Deep neural networks based visual object detection in videos

Jin, Ruibing

2020

Jin, R. (2020). Deep neural networks based visual object detection in videos. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/144136>

<https://doi.org/10.32657/10356/144136>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 22:10:33 SGT

Deep Neural Networks based Visual Object Detection in Videos

Ruibing Jin

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

11/08/2020

.....

Date

Ruibing Jin

.....

Ruibing Jin

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

11/08/2020

.....

Date

.....

Prof. Changyun Wen

Authorship Attribution Statement

This thesis contains material from 3 papers submitted to conferences in which I am listed as an author.

Chapter 3 is submitted as [Ruibing Jin, Guosheng Lin, Changyun Wen, Fayao Liu, and Jianliang Wang. Feature Flow: In-network Feature Flow Estimation for Video Object Detection. IEEE Transactions on Multimedia.](#)

The contributions of the co-authors are as follows:

- Prof. Changyun Wen, Prof. Guosheng Lin and Prof. Jianliang Wang supervised me on the overall research process.
- I proposed the idea, conducted the experiments, and prepared the manuscript.
- I carried out the experiments.
- The manuscript was revised by Prof. Guosheng Lin, Prof. Changyun Wen, Dr. Fayao Liu, and Prof. Jianliang Wang.

Chapter 4 is accepted as [Ruibing Jin, Guosheng Lin, Changyun Wen, and Jianliang Wang. Motion Context Network for Weakly Supervised Object Detection in Videos. IEEE Signal Processing Letters.](#)

The contributions of the co-authors are as follows:

- Prof. Changyun Wen, Prof. Guosheng Lin and Prof. Jianliang Wang supervised me on the overall research process.
- I proposed the idea, conducted the experiments, and prepared the manuscript.
- I carried out the experiments.
- The manuscript was revised by Prof. Guosheng Lin, Prof. Changyun Wen, and Prof. Jianliang Wang.

Chapter 5 is submitted as [Ruibing Jin, Guosheng Lin, and Changyun Wen. Online Active Proposal Set Generation for Weakly Supervised Object Detection. IEEE Transactions on Circuits and Systems for Video Technology.](#)

The contributions of the co-authors are as follows:

- Prof. Changyun Wen and Prof. Guosheng Lin supervised me on the overall research process.
- I proposed the idea, conducted the experiments, and prepared the manuscript.
- I carried out the experiments.
- The manuscript was revised by Prof. Guosheng Lin, and Prof. Changyun Wen.

11/08/2020

.....

Date

Ruibing Jin

.....

Ruibing Jin

Acknowledgements

I would like to express my greatest gratitude to my supervisors, Prof. Wen Changyun and Prof. Wang Jianliang, who provide continuous support and encouragement for my research. Their instructive advice helps me in both academic research and life plan.

I also wish to express my appreciation to my co-supervisor Prof. Lin Guosheng for his selfless help and patient instruction. His insightful suggestions and invaluable research experience greatly inspire my research during the Ph.D study.

I would also like to thank Prof. Yuan Junsong for his kind sharing and unconditional help at the beginning of my Ph.D study, which provides a solid foundation for my Ph.D research.

Based on it, I need to appreciate my elders and friends (Wang Yi, Xue Yuhao, Ning Yajing, Deng Tianwei, Liu Zhichao, Wang Hongyan, Jin Lili, Lin Yunli, Shi Jicheng, et al.) in Immanuel Christian Church (ICC). They give me warm care on my life and study, especially when I meet setbacks and get confused. The time spent with them enable me to learn to understand and consider other people, which is priceless treasure for me.

I also thank to all members in ST Engineering - NTU Corporate Laboratory for enthusiastic help and academic discussion. I also wish to express my gratitude to the research support from ST Engineering, providing sufficient hardware and software.

Last but not least, I must appreciate my beloved family who always help me without a word of complaint. Without you, I would have nothing. Thank you, my family.

“Before destruction the heart of man is haughty; And before honor goeth humility.”

— *Bible, Proverbs 18:12*

To my dear family

Contents

Acknowledgements	ix
Summary	xvii
List of Figures	xix
List of Tables	xxvii
Abbreviations	xxxi
1 Introduction	1
1.1 Background and Challenge	1
1.2 Motivation, Objectives and Contributions	3
1.3 Outline of the Thesis	4
2 Literature Review	7
2.1 Static Image based Object Detection	8
2.1.1 Traditional Methods	8
2.1.2 Deep Learning Based Detection Methods	10
2.2 Video Object Detection	19
2.2.1 Box-level Methods	19
2.2.2 Feature Level Methods	20
2.3 Weakly Supervised Object Detection	25
2.4 Benchmarks and Evaluation Metrics	31
2.4.1 Benchmarks	31
2.5 Evaluation Metrics	33
3 Feature Flow: In-network Feature Flow Estimation for Video Object Detection	37
3.1 Introduction	37
3.2 In-network Feature Flow Network	40
3.2.1 In-network Feature Flow Estimation Module	41
3.2.2 Transformation Residual Loss	44
3.2.3 Modified Seq-NMS	44

3.2.4	Network Architecture	45
3.3	Experiments	45
3.3.1	Experimental Setting	45
3.3.2	Comparison between Optical Flow and Feature Flow	46
3.3.3	Ablation Study	49
3.3.4	Comparison with the state-of-the-art	50
3.3.5	Speed Analysis	52
3.4	Conclusion	54
4	Motion Context Network for Weakly Supervised Object Detection in Videos	55
4.1	Introduction	55
4.2	Motion Context Network	58
4.2.1	Motion Correlation Module	60
4.2.1.1	Self-correlation layer	60
4.2.1.2	Instance pooling layer	61
4.2.2	Temporal Aggregation Module	62
4.2.2.1	Feature alignment	62
4.2.2.2	Feature aggregation	63
4.2.3	Fusion Module	63
4.3	Experiments	64
4.3.1	Experimental Setup	64
4.3.2	Ablation study	66
4.3.3	Analysis on optical flow	67
4.3.4	Comparison with other methods	69
4.3.5	Effectiveness of Motion Context	69
4.4	Conclusion	70
5	Online Active Proposal Set Generation for Weakly Supervised Object Detection	73
5.1	Introduction	73
5.2	Influence of Proposal Set	77
5.3	Online Active Proposal Set Generation	80
5.3.1	Dynamic Proposal Constraint	81
5.3.2	Proposal Ranking	81
5.3.3	Detection Framework	83
5.4	Experiments	85
5.4.1	Experiment Settings	85
5.4.2	Analysis of OPG Algorithm	86
5.4.3	Effectiveness of OPG Algorithm	88
5.4.4	Comparison with the state-of-the-art	91
5.5	Conclusion	93
6	Conclusion and Future Works	95

6.1 Conclusion	95
6.2 Future Works	96
Author's Publications	99
Bibliography	101

Summary

As a fundamental task in computer vision, object detection is to locate visual objects of pre-defined classes in instance level. Most existing object detection methods are proposed for static images, which usually try to utilize visual information to accurately locate and recognize an object. It is usually influenced by the object scale, context information, partial occlusion and so on.

Different from object detection in static images, video object detection is additionally affected by the deteriorated object appearances, like motion blur and objects under extreme viewpoints. Affected by these deteriorated appearances, the performances of static image based methods are not satisfactory. To alleviate this issue, many approaches are proposed to leverage temporal information to enhance the feature map at the current frame, improving the detection accuracy. These methods are generally composed of two components: temporal feature aggregation and object detection.

In temporal feature aggregation, due to the feature displacement across frames, people need to firstly align features across frames and combine these aligned features into an enhanced feature map. Optical flow, which expresses pixel displacement, is very popular in computer vision field. Considering pixel displacement is not strictly consistent to feature displacement, they apply a common approach: pre-train a neural network to predict optical flow and fine-tune this pre-trained network at the task dataset. With this method, they expect the fine-tuned network to produce tensors encoding feature displacement. However, due to domain gap between the optical flow training dataset and the task dataset, these optical flow based methods cannot accurately align feature maps cross frames, which degrades their performances. Furthermore, the pre-training process requires another dataset with optical flow annotations, which results the training process complex and time-consuming.

In the object detection component, existing methods are proposed under full supervision. They require bounding box-level annotations for each frames. Video

dataset usually includes hundreds of videos and each video involves hundreds of frames. Constructing such a large-scale video dataset is labor-intensive.

To address these issues, three novel methods are proposed in this thesis. For the temporal feature aggregation, we rethink this de facto paradigm (pre-train and fine-tune) and analyze its drawbacks. Based on our analyze, we propose a novel network (IFF-Net) with an **In**-network **F**eature **F**low estimation module (IFF) to solve these drawbacks. Without resort to pre-training on any additional dataset, our IFF module is able to directly produce **feature flow** which expresses the feature displacement. Our IFF module consists of a shallow module which shares the features with the detection branches. This compact design enables our IFF-Net to accurately detect objects, while maintain a fast inference speed. Additionally, we propose a transformation residual loss (TRL) based on *self-supervision*, which further improves the performance of our IFF-Net.

For the object detection component, we propose a video object detection framework under weak supervision, where only image-level annotations are required and much labor cost is saved. Existing weakly supervised object detection methods are proposed for images. Lacking of box-level annotations, these methods usually cannot accurately locate objects. Considering an object may show different motion from its surrounding objects or background, we leverage motion information to improve the detection accuracy. However, the motion pattern of an object is generally complex in videos, e.g., different parts of the object may have different motion patterns and some object parts may show similar motion as background regions in some scenarios, which poses great challenges in exploring motion information for accurate object localization. Directly using motion information to refine object regions may lead to degraded localization performance. To overcome these issues, we propose a **M**otion **C**ontext **N**etwork (MC-Net) which effectively leverages motion information to improve object localization in the weakly supervised setting. Our MC-Net generates motion context features by exploiting neighborhood motion correlation information on moving regions. These motion context features are then incorporated with image information to improve the detection accuracy. Benefited from the motion context information, our weakly supervised method is able to accurately localize objects.

Apart from it, weakly supervised object detection methods generally train detectors with a fixed training proposal set, which are dominated by overwhelming negative

samples. Without box-level annotations, it is a challenge for these methods to filter out redundant negative samples. This causes the weakly supervised learning procedure may be biased towards negative samples and degrades the detection accuracy. To investigate the effect of the imbalance between positive and negative samples during training, we conduct a series of analysis experiments and find this imbalance heavily hinders the detection accuracy. To alleviate this issue, we propose an **O**nline Active **P**roposal Set **G**eneration (OPG) algorithm which online generates an active training proposal set according to the prediction of the training model. This active proposal set generated by our OPG maintains a balance between positive and negative samples, which effectively improves the detection accuracy.

List of Figures

1.1	The illustration of deteriorated appearances of objects in videos. In the first row, due to motion blur, the fox is not displayed clearly. Especially for the last image, only part body of this fox is shown. For the second row, various viewpoints of a cat are shown. Affected by some extreme viewpoints, the detection accuracy mat be degraded.	2
2.1	Illustration of features selected by the Adaboost algorithm in VJ detectors.	9
2.2	The architecture of LeNet-5	11
2.3	The illustration of frameworks in R-CNN, SPP-Net, Fast R-CNN, and Faster R-CNN. The first branch indicates the R-CNN framework. SPP-Net and Fast R-CNN are similar in their detection frameworks. We summary them and illustrate their frameworks at the second branch. Faster R-CNN is illustrated in the third branch.	12
2.4	The illustration of anchors. The sub-region is projected to its corresponding region on the original image and is reshaped according to these anchor shapes.	13
2.5	The framework of R-FCN. R-FCN replaces the fully connected layers with convolutional layers. The feature map from the CNN network backbone is passed to a convolution layer with $k^2 * (C + 1)$ kernels to produce a position sensitive score map, where different color feature maps denote the features in differently related locations. The prediction result is computed by average pooling on these corresponding feature responses.	14
2.6	The illustration of YOLO and SSD. The upper row describes YOLO detection framework, which directly predicts detection results in one-stage. The bottom row indicates SSD detection framework. SSD produces multi-scale feature maps in a progressive way and detects objects on these feature maps. Benefiting from multi-scale features, SSD shows a better detection performance than YOLO.	15
2.7	Feature pyramid comparison between SSD and FPN. SSD simply predicts on different scale feature maps, while FPN effectively combines multi-scale features via literal connection and detects objects on these fused feature maps.	17
2.8	The outline of deep learning based detection methods.	18
2.9	The illustration of box-level video object detection methods.	20

