regions. Two modules share the feature maps from the convolutional layers.

Region Proposal Network. A Region Proposal Network (RPN) takes in an image of any size and predicts a set of rectangular object proposals. Inspired by the idea of the sliding window, a small network is slid over the convolutional feature map generated from the backbone model. The small network looks at an $n \times n$ spatial window at each location of the input feature map and extracts a feature vector. This feature is then fed into two sibling 1×1 convolutional layers with one for box localization (*loc*) and the other for box classification (*cls*), as shown in Figure 1.2. At each sliding window location, multiple region proposals are predicted, where the number of maximum possible proposals for each location is denoted as p. Therefore, the *cls* layer outputs p scores indicating the probability of objectness for each proposal, and the *loc* layer has $4p$ outputs encoding the coordinate deltas of p boxes. The coordinate deltas are concerning p reference boxes which are referred to as *anchors*. All anchor boxes are centered at the sliding window location. The set of these various anchor boxes is denoted as A. Duo to the nature of multiple scales (and aspect ratios) of objects, the anchor boxes are also designed as a multi-scale (and multi-aspect) pyramid as shown in Figure 1.2. This design of multi-sale anchors is a key for addressing scale variation with shared features in object detection.

Per-region prediction network. In the per-region prediction network, the proposed regions from RPN are used as input to extract per-region features from the backbone. For each region of interest (RoI),

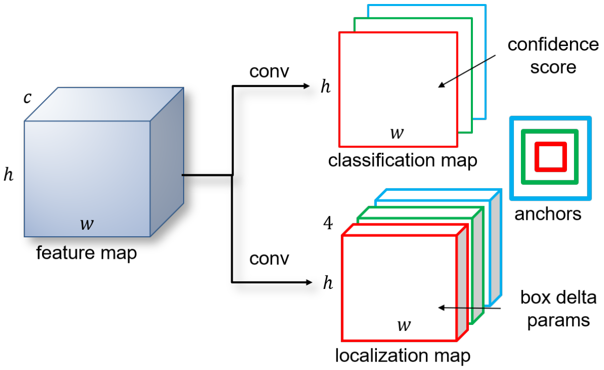


Figure 1.2: Detailed architecture of the Region Proposal Network.

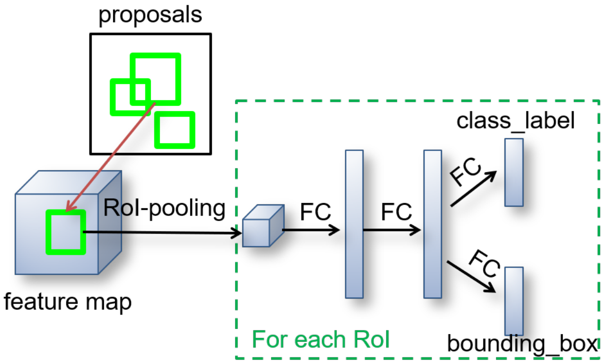


Figure 1.3: Detailed architecture of the per-region prediction network.

it is firstly projected back onto a feature map in the backbone and a feature tensor is extracted with activation functions like RoI-pooling [29] or RoI-align [33]. The extracted feature is then fed into several fully connected (FC) layers followed by two sibling FC layers. One FC layer outputs the classification probabilities of all classes for the RoI in question. The other FC layer further predicts the coordinate deltas w.r.t. the proposed region of this RoI. The whole pipeline is illustrated in Figure 1.3.

1.2.2 Feature Pyramid Network

One long-standing problem for object detection is scale variation. To achieve scale invariability, pre-deep learning approaches often adopt the image pyramid as a basic component. But recent deep learning object detectors have avoided the image pyramid partly because it is computation and memory intensive. The Feature Pyramid Network (FPN) [56] addresses this dilemma by exploiting the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. Specifically, FPN takes in a single-scale image of arbitrary size and outputs proportionally sized feature maps at multiple levels in a fully convolutional fashion. As shown in Figure 1.4, a coarser-

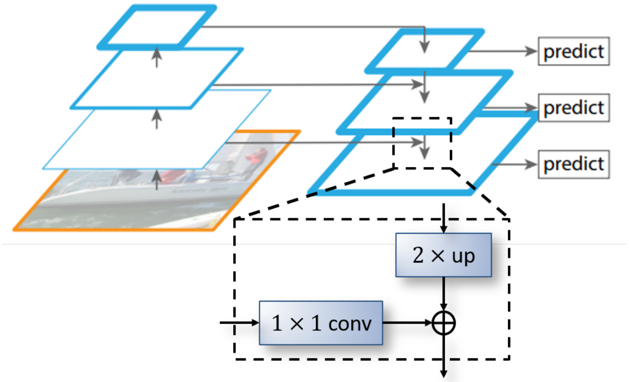


Figure 1.4: Architecture of the Feature Pyramid Network.

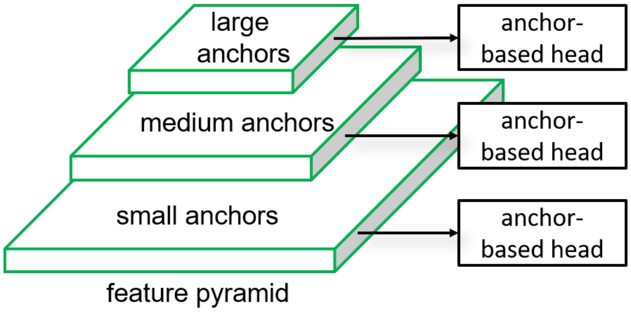


Figure 1.5: Typical way of assigning anchor boxes to multiple levels of FPN. Larger anchor boxes are associated with coarser-resolution map.

resolution feature map is upsampled by a factor of 2 using nearest neighbor upsampling. The upsampled map is then merged with the corresponding bottom-up map by element-wise addition. The bottom-up map goes through a 1×1 convolutional layer to match the channel dimension of the upsampled map. This process is iterated until the finest resolution map is generated. Predictions are independently made on all levels.

When integrating FPN with anchor boxes, large anchor boxes are typically associated with upper feature maps, and small anchor boxes are associated with lower feature maps, see Figure 1.5. This is based on the heuristic that upper feature maps have more semantic information suitable for detecting big instances whereas lower feature maps have more fine-grained details suitable for detecting small instances [32]. The design of feature pyramids integrated with anchor boxes has achieved good performance on several object detection benchmarks [22, 58, 27].

1.2.3 RetinaNet

The single-stage detectors have the advantage of simple network architecture and fast runtime speed when compared with two-stage detectors. But they are less accurate than the two-stage counterparts until the proposal of RetinaNet [57]. RetinaNet keeps the network architecture concise. The detection head is as simple as the RPN in Figure 1.2, except that the classification map for each anchor has multiple channels corresponding to multiple classes. The detection head is attached to each level of the feature pyramid network. It is discovered that the lower accuracy is caused by the extreme foreground vs. background class imbalance during training. Because the predictions are generated from a dense grid, there are a large number of negative anchor boxes. Training with conventional classification loss will make the detector bias to low confidence scores. RetinaNet addresses this with a new loss function named Focal Loss which reshapes the standard cross-entropy loss so that detector will focus more on hard and misclassified examples during training. Focal Loss enables the single-stage detectors to achieve comparable accuracy with two-stage detectors while maintaining high runtime speed.

1.3 Summary of Key Contributions

In this dissertation, we aim at constructing effective solutions to address the varying nature of the real-world data. We build upon previous state-of-the-art methods and provide an in-depth analysis of the issues in the current system. Then novel designs are proposed guided by the analysis. And thorough experiments are conducted to verify the effectiveness of the proposed components. It turns out our methods are more tolerant to the appearance variation, scale variation, and availability variation from the real world. Our key contributions can be summarized as follows:

- For the appearance variation, we propose the multi-resolution feature and the contextual reasoning module for the two-stage object detectors [118, 119]. The multi-resolution feature fuses information from multiple layers from the CNN architecture with different resolutions. And the contextual reasoning module enables explicit reference to the context information surrounding the objects during inference. We explore both fixed context regions and learnable dynamic context regions as well as different information fusion strategies and demonstrate the improvements on human faces.
- For the scale variation, we start with improving the anchor-based face detectors [117]. An in-depth analysis of the anchor matching mechanism is provided under different conditions with the newly proposed Expected Max Overlap (EMO) score to theoretically characterize anchors' ability to achieve high face IoU scores. Guided by the theoretical analysis, we propose several effective techniques of

new anchor design for higher IoU scores especially for tiny faces, which demonstrates significant improvement over the strong baseline. Our final face detection system achieves state-of-the-art performance on WiderFace, AFW, PASCAL Faces, and FDDB with competitive runtime speed.

- Further on the scale variation problem, we discover the inherent limitations of anchor-based detection and reformulate the object detection from an anchor-free perspective [116, 115]. Under the anchor-free formulation, we take advantage of its flexibility for feature selection and propose several novel strategies for dynamic feature level selection. To this end, we come up with a detection prototype with a simpler architecture, faster speed, easier training, and higher accuracy compared to the anchor-based counterpart. Our anchor-free detector demonstrates the philosophy that less is more and achieves a superior balance of accuracy vs. speed. In a graph of detection accuracy vs. runtime, the performances of our detector under different backbone models form an upper envelope of all previous single-stage or two-stage detectors.

- For the availability variation, we study object detection in the few-shot setting to address the class imbalance problem in the data with the long-tailed distribution. In addition to the limited visual information, we propose to use the semantic relation between the common classes and rare classes to aid the learning of rare classes, where the semantic relation information can be acquired from the knowledge in natural language learning with some knowledge graphs. To the best of our knowledge, we are the first work to investigate semantic relation reasoning for the few-shot detection task and show its potential to improve a strong baseline. Our few-shot detector achieves stable performance w.r.t the availability variation, outperforming state-of-the-art methods under several existing settings especially when the rare class data is extremely limited.

- For the few-shot detection setting, we also discover the flaws of existing setups in previous methods. We find the detector can get early access to rare class data through the pretrained model used for initializing the backbone of the detector because the pretrained model is trained on a large-scale classification dataset including many samples from the rare classes, which makes the rare classes not so “rare”. Therefore, we suggest a more realistic setting in which rare classes are removed from the classification dataset for the pretrained model. Our few-shot detector can maintain a more steady performance compared to previous methods if using the new pretrained model.

1.4 Notation and Organization

This thesis is structured as follows. In Chapter 2, we focus on addressing the detection challenges from the appearance variation. We motivate and present Contextual Multi-Resolution R-CNN (CMR-RCNN) for face detection in the wild. The key insight is that the multi-resolution feature and the context information surrounding the face matter a lot for handling the facial appearance variation. And CMR-RCNN demonstrates how to incorporate such information into the detector on top of the Faster R-CNN prototype. In Chapter 3, we study how to address the scale variation. We start with investigating the existing anchor-based detectors. Section 3.1.1 derives the Expected Max Overlapping score to characterize the robustness of anchor design, followed by several design strategies of robust anchor in Section 3.1.2. Section 3.2.1 interprets anchor-object matching from the feature selection's perspective. and reveals its inherent limitations, which inspires the proposal of the anchor-free detection prototype (Section 3.2.2) and several advanced feature selection methods (Section 3.2.3). In Chapter 4, we focus on addressing the availability variation in the few-shot setting. Section 4.2 introduces the standard setting for few-shot object detection and our semantic relation reasoning is proposed in Section 4.3. Chapter 5 summarizes the work with the major findings and discusses some feature research ideas at a high level.

We use the following mathematical notation for the remainder of this dissertation:

m	A scalar value
$\mathbf{m}$	A column vector
$\mathbf{M}$	A matrix

Chapter 2

Contextual Multi-Resolution R-CNN for Appearance Variation

Detection and analysis on human subjects using biometrics based on facial features for access control, surveillance systems, and other security applications have gained popularity over the past few years. Several such biometrics systems are deployed in security checkpoints across the globe with more being deployed every day. Particularly, face recognition has been one of the most popular biometrics modalities attractive to security departments. Indeed, the uniqueness of facial features across individuals can be captured much more easily than other biometrics. To take into account a face recognition algorithm, however, face detection usually needs to be done first.

The problem of face detection has been intensely studied for decades to ensure the generalization of robust algorithms to unseen face images [84, 106, 120, 51, 47, 63, 48, 64, 7, 97, 28, 55]. Although the detection accuracy in recent face detection algorithms [49, 24, 100, 73, 98, 74] has been highly improved due to the advancement of deep Convolutional Neural Networks (CNN), they are still far from achieving the same detection capabilities as a human due to many challenges from facial appearance variation. For example, off-angle faces, large occlusions, low-resolutions, and strong lighting conditions, as shown in Figure 2.1, are always the important factors that need to be considered.

This chapter presents an advanced R-CNN-based approach named Contextual Multi-Resolution Region-based CNN (CMR-RCNN) to handle the problem of face detection in digital face images collected under severe appearance variation. Our designed region-based CNN architecture allows the network to simultaneously look at multi-resolution features, as well as to explicitly look outside facial regions as the potential body regions. In other words, this process tries to mimic the way of face detection by humans in the sense that when humans are not sure about a face, seeing the body will increase our confidence. Addition-

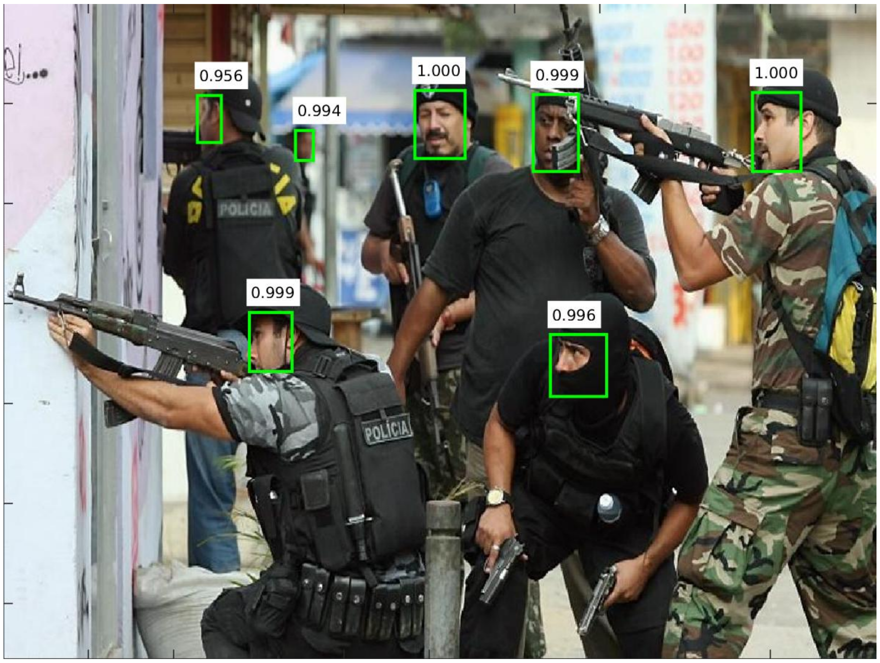


Figure 2.1: An example of face detection results using our proposed CMR-RCNN method. The proposed method can robustly detect faces across occlusion, facial expression, pose, illumination, scale and low resolution conditions from the WiderFace dataset [101].

ally, this architecture also helps to synchronize both the global semantic features in high-level layers and the localization features in low-level layers for facial representation. Specifically, our method introduces the Multi-Resolution Region Proposal Network (MR-RPN) to generate a set of region candidates and the Contextual Multi-Resolution Convolution Neural Network (CMR-CNN) to do inference on the region candidates of facial regions. A confidence score and bounding box regression are computed for every candidate.

In the experiments, we first conduct ablation studies to verify the effect of the proposed components. Then our best model is evaluated on four public face detection databases, the WiderFace [101], the Face Detection Dataset and Benchmark (FDDB) [39], the Annotated Faces in the Wild (AFW) [120] and the PASCAL Faces [99]. It is compared against many other recent face detection methods. Experimental results show that our approach outperforms previous approaches by a considerable margin and is robust when dealing with faces under extreme conditions.

2.1 Background in Deep-Learning-Based Face Detection

Apart from the general methods introduced in Section 1.2, there are many detection approaches designed especially for human faces. Li et al. [49] utilized a cascade of CNNs to perform face detection. The cascading networks allowed them to process different scales of faces at different levels of the cascade while also allowing for false positives from previous networks to be removed at later layers in a similar approach to other cascade detectors. Yang et al. [100] approached the problem from a different perspective more similar to a DPM approach. In their method, the face is broken into several facial parts such as hair, eyes, nose, mouth, and beard. By training a detector on each part and combining the score maps intelligently, they were able to achieve accurate face detection even under occlusions. Both of these methods require training several networks to achieve high accuracy. Our method, on the other hand, can be trained as a single network, end-to-end, allowing for less annotation of training data needed while maintaining highly accurate face detection.

The idea of using contextual information in object detection has been studied in several recent works with very high detection accuracy. Divvala et al. [18] reviewed the role of context in contemporary and challenging object detection in their empirical evaluation analysis. In their conclusions, the context information not only reduces the overall detection errors, but also the remaining errors made by the detector are more reasonable. Bell et al. [2] introduced an advanced object detector method named Inside-Outside Network (ION) to exploit information both inside and outside the region of interest. In their approach, the contextual information outside the region of interest is incorporated using spatial recurrent neural networks. Inside the network, skip pooling is used to extract information at multiple scales and levels of abstraction. Recently, Zagoruyko et al. [105] have presented the MultiPath network with three modifications to the standard Fast R-CNN object detector, i.e. skip connections that give the detector access to features at multiple network layers, a foveal structure to exploit object context at multiple object resolutions, and an integral loss function and corresponding network adjustment that improve localization. The information in their proposed network can flow along multiple paths. Their MultiPath network is combined with DeepMask object proposals to solve the object detection problem.

2.1.1 Limitations of Faster R-CNN

The Region-based CNN family, e.g. Faster R-CNN [77] and its variants, achieves the state-of-the-art performance results in object detection on the PASCAL VOC dataset. These methods can detect objects such as vehicles, animals, people, chairs, etc. with very high accuracy. In general, the defined objects often occupy the majority of a given image. However, when these methods are tested on the challenging Microsoft COCO dataset [58], the performance drops a lot, since images contain more small, occluded, and incomplete objects. Similar situations happen in the problem of face detection. We focus on detecting only facial regions that are sometimes small, heavily occluded, and of low resolution.

The detection network in designed Faster R-CNN is unable to robustly detect such tiny faces. The intuition point is that the Regions of Interest pooling layer, i.e. ROI-pooling layer, builds features only from the last single high-level feature map. For example, the global stride of the ′conv5′ layer in the VGG-16 model is 16. Therefore, given a facial region with the sizes less than 16×16 pixels in an image, the projected ROI-pooling region for that location will be less than 1 pixel in the ′conv5′ layer, even if the proposed region is correct. Thus, the detector will have much difficulty predicting the object class and the bounding box location based on information from only one pixel.

2.1.2 Other Face Detection Method Limitations

Other challenges in object detection in the wild include occlusion and low resolution. For face detection, it is very common for people to wear stuff like sunglasses, scarves, and hats, which occlude the face. In such cases, the methods that only extract features from faces do not work well. For example, Faceness [100] consider finding faces through scoring facial parts responses by their spatial structure and arrangement, which works well on clear faces. But when facial parts are missing due to occlusion or when the face itself is too small, facial parts become harder to detect. Therefore, the body context information plays its role. As an example of context-dependent objects, faces often come together with the human body. Even though the faces are occluded, we can still locate them only by seeing the whole human body. Similar advantages for faces at low resolution, i.e. tiny faces. The deep features can not tell much about tiny faces since their receptive field is too small to be informative. Introducing context information can extend the area to extract features and make them meaningful. On the other hand, the context information also helped with reducing false detection as discussed previously, since context information tells the difference between real faces with bodies and face-like patterns without bodies.

2.2 Contextual Multi-Resolution R-CNN

Our goal is to detect human faces captured under strong appearance variations such as strong illumination, heavy occlusion, extreme off-angles, and low resolution. Under these conditions, the current CNN-based detection systems suffer from two major problems, i.e. 1) low-res faces are hard to identify; 2) only face region is taken into consideration for classification. In this section, we show why these problems hinder the ability of a face detection system. Then, our proposed network is presented to address these problems by using the Multi-Resolution Region Proposal Network (MR-RPN) and the Contextual Multi-Resolution Convolution Neural Network (CMR-CNN), as illustrated in Figure 2.2. Similar to Faster R-CNN, the MR-RPN outputs several region candidates and the CMR-CNN computes the confidence

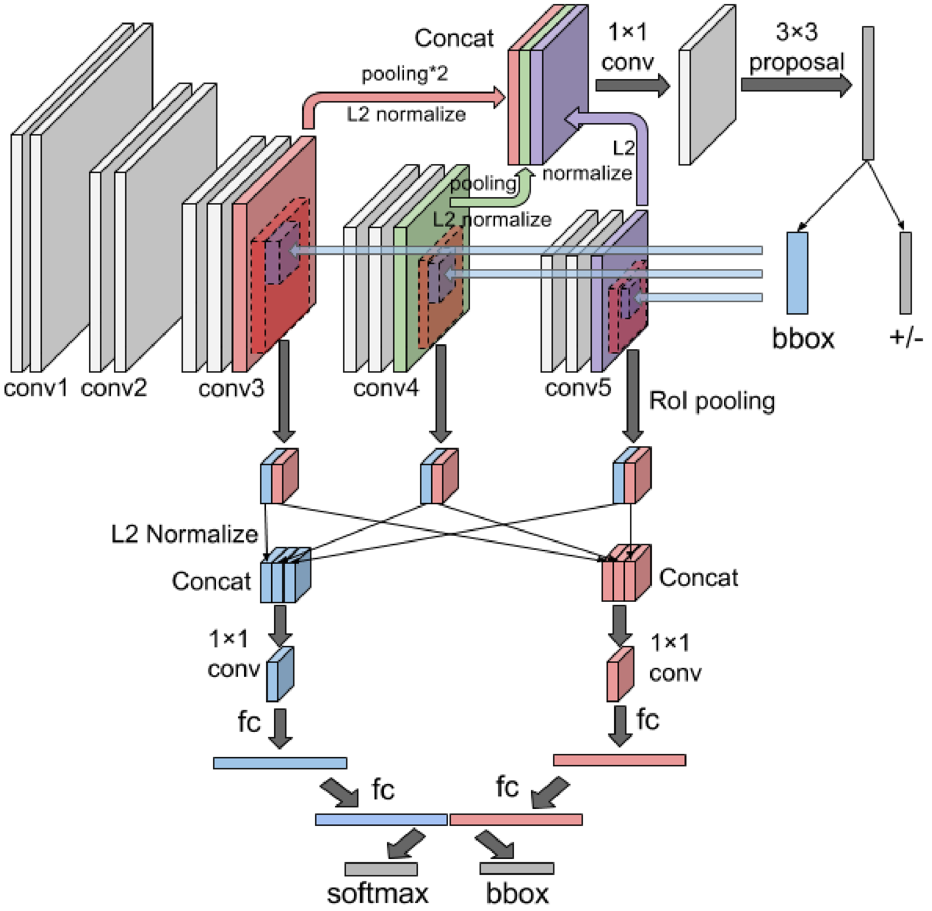


Figure 2.2: Our proposed Contextual Multi-Resolution Region-based CNN model. It is based on the VGG-16 model [81], with 5 sets of convolution layers in the middle. The upper part is the Multi-Resolution Region Proposal Network (MR-RPN) and the lower part is the Contextual Multi-Resolution Convolution Neural Network (CMR-CNN). In the CMR-CNN, the face features labeled as blue blocks and the body context features labeled as red blocks are processed in parallel and combined in the end for final outputs, i.e. confidence score and bounding box.

score and bounding box for each candidate.

2.2.1 Multi-Resolution Feature

Why low-res faces are hard to be robustly detected by the previous region-based CNNs? The reason is that in these networks both the proposed region and the classification score are produced from one single high-level convolution feature map. This representation doesn't have enough information for the multiple tasks, i.e. region proposal and RoI detection. For example, Faster R-CNN generates region candidates and does RoI-pooling from the 'conv5' layer of the VGG-16 model [81], which has an overall stride of 16. One issue is that the reception field in this layer is quite large. When the face size is less than 16-by-16 pixels, the corresponding output in the ’conv5’ layer is less than 1 pixel, which is insufficient to encode informative features. The other issue is that as the convolution layers go deeper, each pixel in the feature map gathers more and more information outside the original input region so that it contains a lower proportion of information for the region of interest. These two issues together make the last convolution layer less representative for tiny faces.

Multi-Resolution Faster-RCNN Our solution for this problem is a combination of both global and local features, i.e. multiple resolutions. In this architecture, the feature maps are incorporated from low-level convolution layers with the last convolution layer for both MR-RPN and CMR-CNN. Features from lower convolution layers help get more information for the low-res faces because stride in the lower convolution layer will not be too small. Another benefit is that both low-level features with localization capability and high-level features with semantic information are fused [32] since face detection needs to localize the face as well as to identify the face. In the MR-RPN, the whole lower-level feature maps are down-sampled to the size of the high-level feature map and then concatenated with it to form a unified feature map. Then we reduce the dimension of the unified feature map and use it to generate region candidates. In the CMR-CNN, the region proposal is projected into feature maps from multiple convolution layers. And RoI-pooling is performed in each layer, resulting in a fixed-size feature tensor. All feature tensors are normalized, concatenated, and dimension-reduced to a single feature blob, which is forwarded to two fully connected layers to compute a representation of the region candidate.

L2 Normalization In both MR-RPN and CMR-CNN, concatenation of feature maps is done with L2 normalization layer [60], shown in Fig. 2.2, since the feature maps from different layers have generally different properties in terms of numbers of channels, scales of value, and norm of feature map pixels. Generally, compared with values in shallower layers, the values in deeper layers are usually too small, which leads to the dominance of shallower layers. In practice, the system cannot readjust and tune values from each layer for best performance. Therefore, L2 normalization layers before concatenation are crucial for the robustness of the system because it keeps the value from each layer in roughly the same scale.

The normalization is performed within each pixel, and all feature map is treated independently:

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \quad \|\mathbf{x}\|_2 = (\sum_{i=1}^{d} |x_i|)^{\frac{1}{2}} \tag{2.1}$$

where the $\mathbf{x}$ and $\hat{\mathbf{x}}$ stand for the original pixel vector and the normalized pixel vector respectively. d stands for the number of channels in each feature map tensor.

During training, scaling factors γ_i will be updated to readjust the scale of the normalized features. For each channel i, the scaling factor follows:

$$y_i = \gamma_i \hat{x}_i \tag{2.2}$$

where y_i stand for the re-scaled feature value.