2.10	The illustration of optical flow based video object detection methods. Raw images are passed into two deep neural networks: a convolutional neural network (CNN) and FlowNet. The CNN serves as a feature extractor and FlowNet is used to produce optical flow for feature alignment. The adaption module is used to improve the performance of feature alignment.	21
2.11	The illustration of MatchTrans module in STMN.	22
2.12	Comparison between MatchTrans and PSLA. MatchTrans and PSLA are shown at the left and right side, respectively. Compared with MatchTrans, PSLA compares the feature similarity in a progressive sparser stride manner. This method is able to increase the match field, while saves much computation.	23
2.13	Summary of video object detection methods.	25
2.14	Illustration of Weakly Supervised Deep Detection Networks. WSDN passes images to a convolutional neural network (CNN) and the selective search algorithm to predict the feature map and proposals, respectively. These proposals and the predicted feature maps are forwarded into a ROI pooling layer to generate proposal features, which are then passed to two branches, predicting the final results. .	27
2.15	Illustration for ContextLocNet. ContextLocNet extracts features on the proposal region and its corresponding context region. Features on these two regions are fused together by two different ways: addition or contrast, providing strong feature representation for object detection.	28
2.16	Illustration for OICR	29
2.17	Illustration for the methods which jointly train weakly supervised segmentation and detection networks.	30
2.18	Sample images in PASCAL VOC and COCO benchmarks.	32
2.19	Sample images in YTO benchmark.	34
2.20	Sample images in ImageNet VID benchmark.	34
2.21	Relationship among true positive (TP), false positive (FN), true negative (TN), and false negative (FN).	35
3.1	The Pipeline of IFF-Net (in-network feature flow estimation network) is illustrated in the case of fusing the reference features at frame t with its neighboring features at frame $t - 5$. The features at frame $t - 5$ are warped to the features at frame t via the FFM (feature flow maps) produced by IFF (feature flow estimation module). Then, aggregated features are generated by fusing the features at frame t and the warped features at frame $t - 5$. After that, the aggregated features are forwarded to RPN and a detector to produce results. The transformation residual loss (TRL) is applied only in the training process.	40

3.2	Different structures for feature flow estimation module (IFF). F_i denotes the feature of the reference frame and F_j denotes the feature of its neighboring frame. The structure of the basic IFF is illustrated in (a). An advanced version of IFF is illustrated in (b).	42
3.3	Illustration of the structure of EB-1, shown in (a), and EB-2, shown in (b).	43
3.4	Visualization of feature flow maps. The first to the fifth rows respectively list original images, corresponding feature map visualizations, optical flow from Flownet, feature flow maps from our IFF and detection visualizations.	48
3.5	Examples of detection results based on IFF-Net.	53
4.1	Illustration of optical flows for two objects. RGB images and optical flows are shown on the left and right, respectively. The airplane is clearly highlighted in the optical flow. In contrast, as a non-rigid object, this bear's motion pattern is complex and the region of the whole object cannot be directly obtained from its corresponding optical flow.	56
4.2	Detection comparison between the baseline method OICR (left) and our MC-Net (right). One optical flow image is shown in the middle, where two moving regions of this hamster are marked in dashed boxes. The baseline method fails to locate the object. By incorporating motion context information and appearance information from images, our MC-Net accurately detect this hamster.	57
4.3	Architecture of MC-Net (Motion Context Network). It illustrates the case of using two neighboring frames (I_{t-5} and I_{t+5}) as inputs. Frame, I_t , and its neighboring frames, I_{t-5} and I_{t+5} , are forwarded to MC-Net. The Network backbone and the FLOWNet produce image feature maps and optical flows, respectively. The concatenated optical flows are passed to Motion Correlation Module (MCM). The feature maps from neighboring frames and optical flows are passed to Temporal Aggregation Module (TAM) to generate enhanced features. Finally, the aggregated appearance features and the motion context features are fused and forwarded to OICR to generate the final results.	59
4.4	Illustration of self-correlation layer. Given a spatial location and a predefined neighborhood region, self-correlation layer computes the motion correlation between this location and its neighborhood. The central location, its neighborhood region and the motion correlation are visualized in purple, blue, and gray colors, respectively. The results are then reshaped into a vector.	60

4.5	The structure of fusion module. The features of appearance information are passed to two fully connected (FC) layers. The features from motion context information are passed to one FC layer. Then, both outputs are concatenated together and forwarded to OICR. With this module, our MC-Net is able to effectively incorporate appearance and motion context information together.	64
4.6	Visualization of some detection results. The detection results of OICR (1st row), our MC-Net (2nd column), and the corresponding optical flows (3rd column) are shown in the figure. With the proposed motion context component, our MC-Net effectively improves the detection results of OICR. It shows that our MC-Net effectively incorporates motion context and appearance information to improve the detection accuracy. It generalizes well on both rigid and non-rigid objects.	71
5.1	Diagram of the active training proposal set from our Online Active Proposal Set Generation (OPG) during a training process. Circle and rectangle points indicate negative and positive samples, respectively. Our active training proposal set is initialized with the fixed set in weakly supervised detection methods. During the training process, our OPG online generates the active proposal set according to the predictions of the training model. The ratio between positive and negative samples in our active set gradually becomes similar to that of fully supervised methods, which mitigate the issue arised by the imbalance between positive and negative samples, improving the detection accuracy.	75
5.2	Illustration of the basic detection framework under weak supervision. To investigate the influence of the ratio between positive and negative samples, we firstly train a network with only blue parts under weak supervision to generate pseudo ground truth. Then, we train the whole network including blue and yellow parts without sharing the parameters of the trained network above. During the second training process, we control the ratio between positive and negative samples according to pseudo ground truths. The original proposal set is denoted as R and the filtered set is indicated by R_f	78
5.3	The performances of the trained models under different N_r values. It shows that the performance of the trained model is significantly affected by the ratio between positive and negative samples.	79
5.4	Visualization of the proposal number in our active training proposal set during the training process. It can be seen that nearly all proposals are involved at the warm up stage, which provides a solid foundation for the performance improvement as the following stage. At the transition stage, the number of proposals gradually decreases. At the stable stage, only few proposals are in involved, which prevents the training model from biasing to negative samples.	82

5.5	Illustration of our weakly supervised object detection framework. The basic branch is the same as the FC layers of the network in Sec. 5.2. We firstly forward images and corresponding proposals to the basic branch. Then, our OPG-1 generates an active proposal set according to the predictions from the basic branch. The pseudo ground truth, PGT-1 and the active proposal set from OPG-1 are passed to the OICR branch, OICR-1. We repeat this process by 2 iterations. The predictions of these three OICR branches are averaged to obtained the final results.	84
5.6	Visualization of the proposal number in our active training proposal set under different α values. It shows that α controls the occupation of the transition stage in the training process.	86
5.7	Visualization of the proposal number in our active training proposal set under different β values. It can be seen that β changes the ratio between the epochs in the warm up stage and that in the stable stage.	87
5.8	Visualization of our detection results.	92

List of Tables

3.1	Comparison between optical flow and feature flow in feature alignment. OptFlow-A and OptFlow-B apply optical flow to feature warping, while IFF-Net uses feature flow in feature alignment. This shows that optical flow is not suitable for feature alignment, while our feature flow is able to effectively align temporal features.	47
3.2	Ablation study of our IFF-Net and comparison with optical flow based methods. IFF, TRL and Seq-NMS+ improve the performance of IFF-Net. OptFlow-A and OptFlow-B apply optical flow to feature warping, while IFF-Net uses feature flow in feature warping. All IFF-Net variants are better than these optical flow based methods. It shows that our IFF is better than optical flow in feature warping.	49
3.3	Comparison with state-of-the-art methods without any post-processing method. It shows that our IFF-Net-C achieves the best performance among other methods without post-processing method.	51
3.4	Comparison with state-of-the-art methods under different settings. It shows that our IFF-Net-H achieves the best performance among other methods with less training dataset and smaller network on ImageNet VID validation set.	51
3.5	Comparison with optical flow based methods on feature alignment per frame. It shows that our IFF-Net aligns feature maps much faster than FGFA.	52
4.1	Ablation study in mAP for our MC-Net. MCM (max) is MCM with ROI pooling and MCM (avg) is MCM with ROI Align. It shows that each component of our MC-Net effectively improves the performance on all types of motion. In mAP, almost all MC-Net variants perform better than the baseline (OICR).	66
4.2	Ablation study in CorLoc of our MC-Net. From this table, It can be seen that each component of our MC-Net effectively improves the CorLoc performance. Almost all MC-Net variants perform better than the baseline (OICR).	66

4.3	Analysis between optical flow and motion context. MC-Net-OptFlow is an alternative method to motion context, directly fusing optical flow features with image features for object detection. MC-Net-MotionContext denotes using motion correlation module to capture motion context information. MC-Net-OptFlow show better performance on mAP and CorLoc in VID than the baseline, while performs worse on CorLoc in DET. MC-Net-MotionContext performs better on all mAP, CorLoc in VID and DET. It shows that motion correlation module is more effective for object location compared to directly using optical flow.	68
4.4	Comparisons with other methods. Fast RCNN denotes a model trained in full supervision, serving as a performance upper-bound. MC-Net + PGT denotes Fast RCNN trained with the pseudo ground truth from our MC-Net. It shows that our MC-Net effectively improves the performance of OICR and shows the best performance under weak supervision on ImageNet VID.	69
5.1	The ratio between positive and negative samples in training proposal sets. WSOD is short for weakly supervised object detection methods. Since nearly all weakly supervised object detection use a fixed training proposal set including all proposals, we use WSOD to indicate them. The table shows that this ratio for fully supervised methods are 1:3 or 1:1, while the ratio for weakly supervised methods is unreasonable (around 1:100). This unreasonable ratio may degrade the detection accuracy of weakly supervised detection methods.	74
5.2	Experiments for various α on PASCAL VOC 2007 dataset, where β is fixed at 0.8. Multi-scale training is only applied in training process. It shows that nearly all model trained with our OPG show better performances than our baseline. The best performance is obtained when α is 10.0.	87
5.3	Experiments for various β on PASCAL VOC 2007 dataset, where α is fixed at 10.0. It shows that all model trained with our OPG show better performances than our baseline. The best performance is obtained when β is 0.8.	88
5.4	Compare our OPG method with the baseline in <i>mAP</i> on PASCAL VOC 2007 <i>test</i> set. After employing our OPG, the performance of our method surpasses the baseline in nearly all classes. It shows that our OPG algorithm significantly improves the performance of the baseline in mAP.	89
5.5	Compare our OPG method with the baseline in <i>CorLoc</i> on PASCAL VOC 2007 <i>trainval</i> set. After employing our OPG, the performance is improved. It shows that our OPG algorithm effectively improves the performance of the baseline in CorLoc.	89

5.6	Comparison with other methods proposed for improve proposal quality in PASCAL VOC 2007 and VOC 2012 dataset. It shows that our OPG performs much better than other two methods in mAP. For CorLoc, our OPG shows a comparable performance to W-RPN and a better performance than WeakRPN[1].	90
5.7	Comparison with other methods. It demonstrates that our method (OICR+OPG) shows a comparable performance to other state-of-the-art methods. Most listed methods are trained in a fixed training proposal set. Our OPG can be applied to these listed methods to further improve their performances.	91

Abbreviations

CNN	C onvolutional N eural N etworks
DCN	D eformable C onvolutional N etworks
DPC	D ynamic P roposal C onstraint
DPM	D eformable P art-based M odel
FC layers	F ully C onnected layers
FFM	F eature F low M ap
FN	F alse N egative
FP	F alse P ositive
FPN	F eature P yramid N etwork
HOG	H istogram of O riented G radients
IFF	I n-network F eature F low
IOU	I ntersection O ver U nion
LSTM	L ong S hort- T erm M emory
mAP	m ean A verage P recision
MCM	M otion C orrelation M odule
MIL	M ultiple I nstance L earning
NMS	N on- M aximum S uppression
OPG	O nline A ctive P roposal S et G eneration
PGT	P seudo G round T ruth
PR	P roposal R anking
PSLA	P rogressive S pase L ocal A ttention module
ROI	R egion O f I nterest
RPN	R egion P roposal N etwork

SGD	S tochastic G radient D escent
SPP	S patial P yramid P ooling
SVM	S upport V ector M achine
TAM	T emporal A ggregation M odule
TN	T rue N egative
TP	T rue P ositive
TPN	T ubelet P roposal N etwork
TRL	T ransformation R esidual L oss
WSDDN	W eakly S upervised D eep D etection N etworks
WSOD	W eakly S upervised O bject D etection
YTO	Y ou T ube- O bjects

Chapter 1

Introduction

In this chapter, the background and challenges in video object detection are firstly presented in Sec 1.1. The objectives of this thesis and our contributions are then discussed in Sec 1.2. This outline of the thesis is provided in Sec 1.3.

1.1 Background and Challenge

With the remarkable progress of convolutional neural networks (CNN), many object detection approaches, which are proposed for static images, show significant performance on some benchmarks, like PASCAL VOC and COCO. The objective of these methods is to utilize the visual information in images to improve the feature representation, accurately locating and recognizing objects. However, suffering from deteriorated appearances of objects (see Fig. 1.1), their performances are degenerated when directly applying them in video object detection. From Fig. 1.1, two videos are illustrated, where the first and the second rows indicate a fox and a cat in a video, respectively. In the first row, the fox is not displayed clearly due to motion blur. Especially for the last image, only partial body of this fox is shown, which poses a challenge to detect this object. For the second row, various viewpoints of a cat are illustrated. Although this cat is shown clearly, some extreme viewpoints like the side view, affect the detection performance.

To alleviate this issue, many approaches are proposed to leverage temporal information and enhance the feature representation at the current frame. An intuitive

FIGURE 1.1: The illustration of deteriorated appearances of objects in videos. In the first row, due to motion blur, the fox is not displayed clearly. Especially for the last image, only part body of this fox is shown. For the second row, various viewpoints of a cat are shown. Affected by some extreme viewpoints, the detection accuracy may be degraded.

method is to apply an off-the-shelf detector, which is originally proposed for static images, to each frame and use a tracking method to link the detected bounding boxes across frames. According to the distribution of the linked detection boxes, the detection results are re-scored to improve detection performance. Although these box-level approaches show better performance than those static image based methods, the performances in these box-level methods are not promising. For further improvement, some methods are proposed to capture temporal information in feature level.

Feature displacement usually exists across frames. To aggregate temporal information in feature level, it is important to accurately align features across frames. Optical flow is very popular in the computer vision field. It is used to express pixel displacement and is usually applied in many video based tasks to provide motion information. However, pixel displacement is not equal to feature displacement. To accurately align feature maps across frames, people follow a common method based on transfer learning. They firstly pre-train a neural network to predict optical flow. After that, they fine-tune this pre-trained network on the task dataset by jointly training it with the object detection network. By employing this method, they wish to enable the fine-tuned network to predict feature displacement. However, domain gap exists between optical flow training dataset and the task dataset. Although they fine-tune the pre-trained network on the target dataset, these optical flow based methods cannot accurately align feature maps across frames. This hinders their detection performances. Furthermore, the pre-training progress requires an

additional dataset with pixel-level annotations for optical flow, which makes their training process complex and time-consuming.

Apart from it, existing approaches in video object detection are proposed under full supervision. To train a neural network to capture spatial-temporal features, these methods require bounding box-level annotations for each frame in a video. In general, a video consists of hundreds of frames and a video dataset includes hundreds of videos, which results that constructing the box-level annotations for such a large-scale dataset is labor-intensive.

1.2 Motivation, Objectives and Contributions

According to the challenges presented in Sec 1.1, we propose three novel methods to address these issues. For the feature alignment, existing methods use optical flow based or tracking based methods to explore temporal information in videos. However, the performances of these approaches are hindered by some factors. In Chapter 3, we propose a new network (IFF-Net), which mitigates the limitations in optical flow based methods and improves the detection accuracy.

In Chapter 3, we firstly analysis the drawbacks in those common approaches (pre-train and fine-tune). To alleviate these drawbacks, we propose a new video object detection framework with an **In-network Feature Flow** estimation module (IFF). Different from the general training pipeline, our proposed IFF-Net does not require any additional dataset. By sharing the network backbone with the detection branches, our IFF module is only compared of several layers, which can directly produce feature flow with nearly free cost. Besides, inspired by *self-supervision*, a transformation residual loss (TRL) is proposed to further improves the performance of our IFF-Net. Extensive experiments are carried out on the ImageNet VID dataset, which demonstrates that our IFF-Net shows a comparable performance to state-of-the-art methods, while maintains a fast speed.

Apart from it, box-level annotations in video object detection dataset requires manually labeling, which is time-consuming. To prevent video object detection methods from the labor-intensive annotations, we propose a weakly supervised object detection methods for videos in Chapter 4. Since weak supervision only

requires image-level annotations, much labor cost is saved. Different from existing image based methods in weak supervision, our proposed weakly supervised detection method is proposed, exploiting motion information and improving the detection accuracy.

Motion patterns of objects are complex. It is difficult to directly use motion information to refine object regions. To solve this problem, we propose a **Motion Context Network** (MC-Net) in Chapter 4. Our MC-Net captures motion context features according to neighborhood motion correlation on moving regions. By combining motion context features with image information, the detection performance is improved. Experiments are conducted on the ImageNet VID dataset, which shows that our MC-Net significantly improves the performance of the image based baseline method.

Furthermore, existing neural networks in weakly supervised object detection are trained in a fixed proposal sets. This fixed proposal set is generally produced based on low-level features and are dominated by negative samples. Directly training on this set, the training network may be biased towards negative samples, affecting its detection accuracy. Without box-level annotations, it is difficult for weakly supervised methods to filter out the redundant negative samples. To further improve the detection accuracy in weak supervision, we propose a novel algorithm to solve this issue in Chapter 5.

We propose an **Online Active Proposal Set Generation** (OPG) algorithm in Chapter 5. Different from the fixed proposal set, our OPG algorithm is able to online generate an **active** proposal set based on the detection results of the training network. During the training progress, our OPG algorithm gradually removes the redundant negative samples and improves the training network performance. Through experiments, our OPG algorithm significantly improves the performance of the baseline.

1.3 Outline of the Thesis

Chapter 1 introduces the background of video object detection and presents the current challenges by briefly analyzing the limitations of existing methods in video

object detection. Following it, the objectives and contributions in this thesis are briefly presented.

Chapter 2 gives a systematical review on the related works including the static image based object detection methods, video object detection methods, and weakly supervised object detection methods. Some drawbacks in these related works are also discussed, which shows the motivation behind our research work.

Chapter 3 proposes a video object detection network (IFF-Net) with an In-network Feature Flow estimation module (IFF). Two kinds of IFF module structures are presented. IFF module does not require any additional training dataset. By sharing the network backbone with the detection branches, IFF module is able to directly predict feature flow in a fast speed. Moreover, a transformation residual loss (TRL) is proposed in a self-supervision manner, further improving the detection accuracy.

Chapter 4 presents a video object detection network in weakly supervised setting. Only image-level annotations are required and much labor cost for box-level annotations is saved. According to motion information in videos, a **Motion Context Network** (MC-Net) is proposed. It utilizes the motion context feature, which is generated from neighborhood motion correlation on moving regions, to indicate the potential object regions. By fusing the motion context feature with appearance information, the detection performance of the baseline is effectively improved.

Chapter 5 solves the problem arose by the fixed training proposal set. The training proposal set in existing weakly supervised detection methods are dominated by negative samples. To verify whether this imbalance between positive and negative samples affecting performance, a series of analysis experiments are firstly presented to show the effect of the overwhelming negative samples on the detection accuracy. After that, an **Online Active Proposal Set Generation** (OPG) algorithm is proposed to filter out redundant negative samples by generating an active proposal set. Experimental results show that the proposed OPG algorithm significantly improves the performance of the baseline.

Chapter 6 summarizes this thesis and discusses the future work.

Chapter 2

Literature Review

In this chapter, we give an overview on the related fields to our research. Our work is related to static image based object detection, video object detection, and weakly supervised object detection. To present our research comprehensively, we present the recent developments in these three fields and discuss their relations to our work.

From the aspect of application, object detection can be roughly divided into two categories: general object detection and detection application [2]. General object detection refers to detect some pre-defined classes of objects with a detection method. Detection application indicates the some detection methods proposed for special objects like face, pedestrian and text. In this chapter, we focus on presenting detection methods in general object detection.

In Sec 2.1, we review the static image based object detection methods. Our thesis work focuses on deep learning object detection. Traditional detection methods are briefly presented and the deep learning based detection methods are exhaustively discussed. In Sec 2.2, the video object detection approaches are presented. Different video object detection methods are categorized according to the methods of exploring temporal information. The limitations for each category are also discussed. In Sec 2.3, weakly supervised object detection methods are also reviewed. In Sec 2.4, some related benchmarks and evaluation metrics are introduced.

2.1 Static Image based Object Detection

As a basic research field in computer vision, object detection has been studied for tens of years and static image based object detection is mainly researched. Previously, static image based object detection methods aim to design some handcrafted features in a sophisticated way to effectively represent objects. Limited by the effectiveness of the handcrafted feature representation, the performances in these traditional methods are unsatisfactory. Recently, benefiting from the deep learning technique, static image based object detection witnesses a remarkable progress and shows significant performances in some benchmarks, like PASCAL VOC [3] and COCO [4]. Since this thesis focuses on deep learning based object detection, we briefly introduce the traditional object detection methods and comprehensively present the object detection methods based on the deep learning technique.

2.1.1 Traditional Methods

Static image based object detection methods can be traced back to the "recognition-by-components" theory [5]. Based on this theory, some early methods are proposed following a similarity measurement method. These methods frame the object detection as a rule-based or template based framework [6, 7]. However, these similarity based methods only work on some simple scenarios and fail to extent to some complicated detection tasks.

To solve this problem, some researchers try to propose detection methods according to machine learning. Eigenfaces [8], as the first learning wave based method, provides better object description via machine learning. After that, wavelet feature transformation [9–11] becomes flourishing, which transfers an image from pixels to a set of wavelet coefficients.

Among these methods, Viola Jones detectors (VJ detectors) [9, 10] show satisfactory performances, which remaining a fast inference speed. This method is proposed by sliding a window on all possible locations and scales in an image. To speed up the detection process, an integral image technique is applied to compute the haar feature. It also adopts the Adaboost algorithm [12] to find the features, which are effective in face recognition. Some features selected by the Adaboost

algorithm are illustrated in Figure. 2.1 (Figure Source: Viola et al. [10]). A cascade paradigm is proposed in VJ detectors to reduce much computation cost on background windows. By using these techniques, VJ detectors achieve a real-time face detection for the first time.

FIGURE 2.1: Illustration of features selected by the Adaboost algorithm in VJ detectors.

After that, Histogram of Oriented Gradients (HOG) feature [13] is proposed, which significantly improves the feature representation from two aspects of scale-invariant and shape context. HOG feature is computed based on some spatial regions (called cells) and uses the contrast normalization on the local responses for better performance. HOG feature has served as an important foundation in many object detection methods [14–16] for a long time. As an extension of HOG feature based detectors, Deformable Part-based Model (DPM) [14] was originally proposed in 2008. It formulates the object detection task as an ensemble detection on different parts of an object and consists of a root-filter and some part-filters. By decomposing an object, DPM and its variants [15, 17, 18] show promising results among traditional object detection methods.

Although current deep learning based methods have surpassed these traditional methods, the core thoughts behind traditional methods have a great influence on the current deep learning based methods. These classic object detection methods still inspire researchers to propose excellent object detection approaches.

2.1.2 Deep Learning Based Detection Methods

Deep learning based object detection methods can be traced back to convolution neural network (CNN) based methods [19, 20] published in 1990s. However, limited by the computation resources at that time, these CNN based approaches do not show impressive results. Until 2010s, the performances of traditional methods gradually become saturate. Benefiting from the progress of GPU, the deep learning based object detection methods begin to rebirth. Different from traditional methods, deep learning based approaches do not resort manually designing complex features. By using some deep CNNs [21–25], they are able to generate robust and high-level features to represent objects. Based on these high-level features, the feature representation is significantly improved and the better performance is obtained.

Deep learning for computer vision. Deep learning approaches are proposed to solve problems in the artificial intelligence field. These methods try to train a multi-layer model to learn knowledge from the work with a hierarchical construction. This model generally consists of many layers. Among various multi-layer models, the convolution neural network (CNN), which is firstly proposed in [26], shows excellent feature capacity in the computer vision field and is widely applied in many computer vision tasks, like image classification, object detection and segmentation. LeNet-5 [20] proposed in the 1990s successes in recognizing handwritten zip code. Its architecture is illustrated in Figure. 2.2(Figure Source: Lecun et al. [20]). In LeNet-5, there are two convolutional layers, two pooling layers and three fully-connected layers. However, its performance is still inferior to support vector machine (SVM) based methods and boosting based methods. After that, AlexNet [27] is proposed and performs the best in the ImageNet ILSVRC2012 image classification competition. This network draws many researcher’s attention and greatly promotes the progress of CNN.