Following the back-propagation and chain rule, the update for scaling factor γ is:

$$\frac{\partial l}{\partial \hat{\mathbf{x}}} = \frac{\partial l}{\partial \mathbf{y}} \cdot \gamma \frac{\partial l}{\partial \mathbf{x}} = \frac{\partial l}{\partial \hat{\mathbf{x}}} \left(\frac{\mathbf{I}}{\|\mathbf{x}\|_2} - \frac{\mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|_2^3} \right) \frac{\partial l}{\partial \gamma_i} = \sum_{y_i} \frac{\partial l}{\partial y_i} \hat{x}_i \tag{2.3}$$

where $\mathbf{y} = [y_1, y_2, ..., y_d]^T$.

The system integrates information from feature maps of lower layers, i.e. third and fourth convolution layers, to extract determinant features for low-res faces. For both parts of our system, i.e. MR-RPN and CMR-CNN, the L2 normalization layers are inserted before concatenation of feature maps from the three layers. The features were re-scaled to proper values and concatenated to a single feature map. We specially set the initial scaling factor, following two rules. First, the average scale for each feature map is roughly identical; second, after the following 1 × 1 convolution, the resulting tensor should have the same average resolution as the conv5 layer in the work of Faster R-CNN. As implied, after the following 1 × 1 convolution, the tensor should be the same as the original architecture in Faster R-CNN, in terms of its size, the scale of values, and function for the downstream process.

2.2.2 Integrating Body Context

When humans are searching for faces, they try to look for not only the facial patterns, e.g. eyes, nose, mouth but also the human bodies, such as hair, shoulders, torso, etc, as the supporting information. Sometimes a human body makes us more convinced about the existence of a face. In addition, sometimes the human body helps to reject false positives. If we only look at face regions, we may make mistakes identifying them. For example, Figure 2.3 shows two cases where body region plays a significant role for correct detection. This intuition is not only true for humans but also valid in computer vision. Previous research has shown that contextual reasoning is a critical piece of the object recognition puzzle, and that context not only reduces the overall detection errors but, more importantly, the remaining errors made by the detector are more reasonable [18]. Based on this intuition, our network is designed to make explicit reference to the human body context information in the RoI detection.

In our proposed network, the contextual body reasoning is implemented by explicitly grouping body information from convolution feature maps shown as the red blocks in Figure 2.2. Specifically, additional RoI-pooling operations are performed for each region proposal in convolution feature maps to represent the body context features. Then same as the face feature tensors, these body feature tensors are normalized, concatenated, and dimension-reduced to a single feature blob. After two fully connected layers, the final body representation is concatenated with the face representation. They together contribute to the computation of confidence score and bounding box regression.

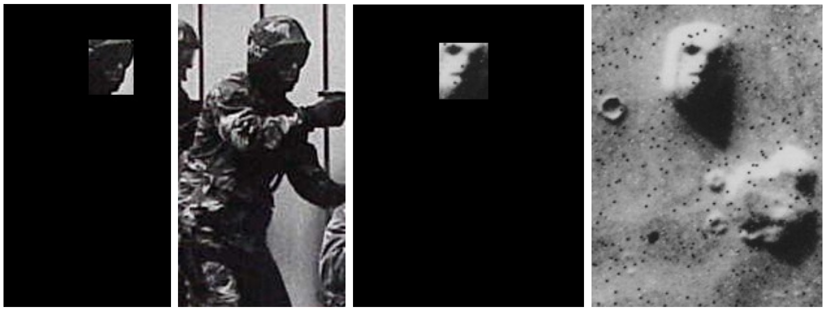


Figure 2.3: Examples of body context helping face identification. The first two figures show that existence of a body can increase the confidence of finding a face. The last two figures show that what looks like a face turns out to be a mountain on the planet surface when we see more context information.

With the projected region proposal as the face region, the additional RoI-pooling region represents the body region and satisfies a certain spatial relation with the face region. A natural question to ask is: how to model the one-to-one spatial relation between the face region and the body region? We borrow the idea from the bounding box regression in Faster R-CNN [77] to parameterize this spatial relation. The body regions are parameterized as box offsets w.r.t. their associated face regions. Mathematically, this spatial relation can be represented by four parameters presented in Equation (2.4).

$$\begin{aligned} t_x &= (x_b - x_f)/w_f \\ t_y &= (y_b - y_f)/h_f \\ t_w &= \log(w_b/w_f) \\ t_h &= \log(h_b/h_f) \end{aligned} \tag{2.4}$$

where $x_{(*)}$, $y_{(*)}$, $w_{(*)}$, and $h_{(*)}$ denote the two coordinates of the box center, width, and height respectively. b and f stand for body and face respectively. t_x, t_y, t_w, and t_h are the parameters.

To decide the four spatial parameters, we propose two approaches. One is Fixed Body Context where the parameters are fixed throughout the experiments. The other is Dynamic Body Context, where the parameters are generated by a sub-network trained jointly with the detector.

Fixed Body Context In this setting, we make a simple hypothesis that if there is a face, there must exist a body, and the spatial relationship between each face and body is fixed. This assumption may not be true all the time but should cover most of the scenarios since most people we see in the real world are either standing or sitting. Therefore, the spatial relation is roughly fixed between the face and the vertical body. To choose a fixed set of parameters, we go through all the person instances in the PASCAL-Part dataset [11]. Then face boxes and body boxes are tightly retrieved from the person instances with visible face and torso parts. Note that the face box encloses only the face part while the body box encloses both the face and the torso parts. We ignore the arms and legs because they may be self-occluded. For each face and

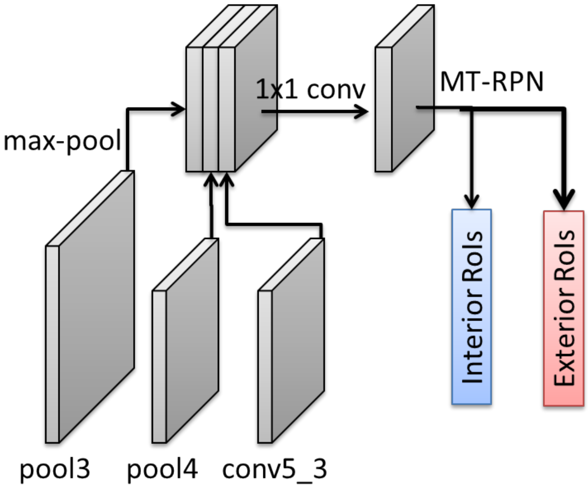


Figure 2.4: The architecture of Multi-Task RPN which outputs both interior and exterior region candidates for face-body pairs.

body box pair, parameters are computed using Equation (2.4). In the end, all parameters are averaged across all the instances. Eventually, we have $t_x = 0.07$, $t_y = 1.53$, $t_w = 0.95$, and $t_h = 1.34$.

Dynamic Body Context In the dynamic setting, the special parameters are predicted from the detector. This is realized by modifying the MR-RPN into a multi-task version, which is termed as Multi-Task Region Proposal Network (MT-RPN). The MT-RPN generates both interior and exterior region candidates in a multi-task fashion from the shared multi-resolution features as shown in Figure 2.4. The interior region and the exterior region correspond to the face region and the body region respectively. The multi-task fashion allows two tasks to benefit each other [74]. Note that MT-RPN still relies on the multi-resolution feature as described in Section 2.2.1.

The MT-RPN essentially predicts the spatial parameters defined in Equation (2.4) for the exterior region candidates. There is no ground truth of body boxes so no direct supervision exists for exterior regions. We propose to learn the spatial parameters in a weakly supervised manner. Specifically, we take the loss for face detection from the very end of the detector and back-propagate the gradients back to the conv layer for spatial parameters. This can be interpreted as the detector is trained to estimate the informative body context region for each face candidate, aiming at minimizing the network loss for face detection. The back-propagation rules are straightforward as presented in Equation (2.5) derived from Equation (2.4).

$$\begin{aligned}
\frac{\partial l}{\partial t_x} &= \frac{\partial l}{\partial x_b} w_f \\
\frac{\partial l}{\partial t_w} &= \frac{\partial l}{\partial w_b} w_f \exp(t_w) \\
\frac{\partial l}{\partial t_y} &= \frac{\partial l}{\partial y_b} h_f \\
\frac{\partial l}{\partial t_h} &= \frac{\partial l}{\partial h_b} h_f \exp(t_h)
\end{aligned} \tag{2.5}$$

where $\frac{\partial l}{\partial x_b}$, $\frac{\partial l}{\partial y_b}$, $\frac{\partial l}{\partial w_b}$, and $\frac{\partial l}{\partial h_b}$ are back-propagated from the RoI-pooling layer. During training, the spatial parameters are initialized as the values in Fixed Body Context.

2.3 Experiments

This section presents the experimental results in face detection. We first conduct the ablative study to analyze the effectiveness of each of our proposed components. Then we evaluate the final model on common face detection benchmarks. Finally, we provide the reference time.

2.3.1 Ablation Study

To investigate the effect of each component in our network, we conduct several ablation experiments. All models are trained on the WiderFace [101] training set and evaluated on the validation set with 3,226 images. The validation images are divided into three parts based on their detection rates on EdgeBox [121]. In other words, face images are divided into three levels according to the difficulties of the detection, i.e. Easy, Medium and Hard. The Hard level includes more challenging faces such as small faces or heavily occluded faces. The performance comparison are presented in Table 2.1.

Baseline. Our detector is a generalization of Faster R-CNN [77], so we directly train a slightly modified version as our baseline method. Different from [77], we only use square anchors, i.e. the aspect ratio of all anchors is set to 1.0. All other hyper-parameters are kept the same. It can deliver decent performance on Easy and Medium cases, but unsatisfying results on the Hard level (shown as the first entry of Table 2.1).

The effect of multi-resolution feature. Then we apply the multi-resolution feature (MRF). This is implemented by adding multi-scale features in the region proposal network and the Fast R-CNN stage as mentioned in Section 2.2.1, which improves the performance on all three levels (shown as the second entry of Table 2.1). Especially, we observe about 20% absolute improvement on the Hard level. This suggests that the multi-resolution feature helps to find more challenging faces.

Table 2.1: Ablation studies on the easy, medium, and hard level of **Wider Face validation set**. Numbers are the average precision scores. MRF: multi-resolution feature; Context: body context (fixed or dynamic); MST: multi-scale testing.

Methods	Easy	Medium	Hard
baseline	84.3%	73.1%	40.9%
+MRF	88.7%	86.1%	61.2%
+MRF +Context (Fixed)	90.2%	87.4%	64.3%
+MRF +Context (Dynamic)	**90.6%**	**88.3%**	**66.1%**
HR-VGG16 [38]	86.2%	84.4%	74.9%
baseline +MRF +Context (Dynamic) +MST	**91.6%**	**89.8%**	**79.2%**

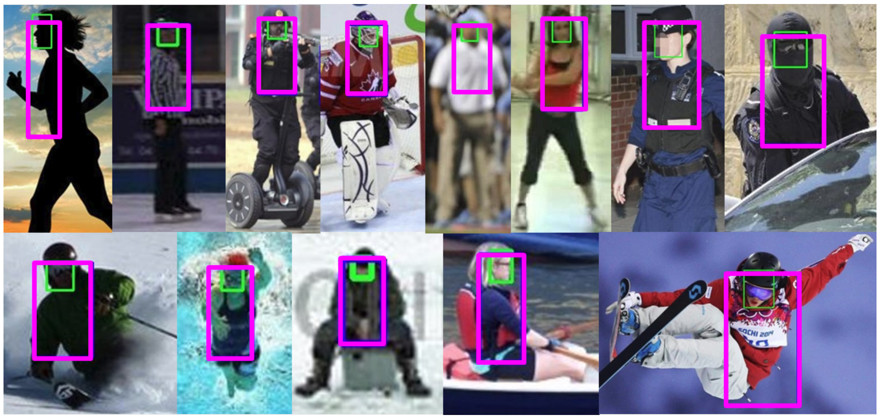


Figure 2.5: Visualization of body context regions (magenta) predicted by the network. Green boxes are the detection of faces.

The effect of body context. Next, we incorporate the body context. This is implemented by predicting additional body region proposals from the region proposal network and fusing the face and body features. We observe another improvement on all three levels no matter whether the context is fixed or dynamic, especially the Hard case (shown as the third and fourth entries of Table 2.1). This is clear evidence that explicit reference to exterior body features can support the detection of challenging faces. Additionally, the dynamic body context gives the detector more freedom to search for better body regions, thus yielding another improvement over the fixed body context. To better understand how body context features help improve the performance, we visualize all the interior and exterior region pairs from the dynamic body context, i.e. the face and body box pairs, predicted by the network. Some examples of challenging faces and their corresponding body regions are illustrated in Figure 2.5. It shows the body regions roughly focus on the face and torso parts and adapt across instances. This means the features from the torso part provide visual cues for finding hard faces.

The effect of multi-scale testing. Inspired by [38], we apply the multi-scale testing technique (MST) to our method and compare it with the VGG version of [38] termed as HR-VGG16. We rescale each image to 0.5X, 1.0X, and 2.0X and run our detector. The detection results are gathered from three images and combined. We use non-maximum suppression with a threshold of 0.3 to remove redundant detections. This gives another performance boost, outperforming [38] on all three cases (see 5th and 6th entries of Table 2.1). This indicates that our method can find more faces on various scale levels.

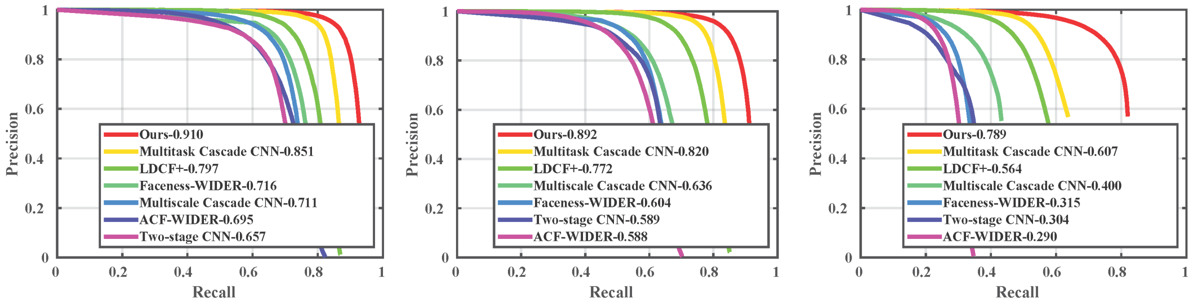
2.3.2 Benchmark Evaluation

We run our CMR-RCNN with multi-resolution feature, body context and multi-scale testing on the WiderFace dataset [101]. Under this database, our approach robustly outperforms strong baseline methods, including Two-stage CNN [101], Multiscale Cascade CNN [101], Faceness [100] and Aggregate Channel Features (ACF) [97], Multitask Cascade CNN [107] by a considerable margin.

We also show that our CMR-RCNN trained on the WiderFace dataset generalizes well to other standard face detection datasets including the AFW [120], the PASCAL faces [99], and the FDDB [39]. Our detector *without MST* alone is able to consistently achieve state-of-the-art results against other popular face detection methods, including MTCNN [107], Conv3D [53], HyperFace [74], DP2MFD [73], CCF [98], Faceness [100], NPDFace [55], MultiresHPM [28], DDFD [24], CascadeCNN [49], ACF [97], Pico [63], HeadHunter [64], Joint Cascade [7], Boosted Exemplar [48], and PEP-Adapt [47], Face++ [111], SURF Cascade multiview [50], XZJY [79].

WiderFace dataset. WiderFace is a public face detection benchmark dataset [101]. It contains 393,703 labeled human faces from 32,203 images collected based on 61 event classes from the Internet. The database has many human faces with a high degree of pose variation, large occlusions, low-resolutions, and strong lighting conditions. The images in this database are organized and split into three subsets, i.e. training, validation, and testing. Each contains 40%, 10%, and 50% respectively of the original databases. The images and the ground-truth labels of the training and the validation sets are available online for experiments. In the testing set, only the testing images are available online. All detection results are sent to the database server for evaluating and receiving the Precision-Recall curves. In our face detection experiments, our model is trained on the training set of the WiderFace dataset containing 159,424 annotated faces collected in 12,880 images. The trained model on this database is used in the testing of all databases *without further fine-tuning*.

Testing and Comparison. We run our CMR-RCNN on the Wider Face testing set. Our proposed method is compared against all published methods , i.e. Multitask Cascade CNN [107], Two-stage CNN [101], Multiscale Cascade CNN [101], Faceness [100], and Aggregate Channel Features (ACF) [97]. All these methods are trained and tested following the same evaluation protocols. We don't compare with [38] because it doesn't report the performance of the VGG version detector on the testing set. The Precision-



(a) Easy level (b) Medium level (c) Hard level

Figure 2.6: Precision-Recall curves obtained by our proposed method (red) and the other published strong baselines on the **WiderFace testing set**. Numbers show the average precision scores.

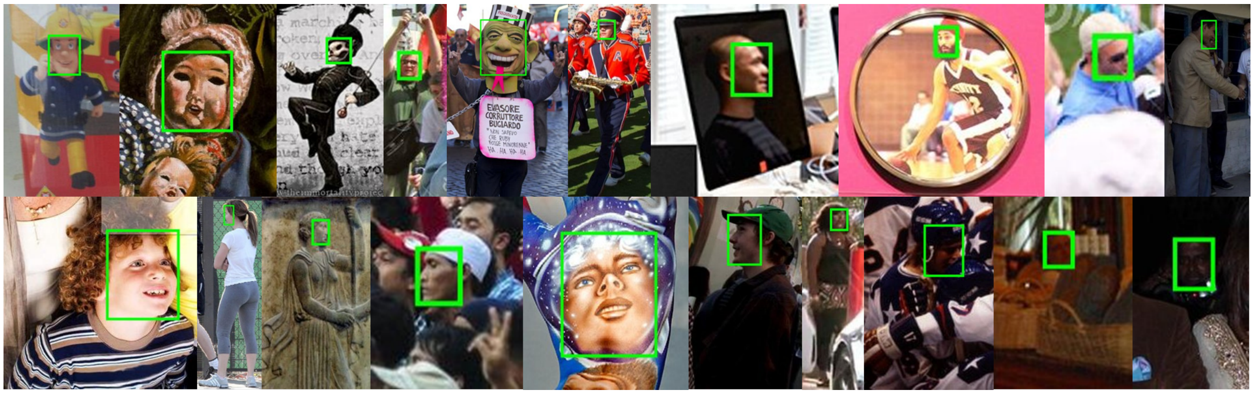
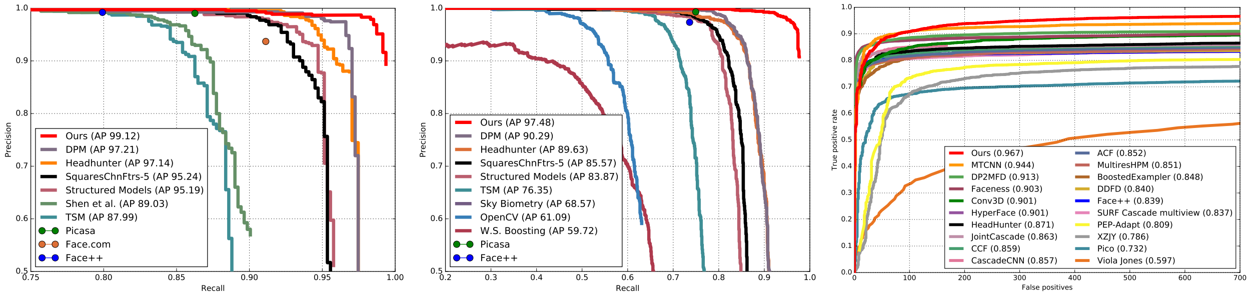


Figure 2.7: Examples of the top 20 false positives from our model tested on the WiderFace validation set. In fact these false positives include many human faces not in the dataset due to mislabeling, which means that our method is robust to the noise in the data.

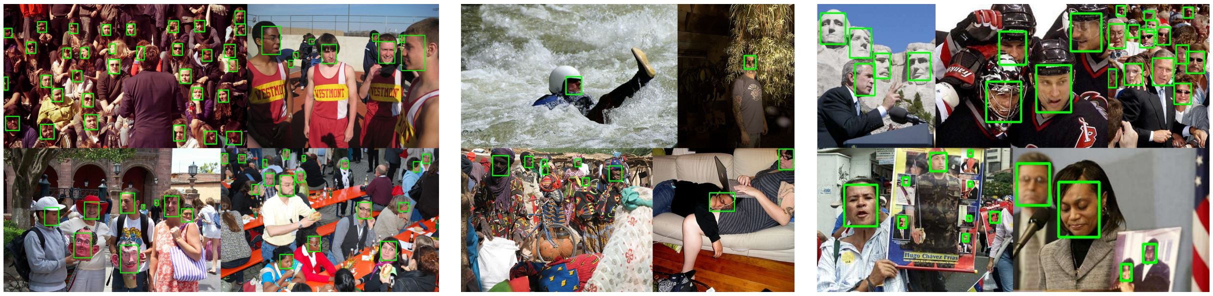
Recall curves and average precision scores are shown in Figure 2.6. Our method outperforms those strong baselines by a considerable margin. It achieves the best average precision in all level faces and outperforms the second-best baseline by 7.06% (Easy), 8.78% (Medium), and 29.98% (Hard). These results suggest that as the difficulty level goes up, our model can detect challenging faces better. So it can handle difficult conditions hence is more closed to human detection performance.

Visualization of False Positives. As it is well known that precision-recall curves degrade due to false positives, we are interested in the false positives produced by our model. We are curious about what object can fool our model to treat it as a face. Is it due to over-fitting, data bias, or miss labeling? To visualize the false positives, we test our model on the Wider Face validation set and pick all the false positives according to the ground truth. Then those positives are sorted by the confidence score in descending order. We choose the top 20 false positives as illustrated in Figure 2.7. Because their confidence scores are high, they are the objects most likely to cause our model to make mistakes. It turns out that most of the false positives are real human faces caused by miss labeling in the dataset. For other false positives, we



(a) AFW (b) PASCAL Faces (c) FDDB

Figure 2.8: Comparison between our method (red) and popular state-of-the-art methods on the AFW, PASCAL Faces, and FDDB datasets.



(a) AFW (b) PASCAL Faces (c) FDDB

Figure 2.9: Some examples of face detection results using our approach on the AFW, PASCAL Faces, and FDDB databases.

find the errors made by our model are rather reasonable. They all have the pattern of the human face as well as the shape of the human body.

AFW, PASCAL Faces and FDDB datasets. To show that our method generalizes well to other databases, the proposed method is also benchmarked on three challenging face detection datasets, i.e. AFW [120], PASCAL Faces [99] and FDDB database [39]. The AFW dataset consists of 205 images with 473 annotated faces. The PASCAL Faces dataset is a subset of the PASCAL VOC [22] testing set. It has 851 images with 1,335 annotated faces. The FDDB dataset has 2,845 images with 5,171 annotated faces. Instead of rectangle boxes, faces in FDDB are represented by ellipses. So we learn a regressor to transform rectangle boxes predicted by our detector to ellipses. Again, we test on the original images without rescaling. Our CMR-RCNN *without* MST can already achieve state-of-the-art performance. The performance curves are generated using the tool from [64].

On the PASCAL Faces and AFW datasets we compare with DPM [64], HeadHunter [64], SquaresChnFtrs-5 [64], Structured Models [95], Shen et al. [79], TSM [120], Picasa, Face++ [111].

On FDDB dataset we compare with several published methods including MTCNN [107], Conv3D [53], HyperFace [74], DP2MFD [73], CCF [98], Faceness [100], NPDFace [55], MultiresHPM [28], DDFD [24], CascadeCNN [49], ACF [97], Pico [63], HeadHunter [64], Joint Cascade [7], Boosted Exemplar [48], and PEP-Adapt [47], Face++ [111], SURF Cascade multiview [50], XZJY [79].

The proposed method outperforms all previous methods as shown in Figure 2.8. This is concrete evidence to demonstrate that our method robustly detects unconstrained faces. Figure 2.9 shows some examples of the face detection results on three datasets.

We present more qualitative results in Figure 2.10 and 2.11.

2.3.3 Inference Time

During inference, our method is running on a single Titan X GPU machine with Intel Core i7-6700 CPU @ 3.40GHz in a batch size of 1. It takes 0.168s per frame running on WiderFace or AFW (XGA quality) and 0.045s per frame on FDDB or PASCAL Faces (VGA quality).

2.4 Summary

This chapter presents our proposed Contextual Multi-Resolution R-CNN (CMR-RCNN) for unconstrained face detection. The CMR-RCNN introduces the multi-resolution feature and the explicit body context reasoning into the conventional two-stage detectors. The superior performance on four face detection datasets shows its capability to extract robust facial feature representation which is tolerant to strong appearance variation in the real-world scenario. The multi-resolution feature greatly improves the detection of faces with low-resolution degradation and the body context reasoning further enhances the facial representation with contextual clue.