With the rapid development of CNNs, deep learning based methods show significant performance in many computer vision challenges. Their performances are increasingly improved. VGG16 [21] shows that the performance of a CNN can be improved by increasing the depth of a neural network. GoogLeNet [22] proposes the Inception module to further improve the CNN ability by stacking these Inception modules. Following it, three Inception modules: Inception v2 [24], Inception v3 [24], and Inception v4 [28] are proposed for better performance.

FIGURE 2.2: The architecture of LeNet-5

Although these deep CNNs significantly improves the performance of computer vision methods, these methods suffer from gradient vanishing and gradient exploding problems with increasing the depth of CNNs. To tackle this problem, the Residual network [23] (ResNet) is proposed by adopting a shortcut connection across different layers. ResNet-34 also wins the champion in the ILSVRC 2015 image classification competition. Based on ResNet, ResNet v2 [29] and ResNeXt [25] are proposed to further increase the network feature capacity.

Deep CNNs requires huge GPU memory, which makes it difficult to apply these neural networks to some embedded systems and mobile devices. To address this issue, some lightweight neural networks [30–36] are proposed. MobileNet series [30–32] are proposed by Google. The depth wise and point wise convolution layers are applied in MobileNet v1. Based on MobileNet v1, MobileNet v2 further improves its performance by introducing the inverted residual block and the linear bottleneck module. In MobileNet v3, it uses the network searching technique and lightweight attention model to improve the accuracy. ShuffleNet v1 [33] and ShuffleNet v2 [34] are proposed to reduce the computation cost by adopting point wise convolution and channel shuffle. GhostNet [35] divides the traditional convolution layer into two parts: less convolution kernels and some linear transformations. EfficientNet [36] develops a new scaling method to adjust all dimensions of neural networks. It also uses the neural architecture search technique to design a new network, remaining high accuracy and efficiency.

Deep learning based object detection methods. Object detection methods with deep learning fall into three categories: one-stage methods, two-stage detection methods and other improved techniques. Two-stage methods decompose object detection into two stages: proposal generation and object detection. Two-stage methods firstly utilize some algorithms to produce the proposals, which may

include objects. After that, they apply a detector on these proposals to detect objects. Although these two-stage methods are complex, they show excellent detection accuracy. Among them, R-CNN [37] is firstly proposed to use a CNN to extract feature vectors and its pipeline is illustrated in Fig. 2.3, where the main components of R-CNN are indicated by yellow rectangles. In Fig. 2.3, R-CNN firstly uses the selective search algorithm [38] to generate proposals. For each proposal, the corresponding feature vector is extracted by a CNN and forwarded to a support vector machine (SVM) for image classification. Considering an image includes thousands of proposals, the detection speed in R-CNN is extremely slow. Although R-CNN shows an excellent performance compared with traditional detection methods, the training and testing process are redundant, which also requires huge disk space to store those feature vectors.

FIGURE 2.3: The illustration of frameworks in R-CNN, SPP-Net, Fast R-CNN, and Faster R-CNN. The first branch indicates the R-CNN framework. SPP-Net and Fast R-CNN are similar in their detection frameworks. We summary them and illustrate their frameworks at the second branch. Faster R-CNN is illustrated in the third branch.

To alleviate these issues, SPP-Net [39], Fast R-CNN [40] and Faster R-CNN [41] are developed. To clearly show their improvements based on R-CNN, we illustrate their frameworks and emphasis their main contributions in Fig. 2.3. Instead of forwarding each proposal to a CNN, SPP-Net introduces a spatial pyramid pooling (SPP) layer which only needs to forward an image to a CNN once and is able to output corresponding feature vectors for proposals. Its difference from R-CNN is illustrated with blue rectangles in Fig. 2.3. After employing the SPP layer,

SPP-Net is more than 20 times faster than R-CNN, while shows a better detection accuracy. However, the training process in SPP-Net is multi-stage and it only fine-tunes its fully connected layers, which hinders its performance. Based on SPP-Net, Fast R-CNN is proposed for further improvement. Compared with SPP-Net, Fast R-CNN proposes a ROI pooling layer to replace the SPP layer in SPP-Net and utilizes a multi-task loss to train all layers in the neural network. Benefited with this single stage training process, Fast R-CNN yields a better performance than SPP-Net. Since the pipeline of Fast R-CNN is similar to that of SPP-Net, we summary their frameworks and illustrate them together.

SPP-Net and Fast R-CNN significantly improve the detection speed, but the speed in deep learning based detection methods is still slow. This poses a challenge to realize real-time detection with the deep learning based methods. To further improve detection speed, Faster R-CNN is proposed to reduce the time cost on proposal generation module, which directly produces proposals based on a CNN without resorting to any other algorithm. Its framework is illustrated in Fig. 2.3, where the main contribution is indicated in an orange rectangle. A region proposal network (RPN) is proposed in Faster R-CNN, which shares the network backbone with the detection head and is able to produce proposals with nearly free cost.

FIGURE 2.4: The illustration of anchors. The sub-region is projected to its corresponding region on the original image and is reshaped according to these anchor shapes.

Additionally, Faster R-CNN proposes an anchor based detection framework shown in Fig. 2.4, which pre-defines several types of anchors with different scales and aspect ratios. For each sub-region on a feature map, it projects this sub-region to

its corresponding region on the original image and reshapes this region according to these anchor shapes. Some classifiers and regressors are trained to produce the results for these anchor based regions. This anchor based detection method effectively improves the detection accuracy and is also used by some one-stage detection methods, like SSD and RetinaNet. Faster R-CNN is the first detection method, which is end-to-end trained. It obtains a better performance than previous methods with nearly real-time detection speed. After that, a ROI Align layer [42] is proposed to replace the ROI pooling layer. The ROI Align layer provides better feature representation than that from the ROI pooling layer and improves the detection performance.

By sharing the network backbone with the detection heads, RPN in Faster R-CNN improves the detection speed. However, affected from the inherent classification networks, the fully connected layers, which serve as a classifier or a regressor, are costly used thousands of times per image to predict results for all proposals. To further improve the detection speed, R-FCN [43] is proposed to replace these fully connected layers with convolutional layers. With this improvement, nearly all computation is shared by proposals and the detection speed in R-FCN is 2.5-20x faster than Faster R-CNN.

FIGURE 2.5: The framework of R-FCN. R-FCN replaces the fully connected layers with convolutional layers. The feature map from the CNN network backbone is passed to a convolution layer with $k^2 * (C + 1)$ kernels to produce a position sensitive score map, where different color feature maps denote the features in differently related locations. The prediction result is computed by average pooling on these corresponding feature responses.

The framework of R-FCN is illustrated in Fig. 2.5. Different from R-CNN variants, R-FCN applies convolutional layers and the average pooling to predict the detection results. R-FCN divides the corresponding region in each proposal into $k \times k$ bins. It forwards the feature map from the network backbone into the last convolutional

layer to produce a bank of position sensitive maps with $k^2(C + 1)$ channels, where C denotes the number of classes. As shown in the Fig. 2.5, different color describes different relative position feature maps. In the case of the Fig. 2.5, with $k = 9$, there are 9 bins for each proposal, indicating the object sub-regions: top-left, top-center, top-right, ..., bottom-right. The output of each bin is computed via average pooling on the corresponding sub-region. The final prediction is obtained by average pooling on these $k \times k$ bins. By using this fully convolutional network, R-FCN increases the detection speed and remains a promising detection accuracy.

Compared with two-stage methods, one-stage methods detect objects only in one-step. These approaches are able to detect objects in a fast speed, while sacrificing the detection accuracy. Among deep learning based methods, OverFeat [44] is firstly proposed to use feature map from CNN to detect objects in a sliding window manner. After that, YOLO [45] and SSD [46] are proposed for further improving one-stage detection methods. They are illustrated in Fig. 2.6. YOLO is short for *You Only Look Once*, which is able to detect objects in a extremely fast speed, 155 fps. As shown in Fig. 2.6, YOLO unifies the detection framework into a single neural network. It divides a full image into a grid of sub-regions and predicts detection results for each sub-region. Although its detection speed is satisfactory, its performance is limited by its pre-defined grid and it fails to detect small objects.

FIGURE 2.6: The illustration of YOLO and SSD. The upper row describes YOLO detection framework, which directly predicts detection results in one-stage. The bottom row indicates SSD detection framework. SSD produces multi-scale feature maps in a progressive way and detects objects on these feature maps. Benefiting from multi-scale features, SSD shows a better detection performance than YOLO.

To alleviate this issue, Single Shot multibox Detector (SSD) proposes a multi-scale feature based detection framework illustrated in Fig. 2.6. Compared with YOLO, SSD progressively generates feature maps in different resolutions and applies its detector on each feature maps to provide final results. With this multi-scale feature map, SSD effectively improves the detection accuracy in a fast speed, 59 fps. Two YOLO variants [47, 48] are also proposed to improve the detection accuracy in one-stage.

Although SSD significantly improves the detection accuracy, performances in one-stage methods are still much lower than two-stage methods. To push the envelop further, Focal Loss [49] proposes a detection network called RetinaNet, which reshapes the loss function in one-stage methods and focuses training on these hard samples, yielding a comparable performance to two-stage methods'. Recently, EfficientDet [50] is proposed to improve the efficiency among the one-stage detection methods. EfficientDet uses the EfficientNet [36] as the backbone network. Motivated by FPN [51], a BiFPN is proposed to solve the multi-scale feature fusion problem. EfficientDet shows a better performance while maintaining a faster reference speed than RetinaNet.

Anchor based detection framework is firstly proposed in Faster R-CNN [41] and is applied in both one-stage and two-stage detection methods. Benefiting from pre-defined anchors, these anchor based methods show excellent performances. However, these pre-defined anchors are a little heuristics and depend on the objects to be detected. To obtain a promising performance, anchor based methods usually produce a larger number of proposals, which are densely overlapped each other. This densely overlapped proposals increase the computation in both training and testing. To overcome these drawbacks, some one-stage detection methods without anchor (called anchor free methods) are recently proposed.

CornerNet [52] is proposed to detect objects by locating a pair of corners of objects. In deep learning based methods, it is the first detection method, which does not utilize anchors and achieves a comparable performance to anchor based methods. Hourglass [53] is used as its network backbone for feature extraction. After that, several detection methods [54–58] are proposed to improve detection performances in an anchor free manner. The method in [54] improves CornerNet [52] and detects objects in triplets of points. The approach in [55] models an object as a center point of its corresponding bounding box. In FSAF [56], a feature selective anchor-free

module is proposed to solve the drawbacks in anchor-based methods. RepPoints [57] aims to locate an object with a set of points. FCOS [58] proposes to detect objects in a fully convolutional network, unifying object detection and semantic segmentation in a single network.

To improve the detection accuracy, some other improved techniques are proposed. Feature pyramids are popular in traditional detection methods to capture objects in different sizes. However, in deep learning based methods, low-level feature map in large scale focuses on discriminate information, while high-level feature map in small scale includes semantic information. Although SSD utilizes multi-scale features to improve its performance, the difference between different scale features hinders its detection accuracy. To remedy this problem, Feature Pyramid Network (FPN) [51] is proposed to design a top-down fusion method, which effectively combines multi-scale features and provides a strong feature representation for object detection. To discuss FPN clearly, we illustrate the multi-scale feature hierarchy in SSD at the left column in Fig. 2.7 and show the architecture of FPN at the right column of the Fig. 2.7.

FIGURE 2.7: Feature pyramid comparison between SSD and FPN. SSD simply predicts on different scale feature maps, while FPN effectively combines multi-scale features via lateral connection and detects objects on these fused feature maps.

From the Fig. 2.7, it can be seen that SSD simply detects objects on different scale feature maps. Affected by different properties in multi-scale feature maps, the performance of SSD is limited. In contrast, FPN fuses different feature maps via a lateral connection. Benefiting from this fusion method, the discriminate information is effectively integrated into the semantic information, which significantly improves the detection accuracy. FPN block is firstly applied to a two-stage method, Faster R-CNN and shows its effectiveness. After that, it is also used in RetinaNet and yields a promising performance.