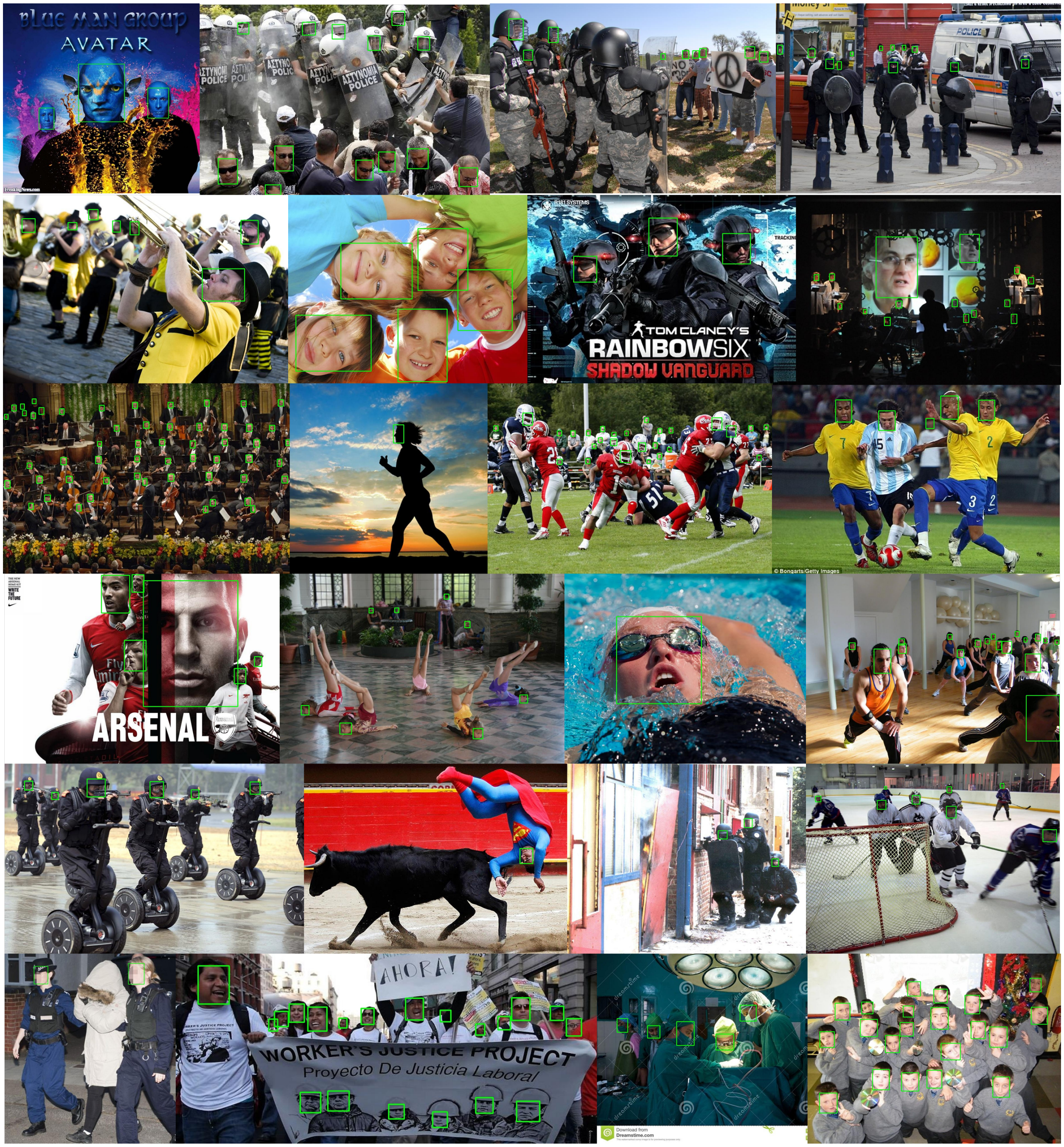


Figure 2.10: Examples of face detection by our approach on the WiderFace dataset. Green rectangles denote the face bounding boxes.

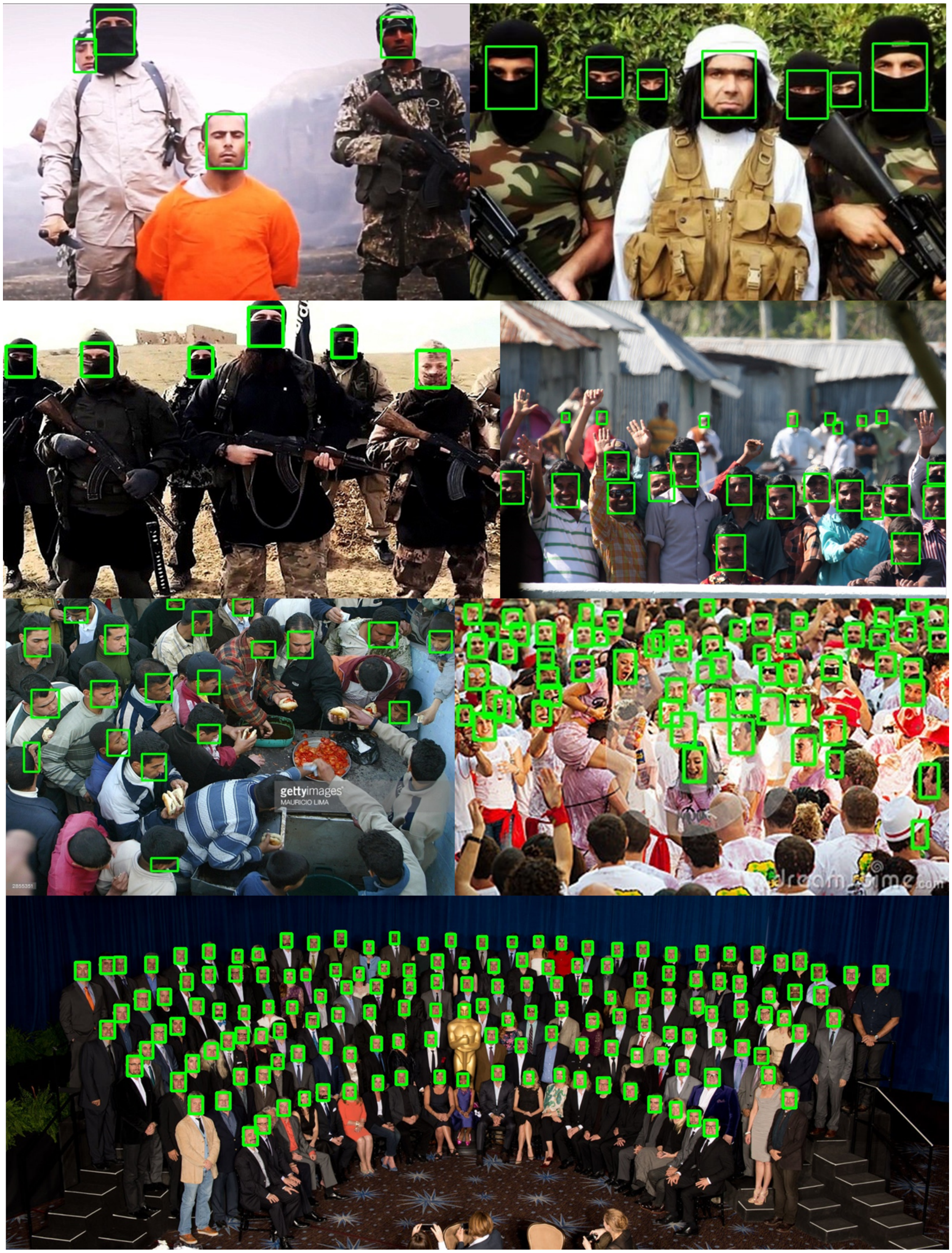


Figure 2.11: Examples of face detection by our approach on some challenging images collected from the Internet. Green rectangles denote the face bounding boxes.

Chapter 3

Scale-Tolerant Detection from Anchor-Based to Anchor-Free

A long-lasting challenge in object detection is the scale variation, meaning the objects can appear with arbitrary size in the image. Ideally, a good object detector should be able to detect all the objects no matter what the scale, but many methods struggle with the scale variation. Indeed, scale variation is a major challenge for object detection so that many important inventions and technical designs are specifically for this challenge.

In the deep learning era, anchor-based detectors are dominating the field [77, 59, 75, 56, 57, 118, 119, 117, 35]. Anchor boxes are designed to deal with scale variation by discretizing the continuous space of all possible instance boxes into a finite number of boxes with predefined locations, scales, and aspect ratios. The detection problem is then formulated as anchor-object matching in the framework of deep convolutional neural networks. Instance boxes are matched to anchor boxes based on the Intersection-over-Union (IoU) overlap. During training, each object instance is matched with one or several highly overlapped anchors. Matched anchors are trained to output high confidence scores and then regress to ground-truth boxes. During inference, objects in a testing image are detected by classifying and regressing anchors.

However, anchor-based detectors are not as scale-invariant as claimed, especially for some extremely small or large objects. The focus of this chapter is to improve the object detector's ability to deal with severe scale variation. To achieve our goal, we start with enhancing anchor-based object detectors by theoretically investigating the anchor-object matching mechanism. The theory inspires us to propose several designs of robust anchor, as well as reveals the inherent limitations of anchor matching from the feature selection's perspective. Therefore, we reformulate the object detection problem from the anchor-free viewpoint. With the anchor-free formulation, we come up with a detection prototype with a simpler architecture, fewer parameters, and easier training compared to the anchor-based counterpart. Under the anchor-free formulation, we discover the importance of advanced feature selection based on object appearance instead of heuristics.

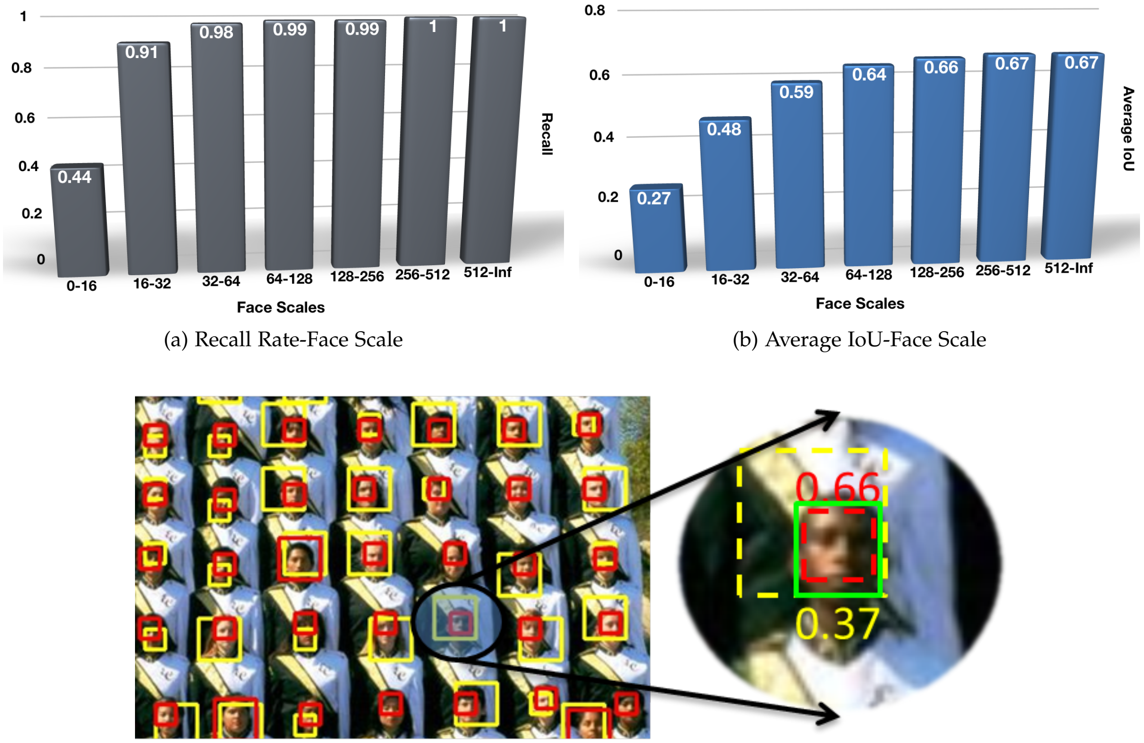
3.1 Improving Anchor-Based Object Detection

We start with analyzing anchor-based detectors and identifying the issues with anchor boxes when handling scale variation. For simplicity and without the loss of generalization, we study the detection of human faces which do not have high shape and appearance variation.

When applying anchor-based methods for face detection, their capability in handling various scales is not satisfactory, i.e. the performance drops drastically on faces with tiny sizes like less than 16×16 pixels. Figure 3.1a shows the face recall rate of a baseline anchor-based detector across different face scales on the Wider Face dataset [101]. While big faces (larger than 64×64 pixels) can be almost 100% recovered, there is a significant drop in recall for smaller faces, especially those with less than 16×16 pixels. In other words, after classifying and adjusting anchor boxes, the new boxes with high confidence scores are still not highly overlapped with enough small faces. This suggests that we look at how anchors are overlapped with faces initially before training. For each face, we compute its highest IoU with overlapped anchors. Then faces are divided into several scale groups. Within each scale group, we compute the averaged highest IoU score, as presented in Figure 3.1b. It's not surprising to find that average IoUs across face scales are positively correlated with the recall rates. We argue that anchors with low IoU overlap with small faces are harder to be adjusted to the ground truth, resulting in the low recall of small faces.

In this section, we focus on new anchor design to support anchor-based detectors for better scale tolerance. Our newly proposed anchors have higher IoU overlaps with faces than the baseline anchors, as shown in Figure 3.1c. Therefore, it is easier for the network to learn how to adjust the new anchors (red boxes) to ground-truth faces than the original anchors (yellow boxes). To achieve this, we look deep into how faces are matched to anchors with various configurations and propose the Expected Max Overlapping (EMO) score to characterize anchors' ability to achieve high IoU overlaps with faces. Specifically, given a face of known size and a set of anchors, we compute the expected max IoU of the face with the anchors, assuming the face's location is drawn from a 2D distribution on the image plane. The EMO score theoretically explains why larger faces are easier to be highly overlapped by anchors and serves as a guide for our anchor design principle.

The EMO score enlightens several simple but effective strategies of new anchor design for higher face



(a) Recall Rate-Face Scale

(b) Average IoU-Face Scale

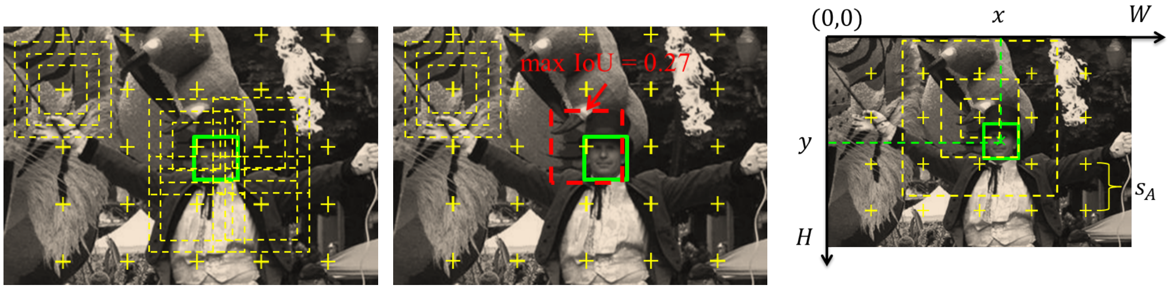
(c) Baseline anchors (yellow) vs. our anchors (red) with higher face IoUs

Figure 3.1: Problems of current anchor-based detectors and our solution. **(a):** A baseline anchor-based detector, trained and evaluated on the WiderFace dataset (see Section 3.1.3 for details), has significantly lower recall rate at IoU of 0.5 on tiny faces (16×16) than larger faces. **(b):** Maximum IoU with anchors is computed for each face and averaged in each scale group, showing a positive correlation with the recall rate across the scales. **(c):** Visualization of the anchor boxes with the highest IoUs for each face. Our anchors have much higher IoU with faces than the baseline anchor. The right side shows an enlarged example. (Best viewed in color)

IoU scores without introducing much complexity to the network. Specifically, we propose to reduce the anchor stride with various network architecture designs. We also propose to add anchors shifted away from the canonical center so that the anchor distribution becomes denser. In addition, we propose to stochastically shift the faces to increase the chance of getting higher IoU overlaps.

3.1.1 Expected Max Overlapping Score

Recap of anchor-based detectors. Anchor-based detection methods classify and regress anchor boxes to detect objects. Anchors are a set of pre-defined boxes with multiple scales and aspect ratios tiled regularly on the image plane. During training, anchors are matched to the ground-truth boxes based on the IoU overlap. An anchor will be assigned to a ground-truth box if a) its IoU with this box is the highest than other anchors, or b) its IoU is higher than a threshold T_h. An anchor will be labeled as background if its



(a) Anchor setup and distribution (b) Anchor matching mechanism (c) Computing the EMO score

Figure 3.2: **(a):** Anchors are a set of boxes with different sizes (yellow dashed boxes) tiled regularly (centered on "+" crosses) on the image plane. A face (green) overlaps with multiple anchors. **(b):** A face is matched to an anchor with the max IoU score. The matched anchor is highlighted as the red dashed box. **(c):** The EMO score characterizes the anchors' ability to capture a face by computing the expected max IoU of the face with anchors w.r.t. the distribution of the face's location (Best viewed in color).

IoU overlap scores with all boxes are lower than a threshold T_l.

Anchors are associated with certain feature maps which determine the location and stride of anchors. A feature map is a tensor of size $c \times h \times w$, where c is the number of channels, h and w are the height and the width respectively. It can also be interpreted as c-dimensional representations corresponding to $h \cdot w$ sliding-window locations distributed regularly on the image. The distance between adjacent locations is the feature stride s_F and decided by $\frac{H}{h} = \frac{W}{w} = s_F$. Anchors take those locations as their centers and use the corresponded representations to compute confidence scores and bounding box regression. So, the anchor stride s_A is equivalent to the feature stride, i.e. $s_A = s_F$.

Anchor setup and matching. We consider the standard anchor setup as shown in Figure 3.2. Formally, let S be a pre-defined scale set representing the scales of anchor boxes, and R be a pre-defined ratio set representing the aspect ratios of anchor boxes. Then, the number of different boxes is $|S \times R| = |S||R|$, where $\times$ is the Cartesian product of two sets and $| * |$ is the set's cardinality. For example, anchor boxes with 3 scales and 1 ratio are shown as the yellow dashed rectangles in the top-left corner in Figure 3.2a. Let L be the set of regularly distributed locations shown as the yellow crosses "+", with the distance between two adjacent locations as anchor stride s_A. Then the set of all anchors A is constructed by repeatedly tiling anchor boxes centered on those locations, i.e. $A = S \times R \times L$.

Given a face box B_f shown as the green rectangle, it is matched to an anchor box B_a with the max IoU overlap shown as the dashed red rectangle (Figure 3.2b). The max IoU overlap is computed as Equation (3.1).

$$\max_{a \in A} \frac{|B_f \cap B_a|}{|B_f \cup B_a|} \tag{3.1}$$

where $\cap$ and $\cup$ denote the intersection and union of two boxes respectively.

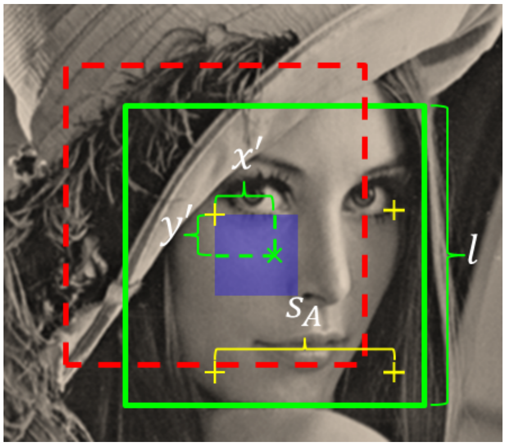


Figure 3.3: The EMO score is a function of face scale l and anchor stride s_A (Best viewed in color).

Computing the EMO score. A face can randomly appear at any location on the image plane $W \times H$, where H and W are the height and width respectively. In Figure 3.2c, we denote the center point of a face as a pair of random variables (x, y), i.e. the green cross "×". Let $p(x, y)$ be the probability density function of the location of the face, it then satisfies $\int_0^H \int_0^W p(x, y)dxdy = 1$. Plugging in Equation (3.1), the EMO score is defined as Equation (3.2).

$$EMO = \int_0^H \int_0^W p(x, y) \max_{a \in A} \frac{|B_f \cap B_a|}{|B_f \cup B_a|} dxdy \tag{3.2}$$

In practice, we consider the anchor setting according to the Wider Face dataset. We set $S = \{16, 32, 64, 128, 256, 512\}$ to cover the face scale distribution of the dataset, and set $R = \{1\}$ since face boxes are close to squares. Therefore, there is a total of six anchors for each "+" location. In addition, we assume each face can randomly appear anywhere on the image plane with equal probability. Thus, (x, y) are drawn from uniform distributions, i.e. $x \sim U(0, W)$ and $y \sim U(0, H)$.

Since anchors are repeatedly tiled on an image, the overlapping pattern is also periodic w.r.t. the face location. So we consider only one period where the face center is enclosed by 4 anchor centers as shown in Figure 3.3. Then, the face will have the highest IoU with the anchor centered on the closest location to the face center. Due to symmetry, we focus on the cases where the face gets matched to the top-left anchor (dashed red box) with the highest IoU and the blue square shows the $\frac{s_A}{2} \times \frac{s_A}{2}$ region where the face center can appear. Face with the center outside that region will be matched to one of the other three anchors. The relative location of the face center to the anchor center is denoted as (x', y'), where x', y' are drawn from the distribution $U(0, s_A/2)$.

Given a $l \times l$ face with the same size as the anchor box, i.e. $l \in S$, it will be matched to the $l \times l$ anchor.

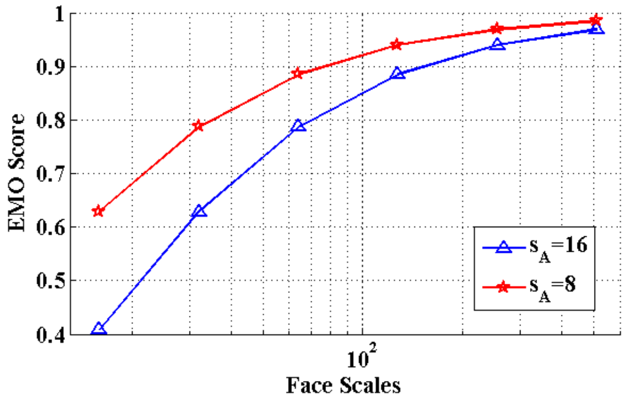


Figure 3.4: The effect of face scale and anchor stride on the EMO score. Small anchor stride is crucial for tiny faces.

So the IoU score between the face and the matched anchor is

$$IoU = \frac{(l-x')(l-y')}{2l^2-(l-x')(l-y')} \tag{3.3}$$

IoU is a function of the relative location (x', y'). A closer distance from face center to anchor center leads to higher IoU. The EMO score of this face is the expected IoU w.r.t. the distribution of (x', y') derived as in Equation (3.4).

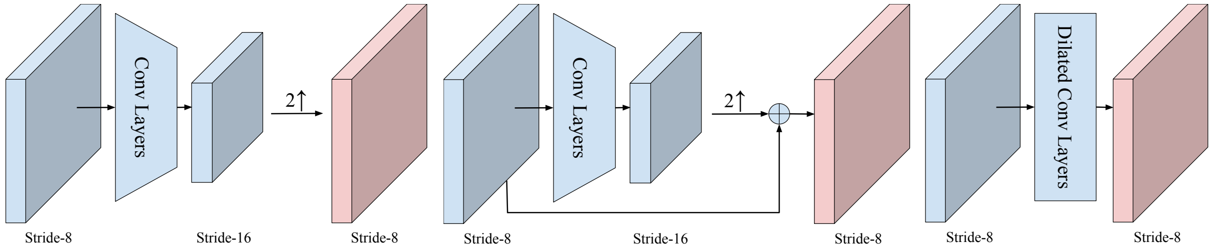
$$EMO = \int_0^{\frac{s_A}{2}} \int_0^{\frac{s_A}{2}} (\frac{2}{s_A})^2 \frac{(l-x')(l-y')}{2l^2-(l-x')(l-y')} dx'dy' \tag{3.4}$$

Figure 3.4 shows the EMO scores given different face scales and anchor strides. It explains why larger faces tend to have higher IoU overlap with anchors. When the face size is fixed, s_A plays an important role in reaching high EMO scores. Given a face, the smaller s_A is, the higher the EMO score achieves, especially for small faces. Hence the average max IoU of all faces can be statistically increased.

3.1.2 Robust Anchor Design

This section introduces our newly designed anchors for finding more tiny faces. We aim at improving the average IoU especially for tiny faces from the view of theoretically improving the EMO score since average IoU scores are correlated with face recall rate. Based on the analysis in Section 3.1.1, we propose to increase the average IoU by reducing the anchor stride as well as the distance between the face center and the anchor center.

For anchor stride reduction, we look into new network architectures to change the stride of the feature map associated with anchors. Three architectures are proposed. Additionally, we redefined the anchor locations such that the anchor stride can be further reduced. Moreover, we propose the face shift jittering method which can statistically reduce the distance between the face center and the anchor center, the other



(a) Bilinear upsampling (b) Bilinear upsampling with skip connection (c) Dilated convolution

Figure 3.5: Three types of network architecture for reducing the anchor stride by enlarging the feature map (red).

important factor to increase the IoU overlap. With these methods, the EMO score can be improved which theoretically guarantees a higher average IoU.

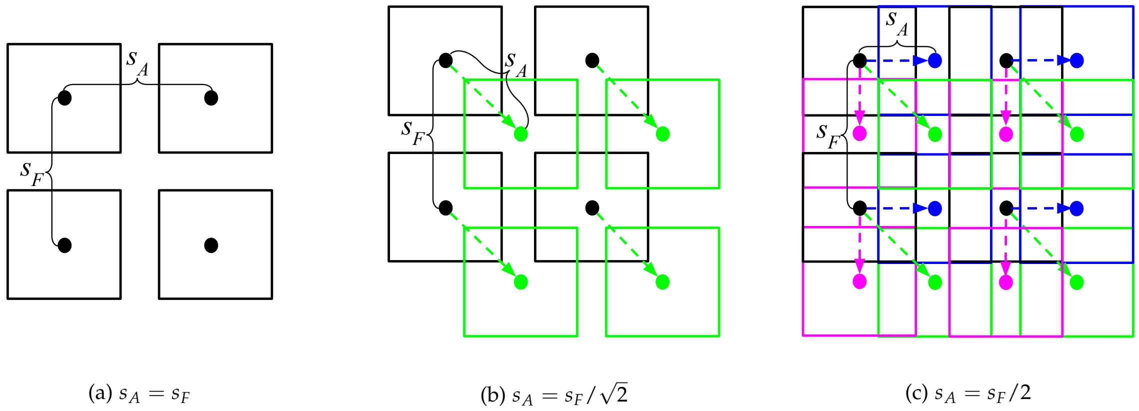
Stride reduction with enlarged feature maps. As discussed before, anchor stride equals feature stride in current anchor-based detectors. Therefore, one way to increase the EMO scores is to reduce the anchor stride by enlarging the feature map. This section presents three different architectures that double the height and width of the feature maps as illustrated in Figure 3.5.

Bilinear upsampling upscales the feature map by a factor of 2 as shown in Figure 3.5a. In this network, a deconvolution layer is attached to the feature map and its filters are initialized to have weights of a bilinear upsampler. During training, the filters are updated to adapt to the data.

Figure 3.5b shows the upscaled feature map augmented with the features from shallower large feature map by skip connection. The intuition in this design is to combine high-level features for semantical information and low-level features for localization precision [56]. In the actual networks, the low-level and high-level feature maps have different numbers of channels. Thus, two 1×1 convolution layers are first added to reduce the number of channels to the same size. Then, after the element-wise addition, another 3×3 convolution layer is appended to the final feature map for detection (not shown in Figure 3.5b).

The architectures in Figures 3.5a and 3.5b introduce additional parameters to the networks when enlarging the feature maps, hence increasing the model complexity. However, the same goal can be achieved without additional parameters by using dilated convolutions [62] as shown in Figure 3.5c. Specifically, we take out the stride-2 operation (either pooling or convolution) right after the shallower large map and dilate the filters in all the following convolution layers. Note that 1×1 filters are not required to be dilated. In addition to not having any additional parameters, dilated convolution also preserves the size of receptive fields of the filters.

With halved anchor stride, the average IoU of tiny faces increases by a large margin as shown in Figure



(a) $s_A = s_F$

(b) $s_A = s_F/\sqrt{2}$

(c) $s_A = s_F/2$

Figure 3.6: Anchor stride reduction by adding shifted anchors.

3.7, compared to the original one. In addition, we show in Section 3.1.3 the performance comparison of the three architectures.

Extra shifted anchors. Reducing anchor strides by enlarging feature maps doesn't change the condition that $s_A = s_F$. This section presents a new approach such that $s_A < s_F$. We further reduce s_A by adding extra supportive anchors not centered on the sliding window locations, i.e. *shifted anchors*. This strategy can help to increase the EMO scores without changing the resolution of feature maps. These shifted anchors share the same feature representation with the anchors in the centers.

Specifically, given a feature map with stride s_F, the distance between two adjacent sliding window locations is s_F, and labeled by black dots in Figure 3.6. In Figure 3.6a, each location has a single anchor (black) centered on it, giving the anchor stride of $s_A = s_F$. When extra supportive (green) anchors are added to the bottom-right of the center for all locations, the anchor stride can be reduced to $s_A = s_F/\sqrt{2}$ (Figure 3.6b). In addition, two other supportive anchors (blue and magenta) can be sequentially added to further reduce the anchor stride to $s_A = s_F/2$ (Figure 3.6c). Note that no matter how many anchors are added, all anchors are regularly tiled in the image plane. Indeed, we only need to add small shifted anchors since large anchors already guarantee high average IoUs, which saves the computational cost. For example, three shifted anchors of the smallest size (16 × 16) are added on top of enlarged feature maps to show further improvements of the average IoUs in Figure 3.7.

Face Shift Jittering. When computing the EMO score for each face, the center of the face is assumed to be drawn from a 2D uniform distribution. However, in the actual datasets, each face has a fixed location. Some of them are closed to the anchor centers so they are more likely to have high IoU overlaps with anchors. While some others far from anchor centers will always get low IoU scores. To increase the probability for those faces to get high IoU overlap with anchors, they are randomly shifted in each

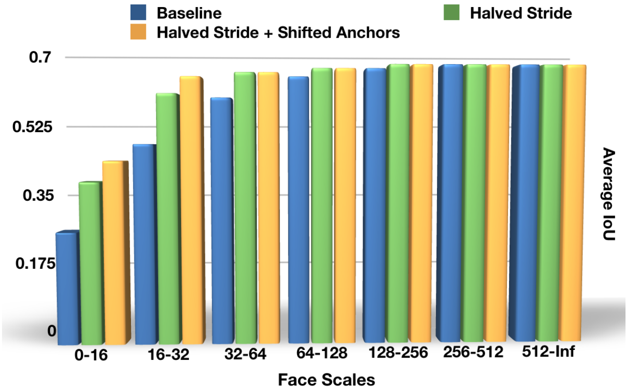


Figure 3.7: Comparison of the average IoU between original and our anchor design. With our proposed stride reduction techniques, the average IoU is significantly improved for small faces.

iteration during training.

Specifically, the image is shifted by a random offset (δ_x, δ_y) in every iteration. δ_x and δ_y are the pixels where the image is shifted right and down respectively so that locations of all faces in that image are added by (δ_x, δ_y). The offset is sampled from a discrete uniform distribution, i.e. $\delta_x, \delta_y \in \{0, 1, ..., s_A/2 - 1\}$. We use the discrete uniform distribution of offsets to approach the continuous uniform distribution of face locations. We set the maximum offset to be $s_A/2 - 1$ because the period of overlap pattern is $s_A/2$.

Hard Face Compensation. As shown in Figure 3.7, even with halved feature strides and shifted small anchors, very tiny faces still have lower average IoU than bigger faces. It is because face scales and locations are continuous whereas anchor scales and locations are discrete. Therefore, there are still some faces whose scales or locations are far away from the anchor. These faces are hard to be matched to anchors.

We propose a compensation strategy of anchor matching to assign hard faces to multiple anchors. Specifically, we first label anchors as positive if their overlapping scores with faces are higher than a threshold T_h, same as the current anchor matching strategy. Then faces whose highest overlapping scores are below T_h are the hard faces. For hard faces, the top N anchors with the highest overlap with them are labeled as positive. We find the optimal N empirically as described in Section 3.1.3.

3.1.3 Ablation Study

We conduct ablative experiments on the WiderFace dataset [101]. This dataset has 32,203 images with 393,703 labeled faces with a high degree of variability in scales, occlusions, and poses. The images are split into training (40%), validation (10%), and testing (50%) set. Faces in this dataset are classified into Easy, Medium, and Hard subsets according to the difficulties of detection. The hard subset includes a lot

Table 3.1: Ablative study of each components in our proposed method on Wider Face validation set. Network architectures: **BU** - bilinear upsampling; **BUS** - bilinear upsampling with skip connection; **DC** - dilated convolution. Extra shifted anchors: $s \times n$ - adding n shifted s-by-s anchors. **SJ** - Face Shift Jittering. **HC(N)** - assigning each hard face to top N anchors.

	Easy	Medium	Hard
Baseline	0.934	0.895	0.705
+BU	0.933	0.901	0.710
+BUS	0.926	0.899	0.778
+DC	0.936	0.911	0.779
+DC+16x1	0.934	0.908	0.781
+DC+16x1&32x1	0.936	0.910	0.782
+DC+16x3	0.938	0.912	0.786
+DC+16x3&32x1	0.937	0.909	0.781
+DC+16x3&32x3	0.938	0.913	0.779
+DC+SJ	0.939	0.910	0.788
+DC+16x3+SJ	**0.940**	**0.914**	**0.789**
+DC+16x3+SJ+HC(3)	0.938	0.912	0.793
+DC+16x3+SJ+HC(5) (Final)	**0.940**	**0.914**	**0.795**
+DC+16x3+SJ+HC(7)	0.936	0.912	0.791
Baseline+Pyramid	0.943	0.927	0.840
Final+Pyramid	**0.949**	**0.933**	**0.861**

of tiny faces. All networks are trained on the training set and evaluated on the validation set. Average Precision (AP) score is used as the evaluation metric. Note that we train and evaluate on the *original* images *without* rescaling since we want to test the detector's ability to find real tiny faces instead of upscaled tiny faces.

Baseline setup. We build an anchor-based detector with ResNet-101 [34] inspired by the R-FCN [14] as our baseline detector. It differs from the original R-FCN in the following aspects. Firstly we set 6 anchors whose scales are from the set $\{16, 32, 64, 128, 256, 512\}$ and all anchors' aspect ratios are 1. This setting matches the face boxes in the Wider Face dataset. Secondly, "res5" is used to generate region proposals instead of "res4". Thirdly, the threshold of IoU for positive anchors is changed to 0.5. All the other settings follow the original [14]. The baseline detector is trained on the Wider Face training set for 8 epochs. The initial learning rate is set to 0.001 and decreases to 0.0001 after 5 epochs. During training we applied online hard example mining [80] with the ratio between positives and negatives as 1:3. The detector is implemented in the MXNet framework [10] based on the open-source code from [15].

The effect of the enlarged feature map. The enlarged feature map reduces the anchor stride so that it helps to increase the EMO scores and the average IoU, especially for tiny faces. To better understand its effect on the final detection performance, we apply three architectures in Figure 3.5 to the ResNet backbone architecture of the baseline detector. The corresponding stride-8 feature and stride-16 feature in the backbone architecture are "res3" and "res5" respectively. For bilinear upsampling (denoted as **BU**), "res5" is appended with a stride-2 deconvolution layer with filters initialized by a bilinear upsampler. For bilinear upsampling with skip connection (denoted as **BUS**), all the additional convolution layers have 512 output channels. For dilated convolution (denoted as **DC**), all the 3×3 filters in "res4" and "res5" are dilated. Evaluation results are presented as "BU", "BUS" and "DC" in Table 3.1. Compared to the baseline, the enlarged feature map provides significant improvements on the Hard subset (rising by 7.4% at most), which mainly consists of tiny faces. Among the three architectures, BUS is better than BU at finding tiny faces, but has more false positives, because BUS uses features from the shallower "res3" layer that is less discriminative. DC achieves the best performance on all three subsets without introducing extra parameters. Thus, we fix the architecture to DC in the following experiments.

The effect of additional shifted anchors. Adding shifted anchors can further reduce the anchor stride without changing the resolution of the feature map. In Figure 3.7 we can see with halved anchor stride, faces larger than 32×32 pixels already have the same average IoU overlap. So we mainly focus on adding anchors with scales of 16 and 32. As shown in Figure 3.6, there are two settings of shifted anchors, i.e. added by one to reduce the stride by $1/\sqrt{2}$ or added by three to reduce the stride by $1/2$. We denote the shifted anchor setting as $s \times n$, where $s \in \{16, 32\}$ is the scale of anchor and $n \in \{1, 3\}$ is the number of additional anchors. Noted that we always start with adding anchors of scale 16 since larger anchors cannot affect the average IoU of smaller faces. Hence there are total 5 combinations as presented in the second row section in Table 3.1. It turns out that adding shifted anchors can improve the AP score on the Hard set. However, there is a trade-off between the number of additional anchors and the detection accuracy. More anchors do not always lead to higher average precision, because each anchor is associated with a set of parameters in the network to predict the confidence score and box offsets. As the number of anchors increases, each anchor is matched to fewer faces and the associated parameters are trained from fewer examples. In the end, we find adding just 3 anchors of scale 16 gives the best performance.

The effect of face shift jittering. Next, we look into the effect of randomly shifting the faces for each iteration during training, denoted as SJ for shift jittering. It is applied to both the model with DC and the model with DC and 3 shifted anchors of scale 16. Experiments show that shift jittering can further improve the AP scores of Hard faces. In Table 3.1, the AP rises by 0.9% on the model with DC (+DC vs. +DC+SJ) and by 0.3% on the model with DC and 3 shifted anchors of scale 16 (+DC+16x3 vs. +DC+16x3+SJ).

The effect of hard face compensation. The hard face compensation completes our final model. It is denoted as HC(N) where N is the number of anchors to which the hard faces are assigned. We find $N = 5$ is a proper choice since smaller N leads to a lower recall rate and larger N results in more false positives. To this end, we denote "+DC+16x3+SJ+HC(5)" as our "Final" detector.

The effect of testing size. The size of the testing images has a significant impact on the face detection

Table 3.2: The effect of testing size on the average precision (AP) of Hard faces in Wider Face validation set.

Max size	600x1000	800x1200	1200x1600	1400x1800
SSH [67]	0.686	0.784	0.814	0.810
Ours	**0.757**	**0.817**	**0.838**	**0.835**

precision, especially for tiny faces. Therefore we evaluate our final model on the Hard set of Wider Face validation set with different image sizes, comparing with the state-of-the-art SSH face detector [67]. We show in Table 3.2 that our detector trained with single-scale input outperforms the SSH detector trained with multiple scales at every testing size. Note that at the maximum input size of 600x1000, our detector outperforms SSH by 7.1%, showing the effectiveness of our proposed techniques for detecting tiny faces.

The effect of image pyramid Image pyramid for multi-scale training and testing helps improving the detection performance, as shown by "Baseline+Pyramid" in Table 3.1. By applying our strategies we observe another improvement ("Final+Pyramid"). We follow the same way in [67] to build the pyramid.

3.2 Anchor-Free Object Detection: Less is More

Both anchor boxes and feature pyramid networks are addressing the scale variation in object detection. When combining anchor boxes with feature pyramid network, large anchor boxes are typically associated with upper feature maps, and small anchor boxes are associated with lower feature maps, as shown in Figure 1.5. This is based on the heuristic that upper feature maps have more semantic information suitable for detecting big instances whereas lower feature maps have more fine-grained details suitable for detecting small instances [32]. The design of feature pyramids integrated with anchor boxes has achieved good performance on object detection benchmarks. A natural question to ask is how to further improve the performance of this design for scale variation? When it comes to general objects, not only do the objects have a wider range of size but also they can be of arbitrary aspect ratio. In Section 3.1, we have seen that adding more anchor boxes could be a solution. But this is not an elegant solution as more anchor boxes introduce more manually selected hyperparameters as well as increase the computation cost. In this section, we show how to derive from the limitations of anchors towards the anchor-free detection, a more elegant solution.

3.2.1 Inherent Limitations of Anchors

We argue that the anchor-based detection with feature pyramid has two limitations: 1) heuristic-guided feature selection; 2) overlap-based anchor sampling. During training, each instance is always matched

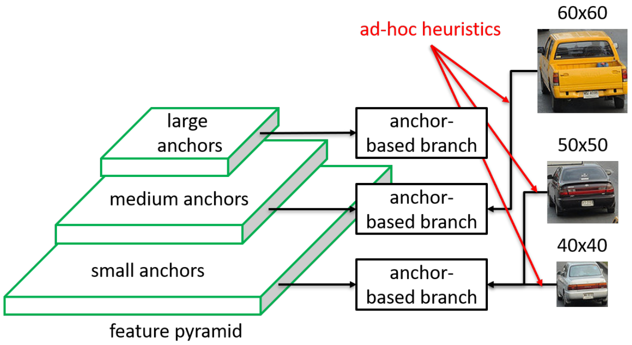


Figure 3.8: Selected feature level in anchor-based branches may not be optimal due to heuristic guided selection.

to the closest anchor box(es) according to IoU overlap. And anchor boxes are associated with a certain level of feature map by human-defined rules, such as box size. Therefore, the selected feature level for each instance is purely based on *ad-hoc heuristics*. For example, a car instance with size 50 × 50 pixels and another similar car instance with size 60 × 60 pixels may be assigned to two different feature levels, whereas another 40 × 40 car instance may be assigned to the same level as the 50 × 50 instance, as illustrated in Figure 3.8. In other words, the anchor matching mechanism is inherently heuristic-guided. This leads to a major flaw that the selected feature level to train each instance may not be optimal.

To this end, we reformulate the detection problem from an anchor-free perspective and propose a simple and effective prototype to address the above two limitations simultaneously. Our motivation is to let each instance select the best level of feature freely to optimize the network, so there should be no anchor boxes to constrain the feature selection in our module. Instead, we encode the instances in an anchor-free manner to learn the parameters for classification and localization. The general concept is presented in Figure 3.9. An anchor-free detection head is built per level of the feature pyramid. An instance can be assigned to an arbitrary level of the anchor-free head. During training, we dynamically select the most suitable level/levels of feature for each instance based on the instance content instead of just the size of the instance box. The selected level of feature then learns to detect the assigned instances. Our anchor-free prototype is agnostic to the backbone network and can be applied to single-shot detectors with a structure of feature pyramid.

3.2.2 A Simple Anchor-Free Prototype

Anchor-based detectors generate detections in a box-to-box style. Differently, anchor-free detectors do not rely on anchor boxes and predictions are directly generated in a point-to-box style. We denote the points from which detection boxes are generated as anchor points. Similarly, anchor points are associated

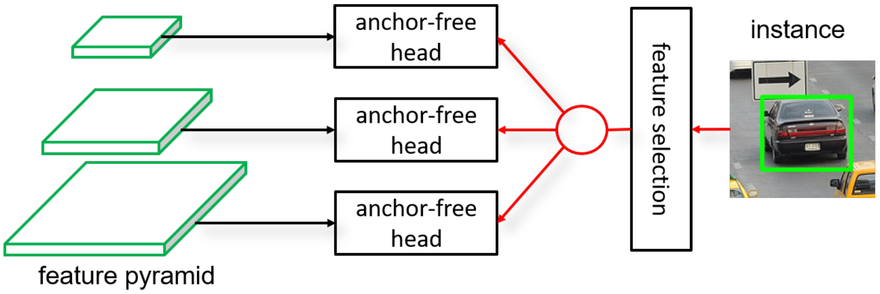


Figure 3.9: The general concept of our anchor-free prototype plugged into feature pyramid network. During training, each instance can be assigned to arbitrary pyramid level/levels for setting up supervision signals.

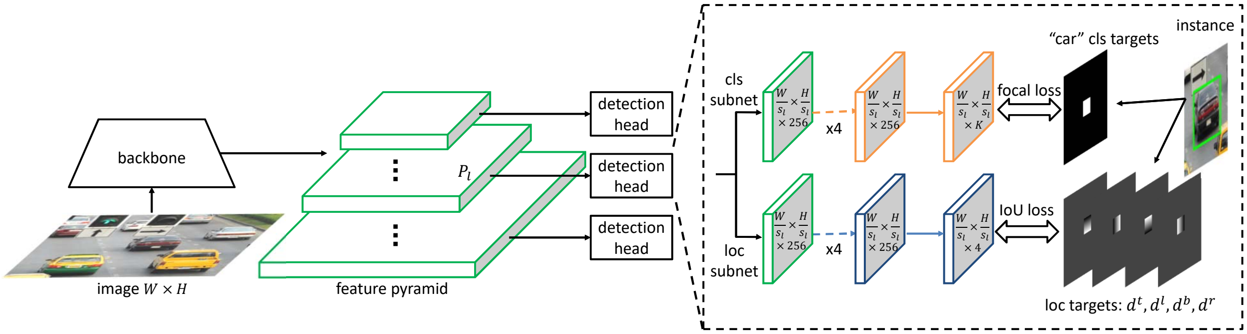

backbone
image $W \times H$
feature pyramid
P_l
detection head
detection head
detection head
cls subnet
loc subnet
$\frac{W}{s_l} \times \frac{H}{s_l} \times 256$
x4
$\frac{W}{s_l} \times \frac{H}{s_l} \times 256$
$\frac{W}{s_l} \times \frac{H}{s_l} \times K$
$\frac{W}{s_l} \times \frac{H}{s_l} \times 256$
x4
$\frac{W}{s_l} \times \frac{H}{s_l} \times 256$
$\frac{W}{s_l} \times \frac{H}{s_l} \times 4$
focal loss
IoU loss
"car" cls targets
instance
loc targets: d^t, d^l, d^b, d^r

Figure 3.10: The network architecture of a vanilla anchor-point detector with a simple detection head.

with the features at their locations just like the anchor boxes. In this section, we formulate the detection pipeline from the anchor point's viewpoint in the setting of a vanilla anchor-free prototype with a simple architecture of the detection head.

Network architecture. As shown in Figure 3.10, the network consists of a backbone, a feature pyramid, and one detection head per pyramid level, in a fully convolutional style. A pyramid level is denoted as P_l where l indicates the level number and it has $1/s_l$ resolution of the input image size $W \times H$. s_l is the feature stride and $s_l = 2^l$. A typical range of l is 3 to 7. A detection head has two task-specific subnets, i.e. classification subnet and localization subnet. They both have five 3×3 conv layers. The classification subnet predicts the probability of objects at each anchor point location for each of the K object classes. The localization subnet predicts the 4-dimensional class-agnostic distance from each anchor point to the boundaries of a nearby instance if the anchor point is positive (defined next).

Supervision targets. We first define the concept of anchor points. An anchor point p_{lij} is a pixel on the pyramid level P_l located at (i, j) with $i = 0, 1, \ldots, W/s_l - 1$ and $j = 0, 1, \ldots, H/s_l - 1$. Each p_{lij} has a corresponding image space location (X_{lij}, Y_{lij}) where $X_{lij} = s_l(i + 0.5)$ and $Y_{lij} = s_l(j + 0.5)$. Next we define the valid box B_v of a ground-truth instance box $B = (c, x, y, w, h)$ where c is the class id, (x, y)

is the box center, and w, h are box width and height respectively. B_v is a central shrunk box of B, i.e. $B_v = (c, x, y, \epsilon w, \epsilon h)$, where ϵ is the shrunk factor. An anchor point p_{lij} is positive if and only if some instance B is assigned to P_l and the image space location (X_{lij}, Y_{lij}) of p_{lij} is inside B_v, otherwise it is a negative anchor point. For a positive anchor point, its classification target is c and localization targets are calculated as the normalized distances $\mathbf{d} = (d^l, d^t, d^r, d^b)$ from the anchor point to the left, top, right, bottom boundaries of B respectively (Equation (3.5)),

$$\begin{aligned} d^l &= \frac{1}{zs_l}[X_{lij} - (x - w/2)] \quad & d^t &= \frac{1}{zs_l}[Y_{lij} - (y - h/2)] \\ d^r &= \frac{1}{zs_l}[(x + w/2) - X_{lij}] \quad & d^b &= \frac{1}{zs_l}[(y + h/2) - Y_{lij}] \end{aligned} \tag{3.5}$$

where z is the normalization scalar. For negative anchor points, their classification targets are background ($c = 0$), and localization targets are set to null because they don't need to be learned. To this end, we have a classification target c_{lij} and a localization target $\mathbf{d}_{lij}$ for all of each anchor point p_{lij}. A visualization of the classification targets and the localization targets of one feature level is illustrated in Figure 3.10.

Loss functions. Given the architecture and the definition of anchor points, the network generates a K-dimensional classification output $\hat{\mathbf{c}}_{lij}$ and a 4-dimensional localization output $\hat{\mathbf{d}}_{lij}$ per anchor point p_{lij}. Focal loss [57] (l_{FL}) is adopted for the training of classification subnets to overcome the extreme class imbalance between positive and negative anchor points. IoU loss [104] (l_{IoU}) is used for the training of localization subnets. Therefore, the per anchor point loss L_{lij} is calculated as Equation (3.6).

$$L_{lij} = \begin{cases} l_{FL}(\hat{\mathbf{c}}_{lij}, c_{lij}) + l_{IoU}(\hat{\mathbf{d}}_{lij}, \mathbf{d}_{lij}), & p_{lij} \in p^+ \\ l_{FL}(\hat{\mathbf{c}}_{lij}, c_{lij}), & p_{lij} \in p^- \end{cases} \tag{3.6}$$

where p^+ and p^- are the sets of positive and negative anchor points respectively. The loss for the whole network is the weighted summation of all anchor point losses divided by the summation of weights of positive anchor points (Equation (3.7)).

$$L = \frac{1}{\sum_{p_{lij} \in p^+} w_{lij}} \sum_{lij} w_{lij} L_{lij} \tag{3.7}$$

where w_{lij} is an attention weight assigned for each anchor point's loss. For each positive anchor point, the weight depends on the distance between its image space location and the corresponding instance boundaries. The closer to a boundary, the more down-weighted the anchor point gets. Thus, anchor points close to boundaries are receiving less attention and the network focuses more on those surrounding the center. For negative anchor points, they are kept unchanged, i.e. their weights are all set to 1.

Mathematically, w_{lij} is defined in Equation (3.8):

$$w_{lij} = \begin{cases} \frac{\min(d^l_{lij}, d^r_{lij}) \min(d^t_{lij}, d^b_{lij})}{\max(d^l_{lij}, d^r_{lij}) \max(d^t_{lij}, d^b_{lij})}, & \exists B, p_{lij} \in B_v \\ 1, & \text{otherwise} \end{cases} \tag{3.8}$$

3.2.3 Feature Selection Matters

Unlike anchor-based detectors, anchor-free methods don't have constraints from anchor matching to select feature levels for instances from the feature pyramid. In other words, we can assign each instance to arbitrary feature level(s) in anchor-free methods during training. And selecting the right feature levels can make a big difference [116]. In this section we consider heuristic guided feature selection and semantic guided feature selection. The heuristic guided feature selection is purely based on ad-hoc heuristics such as the box scale of the instance, which is similar to anchor-based detection. For semantic guided feature selection, we take the actual semantic information of the instance into consideration. A hard version and a soft version are proposed respectively.

Heuristic guided feature selection. Similarly to anchor-based detectors, anchor-free detector can also select the feature levels based on heuristics where the feature selection depends purely on box sizes. We borrow the idea from the FPN detector [56]. An instance I is assigned to the level $P_{l'}$ of the feature pyramid by:

$$l' = \lfloor l_0 + \log_2(\sqrt{wh}/224) \rfloor \tag{3.9}$$

Here 224 is the canonical ImageNet pre-training size, and l_0 is the target level on which an instance with $w \times h = 224^2$ should be mapped. In this work we choose $l_0 = 5$ because ResNet [34] uses the feature map from 5th convolution group to do the final classification.

Semantic guided feature selection: hard version. We believe the semantic guided feature selection should be based on per-instance loss in training because semantic information is encoded in the network loss. For the hard version, we aim to find a single optimal feature level for each instance.

Given an instance B, we define its classification loss and box localization loss on P_l as $L^B_{FL}(l)$ and $L^B_{IoU}(l)$, respectively. They are computed by averaging the weighted focal loss and IoU loss of all positive anchor points on P_l inside the valid box B_v, i.e.

$$\begin{aligned} L^B_{FL}(l) &= \frac{1}{\sum_{p_{lij} \in B_v} w_{lij}} \sum_{p_{lij} \in B_v} l_{FL}(\hat{\mathbf{c}}_{lij}, c_{lij}) \\ L^B_{IoU}(l) &= \frac{1}{\sum_{p_{lij} \in B_v} w_{lij}} \sum_{p_{lij} \in B_v} l_{IoU}(\hat{\mathbf{d}}_{lij}, \mathbf{d}_{lij}) \end{aligned} \tag{3.10}$$

Figure 3.11 shows our hard version feature selection process. First the instance I is forwarded through all levels of the feature pyramid. Then the summation of $L^I_{FL}(l)$ and $L^I_{IoU}(l)$ is computed in all anchor-free

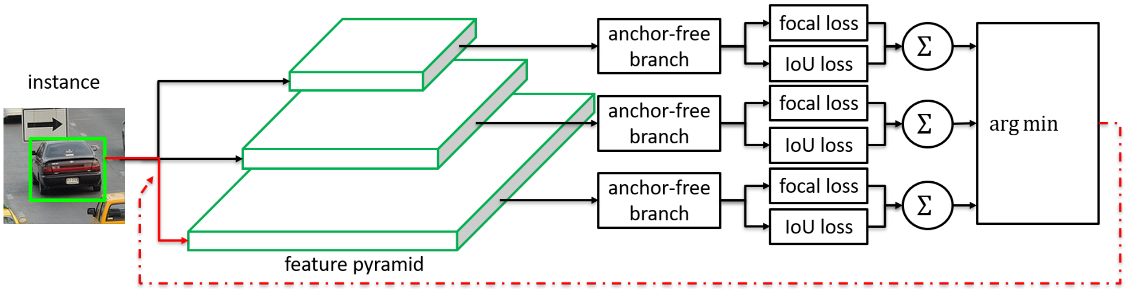


Figure 3.11: Hard version of our semantic guided feature selection. Each instance is passing through all levels of anchor-free branches to compute the averaged classification (focal) loss and localization (IoU) loss over effective regions. Then the level with minimal summation of two losses is selected to set up the supervision signals for that instance.

branches using Equation (3.10). Finally, the best pyramid level P_{l^*} yielding the minimal summation of losses is selected to learn the instance, i.e.

$$l^* = \arg\min_{l} L^B_{FL}(l) + L^B_{IoU}(l) \tag{3.11}$$

For a training batch, features are updated for their correspondingly assigned instances. The intuition is that the selected feature is currently the best to model the instance. Its loss forms a lower bound in the feature space. And by training, we further pull down this lower bound. At the time of inference, we do not need to select the feature because the most suitable level of the feature pyramid will naturally output high confidence scores.