Apart from it, DeepID-Net [59], Deformable ConvNet v1 [60] and v2 [61] are proposed to capture the geometric variations of objects. Libra R-CNN [62] is proposed for balance learning in two-stage object detection methods. It aims to solve the imbalance training from three aspects: proposal sampling, feature fusion and optimization loss. GBD-Net [63] is proposed to utilize context information to improve detection accuracy. Relation Network [64] is proposed to accurately detect objects according to the relationships between objects. OHEM [65] is designed for two-stage detection methods to improve detection performance by focus training on hard examples. SNIP [66] solves the scale variant problem in object detection and improves the detection accuracy. Cascade R-CNN [67] designs a multi-stage detection pipeline to accurately detect objects. Bounding Box Voting in [68] is proposed as a post-processing method to improve the detection performance according to the overlapped bounding boxes. Soft-NMS [69] is designed to alleviate limitations of the traditional NMS algorithm and is widely used in the COCO detection competition.

Overall, the deep learning based object detection methods can be outlined in Fig. 2.8. In Fig. 2.8, some detection methods under each category are also listed.

FIGURE 2.8: The outline of deep learning based detection methods.

2.2 Video Object Detection

Static image based object detection focuses on using image information to detect objects. Different from it, video object detection aims to explore temporal information to improve detection performance. Video object detection methods are composed of two parts: temporal feature aggregation and object detection. From the aspect of temporal feature aggregation, approaches in video object detection fall into two categories: box-level and feature level feature aggregation methods.

2.2.1 Box-level Methods

For box-level methods, they firstly employ the static image based detector on frames and store these detection results. After that, various methods are proposed to link these detected boxes and re-score these boxes according to the distribution of linked boxes. In [70], it utilizes a tracking method to bidirectionally link the boxes, which are detected by a static image based detector, generating some tubelets. Then, a temporal convolutional network is proposed to refine the detection results. Based on [70], T-CNN is proposed to optimize this box-level method for better performance.

In [71], it divides the temporal re-scoring step into two parts: short and long range. This method firstly refines the detection results in a short segment in a video. Then, it further improves the detection accuracy by utilizing long segment information. With this method, the detection accuracy is improved.

Seq-NMS [72] is another box-level method, which is widely used in many video object detection methods. It is composed of three steps: sequence selection, sequence re-scoring, and suppression. In sequence selection, detected boxes across frames are linked, if the IOU between these two boxes is larger than 0.5. These linked boxes form into a sequence. Then, it tries to find the sequence with the maximum sum of box scores. After selecting the sequence, the sequence re-scoring is applied to this sequence. Two re-scoring function are used: average and max. In the third stage, other boxes, which are overlapped with those linked boxes larger than a threshold value, are removed. These three steps are repeated until no boxes left. Seq-NMS is able to generalize well on many video object detection methods and show effective improvement. In summary, box-level methods generally consist of three phases:

applying static image based detectors on frames and linking detection results and re-scoring results. They are illustrated in Fig. 2.9.

FIGURE 2.9: The illustration of box-level video object detection methods.

2.2.2 Feature Level Methods

Although box-level methods effectively improves the detection accuracy in videos, their performances are not satisfactory. To further improve detection performance in videos, many approaches [73–81] are proposed to explore temporal information in feature level. These methods are composed of two components: temporal feature aggregation and object detection. For object detection, they directly use Faster R-CNN or R-FCN as their detectors.

For feature aggregation, they can be roughly divided into two categories: proposal-level and whole feature map level feature aggregation. In proposal-level aggregation, methods in [73, 82] train a network to predict the box offsets. In [82], it is composed of two parts: a Tubelet Proposal Network (TPN) and a CNN-LSTM network. It firstly applies the static image based detector on frames and uses TPN on the features within proposals to predict object motion. According to object motion states, it aligns these proposals across frames and utilizes the CNN-LSTM network to aggregate temporal information, predicting the detection results. Compared with [82], the method in [73] is more simple and effective. It uses a correlation layer to predict the proposal motion states and applies a post-process method to improve the detection results.

Approaches in [80, 81] fuse the features in the related proposals based on self-attention or similarity mechanism. A static image based detector is applied to predict results on each frames and features within these proposals are used to compute the similarities between each other. After that, the related proposals are fused together to enhance the proposal feature at the current frame. These similarity based fusion method does not occupy much memory and is able to aggregate

long-range temporal information. However, limited by the proposal regions, they cannot utilize the context region to improve the detection accuracy, which hinders their performances.

FIGURE 2.10: The illustration of optical flow based video object detection methods. Raw images are passed into two deep neural networks: a convolutional neural network (CNN) and FlowNet. The CNN serves as a feature extractor and FlowNet is used to produce optical flow for feature alignment. The adaption module is used to improve the performance of feature alignment.

Compared with proposal-level feature aggregation, some methods [74–76, 78, 83] are proposed to aggregate temporal features on the whole feature map level. Methods in [74, 75, 83] are proposed based on optical flow and are illustrated in Fig. 2.10. These optical flow based methods forward raw images to two deep neural networks: FlowNet [84] and a convolutional neural network (CNN). A CNN is used as a network backbone to extract feature map for each frame. For effective temporal feature aggregation, FlowNet is used to produce optical flow, which provides motion information for feature alignment. Optical flow based methods firstly align features across frames according to optical flow and then, fuse these aligned features according to the feature similarities. This aligned feature fusion is defined as follows:

$$\tilde{X}_i = \sum_{t=-\tau}^{\tau} \mathbf{W}_t \cdot X_t, \quad (2.1)$$

where X_t denotes the feature map at the neighboring frame t , τ expresses the maximum temporal offset in feature aggregation, $\mathbf{W}_t \in R^{w \times h}$ is the weight matrix, and $\tilde{X}_i$ is the fused feature map at the current frame i . To compute the weight matrix $\mathbf{W}_t$, the similarity $\overline{W}_t(x, y)$ at (x, y) are firstly computed as:

$$\overline{W}_t(x, y) = \frac{X_{t \rightarrow i}(x, y) \cdot X_i(x, y)}{|X_{t \rightarrow i}(x, y)| \cdot |X_i(x, y)|}, \quad (2.2)$$

where $X_{t \rightarrow i}$ is the aligned feature map. Then, $\bar{W}_t$ is normalized along the temporal axis according to (2.3) as follows.

$$W_t(x, y) = \frac{\bar{W}_t(x, y)}{\sum_{k=-\tau}^{\tau} \bar{W}_k(x, y)}, \quad (2.3)$$

With this method, they enhance the feature map at the current frame and improve the detection accuracy.

However, optical flow describes pixel displacement, which is different from feature displacement. To accurately align feature map across frames, two strategies are employed in these optical flow based methods. At first, they pre-train a neural network (FlowNet [84]) to predict optical flow and fine-tune this pre-trained network on the task dataset. With this methods, they wish the output of this fine-tuned FlowNet is able to express feature displacement, instead of pixel displacement.

Furthermore, they add several extra layers as an adaptation module to improve their optical flow based feature alignment. In DFF [83], it propose a *scale function* to modify optical flow values. In FGFA [74], an *embedded network* is adopt in feature fusion module. In [75], one additional layers are utilized for feature fusion. Although these optical flow based methods show good performances, the FlowNet used in these methods is computation expensive and occupies much GPU memory.

FIGURE 2.11: The illustration of MatchTrans module in STMN.

To solve these issues, several methods [76, 78] are proposed. Instead of optical flow, STMN [76] proposes a MatchTrans module for feature alignment and applies a LSTM for temporal feature fusion. To present the MatchTrans clearly, we illustrate

it in Fig. 2.11. Given a feature vector $X_i(x, y)$ at (x, y) in the current frame i , MatchTrans computes the similarities between $X_i(x, y)$ and feature vectors around (x, y) in X_t . It is defined in 2.4.

$$W_{x,y}(i, j) = \frac{X_i(x, y) \cdot X_t(x + i, y + j)}{\sum_{i,j \in -k, \dots, k} X_i(x, y) \cdot X_t(x + i, y + j)}, \quad (2.4)$$

where k controls the size of the neighboring region. After that, STMN computes the aligned feature map according to 2.5. Then, STMN uses a LSTM to aggregate these aligned feature map, improving the quality of the feature map at the current frame.

$$X_{t \rightarrow i}(x, y) = \sum_{i,j \in -k, \dots, k} W_{x,y}(i, j) \cdot X_t(x + i, y + j) \quad (2.5)$$

Following STMN, the method in [78] proposes some improvements on feature alignment and feature aggregation. In feature alignment, the method in [78] proposes a Progressive Sparse Local Attention module (PSLA) according to the MatchTrans module. To discuss PSLA clearly, we compare PSLA with MatchTrans in Fig. 2.12, where MatchTrans is illustrated on the left side and PSLA is illustrated on the right side. From the Fig. 2.12, it can be seen that PSLA matches the feature vectors in a progressive sparser stride manner, which increases the match field and saves the computation. Benefiting from this modification, PSLA is able to align feature map better than MatchTrans.

FIGURE 2.12: Comparison between MatchTrans and PSLA. MatchTrans and PSLA are shown at the left and right side, respectively. Compared with MatchTrans, PSLA compares the feature similarity in a progressive sparser stride manner. This method is able to increase the match field, while saves much computation.

In feature aggregation, the method in [78] divides frames in videos into key frames and non-key frames. For key frames, it proposes a module called Recursive Feature Updating (RFU) to temporally aggregate features from key frames. For non-key frames, it proposes a module called Densely Feature Transforming (DFT) to fuse the aggregated features of key frames and the features at the current frame together. By using this scheme, this method is able to aggregate temporal features in a fast speed.

Methods in [76, 78] propose feature alignment based on feature similarities and aggregate temporal feature in a fast speed. However, due to their simple feature alignment scheme, features across frames are not aligned accurately, which hinders their detection accuracy in videos.

Although many video object detection methods are proposed for performance improvement, the overhead computation in existing methods limits their applications on mobile devices. To alleviate this issue, some video object detection methods for mobile devices are proposed. The approach in [85] proposes a Bottleneck-LSTM to efficiently leverage temporal information and is able to detect objects at 15 fps on a mobile CPU. The method in [86] is designed based on FGFA [74]. According to the FGFA framework, it re-trains a network backbone with light computation for feature extraction and designs a Light Flow based on MobileNet [30] to replace FlowNet, predicting optical flow for feature alignment.

Overall, we sum up approaches in video object detection and illustrate them in Fig. 2.13. Box-level methods target to explore temporal information on box-level. These methods often apply static image based detector on each frame and link the related detection results together. They re-score the detection results according to the distribution of these linked detection results. Compare with them, feature level methods target to enable detection networks to capture spatial-temporal features. They are roughly divided into two types: proposal-level and whole feature map level. The former type fuses the related proposal features together and detects objects on these fuses proposal features. The second type combines the whole feature maps across frames and detects objects on these fused feature map.

Approaches in proposal-level feature aggregation are able to leverage long-range information. However, they only use features in proposal region, which results that they cannot utilize context information to improve the detection accuracy.

FIGURE 2.13: Summary of video object detection methods.

For existing methods in aggregating the whole feature map level, they cannot align feature map accurately due to their inherent drawbacks as discussed above. To mitigate these issues, we propose a novel network (IFF-Net) with an In-network Feature Flow estimation module (IFF). We present our work on IFF-Net in Sec. 3.

2.3 Weakly Supervised Object Detection

With the development of Convolutional Neural Network (CNN), object detection shows significant performance in some remarkable benchmarks, like PASCAL VOC [3], COCO [4], and ImageNet VID [87]. However, most existing detection methods including static image based detection methods [37, 40, 41, 43] and video object detection methods [74, 75, 80, 81] are proposed under fully supervised training.

These methods are realized based on large-scale datasets [3, 4, 87] with labor-intensive annotations. Especially for video object detection datasets, each frame in a video requires bounding box-level annotations, which results that constructing such a huge dataset is time-consuming. To alleviate this problem, many approaches [88–91] are proposed under weakly supervised training, which only requires image-level annotations. These methods are called weakly supervised object detection methods. Since image-level annotations only indicate which class is included in an image, they are easily obtained by simple web search on image search engines. Weakly supervised object detection methods prevent object detection from human labeling.

In weakly supervised object detection methods, researchers usually formulate this task as a Multiple Instance Learning (MIL) problem, where each image is regarded as a bag and image regions produced by selective search [38] or edge boxes [92] are treated as instances. A detector is trained to iteratively predict instance scores for each bag. A weighted sum based on these instance scores is used to indicate a image-level prediction. The difference between this prediction and the image-level annotation is used as the optimization objective.

Weakly supervised object detection has been studied for a long time. Before the deep learning era, many approaches [93–96] are proposed to solve this problem from the aspect of initialization strategies or regularizing optimization. As this thesis focuses on deep learning based methods, we focus on discussing deep learning based methods in weakly supervised object detection.