Semantic guided feature selection: soft version. If we look into the properties of the feature pyramid, feature maps from different pyramid levels are somewhat similar to each other, especially the adjacent levels. We visualize the response of all pyramid levels in Figure 3.12. It turns out that if one level of feature is activated in a certain region, the same regions of adjacent levels may also be activated in a similar style. But the similarity fades as the levels are farther apart. This means that features from more than one pyramid level can participate together in the detection of a particular instance, but the degrees of participation from different levels should be somewhat different.

Inspired by the above observation, we argue features from multiple levels should be involved in the training and testing for each instance, and each level should make distinct contributions. FoveaBox [44] has shown that assigning instances to multiple feature levels can improve the performance to some extent. But assigning to too many levels may instead hurt the performance severely. We believe this limitation is caused by the hard selection of pyramid levels. For each instance, the pyramid levels in FoveaBox are either selected or discarded. The selected levels are treated equally no matter how different their feature responses are. Therefore, the solution lies in reweighting the pyramid levels for each instance. In other words, a weight is assigned to each pyramid level according to the feature response, making the selection

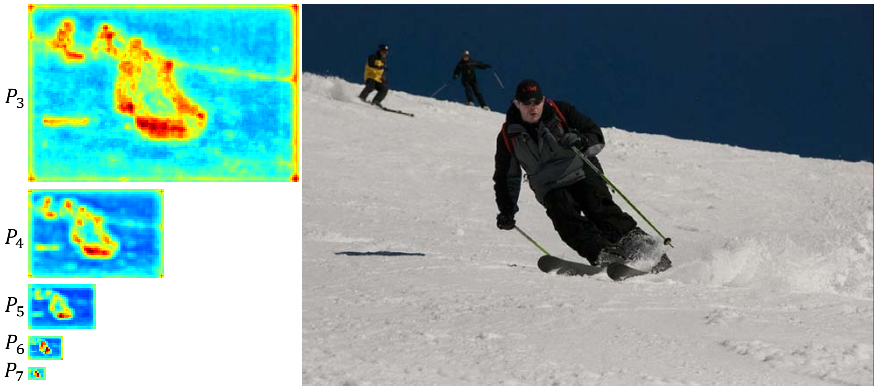


Figure 3.12: Feature responses from P_3 to P_7. They look similar but the details gradually vanish as the resolution becomes smaller. Selecting a single level per instance causes the waste of network power.

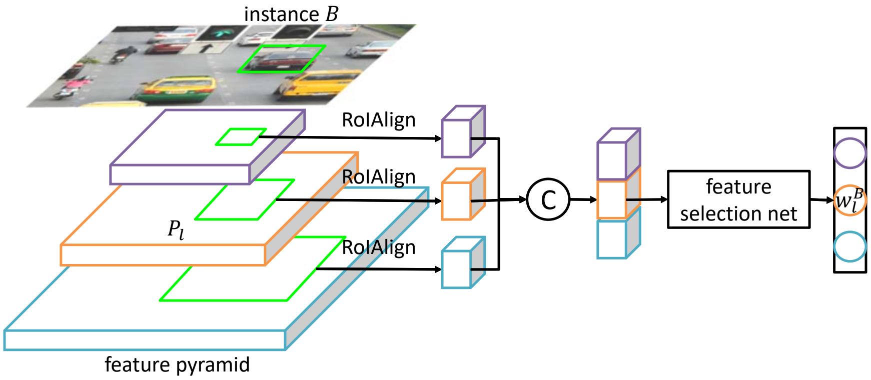


Figure 3.13: The weights prediction for soft-selected pyramid levels. "C" indicates the concatenation operation.

soft. This can also be viewed as assigning a proportion of the instance to a level.

So how to decide the weight of each pyramid level per instance? We propose to train a feature selection network to predict the weights for soft feature selection [115]. The input to the network is instance-dependent feature responses extracted from all the pyramid levels. This is realized by applying the RoIAlign layer [33] to each pyramid feature followed by concatenation, where the RoI is the instance ground-truth box. Then the extracted feature goes through a feature selection network to output a vector of the probability distribution, as shown in Figure 3.13. We use the probabilities as the weights of soft feature selection.

There are multiple architecture designs for the feature selection network. For simplicity, we present a lightweight instantiation. It consists of three 3 × 3 conv layers with no padding, each followed by the ReLU function, and a fully connected layer with softmax. Table 3.3 details the architecture. The feature selection network is jointly trained with the detector. Cross entropy loss is used for optimization and the ground-truth is a one-hot vector indicating which pyramid level has minimal loss as defined in the hard

Table 3.3: Architecture of the feature selection network. The conv layers have no padding.

layer type	output size	layer setting	activation
input	$1280 \times 7 \times 7$	n/a	n/a
conv	$256 \times 5 \times 5$	$3 \times 3, 256$	relu
conv	$256 \times 3 \times 3$	$3 \times 3, 256$	relu
conv	$256 \times 1 \times 1$	$3 \times 3, 256$	relu
fc	5	n/a	softmax

version of feature selection.

So far, each instance B is associated with a per level weight w_l^B via the feature selection network. The anchor point loss L_{lij} is down-weighed further if B is assigned to P_l and p_{lij} is inside B_v. We assign each instance B to top$q$ feature levels with q minimal instance-dependent losses during training. Thus, Eq. (3.8) is augmented into Equation (3.12).

$$w_{lij} = \begin{cases} w_l^B \frac{\min(d_{lij}^l, d_{lij}^r)\min(d_{lij}^t, d_{lij}^b)}{\max(d_{lij}^l, d_{lij}^r)\max(d_{lij}^t, d_{lij}^b)}, & \exists B, p_{lij} \in B_v \\ 1, & \text{otherwise} \end{cases} \tag{3.12}$$

The total loss of the whole model is the weighted sum of anchor point losses plus the classification loss ($L_{\text{select-net}}$) from the feature selection network, as in Equation (3.13).

$$L = \frac{1}{\sum_{p_{lij} \in p^+} w_{lij}} \sum_{lij} w_{lij} L_{lij} + \lambda L_{\text{select-net}} \tag{3.13}$$

where λ is the hyperparameter that controls the proportion of classification loss $L_{\text{select-net}}$ for feature selection.

3.2.4 Experimental Results

We conduct experiments on the more general and challenging COCO [58] detection dataset using the MMDetection [9] codebase. All models are trained on the `train2017` split including around 115k images. We analyze our method by ablation studies on the `val2017` split containing 5k images. When comparing to the state-of-the-art detectors, we report the Average Precision (AP) scores on the `test-dev` split.

Implementation details. We follow [57] for the initialization of the detection network. Specifically, the backbone networks are pre-trained on ImageNet1k [17]. The classification layers in the detection head are initialized with bias $-\log((1-\pi)/\pi)$ where $\pi = 0.01$, and a Gaussian weight. The localization layers in the detection head are initialized with bias 0.1, and also a Gaussian weight. For the newly added feature selection network, we initialize all layers in it using a Gaussian weight. All the Gaussian weights are filled with $\sigma = 0.01$.

Table 3.4: Ablative experiments for our anchor-free prototype on the COCO `minival`. ResNet-50 is the backbone network for all experiments in this table. We study the effect of various feature selection strategies.

	Heuristic guided	Semantic guided		AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		Hard version	Soft version						
RetinaNet (anchor-based)	✓			35.7	54.7	38.5	19.5	39.9	47.5
Ours (anchor-free)	✓			35.9	54.8	38.1	20.2	39.7	46.5
		✓		37.0	55.8	39.5	20.5	40.1	48.5
			✓	**38.0**	**56.9**	**40.5**	**21.0**	**41.1**	**50.2**

The entire detection network and the feature selection network are jointly trained with stochastic gradient descent on 8 GPUs with 2 images per GPU using the COCO `train2017` set. Unless otherwise noted, all models are trained for 12 epochs (∼90k iterations) with an initial learning rate of 0.01, which is divided by 10 at the 9th and the 11th epoch. Horizontal image flipping is the only data augmentation unless otherwise specified. In the soft version of semantic guided feature selection, we don't use the output from the feature selection network for the first 6 epochs. The detection network is trained with the same feature selection strategy as in the hard version, i.e. each instance is assigned to only one feature level yielding minimal loss. We plug in the soft selection weights and choose the top$q$ levels for the second 6 epochs. This is to stabilize the feature selection network first and to make the learning smoother in practice. We use the training hyperparameters for the shrunk factor $\epsilon = 0.2$ and the normalization scalar $z = 4.0$. We set $\lambda = 0.1$ although results are robust to the exact value.

At the time of inference, the network architecture is as simple as in Figure 3.10. The feature selection network is *not* involved in the inference so the runtime speed is not affected. An image is forwarded through the network in a fully convolutional style. Then classification prediction $\hat{\mathbf{c}}_{lij}$ and localization prediction $\hat{\mathbf{d}}_{lij}$ are generated for all each anchor point p_{lij}. Bounding boxes can be decoded using the reverse of Equation (3.5). We only decode box predictions from at most 1k top-scoring anchor points in each pyramid level, after thresholding the confidence scores by 0.05. These top predictions from all feature levels are merged, followed by non-maximum suppression with a threshold of 0.5, yielding the final detections.

Ablation Study

For all ablation studies, we use an image scale of 800 pixels for both training and testing. We compare our anchor-free prototype with the anchor-based counterparts, i.e. RetinaNet [57]. We also study the effect of different proposed feature selection strategies. Results are reported in Table 3.4.

Anchor boxes may not be necessary. We first compare the performance of our anchor-free prototype with the anchor-based counterpart, i.e. RetinaNet. It turns out even with the naive heuristic guided feature selection, comparable results can be achieved (Table 3.4 1st vs. 2nd entry) with the anchor-free detection head which has fewer parameters and simpler network architecture than the anchor-based detection head. Thus we encourage the community to rethink the necessity of anchor boxes which are currently considered as the de facto standard in object detection.

Semantic guided feature selection is essential. Our anchor-free prototype with semantic guided feature selection outperforms the heuristic guided one by a considerable margin (Table 3.4 2nd vs. 3rd entry). This indicates selecting the right feature to learn plays a fundamental role in detection. To understand the optimal pyramid level selected for instances, we visualize some qualitative detection results from the anchor-free detection heads in Figure 3.14. The number before the class name indicates the feature level that detects the object. It turns out the hard version of the semantic guided feature selection somehow follows the rule that upper levels select larger instances, and lower levels are responsible for smaller instances, which is the same principle in anchor-based branches. However, there are quite a few *exceptions*, i.e. semantic guided feature selection chooses pyramid levels different from the choices of heuristic guided selection. We label these exceptions as red boxes in Figure 3.14. Green boxes indicate agreement between the semantic guided and heuristic guided ones. By capturing these exceptions, our anchor-free detector can use better features to detect challenging objects.

Soft-selected pyramid levels utilize the feature power better. We further compare the soft version of semantic guided feature selection with the hard version (Table 3.4 3rd vs. 4th entry). We find that as long as each instance is assigned to more than one pyramid level, a robust ∼1.0% absolute AP improvement over the hard version can be observed. This indicates that allowing instances to optimize multiple pyramid levels is essential to utilize the feature power as much as possible. Empirically, we assign each instance to the top 3 feature levels with the minimal instance-dependent losses according to Table 3.5. To understand how does the feature selection network assign instances, we visualize the predicted soft selection weights in Figure 3.15. It turns out that larger instances tend to be assigned high weights for higher pyramid levels. The majority of instances can be learned with no more than two levels. Very rare instances need to be modeled by more than two levels, e.g. the sofa in the top right sub-figure of Figure 3.15. This is consistent with the results in Table 3.5.

Semantic guided feature selection works well with augmented feature pyramids. To evaluate the generalization ability of the semantic guided feature selection, we apply it to augmented feature pyramids. Here we adopt the Balanced Feature Pyramid (BFP) [68] and achieve further improvement. The BFP pushes our model with ResNet-50 to a 38.8% AP. More importantly, our proposed semantic guided feature selection can robustly work with advanced feature pyramids, offering a steady 2% AP gain as shown in

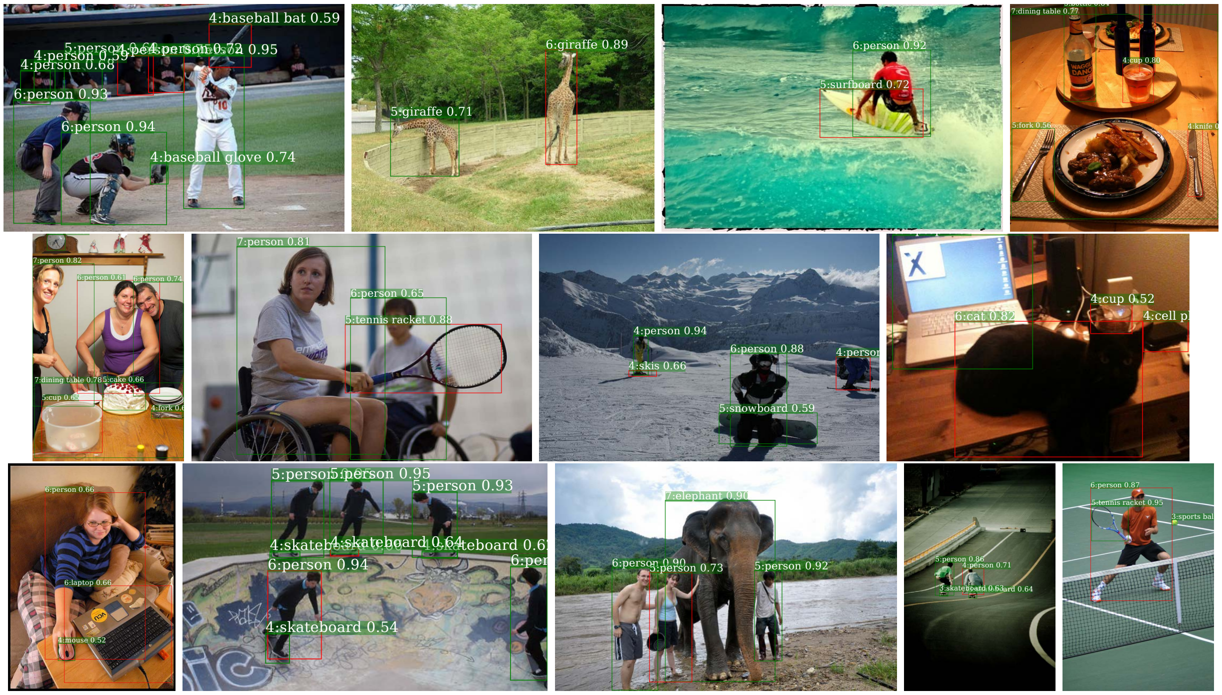


Figure 3.14: Visualization of online feature selection from anchor-free branches. The number before the class name is the pyramid level that detects the instance. We compare this level with the level to which as if this instance is assigned in the anchor-based branches, and use *red* to indicate the disagreement and *green* for agreement.

Table 3.5: Varying q for number of selected levels in soft version of semantic guided feature selection.

top$q$	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
2	37.9	56.9	40.5	21.1	41.0	50.1
3	**38.0**	56.9	40.5	21.2	41.2	50.2
4	37.9	56.9	40.3	21.2	41.1	50.2
5	37.9	56.8	40.5	21.0	41.1	50.2

Table 3.6: The effect of semantic guided selection on different feature pyramids.

Feature pyramid	Heuristic guided selection	Semantic guided selection	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
FPN	✓		35.9	54.8	38.1	20.2	39.7	46.5
		✓	38.0	56.9	40.5	21.0	41.1	50.2
BFP	✓		36.8	57.2	39.0	22.0	41.0	45.9
		✓	38.8	58.7	41.3	22.5	42.6	50.8

Table 3.6.

Our anchor-free detector is robust and efficient. Our anchor-free detector can consistently provide robust performance using deeper and better backbone networks, while at the same time keeping the detection head as simple as possible. We report the head-to-head comparisons with anchor-based RetinaNet in terms of detection accuracy and speed in Table 3.7. Except for the head architectures, all other settings

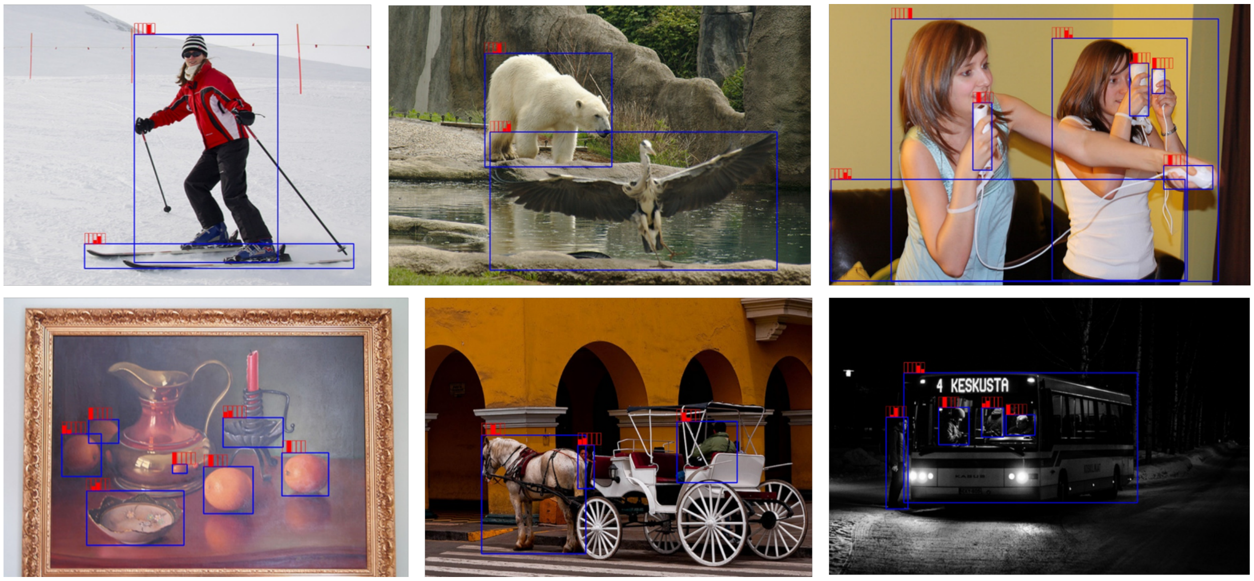


Figure 3.15: Visualization of the soft feature selection weights from the feature selection network. Weights (the top-left red bars) ranging from 0 to 1 of five pyramid levels (P_3 to P_7) are predicted for each instance (blue box). The more filled a red bar is, the higher the weight. *Best viewed in color when zoomed in.*

Table 3.7: Head-to-head comparisons of anchor-based RetinaNet, and our anchor-free detector with different backbone networks on the COCO `val2017` set. **R**: ResNet. **X**: ResNeXt.

Backbone	Method	AP	AP$_{50}$	FPS
R-50	RetinaNet (anchor-based)	35.7	54.7	11.6
	Ours (anchor-free)	**38.8**	**58.7**	**14.9**
R-101	RetinaNet (anchor-based)	37.7	57.2	8.0
	Ours (anchor-free)	**41.0**	**60.7**	**11.2**
X-101-64x4d	RetinaNet (anchor-based)	39.8	59.5	4.5
	Ours (anchor-free)	**43.1**	**63.7**	**6.1**

are the same. All detectors run on a single GTX 1080Ti GPU with CUDA 10 using a batch size of 1. It turns out that our detector gets both sides of two worlds. Ours with purely anchor-free heads can not only run faster than the anchor-based counterparts due to simpler head architecture, but also outperform the anchor-based detector by significant margins, i.e. 3.1%, 3.3%, and 3.3% absolute AP increases on ResNet-50, ResNet-101, and ResNeXt-101-64x4d backbones respectively.

Comparison to State of the Art

We evaluate our best anchor-free detector on the COCO `test-dev` set to compare with recent state-of-the-art methods. All of our models are trained using scale jitter by randomly scaling the shorter side of images in the range from 640 to 800 and for 2× number of epochs as the models in the ablation study with the learning rate change points scaled proportionally. Other settings are the same.

For a fair comparison, we report the results of single-model single-scale testing for all methods, as

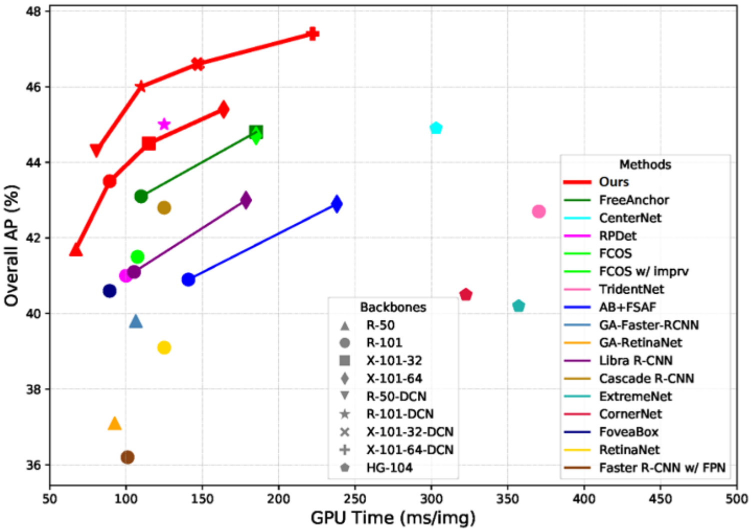


Figure 3.16: Single-model single-scale speed (ms) vs. accuracy (AP) on COCO `test-dev`. We show variants of our anchor-free detector with and without DCN [15]. Without DCN, our fastest version can run up to 5× faster than other methods with comparable accuracy. With DCN, our accurate version forms an upper envelop of all recent detectors.

well as their corresponding inference speeds in Table 3.8. A visualization of the accuracy-speed trade-off is shown in Figure 3.16. The inference speeds are measured by Frames-per-Second (FPS) on the same machine with a GTX 1080Ti GPU using a batch size of 1 whenever possible. A "n/a" indicates the case that the method doesn't provide trained models nor self-timing results from the original paper.

Our detector pushes the envelope of accuracy-speed boundary to a new level. We report the results of two series of backbone models, one without DCN and the other with DCN. Without DCN, our fastest version based on ResNet-50 can reach a 14.9 FPS while maintaining a 41.7% AP, outperforming some of the methods [56, 57, 44, 82, 68] using ResNet-101. With DCN, our accurate version forms an upper envelope of state-of-the-art anchor-free detectors and some recent anchor-based detectors. The closest competitor, RPDet [102], is 1.0% AP worse and 15ms slower than ours.

3.3 Summary

This chapter studies how to address the scale variation problem for object detection. Our exploration starts with anchor-based detection and evolves to anchor-free detection. For the anchor-based detection, we identify low object-anchor overlap as the major reason hindering anchor-based detectors to detect tiny objects. We proposed the new EMO score to characterize anchors' capability of getting high overlaps with objects, providing an in-depth analysis of the anchor matching mechanism. This inspired us to come up with several simple but effective strategies for a new anchor design for higher IoU scores. Consequently,

Table 3.8: *Single-model and single-scale* accuracy and inference speed of SAPD vs. recent state-of-the-art detectors on the COCO `test-dev` set. FPS is measured on the same machine with a single *GTX 1080Ti* GPU using the official source code whenever possible. "n/a" means that both trained models and timing results from original papers are not available. **R**: ResNet. **X**: ResNeXt. **HG**: Hourglass. For a visualized comparison, please refer to Figure 3.16.

Method	Backbone	Anchor free?	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Multi-stage detectors									
Faster R-CNN w/ FPN [56]	R-101		9.9	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN [15]	Inception ResNet		n/a	37.5	58.0	n/a	19.4	40.1	52.5
Soft-NMS [4]	Inception ResNet		n/a	40.9	62.8	n/a	23.3	43.6	53.3
Cascade R-CNN [5]	R-101		8.0	42.8	62.1	46.3	23.7	45.5	55.2
GA-Faster-RCNN [85]	R-50	✓	9.4	39.8	59.2	43.5	21.8	42.6	50.7
Libra R-CNN [68]	R-101		9.5	41.1	62.1	44.7	23.4	43.7	52.5
Libra R-CNN [68]	X-101-64x4d		5.6	43.0	64.0	47.0	25.3	45.6	54.6
RPDet [102]	R-101	✓	10.0	41.0	62.9	44.3	23.6	44.1	51.7
RPDet [102]	R-101-DCN	✓	8.0	45.0	66.1	49.0	26.6	48.6	57.5
TridentNet [52]	R-101		2.7	42.7	63.6	46.5	23.9	46.6	56.6
TridentNet [52]	R-101-DCN		1.3	46.8	67.6	51.5	28.0	51.2	60.5
Single-stage detectors									
YOLOv2 [76]	DarkNet-19		n/a	21.6	44.0	19.2	5.0	22.4	35.5
SSD [59]	ResNet-101		n/a	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet [57]	R-101		8.0	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet [108]	R-101		n/a	36.4	57.5	39.5	16.6	39.9	51.4
CornerNet [46]	HG-104	✓	3.1	40.5	56.5	43.1	19.4	42.7	53.9
AB+FSAF [116]	R-101		7.1	40.9	61.5	44.0	24.0	44.2	51.3
AB+FSAF [116]	X-101-64x4d		4.2	42.9	63.8	46.3	26.6	46.2	52.7
GA-RetinaNet [85]	R-50	✓	10.8	37.1	56.9	40.0	20.1	40.1	48.0
ExtremeNet [113]	HG-104	✓	2.8	40.2	55.5	43.2	20.4	43.2	53.1
FoveaBox [44]	R-101	✓	11.2	40.6	60.1	43.5	23.3	45.2	54.5
FoveaBox [44]	X-101	✓	n/a	42.1	61.9	45.2	24.9	46.8	55.6
FCOS [82]	R-101	✓	9.3	41.5	60.7	45.0	24.4	44.8	51.6
FCOS [82] w/ imprv	X-101-64x4d	✓	5.4	44.7	64.1	48.4	27.6	47.5	55.6
CenterNet [21]	HG-52	✓	4.4	41.6	59.4	44.2	22.5	43.1	54.1
CenterNet [21]	HG-104	✓	3.3	44.9	62.4	48.1	25.6	47.4	57.4
CenterNet(OAP) [112]	HG-104	✓	n/a	42.1	61.1	45.9	24.1	45.5	52.8
FreeAnchor [109]	R-101		9.1	43.1	62.2	46.4	24.5	46.1	54.8
FreeAnchor [109]	X-101-32x8d		5.4	44.8	64.3	48.4	27.0	47.9	56.0
Ours	R-50	✓	14.9	41.7	61.9	44.6	24.1	44.6	51.6
Ours	R-101	✓	11.2	43.5	63.6	46.5	24.9	46.8	54.6
Ours	X-101-32x4d	✓	8.7	44.5	64.7	47.8	26.5	47.8	55.8
Ours	X-101-64x4d	✓	6.1	45.4	65.6	48.9	27.3	48.7	56.8
Ours	R-50-DCN	✓	12.4	44.3	64.4	47.7	25.5	47.3	57.0
Ours	R-101-DCN	✓	9.1	46.0	65.9	49.6	26.3	49.2	59.6
Ours	X-101-32x4d-DCN	✓	6.8	46.6	66.6	50.0	27.3	49.7	60.7
Ours	X-101-64x4d-DCN	✓	4.5	47.4	67.4	51.1	28.1	50.3	61.5

our method outperforms the baseline anchor-based detector by a considerable margin and achieves the state-of-the-art performance on the challenging WiderFace dataset. For the anchor-free detection, we discover heuristic feature selection as the primary limitation for anchor-based detectors with feature pyramids. Our insight is that flexible feature selection is the major advantage of anchor-free detection over anchor-based counterparts. Semantic-guided feature selection is the key to utilize such an advantage. Eventually, we achieve the goal that less is more. Our anchor-free detector both faster and more accurate than previous state-of-the-art methods.

Chapter 4

Semantic Relation Reasoning for Availability Variation

Deep learning algorithms usually require a large amount of annotated data to achieve superior performance. To acquire enough annotated data, one common way is by collecting abundant samples from the real world and paying annotators to generate ground-truth labels. However, even if all the data samples are well annotated based on our requirements, we still face the problem of availability variation. Because long-tail distribution is an inherent characteristic of the real world, there always exist some rare cases that have just a few samples available, such as rare animals, uncommon road conditions. In other words, we are unable to alleviate the situation of scarce cases by simply spending more money on annotation even big data is accessible. Therefore, how to address data availability variation is an imperative and long-lasting task.

The availability variation problem can be formulated as a few-shot learning task. Recently, efforts have been put into the study of few-shot object detection (FSOD) [8, 41, 20, 40, 96, 91, 23, 103, 90, 92, 93]. In FSOD, there are base classes in which sufficient objects are annotated with bounding boxes and novel classes in which very few labeled objects are available. The novel class set does not share common classes with the base class set. The few-shot detectors are expected to learn from limited data in novel classes with the aid of abundant data in base classes and to be able to detect all novel objects in a held-out testing set. To achieve this, most of the recent few-shot detection methods adopt the ideas from meta-learning and metric learning for few-shot recognition and apply them to conventional detection frameworks, e.g. Faster R-CNN [77], YOLO [75].

Although recent FSOD methods have improved the baseline considerably, data scarcity is still a bottleneck that hurts the detector's generalization from a few samples. In other words, the performance is very

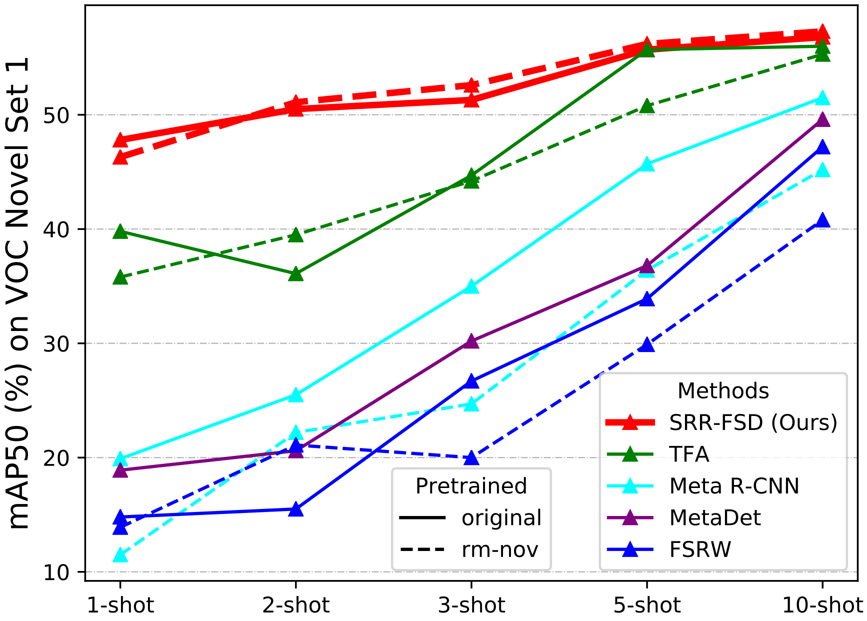


Figure 4.1: FSOD performance (mAP50) on VOC [22] Novel Set 1 at different shot numbers. Solid line (original) means the pretrained model used for initializing the detector backbone is trained on the original ImageNet [17]. Dashed line (rm-nov) means classes in Novel Set 1 are removed from the ImageNet for the pretrained backbone model. Our SRR-FSD is more stable to the variation of explicit shots (x-axis) and implicit shots (original vs. rm-nov).

sensitive to the number of both explicit and implicit shots and drops drastically as data becomes limited. The explicit shots refer to the available labeled objects from the novel classes. For example, the 1-shot performance of some FSOD methods is less than half of the 5-shot or 10-shot performance, as shown in Figure 4.1. In terms of implicit shots, initializing the backbone network with a model pretrained on a large-scale image classification dataset is a common practice for training an object detector. However, the classification dataset contains many implicit shots of object classes overlapped with the novel classes. So the detector can have early access to novel classes and encode their knowledge in the parameters of the backbone. Removing those implicit shots from the pretrained dataset also hurts the performance as shown in Figure 4.1. The variation of explicit and implicit shots could potentially lead to system failure when dealing with extreme cases in the real world.

We believe the reason for shot sensitivity is due to exclusive dependence on visual information. Novel objects are learned through images only and the learning is independent between classes. As a result, visual information becomes limited as image data becomes scarce. However, one thing remains constant regardless of the availability of visual information, i.e. the semantic relation between base and novel

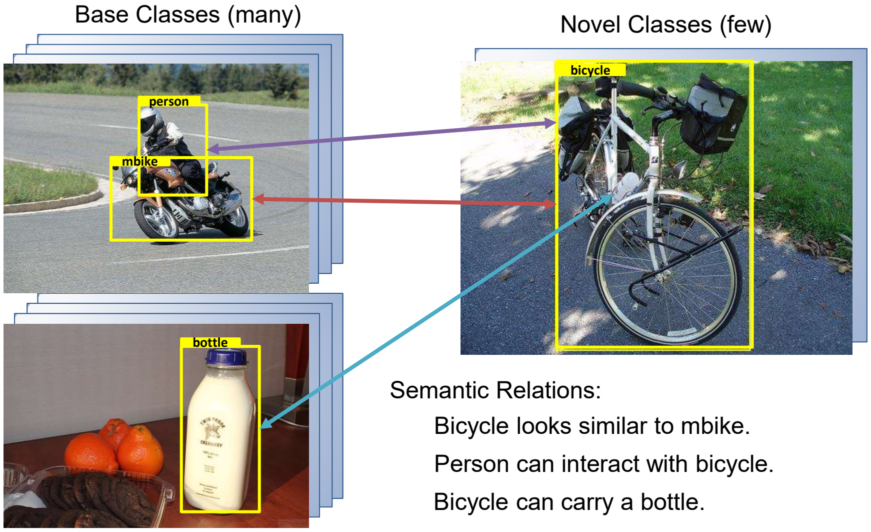

Base Classes (many)
Novel Classes (few)
person
mbike
bicycle
bottle
Semantic Relations:
Bicycle looks similar to mbike.
Person can interact with bicycle.
Bicycle can carry a bottle.

Figure 4.2: Key insight: the semantic relation between base and novel classes is constant regardless of the data availability of novel classes, which can aid the learning together with visual information.

classes. For example in Figure 4.2, if we have the prior knowledge that the novel class "bicycle" looks similar to "motorbike", can have interaction with "person", and can carry a "bottle", it would be easier to learn the concept "bicycle" than solely using a few images. Such explicit relation reasoning is even more crucial when visual information is hard to access [89].

So how can we introduce semantic relation to few-shot detection? In natural language processing, semantic concepts are represented by word embeddings [65, 70] from language models, which have been used in zero-shot learning methods [89, 1]. And explicit relationships are represented by knowledge graphs [66, 6], which are adopted by some zero-shot or few-shot recognition algorithms [89, 69]. However, these techniques are rarely explored in the FSOD task. Also, directly applying them to few-shot detectors leads to non-trivial practical problems, i.e. the domain gap between vision and language, and the heuristic definition of knowledge graph for classes in FSOD datasets (see Section 4.3.1 and 4.3.2 for details).

In this chapter, we explore the semantic relation for FSOD. We propose a Semantic Relation Reasoning Few-Shot Detector (SRR-FSD), which learns novel objects from both the visual information and the semantic relation in an end-to-end style. Specifically, we construct a semantic space using word embeddings. Guided by the word embeddings of the classes, the detector is trained to project the objects from the visual space to the semantic space and to align their image representations with the corresponding class embeddings. To address the aforementioned problems, we propose to learn a dynamic relation graph driven by the image data instead of predefining one based on heuristics. Then the learned graph is used to perform relation reasoning and augment the raw embeddings for reduced domain gap.

With the help of the semantic relation reasoning, our SRR-FSD demonstrates the shot-stable property in two aspects, see the red solid and dashed lines in Figure 4.1. In the common few-shot settings (solid lines), SRR-FSD achieves competitive performance at higher shots and significantly better performance at lower shots compared to state-of-the-art few-shot detectors. In a more realistic setting (dashed lines) where implicit shots of novel concepts are removed from the classification dataset for the pretrained model, SRR-FSD steadily maintains the performance while some previous methods have results degraded by a large margin due to the loss of implicit shots. We hope the suggested realistic setting can serve as a new benchmark protocol for future research.

4.1 Background of Semantic Reasoning in Vision Tasks

Semantic word embeddings have been used in zero-shot learning tasks to learn a mapping from the visual feature space to the semantic space, such as zero-shot recognition [89] and zero-shot object detection [1, 71]. In [13], semantic embeddings are used as the ground-truth of the encoder TriNet to guide the feature augmentation. In [31], semantic embeddings guide the feature synthesis for unseen classes by perturbing the seen feature with the projected difference between a seen class embedding and an unseen class embedding. In zero-shot or few-shot recognition [89, 69], word embeddings are often combined with knowledge graphs to perform relation reasoning via the graph convolution operation [43]. Knowledge graphs are usually defined based on heuristics from databases of common sense knowledge rules [66, 6]. [12] proposed a knowledge graph based on object co-occurrence for the multi-label recognition task. To our knowledge, the use of word embeddings and knowledge graphs are rarely explored in the FSOD task. Any-Shot Detector (ASD) [72] is the only work that uses word embeddings for the FSOD task. But ASD focuses more on the zero-shot detection and it does not consider the explicit relation reasoning between classes because each word embedding is treated independently.

4.2 The Setting of Few-Shot Object Detection

Conventional object detection problem has a base class set $\mathcal{C}_b$ in which there are many instances, and a base dataset $\mathcal{D}_b$ with abundant images. $\mathcal{D}_b$ consists of a set of annotated images $\{(x_i, y_i)\}$ where x_i is the image and y_i is the annotation of labels from $\mathcal{C}_b$ and bounding boxes for objects in x_i. For few-shot object detection (FSOD) problem, in addition to $\mathcal{C}_b$ and $\mathcal{D}_b$ it also has a novel class set $\mathcal{C}_n$ and a novel dataset $\mathcal{D}_n$, with $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. In $\mathcal{D}_n$, objects have labels belong to $\mathcal{C}_n$ and the number of objects for each class is k for k-shot detection. A few-shot detector is expected to learn from $\mathcal{D}_b$ and to quickly generalize to $\mathcal{D}_n$ with a small k such that it can detect all objects in a held-out testing set with object classes in $\mathcal{C}_b \cup \mathcal{C}_n$.

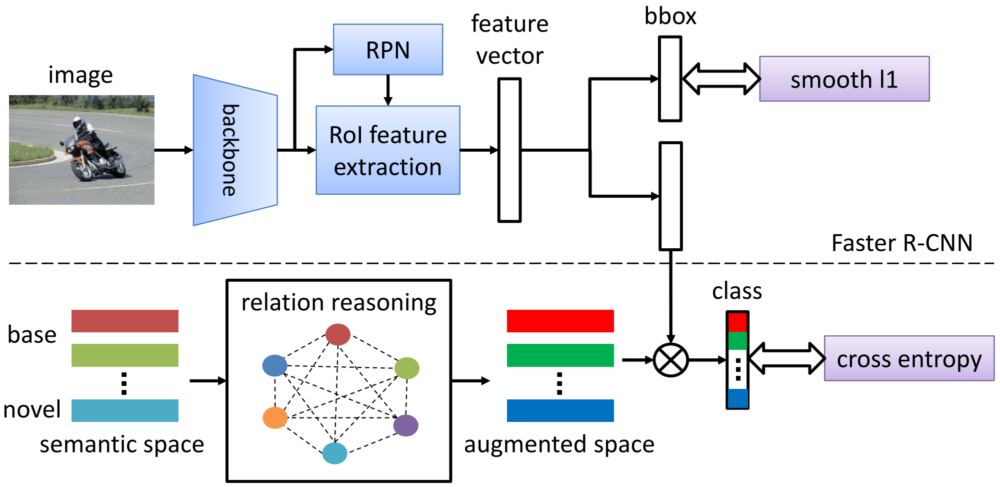


Figure 4.3: Overview of the SRR-FSD. A semantic space is built from the word embeddings of all corresponding classes in the dataset and is augmented through a relation reasoning module. Visual features are learned to be projected into the augmented space. "⊗": dot product.

A typical few-shot detector has two training phases. The first one is the base training phase where the detector is trained on $\mathcal{D}_b$ similarly to conventional object detectors. Then in the second phase, it is further finetuned on the union of $\mathcal{D}_b$ and $\mathcal{D}_n$. To avoid the dominance of objects from $\mathcal{D}_b$, a small subset is sampled from $\mathcal{D}_b$ such that the training set is balanced concerning the number of objects per class. As the total number of classes is increased by the size of $\mathcal{C}_n$ in the second phase, more class-specific parameters are inserted in the detector and trained to be responsible for the detection of novel objects. The class-specific parameters are usually in the box classification and localization layers at the very end of the network.

4.3 Semantic Relation Reasoning for Few-Shot Detector

Our proposed few-shot detector is built on top of Faster R-CNN [77], a popular two-stage general object detector. An overview of our method is presented in Figure 4.3. We modify the classification output in the second stage of Faster R-CNN with our semantic relation reasoning module [114].

In the second stage of the original Faster R-CNN, a feature vector is extracted for each region proposal and forwarded to a classification subnet and a regression subnet. In the classification subnet, the feature vector is transformed into a d-dimentional vector $\mathbf{v} \in \mathcal{R}^d$ through fully-connected layers. Then $\mathbf{v}$ is multiplied by a learnable weight matrix $\mathbf{W} \in \mathcal{R}^{N \times d}$ to output a probability distribution as in Equation (4.1).

$$\mathbf{p} = softmax(\mathbf{W}\mathbf{v} + \mathbf{b}) \tag{4.1}$$

where N is the number of classes and $\mathbf{b} \in \mathcal{R}^N$ is a learnable bias vector. Cross-entropy loss is used during training.

4.3.1 Semantic Space Projection

To learn objects from both the visual information and the semantic relation, we first construct a semantic space and project the visual feature $\mathbf{v}$ into this semantic space. Specifically, we represent the semantic space using a set of d_e-dimensional word embeddings $\mathbf{W}_e \in \mathcal{R}^{N \times d_e}$ [65] corresponding to the N object classes (including the background class). And the detector is trained to learn a linear projection $\mathbf{P} \in \mathcal{R}^{d_e \times d}$ in the classification subnet (see Figure 4.3) such that $\mathbf{v}$ is expected to align with its class's word embedding after projection. Mathematically, the prediction of the probability distribution turns into Equation (4.2) from Equation (4.1).

$$\mathbf{p} = softmax(\mathbf{W}_e \mathbf{P} \mathbf{v} + \mathbf{b}) \tag{4.2}$$

During training, $\mathbf{W}_e$ is fixed and the learnable variable is $\mathbf{P}$. A benefit is that generalization to novel objects involves no new parameters in $\mathbf{P}$. We can simply expand $\mathbf{W}_e$ with embeddings of novel classes. We still keep the $\mathbf{b}$ to model the category imbalance in the detection dataset.

Domain gap between vision and language. $\mathbf{W}_e$ encodes the knowledge of semantic concepts from natural language. While it is applicable in zero-shot learning, it will introduce the bias of the domain gap between vision and language to the FSOD task. Because the few-shot detector can rely on both the images and the embeddings to learn the concept of novel objects, which is different from the zero-shot learning where unseen classes have no support from images. When there are very few images to rely on, the knowledge from embeddings can guide the detector towards a decent solution. But when more images are available, the knowledge from embeddings may be misleading due to the domain gap, resulting in a suboptimal solution. Therefore, we need to augment the semantic embeddings to reduce the domain gap. Some previous works like ASD [72] apply a trainable transformation to each word embedding *independently*. But we find leveraging the explicit relationship between classes is more effective for embedding augmentation, leading to the proposal of the dynamic relation graph in Section 4.3.2.

4.3.2 Relation Reasoning

The semantic space projection learns to align the concepts from the visual space with the semantic space. But it still treats each class independently and there is no knowledge propagation among classes. Therefore, we further introduce a knowledge graph to model their relationships. The knowledge graph $\mathbf{G}$ is a $N \times N$ adjacency matrix representing the connection strength for every neighboring class pair. $\mathbf{G}$ is in-

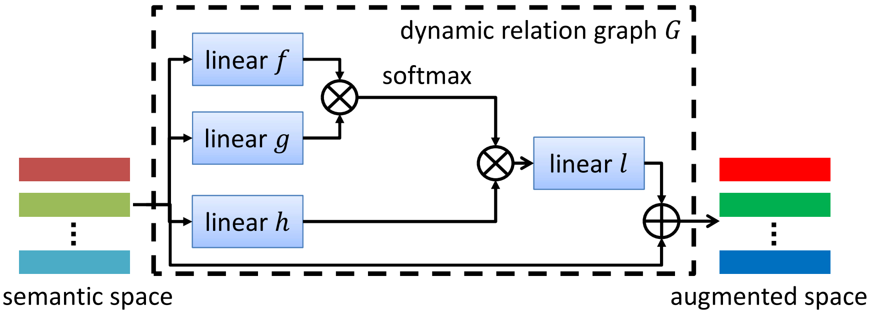


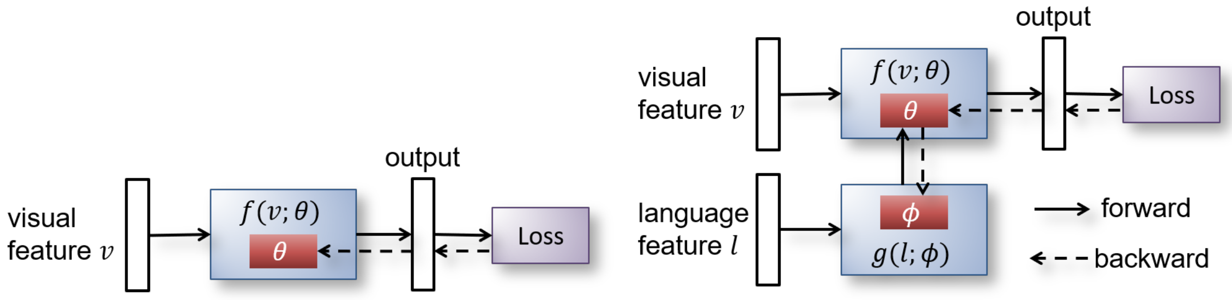
Figure 4.4: Network architecture of the relation reasoning module for learning the relation graph. "$\otimes$": dot product. "$\oplus$": element-wise plus.

volved in classification via the graph convolution operation [43]. Mathematically, the updated probability prediction is shown in Equation (4.3).

$$\mathbf{p} = softmax(\mathbf{G}\mathbf{W}_e\mathbf{P}\mathbf{v} + \mathbf{b}) \tag{4.3}$$

The heuristic definition of the knowledge graph. In zero-shot or few-shot recognition algorithms, the knowledge graph **G** is predefined base on heuristics. It is usually constructed from a database of common sense knowledge rules by sampling a sub-graph through the rule paths such that semantically related classes have strong connections. For example, classes from the ImageNet dataset [17] have a knowledge graph sampled from the WordNet [66]. However, classes in FSOD datasets are not highly semantically related, nor do they form a hierarchical structure like the ImageNet classes. The only applicable heuristics we found are based on object co-occurrence from [12]. Although the statistics of the co-occurrence are straightforward to compute, the co-occurrence is not necessarily equivalent to semantic relation.

Instead of predefining a knowledge graph based on heuristics, we propose to learn a *dynamic* relation graph driven by the data to model the relation reasoning between classes. The data-driven graph is also responsible for reducing the domain gap between vision and language because it is trained with image inputs. Inspired by the concept of the transformer, we implement the dynamic graph with the self-attention architecture [83] as shown in Figure 4.4. The original word embeddings $\mathbf{W}_e$ are transformed by three linear layers f, g, h, and a self-attention matrix is computed from the outputs of f, g. The self-attention matrix is multiplied with the output of h followed by another linear layer l. A residual connection [34] adds the output of l with the original $\mathbf{W}_e$. Another advantage of learning a dynamic graph is that it can easily adapt to new coming classes. Because the graph is not fixed and is generated on the fly from the word embeddings. We do not need to redefine a new graph and retrain the detector from the beginning. We can simply insert corresponding embeddings of new classes and fine-tune the detector.



(a) Conventional approaches

(b) Our approach

Figure 4.5: Comparison between conventional FSOD approaches and our SRR from a meta-learning's perspective.

4.3.3 Decoupled Fine-tuning

In the second fine-tuning phase, we only unfreeze the last few layers of our SRR-FSD similar to TFA [90]. For the classification subnet, we fine-tune the parameters in the relation reasoning module and the projection matrix **P**. For the localization subnet, it is not dependent on the word embeddings but it shares features with the classification subnet. We find that the learning of localization on novel objects can interfere with the classification subnet via the shared features, leading to many false positives. Decoupling the shared fully-connected layers between the two subnets can effectively make each subnet learn better features for its task. In other words, the classification subnet and the localization subnet have individual fully connected layers and they are fine-tuned independently.

4.3.4 Generalized Meta-Solution

The proposed Semantic Relation Reasoning (SRR) can be interpreted from a meta-learning perspective. Under the meta-learning view, our SRR differentiates from conventional FSOD approaches by the way of parameter optimization. Figure 4.5 visualizes this major difference.

Conventional approaches address FSOD by learning exclusively from visual information as shown in Figure 4.5a. Given a visual feature $\mathbf{v}$, it is fed into a network f parameterized by trainable class-specific θ to predict the output $\mathbf{y}$. Then the loss function compares the output with the ground truth and computes the gradient, which is back-propagated to update θ. Mathematically, the output is computed as in Equation (4.4).

$$\mathbf{y} = f(\mathbf{v};\theta) \tag{4.4}$$

Under such parameter optimization, data scarcity is a bottleneck that hurts the detector's generalization from a few samples. In other words, the performance is very sensitive to the shots variation and drops

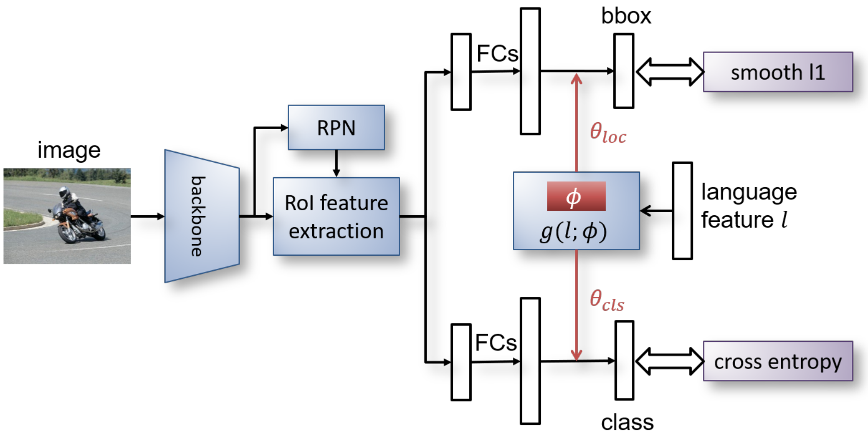


Figure 4.6: The architecture of SRR-FSD++, a more general version of SRR-FSD and a unified solution for few-shot classification and localization.

drastically as data becomes limited.

In our approach, the cross-domain knowledge is leveraged for parameter generation from semantic concepts in natural language as shown in Figure 4.5b. Instead of directly learning θ, θ is generated from a dynamic network g parameterized by class-agnostic ϕ. The network g takes in the language feature $\mathbf{l}$ representing the semantic concept of a class and outputs the class-specific parameter for the detector, as presented in Equation (4.5).

$$\mathbf{y} = f(\mathbf{v}; g(\mathbf{l}; \phi)) \tag{4.5}$$

One benefit of viewing SRR in the form of meta-learning is that we can easily generalize it to not only few-shot classification but also few-shot localization. In other words, our approach can be general enough to be applied to both the classification subnet and the localization subnet in the few-shot detector in a unified and coherent way. Notice that our SRR-FSD in Figure 4.3 still has the localization subnet trained conventionally, i.e. exclusively dependent on visual information. In a more general scenario, the network g can output two sets of parameters for the classification branch and the localization branch, respectively. Figure 4.6 demonstrates the architecture of this general solution of FSOD, which we call SRR-FSD++.