In weakly supervised object detection methods based on the deep learning technique, WSDDN [88] proposes a Convolutional Neural Network (CNN) based frameworks, which adopts a two stream construction and uses a spatial regulariser to further improve the performance. Its framework is illustrated in Fig. 2.14.

Weakly Supervised Deep Detection Networks (WSDDN) applies selective search or edge box to produce proposals and forwards the corresponding image to a CNN, generating feature maps. These proposals and the corresponding feature map are passed to a ROI pooling layer to predict proposal features. These features are further encoded by two consecutive fully connected (FC) layers. After that, these encoded features are connected with two streams: a classification stream and a detection stream.

FIGURE 2.14: Illustration of Weakly Supervised Deep Detection Networks. WSDDN passes images to a convolutional neural network (CNN) and the selective search algorithm to predict the feature map and proposals, respectively. These proposals and the predicted feature maps are forwarded into a ROI pooling layer to generate proposal features, which are then passed to two branches, predicting the final results.

In the classification stream, encoded features are passed to a FC layer called FC-8c and normalized by a *softmax* layer along the class axis: $\delta_c(\mathbf{x}^c)(i, j) = \frac{e^{x_{i,j}^c}}{\sum_{m=1}^C e^{x_{m,j}^c}}$, where $x_{i,j}^c$ denotes the output of FC-8c layer for the j th proposal at class i and C indicates the number of classes. This classification stream is to classify each region into C classes.

In the detection stream, encoded features are passed to a FC layer called FC-8d and normalized by another *softmax* layer along the proposal dimension: $\delta_d(\mathbf{x}^d)(i, j) = \frac{e^{x_{i,j}^d}}{\sum_{m=1}^{|R_r|} e^{x_{i,m}^d}}$, where $x_{i,j}^d$ denotes the output of FC-8d layer for the j th proposal at class i and $|R_r|$ is the number of proposals. This detection stream is proposed to score one region relative to others.

The final score for each proposal is computed by an element-wise production: $\mathbf{x}^R = \delta_c(\mathbf{x}^c) \odot \delta_d(\mathbf{x}^d)$. In the end, the proposal score metric at c th class are converted to a image score at c th class: $p_c = \sum_{k=1}^{|R_r|} x_{ck}^R$. WSDDN trains this network using a multi-class cross entropy loss L_{base} , as listed in Eq. 2.6.

$$L_{base} = -\frac{1}{|C|} \sum_{c=1}^C [y_c \ln p_c + (1 - y_c) \ln(1 - p_c)], \quad (2.6)$$

where y_c is 1 or 0, indicating whether the image includes class c and C is the number of classes.

Based on this framework, WSDDN add a spatial regulariser on Eq. 2.6 for further improvement. It computes the feature differences between the top-scoring region and other regions with IOU higher than 0.6. After employing this regularization, its performance is further improved.

According to WSDDN, ContextLocNet [97] and OICR [89] are developed for better performances. ContextLocNet [97] proposes context-aware models, which uses the context region to improve the detection accuracy. Two different models, an additive model and a contrastive model are designed in ContextLocNet [97].

As illustrated in Fig. 2.15, given a proposal from an image, ContextLocNet firstly extracts features in the corresponding proposal region and the context region of this proposal. After that, it encodes these two features into the same size feature vector and fuses them via two different ways: addition and contrast. The additive model is proposed to recognize objects according to the feature consistency between the proposal region and its context region. The contrastive model detects object based on the semantic difference between the proposal region and its context region. With these methods, ContextLocNet improves the detection accuracy under weak supervision.

FIGURE 2.15: Illustration for ContextLocNet. ContextLocNet extracts features on the proposal region and its corresponding context region. Features on these two regions are fused together by two different ways: addition or contrast, providing strong feature representation for object detection.

Different from ContextLocNet, OICR [89] improves the detection accuracy by proposing an online refinement algorithm according to spatial information, which is shown in Fig. 2.16. In the Fig. 2.16, the basic module is the same as the two branch part in WSDDN as shown in Fig. 2.14. For the PGT module, it selects the top scoring proposal as the pseudo ground truth (PGT). In the k th OICR part, it predicts proposal scores by forwarding proposal features from a ROI pooling layer to a FC layer and normalizes these proposal scores via a *softmax* layer along the class axis. The proposals, whose IOUs with pseudo ground truths are larger than 0.5, are assigned to the same labels as the pseudo ground truths’.

To train this neural network stable, the predicted scores for a PGT is used as a weight in the loss function. Given the pseudo ground truth with a scores w_{ik} for the i th proposal at the k th OICR branch, the loss function L_r^k is defined as:

$$L_i^k = -\frac{1}{|R|} \sum_{i=1}^R \sum_{c=1}^{C+1} w_{ik} y_{ik}^c \ln x_{ik}^c, \quad (2.7)$$

where y_{ik}^c is the class c label for the i th proposal at the k th OICR branch and x_{ik}^c is the predicted scores for i th proposal in class c at the k th OICR branch.

FIGURE 2.16: Illustration for OICR

In the Figure 2.16, OICR firstly generates the pseudo ground truths, PGT-1 according to the predictions of the basic branch and passes PGT-1 to OICR-1. These operations are repeated iteratively for the following two OICR branches. Finally, the predictions from three OICR branches are averaged to obtain the final results. The loss function for OICR detection framework is defined as:

$$L = L_{base} + \sum_{m=1}^M L_r^m, \quad (2.8)$$

where L_{base} is presented in 2.6 and M is 3. With this online refinement method, OICR significantly improves the detection accuracy and is widely used as the baseline in many recently published methods, like WSOD2 [98], C-WSL [99], and TS²C [100].

Apart from it, many approaches are proposed for weakly supervised object detection. ZNDN [91] attempts to solve weakly supervised object detection problem with a zigzag learning strategy. It develops a criterion to evaluate reliable samples for training a neural network. W2F [101] proposes an algorithm for generating instance-level pseudo ground truths in an iterative way. In MELM [90], an entropy metric is defined to indicate the randomness of object location. A min-entropy latent model based on this metric is proposed to mine potential regions including objects. To solve the non-convexity problem in weakly supervised loss function, C-MIL [102] proposes an optimization algorithm in a continuation way. The approach proposed in [103] proposes a method to label objects in instance-level using spatial constrain.

FIGURE 2.17: Illustration for the methods which jointly train weakly supervised segmentation and detection networks.

Recently, weakly supervised segmentation methods are widely applied in some weakly supervised detection methods [100, 104, 105], which are illustrated in Fig. 2.17. These methods jointly train segmentation and detection neural networks under weak supervision and utilize the predicted results from both networks as regularization. By simultaneously training both neural networks, their performances on weakly supervised detection and segmentation are improved.

Most existing weakly supervised object detection methods are proposed for images. Lacking of box-level annotations, the performances on these methods are limited. To obtain a better performance, W-RPN [106] proposes to leverage videos to improve the quality of proposals in weakly supervised detection methods. It uses an

external video dataset to train a region proposal network (RPN), which is applied to the image based object detection under weak supervision, removing the noisy proposals. However, the instance-level information is not encoded into its detector. Furthermore, W-RPN requires pre-training on an additional video dataset and fine-tuning on the task dataset. This training process is complex. Different from W-RPN, we propose a Motion Context network (MC-Net) in this thesis. Our MC-Net is able to directly capture the instance-level information from videos and encodes this information into the detector. Our MC-Net is end-to-end trained and does not require the complex training process. We present our MC-Net in Sec. 4.

2.4 Benchmarks and Evaluation Metrics

To evaluate the performances on proposed methods, many benchmarks are established and the corresponding evaluation metrics for specific tasks are defined. In this part, we present some datasets, which are widely used for evaluation in static image based object detection, video object detection and weakly supervised object detection. Two evaluation metrics used in object detection and weakly supervised object detection are also introduced.

2.4.1 Benchmarks

Static image based object detection aims to use image information to detect objects, while video object detection utilizes temporal information to improve detection accuracy. To evaluate performances of methods on these two fields, different datasets are established.

We firstly introduce the benchmarks used in static image based object detection methods and illustrate some images from these benchmarks in Fig. 2.18.

- *PASCAL VOC* [3]: PASCAL VOC is short for PASCAL Visual Object Classes, where PASCAL stands for pattern analysis, statistical modeling and computational learning. It provides a public dataset including images and ground truths. Additionally, it also holds annual competition and work shop from 2005 to 2012. Among PASCAL VOC datasets, PASCAL VOC 2007

and 2012 are widely used to evaluate the detection performance in static image based detection methods. In PASCAL VOC datasets, there are 20 class objects to be detected. In PASCAL VOC 2007, there are 9,962 images with 24,640 annotated objects in total. There are 5,011 images in trainval set. In PASCAL VOC 2012, more images are added. There are 11,530 images involving 27,450 annotated objects.

- *COCO* [4]: COCO dataset is also called MS COCO, which is short for Microsoft Common Objects in COntext. There are 91 categories with 328,000 images and 2,500,000 labeled instances. COCO dataset includes several common visual tasks, like object detection, object segmentation and keypoint detection. In object detection, it contains 123,000 images for training and 20,288 images for testing. The images in COCO include more complex scenes, which poses a new challenge to object detection.

FIGURE 2.18: Sample images in PASCAL VOC and COCO benchmarks.

Benchmarks, which are used in static image based object detection, are composed of images. In video object detection, to leverage temporal information, benchmarks in video object detection consist of videos. Each video includes tens or hundreds of frames. There are many benchmarks collected to evaluate detection performance in videos, like YTO [107] and ImageNet VID [108]. We give an introduction on the related benchmarks to our work and show several video segments from these datasets in Fig. 2.19 and Fig. 2.20.

- *YTO* [107]: YTO is short for YouTube-Objects. It is composed of videos, which are collected from YouTube. There are 10 classes and each class contains from 9 to 24 videos. Videos in YTO is weakly annotated. Annotations for each video only ensure including one object among these ten classes. Only few frames are annotated. YTO dataset is collected to study the domain gap between static image based object detection and video object detection. Since frames in this dataset are not fully annotated, it is difficult to use this dataset to accurately measure the detection performance in videos.
- *ImageNet VID* [108]: ImageNet VID dataset is used for the object detection from video competition in ImageNet. There are 30 basic-level categories. There are 1952 videos with full annotations in training set. In the training set, the number of frame in each video varies from 6 frames to 5213 frames. In the validation set, 281 videos with full annotations are involved. There are from 11 to 2898 frames in each video. For test set, there are 458 videos and the ground truths for test set are not open to public. ImageNet VID is fully annotated and is widely used in recently published works on video object detection. In this thesis, we also use Image VID to evaluate our method performance.

2.5 Evaluation Metrics

To evaluate the method performance, several evaluation metrics are computed according to the prediction results. In fully supervised detection, mAP is usually used to evaluate the detection performance. In weakly supervised detection, mAP and $CorLoc$ are used to evaluate the detection method performance.

FIGURE 2.19: Sample images in YTO benchmark.

FIGURE 2.20: Sample images in ImageNet VID benchmark.

mAP is short for mean average precision. It is computed according to *Precision* (P) and *Recall* (R). Before presenting mAP , we firstly introduce *Precision* and *Recall*. Given many samples including positive and negative samples, a predictor is to predict these samples, classifying these samples into positive and negative samples. The prediction results can be divided into four parts: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The relationship among these four parts are shown in Fig. 2.21. TP indicates that positive samples are correctly identified positive samples, while FP indicates negative samples are wrongly recognized as positive samples. TN means that negative samples are correctly predicted as negative samples, while FN means that positive samples are wrongly recognized are negative samples.

According to the relationship among these four parts, *Precision* (P) is defined as:

FIGURE 2.21: Relationship among true positive (TP), false positive (FN), true negative (TN), and false negative (FN).

$$P = \frac{TP}{TP + FP}, \quad (2.9)$$

and *Recall* (R) is defined as:

$$R = \frac{TP}{TP + FN}. \quad (2.10)$$

Based on these definitions, it can be seen that *Precision* indicates the fraction of correctly predicted positive samples among all predicted positive samples. *Recall* expresses the fraction of all positive samples that are correctly predicted.

In object detection, it not only correctly classify this object into its corresponding class, but also needs to accurately locate this object. To measure whether a detector accurately locates an object, an Intersection over Union (IOU) between a prediction and a ground truth is computed. Given a predicted bounding box B_p and an expected ground truth bounding box GT , the IOU is computed as follows:

$$IOU = \frac{Area(B_p \cap GT)}{Area(B_p \cup GT)}. \quad (2.11)$$

Additionally, to evaluate whether a detector correctly classify an object, the predicted confidence score s is also used. In general, when a prediction with IOU larger than 0.5, it is regarded as true positive.

To compute mAP , average precision (AP) is firstly computed. Given detection results at a certain class. We sort these detection results in a descending order according to their predicted scores. Then, AP is computed on this sorted detection result list as follows:

$$AP = \sum_{k=1}^N P(k) \triangle R(k), \quad (2.12)$$

where N indicates the number of detection results, $P(k)$ means the precision at the cut-off k in the sorted list, and $\triangle R(k)$ is the recall change from $k - 1$ to k . After computing AP for each class, mAP is obtained by computing the average of AP in all classes.

For computing *CorLoc* [109], only the detection results with highest scores in each class per image are recorded. According to its definition in [109], CorLoc expresses the percentage of images which include at least one instance correctly detected among these results. Similar to mAP , the detection results with IOU between ground truths larger than 0.5 is regarded as true positive.

Chapter 3

Feature Flow: In-network Feature Flow Estimation for Video Object Detection

3.1 Introduction

Optical flow which indicates the pixel displacement is very popular in many computer vision tasks. It is usually applied in video based tasks to indicate motion information. Recently, video object detection has witnessed rapid progress by using convolutional neural networks (CNN) [74, 75, 83]. These methods aim to leverage temporal information to enhance the feature representation at the current frame. Due to feature displacements existing across frames, they utilize optical flow (which expresses pixel displacement), which is generated by FlowNet [84] to estimate the feature displacement and align these feature maps according to the generated optical flow. After that, these aligned features are aggregate by weighted sum to generate an enhanced feature map for detection. Although good performances are achieved by these methods, they are limited by three drawbacks as discussed below:

- Optical flow expresses pixel displacement in images. Since the feature displacement is not strictly consistent with pixel displacement, it is inaccurate to align features based on optical flow. The quality of aggregated features

is degraded by this inaccurate alignment scheme, limiting the accuracy of detectors.

- As an external network, FlowNet [84] is used in these methods to generate optical flow. In these methods, they pre-train FlowNet on an additional dataset, Flying Chair [84] and then fine-tune it on the task dataset. This makes the training procedure complex. Additionally, FlowNet, which consists of 23 layers, is computationally expensive and occupies large GPU memory. This also limits the number of aggregated frames during training and testing. Overall, these optical flow based methods are inefficient in training and inference.
- Domain shift exists between optical flow training datasets and the target task datasets. Trained in an external optical flow dataset, FlowNet cannot adapt well to the video object detection task (discussed in Sec. 3.3.3), although it is fine-tuned in the task dataset. This further degenerates the aggregated features and affects the accuracy of detectors.

To tackle the above problems, we propose a novel neural network which is able to predict **feature flow**, indicating the feature displacement. Since an **In-network Feature Flow** estimation module (IFF) is proposed in this network, our proposed method is called IFF-Net. Our IFF-Net predicts feature flow and detection results in a **single** neural network.

Benefited from the feature flow generated by our IFF module, our IFF-Net could align and aggregate features better than optical flow based methods. It provides a solid foundation for accuracy improvement of detectors.

Additionally, our IFF module shares the network backbone with detection branches, which prevents IFF-Net from the overhead computation in FlowNet and makes feature flow estimation nearly cost-free. Without resorting to any additional neural network, our IFF-Net could aggregate temporal feature and predict results in a single network. Furthermore, our IFF module does not need any additional dataset for training. So, our IFF-Net is effective and efficient in both training and inference.

IFF-Net is trained in an end-to-end manner, which makes the feature flow generated by our IFF module adapt well to this task. It improves the quality of aligned features further. In contrast to optical flow based methods, we do not have the issue of domain shift due to networks training in additional optical flow datasets.

Since features highly depend on trained models, it is difficult to do annotations for feature flow in training datasets. Instead of fully supervised training, we propose the transformation residual loss (TRL) to improve the performance of IFF module in *self-supervised* learning. It is used as an additional loss in *training*. After employing TRL, feature flow from IFF is improved with negligibly additional computation cost. The performance of IFF-Net is also improved.

Our contributions can be summarized as follows:

- We propose a novel network (IFF-Net) with an in-network feature flow estimation module (IFF). Compared with optical flow, feature flow is better at feature alignment, which enables IFF-Net to detect video objects more accurately. Additionally, our IFF does not require any additional dataset in training. By sharing features with detection branches, the feature flow estimation by our IFF is nearly cost-free, enabling efficient feature aggregation. Our IFF-Net is efficient in training and inference.
- IFF-Net is trained in an end-to-end manner, which avoids the domain gap problem in existing methods. A transformation residual loss (TRL) is also introduced as a self-supervision method, which improves our IFF module further and boosts up the performance of IFF-Net.
- Seq-NMS [72] is widely used in many approaches for video object detection as a post-processing method. We improve Seq-NMS (denoted as Seq-NMS+) and apply this modified version to further boost the performance of our IFF-Net.
- We have carried out extensive experiments to verify the effectiveness of our method. IFF-Net outperforms all existing methods and achieves a new state-of-the-art accuracy on the ImageNet VID [108] while maintaining a fast inference speed.

FIGURE 3.1: The Pipeline of IFF-Net (in-network feature flow estimation network) is illustrated in the case of fusing the reference features at frame t with its neighboring features at frame $t - 5$. The features at frame $t - 5$ are warped to the features at frame t via the FFM (feature flow maps) produced by IFF (feature flow estimation module). Then, aggregated features are generated by fusing the features at frame t and the warped features at frame $t - 5$. After that, the aggregated features are forwarded to RPN and a detector to produce results. The transformation residual loss (TRL) is applied only in the training process.

3.2 In-network Feature Flow Network

Our proposed framework, IFF-Net is composed of a network backbone N_m and three branches (feature flow estimation module, detector network and region proposal network), with the overall framework shown in Figure 3.1. Given L proceeding frames of a video, i.e. $I_t, t = 1, \dots, L$, N_m produces feature maps F_t for each of them. The number of aggregated features of neighboring frames are denoted as τ . For convenience, we discuss our framework in the case of $\tau = 1$ in the following parts, as it is similar for other values of τ . IFF produces the corresponding feature flow map (FFM), $M_{j \rightarrow i}$, for features from a pair of images I_i and I_j , where I_i is the image of a reference frame and I_j is the image of its neighboring frame. Inspired by FGFA [74], the feature map from I_j is then warped to I_i based on the $M_{j \rightarrow i}$. The warped feature map $F_{j \rightarrow i}$ is obtained as:

$$F_{j \rightarrow i} = \mathcal{T}(F_j, M_{j \rightarrow i}), \quad (3.1)$$

where $\mathcal{T}$ is a bilinear warping function. After that, feature aggregation is performed via a weighted sum operation on features of the reference frame and other warped features according to FGFA. The aggregated feature is then passed to RPN and

a detector to make predictions. As an additional loss, the TRL is used for self-supervised learning in the training process. Our IFF-Net is a single network and can be trained end-to-end.

3.2.1 In-network Feature Flow Estimation Module

To accurately align feature maps across frames, we newly propose in-network feature flow (IFF) estimation modules in this subsection to provide feature flow. With the produced feature flow, our IFF-Net is able to aggregate temporal feature accurately, which provides a solid foundation for detection improvement. Our IFF module are built in IFF-Net itself.

Given two feature maps F_i and F_j , IFF predicts a feature flow map (FFM) which expresses feature flow. Since IFF directly predicts feature displacement, the displacements of features is captured more accurately than optical flow from FLOWNet. As IFF shares network backbone with detection branches, expensive computation on images in FlowNet is avoided, leading to nearly cost-free motion estimation. Two kinds of IFF versions are introduced in this part and are illustrated in Figure 3.2.

Basic IFF. Basic IFF is shown in the Figure 3.2(a). It is a basic version of IFF, and consists of three convolutional layers and a concatenation layer. A feature map of a reference frame F_i and a feature map of its neighboring frame F_j are fed into the first two convolutional layers ($1 \times 1 \times 512$ kernels and $3 \times 3 \times 512$ kernels), separately. After that, these two produced features are concatenated together and passed to a convolutional layer ($3 \times 3 \times 2$ kernels), which predicts the corresponding FFM.

Benefiting from end-to-end train, our proposed IFF-Net with the basic IFF module shows a comparable performance to optical flow based methods like FGFA. FlowNet used in FGFA is composed of 23 layers, whereas our basic IFF only consists of 3 layers. This indicates that our basic IFF module is able to efficiently produce FFM and effectively align feature map across frames. To further improve our IFF module performance, we propose an advanced version as follows.

Advanced IFF. To further improve the performance of the IFF module, an advanced IFF module is proposed. Since the relationship between the reference frame

FIGURE 3.2: Different structures for feature flow estimation module (IFF). F_i denotes the feature of the reference frame and F_j denotes the feature of its neighboring frame. The structure of the basic IFF is illustrated in (a). An advanced version of IFF is illustrated in (b).

and its neighboring frame provides important information for feature flow estimation, we explicitly produce this relation information and fuse it with the feature map of the reference frame (which undergoes an element-wise addition merger) in the advanced IFF module. The structure of this advanced IFF is illustrated in the Figure 3.2(b).

Inspired by ResNet [23], the first two layers in the basic IFF is replaced by a residual block (called embedded block 1 abbreviated as EB-1). It is shown in the Figure 3.3(a). EB-1 consists of three convolutional layers with $1 \times 1 \times 512$ kernels, $1 \times 1 \times 512$ kernels, and $3 \times 3 \times 512$ kernels. After that, the two features are concatenated together and passed to a correlation layer [84] to produce the relation information between two features. The correlation layer is defined as:

FIGURE 3.3: Illustration of the structure of EB-1, shown in (a), and EB-2, shown in (b).

$$o(x, y) = \frac{\mathbf{F}_i(x, y) \cdot \mathbf{F}_j(x + d_x, y + d_y)}{c}, \quad (3.2)$$

where $\mathbf{F}_i$ and $\mathbf{F}_j$ are two feature vectors, c is the channel number in this feature map and $\mathbf{d} = (d_x, d_y)$ denotes the offset between two feature vectors, F_i and F_j . Correlation is operated in a square neighborhood $D \times D$ in pixel with $D = 2\bar{d} + 1$, where $\bar{d}$ is the maximum displacement. A stride s is used in F_j to sample on F_j feature map. So, the size of correlation feature map is $w \times h \times (2 \times \bar{d}/s + 1)^2$, where w and h are the width and height of feature map F_i or F_j .

To perform element-wise addition merging, the relation information and the feature map of the reference frame are converted into the same sized feature map (128-dimension used here). The result of the correlation layer is passed to a convolutional layer ($3 \times 3 \times 128$ kernels). The features of the reference frame produced by EB-1 are passed to another embedded block denoted as EB-2, which is illustrated in Figure 3.3(b). EB-2 is similar to EB-1. The difference is that the channel of kernels for layers in EB-2 are changed to 128. After merging, the fused feature map is passed to two convolutional layers ($3 \times 3 \times 128$ kernels and $3 \times 3 \times 2$ kernels) to produce a FFM.

IFF-Net with this advanced IFF shows a better performance than FlowNet based methods [74, 83], although this advanced IFF only consists of 9 layers, much smaller than Flownet (23 layers). To provide a better performance of IFF-Net, we develop

a Transformation Residual Loss (TRL) inspired by *self-supervision* and present it in the next subsection.

3.2.2 Transformation Residual Loss

Feature maps highly depends on trained models. It is difficult to construct a training dataset with accurate annotations for feature flow. To address this issue and further improve our IFF-Net performance, we develop a transformation residual loss (TRL) in a self-supervised learning. We adapt the main formulation in TV-L1 [110] into our framework. It is used as an additional loss to IFF in training. This modified formulation (TRL) is written as follows:

$$\mathbf{L}_{TRL} = \lambda \frac{1}{N_p * d} \sum_{\mathbf{p} \in \Omega} \text{smoothL}_1[\Phi(F_j(\mathbf{p})) - F_i(\mathbf{p})], \quad (3.3)$$

where $\mathbf{p}$ denotes a location on a feature map F , Ω denotes all locations of a feature map, N_p denotes the number of feature vectors in a feature map and d represents the number of channels for a feature vector. A trade-off weight λ is added. The transformation $\mathcal{T}$ in Eq. (3.1) is used as $\Phi(\cdot)$ in Eq. (3.3). As this additional loss involves a transformation, we refer to it as transformation residual loss (TRL).