4.4 Experiments

4.4.1 Implementation Details

Our method is implemented based on Faster R-CNN [77] with ResNet-101 [34] and Feature Pyramid Network [56] as the backbone using the MMDetection [9] framework. All models are trained with Stochastic Gradient Descent (SGD) and a batch size of 16. For the word embeddings, we use the L2-normalized

Table 4.1: Ablative performance (mAP50) on the VOC Novel Set 1 by gradually applying the proposed components to the baseline Faster R-CNN. **SSP**: semantic space projection. **RR**: relation reasoning. **DF**: decoupled fine-tuning. **GMS**: generalized meta-solution.

	Components				Shots in Novel Set 1				
	SSP	RR	DF	GMS	1	2	3	5	10
Faster R-CNN [77]					32.6	44.4	46.3	49.6	55.6
	✓				40.5	46.8	46.5	47.1	52.2
	✓	✓			44.1	46.0	47.8	51.7	54.7
SRR-FSD	✓	✓	✓		47.8	50.5	51.3	55.2	56.8
SRR-FSD++	✓	✓	✓	✓	50.6	51.2	53.2	57.1	58.8

300-dimensional Word2Vec [65] vectors from the language model trained on large unannotated texts like Wikipedia. In the relation reasoning module, we reduce the dimension of word embeddings to 32 which is empirically selected. In the first base training phase, we set the learning rate, the momentum, and the weight decay to 0.02, 0.9, and 0.0001, respectively. In the second fine-tuning phase, we reduce the learning rate to 0.001 unless otherwise mentioned. The input image is sampled by first randomly choosing between the base set and the novel set with a 50% probability and then randomly selecting an image from the chosen set.

4.4.2 Ablation Study

In this section, we study the contribution of each component. Experiments are conducted on the VOC dataset. Our baseline is the Faster R-CNN [77] with ResNet-101 [34] and FPN [56]. We gradually apply the Semantic Space Projection (SSP 4.3.1), Relation Reasoning (RR 4.3.2), Decoupled Fine-tuning (DF 4.3.3), and Generalized Meta Solution (GMS) to the baseline and report the performance in Table 4.1. We also compare three different ways of augmenting the raw word embeddings in Table 4.2, including the trainable transformation from ASD [72], the heuristic knowledge graph from [12], and the dynamic graph from our proposed relation reasoning module.

Semantic space projection guides shot-stable learning. The baseline Faster R-CNN can already achieve satisfying results at 5-shot and 10-shot. But at 1-shot and 2-shot, performance starts to fall apart due to exclusive dependence on images. The semantic space projection, on the other hand, makes the learning more stable to the variation of shot numbers (see 1st and 2nd entries in Table 4.1). The space projection guided by the semantic embeddings is learned well enough in the base training phase so it can be quickly adapted to novel classes with a few instances. We can observe a major boost at lower shot conditions compared to baseline, i.e. 7.9 mAP and 2.4 mAP gain at 1-shot and 2-shot respectively. However, the raw semantic embeddings limit the performance at higher shot conditions. The performance at 5-shot and 10-shot drops below the baseline. This verifies our argument about the domain gap between vision

Table 4.2: Comparison of three ways of refining the word embeddings, including the trainable transformation from ASD [72], the heuristic knowledge graph from [12], and the dynamic relation graph from our relation reasoning module. **SSP**: semantic space projection. **RR**: relation reasoning. **TT**: trainable transformation. **HKG**: heuristic knowledge graph.

	Shots in Novel Set 1				
	1	2	3	5	10
+SSP	40.5	46.8	46.5	47.1	52.2
+SSP +TT [72]	39.3	45.7	43.9	49.4	52.4
+SSP +HKG [12]	41.6	45.5	47.8	49.7	52.5
+SSP +RR	44.1	46.0	47.8	51.7	54.7

and language. At lower shots, there is not much visual information to rely on so the language information can guide the detector to a decent solution. But when more images are available, the visual information becomes more precise then the language information starts to be misleading. Therefore, we propose to refine the word embeddings for a reduced domain gap.

Relation reasoning promotes adaptive knowledge propagation. The relation reasoning module explicitly learns a relation graph that builds direct connections between base classes and novel classes. So the detector can learn the novel objects using the knowledge of base objects besides the visual information. Additionally, the relation reasoning module also functions as a refinement to the raw word embeddings with a data-driven relation graph. Since the relation graph is updated with image inputs, the refinement tends to adapt the word embeddings for the vision domain. Results in Table 4.1 (2nd and 3rd entries) confirm that applying relation reasoning improves the detection accuracy of novel objects under different shot conditions. We also compare it with two other ways of refining the raw word embeddings in Table 4.2. One is the trainable transformation (TT) from ASD [72] where word embeddings are updated with a trainable metric and a word vocabulary. Note that this transformation is applied to each embedding independently which does not consider the explicit relationships between them. The other one is the heuristic knowledge graph (HKG) defined based on the co-occurrence of objects from [12]. It turns out both the trainable transformation and the predefined heuristic knowledge graph are not as effective as the dynamic relation graph in the relation reasoning module. The effect of the trainable transformation is similar to unfreezing more parameters of the last few layers during fine-tuning as shown in the Section 4.5.4, which leads to overfitting when the shot is low. And the predefined knowledge graph is fixed during training thus cannot be adaptive to the inputs. In other words, the dynamic relation graph is better because it can not only perform explicit relation reasoning but also augment the raw embeddings for a reduced domain gap between vision and language.

Decoupled fine-tuning reduces false positives. We analyze the false positives generated by our SRR-FSD with and without decoupled fine-tuning (DF) using the detector diagnosing tool [36]. The effect of

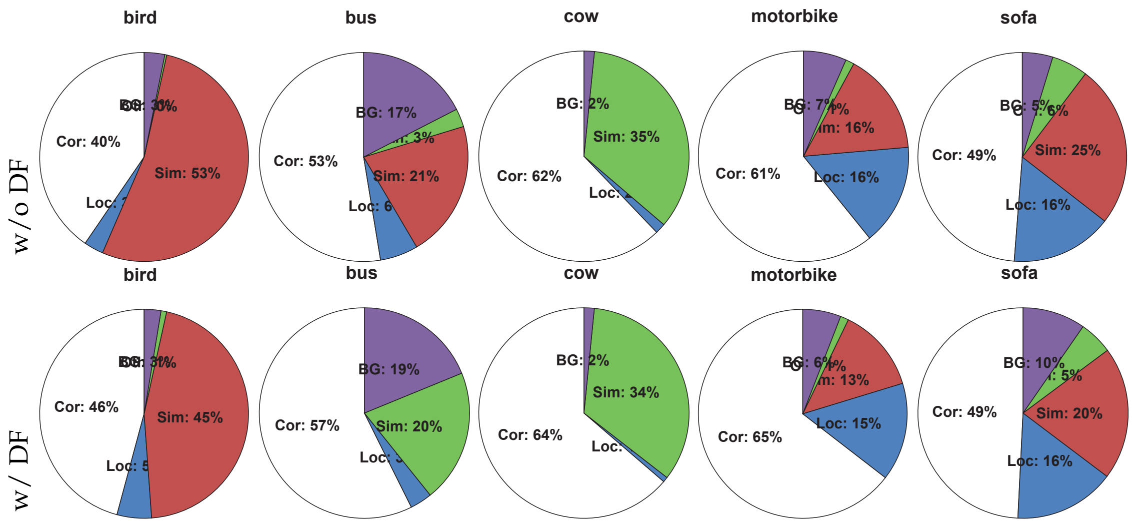


Figure 4.7: Error analysis of false positives in VOC Novel Set 1 with and without decouple fine-tuning (DF). Detectors are trained with 3 shots. Pie charts indicate the fraction of correct detections (Cor) and top-ranked false positives that are due to poor localization (Loc), confusion with similar objects (Sim), confusion with other VOC objects (Oth), or confusion with background or unlabeled objects (BG).

DF on reducing the false positives in novel classes is visualized in Figure 4.7. It shows that most of the false positives are due to misclassification into similar categories. With DF, the classification subnet can be trained independently from the localization subnet to learn better features specifically for classification.

Generalized meta-solution further improves localization. The SRR-FSD only applies semantic relation reasoning to the classification output. The localization subnet still learns the class-specific parameters only from the visual data. So in SRR-FSD++, the semantic relation reasoning is also applied to the localization subnet. The class-specific parameters are generated by a class-agnostic network from the corresponding semantic word embedding of that class. Compared with SRR-FSD, SRR-FSD++ improves the localization accuracy, leading to a more shot-tolerant performance shown as the 5h entry in Table 4.1.

4.4.3 Existing Settings

We follow the existing settings in previous FSOD methods [40, 91, 96, 90] to evaluate our methods on the VOC [22] and COCO [58] datasets. For fair comparison and reduced randomness, we use the same data splits and a fixed list of novel samples provided by [40].

VOC The 07 and 12 train/val sets are used for training and the 07 test set is for testing. Out of its 20 object classes, 5 classes are selected as novel and the remaining 15 are base classes, with 3 different base/novel splits. The novel classes each have k annotated objects, where k equals 1, 2, 3, 5, 10. In the first base training phase, our SRR-FSD is trained for 18 epochs with the learning rate multiplied by 0.1 at the 12th and 15th epoch. In the second fine-tuning phase, we train for $500 \times |\mathcal{D}_n|$ steps where $|\mathcal{D}_n|$ is the

Table 4.3: FSOD evaluation on VOC. We report the mAP with IoU threshold 0.5 (mAP50) under 3 different sets of 5 novel classes with a small number of shots.

	Novel Set 1					Novel Set 2					Novel Set 3				
Method / shot	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW [40]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet [91]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
Meta R-CNN [96]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA [90]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	**49.5**	**49.8**
SRR-FSD (Ours)	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	**40.1**	41.5	**44.3**	46.9	46.4
SRR-FSD++ (Ours)	**50.6**	**51.2**	**53.2**	**57.1**	**58.8**	**34.9**	**35.7**	**39.3**	**41.7**	**46.4**	39.5	**42.2**	42.5	46.5	47.8

Table 4.4: FSOD performance for the base and novel classes on Novel Set 1 of VOC. Our SRR-FSD has the merit of learning without forgetting.

Shot	Method	Base AP50	Novel AP50
3	Meta R-CNN [96]	64.8	35.0
	TFA [90]	79.1	44.7
	Ours base only	77.7	n/a
	SRR-FSD (Ours)	78.2	51.3
10	Meta R-CNN [96]	67.9	51.5
	TFA [90]	78.4	56.0
	Ours base only	77.7	n/a
	SRR-FSD (Ours)	78.2	56.8

number of images in the k-shot novel dataset.

We report the mAP50 of the novel classes on VOC with 3 splits in Table 4.3. In all different base/novel splits, our SRR-FSD achieves a more shot-stable performance. At higher shots like 5-shot and 10-shot, our performance is competitive compared to previous state-of-the-art methods. At more challenging conditions with shots less than 5, our approach can outperform the second-best by a large margin (up to 10+ mAP). Compared to ASD [72] which only reports results of 3-shot and 5-shot in the Novel Set 1, ours is 24.2 and 6.0 better respectively in mAP. We do not include ASD in Table 4.3 because its paper does not provide the complete results on VOC.

Learning without forgetting is another merit of our SRR-FSD. After generalization to novel objects, the performance on the base objects does not drop at all as shown in Table 4.4. Both base AP and novel AP of our SRR-FSD compare favorably to previous methods based on the same Faster R-CNN with ResNet-101. The base AP even increases a bit probably due to the semantic relation reasoning from limited novel objects to base objects.

COCO The `minival` set with 5000 images is used for testing and the rest images in train/val sets are for training. Out of the 80 classes, 20 of them overlapped with VOC are the novel classes with $k = 10, 30$ shots per class and the remaining 60 classes are base. We train the SRR-FSD on the base dataset for 12 epochs using the same setting as MMDetection [9] and fine-tune it for a fixed number of $10 \times |\mathcal{D}_b|$ steps

Table 4.5: FSOD performance of the novel classes on COCO.

Shot	Method	AP	AP50	AP75
10	FSRW [40]	5.6	12.3	4.6
	MetaDet [91]	7.1	14.6	6.1
	Meta R-CNN [96]	8.7	19.1	6.6
	TFA [90]	10.0	-	9.3
	MPSR [92]	9.8	17.9	9.7
	SRR-FSD (Ours)	11.3	**23.0**	9.8
	SRR-FSD++ (Ours)	**11.9**	22.7	**11.4**
30	FSRW [40]	9.1	19.0	7.6
	MetaDet [91]	11.3	21.7	8.1
	Meta R-CNN [96]	12.4	25.3	10.8
	TFA [90]	13.7	-	13.4
	MPSR [92]	14.1	25.4	**14.2**
	SRR-FSD (Ours)	14.7	**29.2**	13.5
	SRR-FSD++ (Ours)	**15.1**	28.8	13.9

where $|\mathcal{D}_b|$ is the number of images in the base dataset. Unlike VOC, the base dataset in COCO contains unlabeled novel objects, so the region proposal network (RPN) treats them as the background. To avoid omitting novel objects in the fine-tuning phase, we unfreeze the RPN and the following layers. Table 4.5 presents the COCO-style averaged AP. Again we consistently outperform previous methods including FSRW [40], MetaDet [91], Meta R-CNN [96], TFA [90], and MPSR [92].

COCO to VOC For the cross-domain FSOD setting, we follow [40, 91] to use the same base dataset with 60 classes as in the previous COCO within-domain setting. The novel dataset consists of 10 samples for each of the 20 classes from the VOC dataset. The learning schedule is the same as the previous COCO within-domain setting except the learning rate is 0.005. Figure 4.8 shows that our SRR-FSD achieves the best performance with a healthy 44.5 mAP, indicating better generalization ability in cross-domain situations.

4.4.4 A More Realistic Setting

The training of the few-shot detector usually involves initializing the backbone network with a model pretrained on large-scale object classification datasets such as ImageNet [17]. The set of object classes in ImageNet, i.e. $\mathcal{C}_0$, is highly overlapped with the novel class set $\mathcal{C}_n$ in the existing settings. This means that the pretrained model can get early access to large amounts of object samples, i.e. *implicit shots*, from novel classes and encode their knowledge in the parameters before it is further trained for the detection task. Even the pretrained model is optimized for the recognition task, the extracted features still have a big impact on the detection of novel objects (see Figure 4.1). However, some rare classes may have highly limited or valuable data in the real world that pretraining a classification network on it is not realistic.

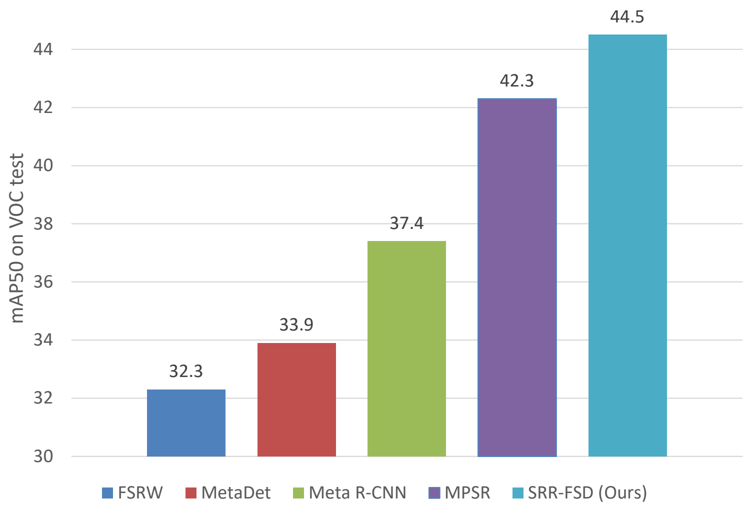


Figure 4.8: 10-shot cross domain performance on the 20 novel classes under COCO to VOC.

Therefore, we suggest a more realistic setting for FSOD, which extends the existing settings. In addition to $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$, we also require that $\mathcal{C}_0 \cap \mathcal{C}_n = \emptyset$. To achieve this, we systematically and hierarchically remove novel classes from $\mathcal{C}_0$. For each class in $\mathcal{C}_n$, we find its corresponding synset in ImageNet and obtain its full hyponym (the synset of the whole subtree starting from that synset) using the ImageNet API [1]. The images of this synset and its full hyponym are removed from the pretrained dataset. And the classification model is trained on a dataset with no novel objects. We provide the list of WordNet IDs for each novel class to be removed in the Appendix A.2.

We notice that CoAE [37] also proposed to remove all COCO-related ImageNet classes to ensure the model does not "foresee" the unseen classes. As a result, a total of 275 classes are removed from ImageNet including both the base and novel classes in VOC [22], which correspond to more than 300k images. We think the loss of this much data may lead to a worse pretrained model in general. So the pretrained model may not be able to extract features strong enough for down-streaming vision tasks compared with the model trained on full ImageNet. Our setting, on the other hand, tries to alleviate this effect as much as possible by only removing the novel classes in VOC Novel Set 1, 2, and 3 respectively, which correspond to an average of 50 classes from ImageNet.

Under the new realistic setting, we re-evaluate previous methods using their official source code and report the performance on the VOC dataset in Table 4.6. Our SRR-FSD demonstrates superior performance to other methods under most conditions, especially at challenging lower shot scenarios. More importantly, our SRR-FSD is less affected by the loss of implicit shots. Compared with results in Table 4.3, our performance is more stably maintained when novel objects are only available in the novel dataset.

[1] http://image-net.org/download-API

Table 4.6: FSOD performance (mAP50) on VOC under a more realistic setting where novel classes are removed from the pretrained classification dataset to guarantee $\mathcal{C}_0 \cap \mathcal{C}_n = \emptyset$. Our SRR-FSD is more robust to the loss of implicit shots comparing with Table 4.3.

	Novel Set 1					Novel Set 2					Novel Set 3				
Method / shot	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW [40]	13.9	21.1	20.0	29.9	40.8	13.5	14.2	20.6	20.7	36.8	16.2	22.2	26.8	37.0	41.5
Meta R-CNN [96]	11.5	22.2	24.7	36.4	45.2	10.1	16.9	22.7	29.6	40.1	10.0	21.7	27.1	32.8	41.6
TFA [90]	35.8	39.5	44.2	50.8	55.3	18.8	26.0	33.2	31.3	39.2	25.6	32.6	36.4	**43.7**	**48.5**
SRR-FSD (Ours)	**46.3**	**51.1**	**52.6**	**56.2**	**57.3**	**31.0**	**29.9**	**34.7**	**37.3**	**41.7**	**39.2**	**40.5**	**39.7**	42.2	45.2

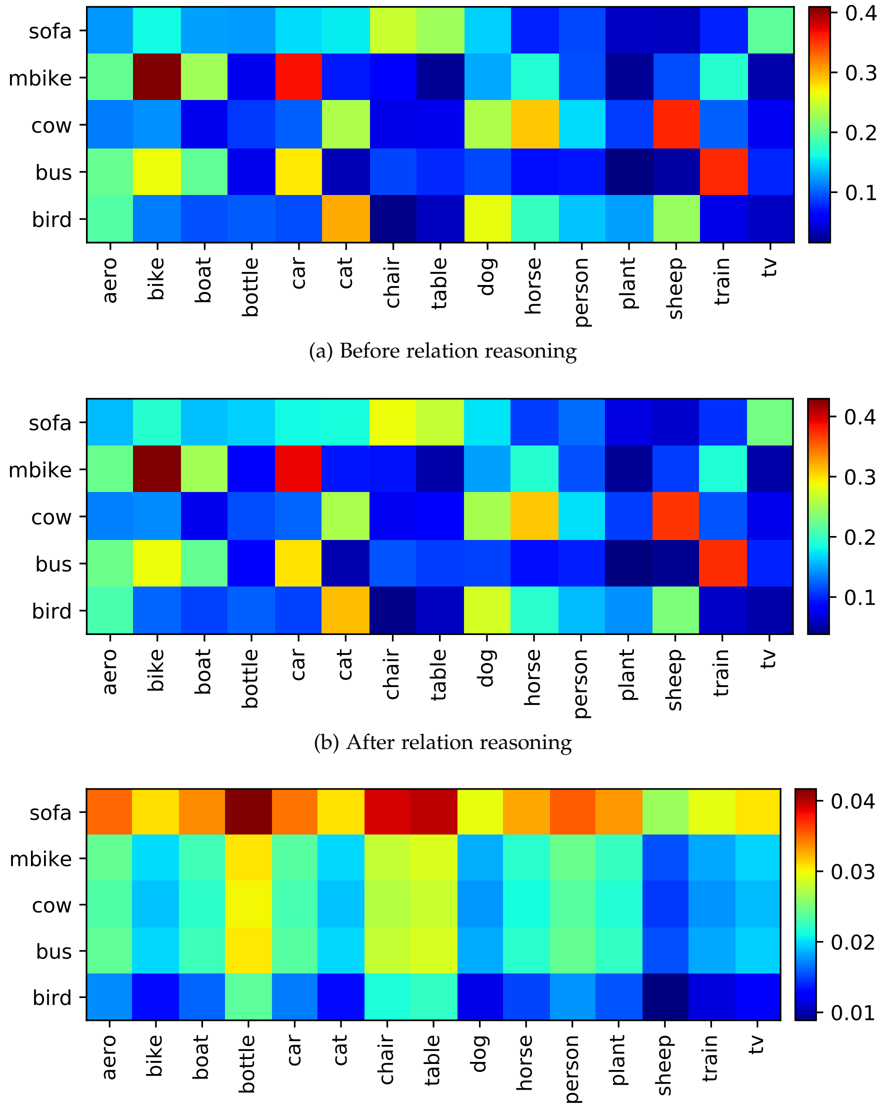
4.5 Discussions

4.5.1 Visualization of Relation Reasoning

Figure 4.9 visualizes the correlation maps between the semantic embeddings of novel and base classes before and after the relation reasoning, as well as the difference between the two maps. Nearly all the correlations are increased slightly, indicating better knowledge propagation between the two groups of classes. Additionally, it is interesting to see that some novel classes get more correlated than others, e.g. "sofa" with "bottle" and "sofa" with "table", probably because "sofa" can often be seen together with "bottle" and "table" in the living room but the original semantic embeddings cannot capture these relationships.

4.5.2 Using Other Word Embeddings

In the semantic space projection, we represent the semantic space using word embeddings from the Word2Vec [65]. We could simply set the $\mathbf{W}_e$ to be random vectors. Additionally, there are other language models for obtaining vector representations for words, such as the GloVe [70]. The GloVe is trained with aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. We also explored using word embedding with different dimensions from the GloVe in the semantic space projection step and compared it with the results by the Word2Vec. Performance on the VOC Novel Set 1 is reported in Table 4.7. The Word2Vec can provide better representations than the GloVe of both 300 dimensions and 200 dimensions. The performance of random embeddings is significantly worse than the meaningful Word2Vec and GloVe, which again verifies the importance of semantic information for shot-stable FSOD.



(a) Before relation reasoning

(b) After relation reasoning

(c) Difference between above correlation maps

Figure 4.9: Correlation of the semantic embeddings before and after the relation reasoning between the base classes and the novel classes on the VOC dataset. The novel classes are from Novel Set 1. The last figure shows how does the correlation change subtly. Some novel classes are getting more correlated with base classes after relation reasoning, e.g. “sofa” with “bottle” and “table”. *Best viewed in color.*

Table 4.7: FSOD performance (mAP50) on the VOC Novel Set 1 under different word embeddings in the semantic space projection. All models are using the ResNet-50 network. 300d and 200d mean the numbers of embedding dimension are 300 and 200 respectively. The Word2Vec provides better representations than the GloVe.

	Novel Set 1				
Word embeddings	shot=1	2	3	5	10
Random-300d	33.2	37.5	43.0	47.0	51.5
Word2Vec-300d [65]	42.8	47.1	49.0	50.8	52.8
GloVe-300d [70]	38.8	44.8	46.6	49.0	54.3
GloVe-200d [70]	39.7	44.6	45.8	49.4	53.0

Table 4.8: FSOD performance (mAP50) on the VOC Novel Set 1 under different reduced feature dimension in the relation reasoning module. Bold font indicates best or second best results. All models are using the ResNet-50 network.

	Novel Set 1				
Dimension	shot=1	2	3	5	10
128	40.9	44.6	44.3	48.1	54.1
64	42.0	**47.4**	**48.9**	51.7	54.1
32	**42.4**	**46.8**	**48.1**	**51.9**	**54.7**
16	**44.1**	46.0	47.8	51.7	**54.7**

4.5.3 Reduced Dimension in Relation Reasoning

In the relation reasoning module, the dimension of word embeddings is reduced by linear layers before computing the attention map, which saves computational time. We empirically test different dimensions and select the one with the best performance, i.e. when the dimension is 32. But other choices are just slightly worse. Table 4.8 reports the results on VOC dataset under different dimensions. All the experiments are following the same setting as in the main paper. The only exception is that we use ResNet-50 [34] to reduce the computational cost of tuning hyperparameters.

4.5.4 Finetuning More Parameters

Similar to TFA [90], we have a finetuning stage to make the detector generalized to novel classes. For the classification subnet, we finetune the parameters in the relation reasoning module and the projection matrix while all the parameters in previous layers are frozen. Some may argue that the improvement of our SRR-FSD over the baseline is due to more parameters finetuned in the relation reasoning module compared to the Faster R-CNN [77] baseline. But we show that finetuning more parameters does not necessarily lead to better results in Table 4.9. We take the TFA model which is essentially a Faster R-CNN finetuned with only the last layer trainable and gradually unfreeze the previous layers. It turns out more parameters involved in finetuning do not change the results substantially and that too many parameters will lead to severe overfitting.

4.5.5 Interpretation of the Dynamic Relation Graph

In the relation reasoning module, we propose to learn a *dynamic* relation graph driven by the data, which is conceptually different from the predefined fixed knowledge graphs used in [89, 12, 69]. We implement the dynamic graph with the self-attention architecture [83]. Although it is in the form of a feedforward network, it can also be interpreted as a computation related to the knowledge graph. If we denote the transformations in the linear layers f, g, h, l as $\mathbf{T}_f$, $\mathbf{T}_g$, $\mathbf{T}_h$, $\mathbf{T}_l$ respectively, we can formulate the relation

Table 4.9: FSOD results (mAP50) on the VOC Novel Set 1 with more and more tunable parameters in the finetuning stage. The baseline is TFA [90] which only finetunes the last classification layer in the Faster R-CNN. We gradually unfreeze more previous layers including two fully-connected layers (FCs) after the RoI-pooling, layers in region proposal network (RPN), and layers in the Backbone. This proves that finetuning more parameters does not guarantee better performance in few-shot detection.

	Novel Set 1				
Tunable Parameters	shot=1	2	3	5	10
Last layer (TFA [90])	39.8	36.1	44.7	55.7	56.0
+FCs	36.9	34.9	45.3	53.0	55.9
+FCs +RPN	37.2	39.8	44.3	52.7	56.2
+FCs +RPN +Backbone	16.2	19.5	24.8	39.2	44.6

reasoning in Equation (4.6)

$$\mathbf{W}'_e = \delta(\mathbf{W}_e \mathbf{T}_f \mathbf{T}_g^T \mathbf{W}_e^T) \mathbf{W}_e \mathbf{T}_h \mathbf{T}_l + \mathbf{W}_e \tag{4.6}$$

where $\mathbf{W}'_e$ is the matrix of augmented word embeddings after the relation reasoning which will be used as the weights to compute classification scores and δ is the softmax function operated on the last dimension of the input matrix. The item $\delta(\mathbf{W}_e \mathbf{T}_f \mathbf{T}_g^T \mathbf{W}_e^T)$ can be interpreted as a $N \times N$ dynamic knowledge graph in which the learnable parameters are $\mathbf{T}_f$ and $\mathbf{T}_g$. And it is involved in the computation of the classification scores via the graph convolution operation [43], which connects the N word embeddings in $\mathbf{W}_e$ to allow knowledge propagation among them. The item $\mathbf{T}_h \mathbf{T}_l$ can be viewed as a learnable transformation applied to each embedding independently.

4.6 Summary

In this chapter, we propose semantic relation reasoning for few-shot object detection. The key insight is to explicitly integrate semantic relation between base and novel classes with the available visual information, which can help to learn the novel concepts better especially when the novel class data is extremely limited. We apply the semantic relation reasoning to the standard two-stage Faster R-CNN and demonstrate robust few-shot performance against the variation of shot numbers. Compared to previous methods, our approach achieves state-of-the-art results on several few-shot detection settings, as well as a more realistic setting where novel concepts encoded in the pretrained backbone model are eliminated. We hope this realistic setting can be a better evaluation protocol for future few-shot detectors. Last but not least, the key components of our approach, i.e. semantic space projection and relation reasoning, can be straightly applied to the classification subnet of other few-shot detectors.

Chapter 5

Conclusion and Future Work

While the problem of real-world object detection has not yet been solved, this work presents several possible directions towards this goal. We identify three major challenges for successful object detection in the real world, i.e. appearance variation, scale variation, and availability variation. The key to this whole work is understanding the varying nature of the real-world data and addressing it in a divide-and-conquer style. While it may be easier, and therefore tempting, to just trust the generalization ability of deep networks to learn everything given enough data, we have shown that imposing the restrictions from prior knowledge can greatly benefit object detection. This is immediately noticeable in how we address the three variation challenges. For the appearance variation, we focus on the most common object, human faces. We enable the detector to construct the multi-resolution feature and to reason about the explicit human body context. For the scale variation, the conventional anchor design is thoroughly analyzed so that it is enhanced with a focus on improving the anchor's ability to achieve high IoU overlap with any object. The analysis also reveals the inherent limitations of the anchor and leads to the reformulation of object detection from an anchor-free perspective, which allows us to discover the importance of semantic guided feature selection and come up with an elegant solution so that less is more. For the availability variation, we propose semantic relation reasoning which makes use of both the semantic word embeddings from the natural language domain and the explicit relation graph for dynamic relation reasoning between semantic concepts.

5.1 Future Research Directions

Of course, the work outlined here is only the first few attempts in the direction of real-world object detection and understanding. For my future research, I want to continue working on the machine understanding of the real world in a much broader scenario, which has a wide range of applicable industries

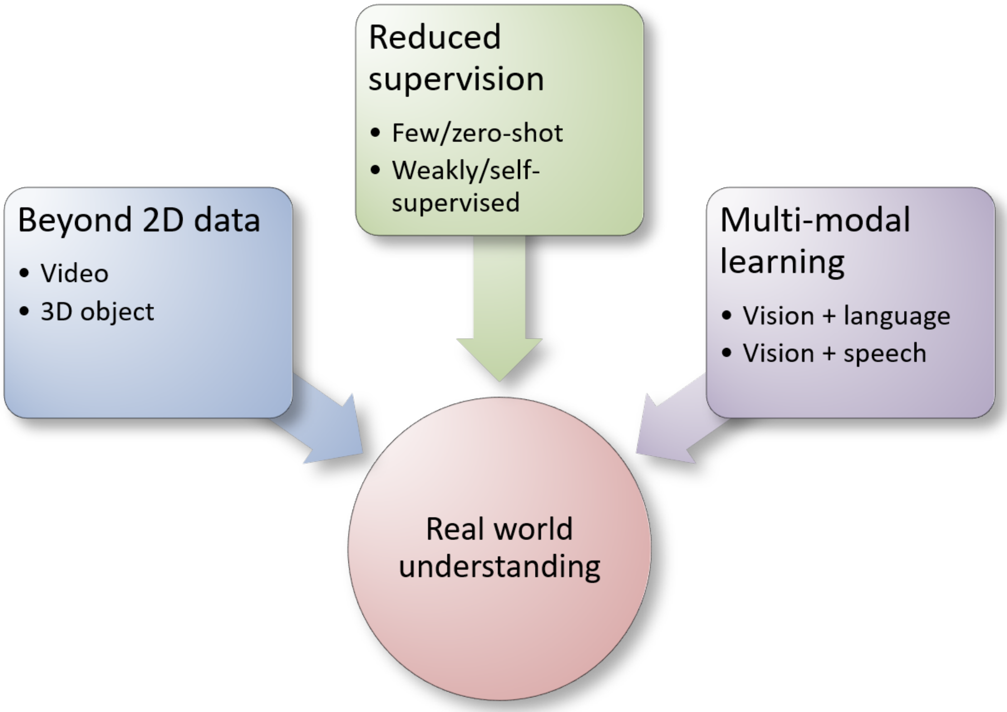


Figure 5.1: Future research directions.

such as augmented reality, autonomous vehicles, service on the cloud, wearable devices, content understanding and recommendation, etc. To be more specific, I think the following research directions have promising potentials to be explored as shown in Figure 5.1.

- **Beyond 2D Data.** Current approaches are operating on static 2D images, which is the projection of the 3D world. The projection causes the loss of information because the object is only captured from a single viewpoint at a single moment. Therefore, it is necessary to go beyond the 2D data to include information from additional dimensions. The additional dimension can be in time or space. Correspondingly, we can have videos or 3D objects as the data format. To represent 3D objects, we can use either point cloud or 3D mesh. With the 3D data, we can not only detect the existence of objects but also measure the distance between the objects and the camera. For few-shot object detection (Chapter 4), the 3D data is particularly helpful as it can provide multiple statuses of a few-shot object.

- **Reduced Supervision.** Most of the algorithms deployed in industrial products rely on supervised learning where data is fully annotated. However, as we move to the wilder aspect of the real world, we inevitably face the problem of reduced supervision. The data may have few-shot or even zero-shot samples not only due to the long-tail distribution but also because the users can customize their items, making the objects unique. Additionally, some real-world data is weakly/partially annotated, e.g. only the name or description of the object is provided but no bounding box or segmentation available. Therefore, weakly supervised learning is necessary to extract useful information from this kind of data. Even so, the majority of the real-world data has no annotation at all and this largely limits the model learning scope. Although self-supervised learning has shown promising progress, it is mostly focusing on preset tasks. If we can figure out an effective way to learn from unsupervised data directly on the down-streaming tasks, the model performance will reach another level.

- **Multi-Modal Learning.** The real world is very complex. Information comes from multiple domains, e.g. vision, language, speech, etc. These modalities are often entangled so that the real world can be represented from different perspectives. Therefore, it is beneficial to use the information from language or speech domains to help the learning of vision methods. Recently, the application of Transformer to the vision tasks has become a trend. While Transformer can achieve similar or even better performance than the conventional CNN family, I believe it is more suitable for multi-modal learning. Because the core part of Transformer is a natural way to transform the feature representation from one domain to another and the transformed feature is compatible with the target domain to perform further inference. This is essentially different from the naive feature fusion methods in current multi-modal learning tasks.

Appendix A

Supplementary Materials

A.1 IoU Loss

We use IoU loss [104] to optimize the box regression subnets for our anchor-free detection methods. As stated in Section 3.2.2, the localization targets can be represented as a 4-dimensional vector:

$$\mathbf{d}_{lij} = [d^l_{lij}, d^t_{lij}, d^r_{lij}, d^b_{lij}], \tag{A.1}$$

where $d^l_{lij}, d^t_{lij}, d^r_{lij}, d^b_{lij}$ are the normalized distances between the current pixel location and the left, top, right, and bottom boundaries of instance B, respectively. For simplicity, we omit l, i, j in the rest of derivation. Accordingly, a predicted projected box is defined as $\hat{\mathbf{d}} = [\hat{d}_l, \hat{d}_t, \hat{d}_r, \hat{d}_b]$. The forward pass and backward pass of the IoU loss l_{IoU} are presented as follows. We denote the area, width, height of the box as A, W, H, and the intersection, union as in, un respectively.

Forward pass:

$$\begin{aligned}
A &= (d_t + d_b) \times (d_l + d_r) \\
\hat{A} &= (\hat{d}_t + \hat{d}_b) \times (\hat{d}_l + \hat{d}_r) \\
H_{in} &= \min(d_t, \hat{d}_t) + \min(d_b, \hat{d}_b) \\
W_{in} &= \min(d_l, \hat{d}_l) + \min(d_r, \hat{d}_r) \\
A_{in} &= H_{in} \times W_{in} \\
A_{un} &= A + \hat{A} - A_{in} \\
IoU &= A_{in} / A_{un} \\
\mathcal{L} &= -\ln IoU
\end{aligned} \tag{A.2}$$

Backward pass: First we compute the derivative of predicted box area $\hat{A}$ w.r.t. $\hat{d}_*$ denoted as $\nabla_{\hat{d}_*}\hat{A}$:

$$\begin{aligned} \frac{\partial \hat{A}}{\partial \hat{d}_t} &= \frac{\partial \hat{A}}{\partial \hat{d}_b} = \hat{d}_l + \hat{d}_r \\ \frac{\partial \hat{A}}{\partial \hat{d}_l} &= \frac{\partial \hat{A}}{\partial \hat{d}_r} = \hat{d}_t + \hat{d}_b \end{aligned} \tag{A.3}$$

Then compute the derivative of intersection area A_{in} w.r.t. $\hat{d}_*$ denoted as $\nabla_{\hat{d}_*}A_{in}$:

$$\begin{aligned} \frac{\partial A_{in}}{\partial \hat{d}_t (\text{or } \partial \hat{d}_b)} &= \begin{cases} W_{in} & \hat{d}_t < d_t (\text{or } \hat{d}_b < d_b) \\ 0 & \text{o.w.} \end{cases} \\ \frac{\partial A_{in}}{\partial \hat{d}_l (\text{or } \partial \hat{d}_r)} &= \begin{cases} W_{in} & \hat{d}_l < d_l (\text{or } \hat{d}_r < d_r) \\ 0 & \text{o.w.} \end{cases} \end{aligned} \tag{A.4}$$

Finally the gradient of IoU loss l_{IoU} w.r.t. $\hat{d}_*$ is:

$$\frac{\partial \mathcal{L}}{\partial \hat{d}_*} = \frac{1}{A_{un}} \nabla_{\hat{d}_*}\hat{A} - \frac{A_{un} + A_{in}}{A_{un}A_{in}} \nabla_{\hat{d}_*}A_{in} \tag{A.5}$$

A.2 Removing Novel Classes from ImageNet

In Section 4.4.4, we propose a realistic setting for evaluating the few-shot object detection methods, where novel classes are completely removed from the classification dataset used for training a model to initialize the backbone network in the detector. This can guarantee that the object concept of novel classes will not be encoded in the pretrained model before training the few-shot detector. Because the novel class data is so rare in the real world that pretraining a classifier on it is not realistic.

ImageNet [17] is widely used for pretraining the classification model. It has 1000 classes organized according to the WordNet hierarchy. Each class has over 1000 images for training. We systematically and hierarchically remove novel classes by finding each synset and its corresponding full hyponym (synset of the whole sub-tree starting from that synset) using the ImageNet API [1]. So each novel class may contain multiple ImageNet classes.

For the novel classes in the VOC dataset [22], their corresponding WordNet IDs to be removed are as follows.

- aeroplane: n02690373, n02692877, n04552348
- bird: n01514668, n01514859, n01518878, n01530575, n01531178, n01532829, n01534433, n01537544, n01558993, n01560419, n01580077, n01582220, n01592084, n01601694, n01608432, n01614925, n01616318,

[1] http://image-net.org/download-API

n01622779, n01795545, n01796340, n01797886, n01798484, n01806143, n01806567, n01807496, n01817953, n01818515, n01819313, n01820546, n01824575, n01828970, n01829413, n01833805, n01843065, n01843383, n01847000, n01855032, n01855672, n01860187, n02002556, n02002724, n02006656, n02007558, n02009229, n02009912, n02011460, n02012849, n02013706, n02017213, n02018207, n02018795, n02025239, n02027492, n02028035, n02033041, n02037110, n02051845, n02056570, n02058221

- boat: n02687172, n02951358, n03095699, n03344393, n03447447, n03662601, n03673027, n03873416, n03947888, n04147183, n04273569, n04347754, n04606251, n04612504
- bottle: n02823428, n03062245, n03937543, n03983396, n04522168, n04557648, n04560804, n04579145, n04591713
- bus: n03769881, n04065272, n04146614, n04487081
- cat: n02123045, n02123159, n02123394, n02123597, n02124075, n02125311, n02127052
- cow: n02403003, n02408429, n02410509
- horse: n02389026, n02391049
- motorbike: n03785016, n03791053
- sheep: n02412080, n02415577, n02417914, n02422106, n02422699, n02423022
- sofa: n04344873

For the novel classes in the COCO dataset [58], they are very common in the real world. Removing them from the ImageNet does not make sense as much as removing data-scarce classes. So we suggest for large-scale datasets like COCO, we should follow the long-tail distribution of their class frequency and select the data-scarce classes on the distribution tail to be the novel classes.

Bibliography

[1] Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 384–400 (2018) 56, 57

[2] Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. arXiv preprint arXiv:1512.04143 (2015) 13

[3] Bhagavatula, C., Zhu, C., Luu, K., Savvides, M.: Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 1

[4] Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. pp. 5561–5569 (2017) 52

[5] Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018) 4, 52

[6] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Aaai. vol. 5. Atlanta (2010) 56, 57

[7] Chen, D., Ren, S., Wei, Y., Cao, X., Sun, J.: Joint cascade face detection and alignment. In: Computer Vision–ECCV 2014, pp. 109–122. Springer (2014) 11, 22, 25

[8] Chen, H., Wang, Y., Wang, G., Qiao, Y.: Lstd: A low-shot transfer detector for object detection. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018) 4, 54

[9] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019) 46, 62, 66

[10] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z.: Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. NIPS Workshop on Machine Learning Systems (LearningSys) (2016) 37

[11] Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 18

[12] Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5177–5186 (2019) ix, 57, 60, 63, 64, 71

[13] Chen, Z., Fu, Y., Zhang, Y., Jiang, Y.G., Xue, X., Sigal, L.: Multi-level semantic feature augmentation for one-shot learning. IEEE Transactions on Image Processing **28**(9), 4594–4605 (2019) 57

[14] Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016) 4, 37

[15] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017) xiii, 37, 51, 52

[16] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. Ieee (2005) 3

[17] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) xiii, 3, 46, 55, 60, 67, 77

[18] Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1271–1278. IEEE (2009) 13, 17

[19] Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 304–311. IEEE (2009) 1

[20] Dong, X., Zheng, L., Ma, F., Yang, Y., Meng, D.: Few-example object detection with model communication. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1641–1654 (2018) 4, 54

[21] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Object detection with keypoint triplets. In: Proceedings of the IEEE International Conference on Computer Vision (2019) 52

[22] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010) xiii, 1, 7, 24, 55, 65, 68, 77

[23] Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W.: Few-shot object detection with attention-rpn and multi-relation detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4013–4022 (2020) 4, 54

[24] Farfade, S.S., Saberian, M.J., Li, L.J.: Multi-view face detection using deep convolutional neural networks. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 643–650. ACM (2015) 11, 22, 25

[25] Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2008) 3

[26] Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence **14**(771-780), 1612 (1999) 3

[27] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 7

[28] Ghiasi, G., Fowlkes, C.C.: Occlusion coherence: Detecting and localizing occluded faces. arXiv preprint arXiv:1506.08347 (2015) 11, 22, 25

[29] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015) 4, 6

[30] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **38**(1), 142–158 (2016) 3

[31] Guan, J., Lu, Z., Xiang, T., Li, A., Zhao, A., Wen, J.R.: Zero and few shot learning with semantic feature synthesis and competitive learning. IEEE transactions on pattern analysis and machine intelligence (2020) 57

[32] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015) 7, 16, 39

[33] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 1, 6, 45

[34] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 1, 37, 43, 60, 62, 63, 71

[35] He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 28

[36] Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: European conference on computer vision. pp. 340–353. Springer (2012) 64

[37] Hsieh, T.I., Lo, Y.C., Chen, H.T., Liu, T.L.: One-shot object detection with co-attention and co-excitation. In: Advances in Neural Information Processing Systems. pp. 2725–2734 (2019) 68

[38] Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 951–959 (2017) 21, 22

[39] Jain, V., Learned-Miller, E.: Fddb: A benchmark for face detection in unconstrained settings. Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst (2010) 12, 22, 24

[40] Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8420–8429 (2019) 4, 54, 65, 66, 67, 69

[41] Karlinsky, L., Shtok, J., Harary, S., Schwartz, E., Aides, A., Feris, R., Giryes, R., Bronstein, A.M.: Repmet: Representative-based metric learning for classification and few-shot object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5197–5206 (2019) 4, 54

[42] Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3128–3137 (2015) 1

[43] Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional network. In: International Conference on Learning Representations (ICLR) (2017) 57, 60, 72

[44] Kong, T., Sun, F., Liu, H., Jiang, Y., Shi, J.: Foveabox: Beyond anchor-based object detector. arXiv preprint arXiv:1904.03797 (2019) 44, 51, 52

[45] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012) 1, 3

[46] Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018) 52

[47] Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J.: Probabilistic elastic part model for unsupervised face detector adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 793–800 (2013) 11, 22, 25

[48] Li, H., Lin, Z., Brandt, J., Shen, X., Hua, G.: Efficient boosted exemplar-based face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1843–1850 (2014) 11, 22, 25

[49] Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5325–5334 (2015) 11, 12, 22, 25

[50] Li, J., Wang, T., Zhang, Y.: Face detection using surf cascade. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. pp. 2183–2190. IEEE (2011) 22, 25

[51] Li, J., Zhang, Y.: Learning surf cascade for fast and accurate object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3468–3475 (2013) 11

[52] Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019) 52

[53] Li, Y., Sun, B., Wu, T., Wang, Y.: Face detection with end-to-end integration of a convnet and a 3d model. In: European Conference on Computer Vision. pp. 420–436. Springer (2016) 22, 24

[54] Liang, X., Wang, T., Yang, L., Xing, E.: Cirl: Controllable imitative reinforcement learning for vision-based self-driving. arXiv preprint arXiv:1807.03776 (2018) 1

[55] Liao, S., Jain, A.K., Li, S.Z.: A fast and accurate unconstrained face detector. IEEE transactions on pattern analysis and machine intelligence **38**(2), 211–223 (2015) 11, 22, 25

[56] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017) 4, 6, 28, 34, 43, 51, 52, 62, 63

[57] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017) 4, 8, 28, 42, 46, 47, 51, 52

[58] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 1, 7, 13, 46, 65, 78

[59] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016) 4, 28, 52

[60] Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579 (2015) 16

[61] Ma, X., He, Y., Luo, X., Li, J., Zhao, M., An, B., Guan, X.: Vehicle traffic driven camera placement for better metropolis security surveillance. IEEE Intelligent Systems (2018) 1

[62] Mallat, S.: A wavelet tour of signal processing. Academic press (1999) 34

[63] Markuš, N., Frljak, M., Pandžić, I.S., Ahlberg, J., Forchheimer, R.: A method for object detection based on pixel intensity comparisons organized in decision trees. arXiv preprint arXiv:1305.4537 (2013) 11, 22, 25

[64] Mathias, M., Benenson, R., Pedersoli, M., Van Gool, L.: Face detection without bells and whistles. In: Computer Vision–ECCV 2014, pp. 720–735. Springer (2014) 11, 22, 24, 25

[65] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013) 56, 59, 63, 69, 70

[66] Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995) 56, 57, 60

[67] Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: Ssh: Single stage headless face detector. In: Proceedings of the IEEE international conference on computer vision. pp. 4875–4884 (2017) 39

[68] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 821–830 (2019) 4, 48, 51, 52

[69] Peng, Z., Li, Z., Zhang, J., Li, Y., Qi, G.J., Tang, J.: Few-shot image recognition with knowledge transfer. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 441–449 (2019) 56, 57, 71

[70] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014) 56, 69, 70

[71] Rahman, S., Khan, S., Barnes, N.: Improved visual-semantic alignment for zero-shot object detection. In: AAAI. pp. 11932–11939 (2020) 57

[72] Rahman, S., Khan, S., Barnes, N., Khan, F.S.: Any-shot object detection. arXiv preprint arXiv:2003.07003 (2020) ix, 57, 59, 63, 64, 66

[73] Ranjan, R., Patel, V.M., Chellappa, R.: A deep pyramid deformable part model for face detection. In: Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. pp. 1–8. IEEE (2015) 11, 22, 25

[74] Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. arXiv preprint arXiv:1603.01249 (2016) 11, 19, 22, 25

[75] Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017) 4, 28, 54

[76] Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017) 52

[77] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015) 4, 13, 18, 20, 28, 54, 58, 62, 63, 71

[78] Van de Sande, K.E., Uijlings, J.R., Gevers, T., Smeulders, A.W.: Segmentation as selective search for object recognition. In: 2011 International Conference on Computer Vision. pp. 1879–1886. IEEE (2011) 3

[79] Shen, X., Lin, Z., Brandt, J., Wu, Y.: Detecting and aligning faces by image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3460–3467 (2013) 22, 24, 25

[80] Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 761–769 (2016) 37

[81] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) x, 1, 15

[82] Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019) 51, 52

[83] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 60, 71

[84] Viola, P., Jones, M.: 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. vol. 1, pp. I–511. IEEE (2001) 3, 11

[85] Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D.: Region proposal by guided anchoring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2965–2974 (2019) 52

[86] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016) 1

[87] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1328–1338 (2019) 1

[88] Wang, X., Gupta, A.: Videos as space-time region graphs. In: The European Conference on Computer Vision (ECCV) (September 2018) 1

[89] Wang, X., Ye, Y., Gupta, A.: Zero-shot recognition via semantic embeddings and knowledge graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6857–6866 (2018) 56, 57, 71

[90] Wang, X., Huang, T.E., Darrell, T., Gonzalez, J.E., Yu, F.: Frustratingly simple few-shot object detection. arXiv preprint arXiv:2003.06957 (2020) x, 4, 54, 61, 65, 66, 67, 69, 71, 72

[91] Wang, Y.X., Ramanan, D., Hebert, M.: Meta-learning to detect rare objects. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9925–9934 (2019) 4, 54, 65, 66, 67

[92] Wu, J., Liu, S., Huang, D., Wang, Y.: Multi-scale positive sample refinement for few-shot object detection. In: European conference on computer vision. Springer (2020) 4, 54, 67

[93] Xiao, Y., Marlet, R.: Few-shot object detection and viewpoint estimation for objects in the wild. In: European conference on computer vision. Springer (2020) 4, 54

[94] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) 1

[95] Yan, J., Zhang, X., Lei, Z., Li, S.Z.: Face detection by structural models. Image and Vision Computing **32**(10), 790–799 (2014) 24

[96] Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., Lin, L.: Meta r-cnn: Towards general solver for instance-level low-shot learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9577–9586 (2019) 4, 54, 65, 66, 67, 69

[97] Yang, B., Yan, J., Lei, Z., Li, S.Z.: Aggregate channel features for multi-view face detection. In: Biometrics (IJCB), 2014 IEEE International Joint Conference on. pp. 1–8. IEEE (2014) 11, 22, 25

[98] Yang, B., Yan, J., Lei, Z., Li, S.Z.: Convolutional channel features. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 82–90 (2015) 11, 22, 25

[99] Yang, B., Yan, J., Lei, Z., Li, S.Z.: Fine-grained evaluation on face detection in the wild. In: Automatic Face and Gesture Recognition (FG), 11th IEEE International Conference on. IEEE (2015) 12, 22, 24

[100] Yang, S., Luo, P., Loy, C.C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3676–3684 (2015) 11, 13, 14, 22, 25

[101] Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5525–5533 (2016) x, 1, 12, 20, 22, 29, 36

[102] Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019) 51, 52

[103] Yang, Z., Wang, Y., Chen, X., Liu, J., Qiao, Y.: Context-transformer: Tackling object confusion for few-shot detection. In: AAAI. pp. 12653–12660 (2020) 4, 54

[104] Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 516–520. ACM (2016) 42, 76

[105] Zagoruyko, S., Lerer, A., Lin, T.Y., Pinheiro, P.O., Gross, S., Chintala, S., Dollár, P.: A multipath network for object detection. arXiv preprint arXiv:1604.02135 (2016) 13

[106] Zhang, C., Zhang, Z.: A survey of recent advances in face detection (2010) 11

[107] Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multi-task cascaded convolutional networks. arXiv preprint arXiv:1604.02878 (2016) 22, 24

[108] Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4203–4212 (2018) 52

[109] Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: Freeanchor: Learning to match anchors for visual object detection. In: Advances in neural information processing systems (2019) 4, 52

[110] Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: Convex feature normalization for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5089–5097 (2018) 1

[111] Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 386–391 (2013) 22, 24, 25

[112] Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 52

[113] Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 850–859 (2019) 52

[114] Zhu, C., Chen, F., Ahmed, U., Shen, Z., Savvides, M.: Semantic relation reasoning for shot-stable few-shot object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 58

[115] Zhu, C., Chen, F., Shen, Z., Savvides, M.: Soft anchor-point object detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 9, 45

[116] Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 9, 43, 52

[117] Zhu, C., Tao, R., Luu, K., Savvides, M.: Seeing small faces from robust anchor's perspective. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 8, 28

[118] Zhu, C., Zheng, Y., Luu, K., Savvides, M.: Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In: Deep Learning for Biometrics, pp. 57–79. Springer (2017) 8, 28

[119] Zhu, C., Zheng, Y., Luu, K., Savvides, M.: Enhancing interior and exterior deep facial features for face detection in the wild. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 226–233. IEEE (2018) 8, 28

[120] Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2879–2886. IEEE (2012) 11, 12, 22, 24

[121] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV, pp. 391–405. Springer (2014) 20