The temporal consistence between warped images and reference images is involved explicitly in TRL. When training with TRL, the performance of IFF-Net is improved, surpassing methods [74, 83].

3.2.3 Modified Seq-NMS

Seq-NMS [72] has shown effectiveness in approaches [74, 76]. It proposes a post-processing method which produces sequences along the temporal dimension in three steps: sequence selection, sequence re-scoring and suppression. In this subsection, we make two major modifications to improve its performance.

- The suppression step is originally proposed in the third step. In our modified version, we move this operation to the first step. With this modification, the post-processing speed is improved.

- Original Seq-NMS only use *average* or *max* operation at the sequence re-scoring step. Average operation is enable to suppress outliers along temporal dimension, while max operation enhances scores of linked boxes. To utilize the advantages of both operations, we jointly use average and max operation to combine their advantages: $0.5 * \text{mean}(\cdot) + 0.5 * \text{max}(\cdot)$ in our modified version.

After employing this modified Seq-NMS, our performance is improved further. For convenience, we denote this modified Seq-NMS as Seq-NMS+ in this thesis.

3.2.4 Network Architecture

The network structures in IFF-Net is presented as follows.

Network backbone. The ResNet-101 [23] pretrained on ImageNet classification is used as our network backbone. To meet the requirement of our task, we modify it according to FGFA [74]. We remove the last average pooling layer and the fully connected layers. Also in the last block, we change the effective stride from 32 to 16, and set the dilation rate of convolutional layers as 2. A convolutional layer ($3 \times 3 \times 1024$ kernels) is added to the last layer and is initialized randomly.

Detection network. R-FCN [43] is used as our detector network. We modify it according to the method in [83]. There are two branches, R-FCN sub-network and RPN sub-network. The first 512-dimension feature map is connected to RPN and the left 512-dimension feature map is passed to R-FCN. 9 anchors (3 scales and 3 aspects) are applied in RPN. RPN produces 300 proposals for each image. The groups in position-sensitive score map is set as 7×7 . The 1024-dimension feature map of the last convolutional layer is fed to IFF module.

3.3 Experiments

3.3.1 Experimental Setting

Datasets. Experiments are conducted on the ImageNet VID [108] dataset which is for video object detection. There are 3862 training videos and 555 validation

videos with annotations of 30 classes. We train our model on a subset of ImageNet DET training set (only the same 30 classes as in ImageNet VID) and VID training set following the setting in previous methods [74, 82]. Following their protocols, training is performed on the training set and performance is evaluated on the validation set.

Motion Categories. Following the protocol of FGFA [74], we classify objects in ImageNet VID into three categories according to object motion speed measured based on the average of object ground truth intersection-over-unions (IOUs) across frames $[t - 10, t + 10]$. The objects with average IOU larger than 0.9 are classified into slow category. For objects with average IOU $\in [0.7, 0.9]$, they belong to middle category and the remaining objects are classified into fast category. According to FGFA, the slow, middle and fast motion categories respectively occupy 37.9%, 35.9% and 26.2%.

Implementation details. SGD training is performed in training. 120K iterations are performed on 4 GPUs. Each GPU holds one image sampled from ImageNet DET or VID. A learning rate of 10^{-3} is used for the first 71.25 K iterations, and a learning rate of 10^{-4} is set for the rest iterations. Images are resized to a shorter side of 600 pixels in both training and testing. Only two neighboring frames are warped during training. For the reference frame t , two frames are sampled randomly in the range of $[t - 10, t + 10]$. λ in Eq. (3.3) is set as 0.65. In Eq. (3.2), the maximum displacement $\bar{d}$ is 10 and the stride s is 2. In testing, the number of warped images is set as 20, which is the same as in FGFA.

3.3.2 Comparison between Optical Flow and Feature Flow

Optical flow indicates pixel-level displacement across frames, which is not consistent to feature-level displacement. To show the limitation of optical flow in feature alignment, we firstly conduct five experiments listed in Table 3.1 and do comparison between optical flow and our proposed feature flow in feature alignment. Then, we illustrate some visualization results in the Figure. 3.4 and briefly discuss them.

Feature Alignment Experiments. In Table 3.1, baseline(R-FCN) serves as our baseline, where we only apply R-FCN, an image based detection method, to each frame alone. There is no temporal information involved in baseline(R-FCN). In

Method	mAP	mAP (slow)	mAP (middle)	mAP (fast)
baseline(R-FCN)	74.1	83.6	71.6	51.2
OptFlow-A	74.4	84.9	74.0	51.5
OptFlow-B	75.4	83.7	74.8	53.5
IFF-Net-B(ours)	76.4	85.3	75.0	54.6
IFF-Net-C(ours)	77.1	85.0	75.9	57.7

TABLE 3.1: Comparison between optical flow and feature flow in feature alignment. OptFlow-A and OptFlow-B apply optical flow to feature warping, while IFF-Net uses feature flow in feature alignment. This shows that optical flow is not suitable for feature alignment, while our feature flow is able to effectively align temporal features.

comparison, two experiments (denoted as OptFlow-A and OptFlow-B) are carried out by using FlowNet for temporal feature aggregation and R-FCN is applied to the aggregated feature. We also conduct two experiments, where we replace FlowNet with our proposed IFF module, which is denoted as IFF-Net-B and IFF-Net-C. All these five experiments are conducted under the same hyper-parameters.

In OptFlow-A, we use a FlowNet, which is **pre-trained** on the optical flow dataset, for feature alignment. To ensure feature alignment conducted with optical flow, we fix FlowNet and only fine-tune detection part in the task dataset. From the Table 3.1, it can be seen that OptFlow-A performs similar to baseline(R-FCN) (74.4 v.s. 74.1). However, in OptFlow-B, we jointly fine-tune FlowNet and the detection parts during training. Compared with OptFlow-A, OptFlow-B shows 1 mAP improvement. *This shows that original optical flow is not suitable for feature alignment due to the difference between pixel-level displacement and feature-level displacement.*

To solve this problem, we replace the FlowNet in OptFlow-B with our proposed IFF module in IFF-Net-B and IFF-Net-C. IFF-Net-B uses our basic IFF module, and IFF-Net-C uses our advanced IFF module. From the Table 3.1, although our basic IFF module only consists of three layers, it effectively improve the detection accuracy than OptFlow-B (76.4 v.s. 75.4). By employing our advanced IFF module in IFF-Net-C, the performance is further improved (77.1 v.s. 75.4).

In the experiment OptFlow-B, although we fine-tune FlowNet on the task dataset, the performance of FlowNet based method (75.4) is still inferior to our IFF-Net

(76.4 and 77.1). *This shows that the domain gap between FlowNet training dataset and the task dataset hinders the performances of FlowNet based methods.*

According to this comparison between optical flow and our feature flow, it demonstrates that optical flow is not suitable for feature alignment. Our proposed IFF modules effectively solve this problem and show significant improvement.

Feature Flow Visualization and Discussion. To show our proposed feature flow clearly, we visualize some feature flow results and compare them with the corresponding optical flow in Figure. 3.4.

FIGURE 3.4: Visualization of feature flow maps. The first to the fifth rows respectively list original images, corresponding feature map visualizations, optical flow from Flownet, feature flow maps from our IFF and detection visualizations.

Original images are shown in the first column, while columns 2 to 4 respectively list their corresponding feature maps, the optical flows of Flownet, and the FFM produced by our IFF module. These feature maps are accumulated along the channel axis. To show the effectiveness of our feature maps, we visualize the corresponding detection results on the last column. To show them more clearly, the regions of objects are zoomed in.

In the video detection task, we need to perform feature alignment for feature aggregation. As feature captures different information (semantics, context) from pixels, the estimated feature flow is different from optical flow (pixel displacement). Compared with optical flow, using feature flow is more suitable for performing warping in the feature level. Moreover, end-to-end training makes our estimated feature flow well adapted to our detection task, which leads to better performance than optical flow based methods.

3.3.3 Ablation Study

To show the effectiveness of our IFF-Net, the performance of baseline and our IFF-Net variants on ImageNet VID are listed in Table 3.2.

Method	IFF module	TRL	Post-process	mAP	mAP (slow)	mAP (middle)	mAP (fast)
baseline(R-FCN)				74.1	83.6	71.6	51.2
IFF-Net-A(ours)	basic			76.0	85.0	74.8	54.7
IFF-Net-B(ours)	basic	✓		76.4	85.3	75.0	54.6
IFF-Net-C(ours)	advanced	✓		77.1	85.0	75.9	57.7
IFF-Net-D(ours)	advanced	✓	Seq-NMS	78.9	86.6	78.1	59.8
IFF-Net-E(ours)	advanced	✓	Seq-NMS+	79.7	87.5	78.7	60.6

TABLE 3.2: Ablation study of our IFF-Net and comparison with optical flow based methods. IFF, TRL and Seq-NMS+ improve the performance of IFF-Net. OptFlow-A and OptFlow-B apply optical flow to feature warping, while IFF-Net uses feature flow in feature warping. All IFF-Net variants are better than these optical flow based methods. It shows that our IFF is better than optical flow in feature warping.

Our baseline consists of ResNet-101 [23] and R-FCN [43] detector. It is a single-frame framework and the same as the baseline in the Table 3.1. Its mAP is 74.1% on overall categories. In Table 3.2, the mAP for fast motion category, 51.2%, is the lowest among three categories. This indicates that fast motion category is challenging. However, as the percentage (26.2%) of fast motion category in the validation set is the lowest in all categories, the mAP for all categories is still very high, that is 74.1%.

After adding the basic IFF module, the performance (denoted as IFF-Net-A) for all categories is 76.0% mAP, 1.9% higher than our baseline. This shows that the basic IFF module improves the performance effectively. There are a 3.5% gain in mAP for fast motion category, 3.2% gain in mAP for middle motion category and 1.4%

gain in mAP for slow category, respectively. After training with TRL (denoted as IFF-Net-B), the performance of the basic IFF module improves further (76.4% mAP vs 76.0% mAP). TRL not only improves the performance, but also makes the convergence faster. IFF-Net-B in Table 3.2 is the same as IFF-Net-B in Table 3.1.

Another variant using the advanced IFF module and TRL, denoted by IFF-Net-C, is shown in the table. The mAP score increases from 76.4% to 77.1%. The performance of slow motion category is nearly unchanged after replacing basic IFF module with advance IFF module, but the mAP of fast motion category increases significantly by 3.1% (57.7 % mAP vs 54.6 % mAP). This indicates that advanced IFF module focuses on the difficult category (fast category), while yielding a similar performance on the easy category (slow category). IFF-Net-C in Table 3.2 is the same as IFF-Net-C in Table 3.1.

Seq-NMS is widely used in existing methods, and we also explore its effectiveness in our framework. After adding Seq-NMS, the performance of our IFF-Net obtains 1.8% point gain. This shows that Seq-NMS is also effective for our framework. By changing the Seq-NMS into Seq-NMS+ (our proposed extension of Seq-NMS), the performance improves further, with 0.8% mAP gain obtained.

3.3.4 Comparison with the state-of-the-art

Comparisons with the state-of-the-art methods are further conducted. Existing methods utilize different post-processing methods. For a fair comparison, we firstly compare IFF-Net with some methods listed in Table 3.3 without ant post-processing method. All methods listed in Table 3.3 apply the same network backbone (ResNet-101). It can be seen that our IFF-Net-C (77.1% mAP) performs better than FGFA [74] (76.3% mAP) and D&T [73] (75.8% mAP) under the same configuration.

For more comprehensive comparison, we discuss other methods in Table 3.4, where detailed configuration of each approach is also listed. Deformable convolutional layers [60] is widely used in existing works, denoted as DCN. We also explore its effectiveness in our IFF-Net. Following DCN v1, we train our IFF-Net with 3 DCN layers (denoted as IFF-Net-F). We also investigate the number of warped images in

Method	Backbone	Post-process	mAP
IFF-Net-C(ours)	ResNet-101	No	77.1
D&T	ResNet-101	No	75.8
FGFA	ResNet-101	No	76.3

TABLE 3.3: Comparison with state-of-the-art methods without any post-processing method. It shows that our IFF-Net-C achieves the best performance among other methods without post-processing method.

training. Previously, we only use two neighboring images in training. In Table 3.4, we further evaluate our method with 4 neighboring frames warped during training. The resulting model is denoted by IFF-Net-G, which improves the performance of our IFF-Net (81.2% mAP). We also train a model with 6 DCN layers denoted as IFF-Net-H.

Method	Backbone	Post-process	mAP
FGFA[74]	ResNet-101 + FlowNet[84]	Seq-NMS	78.4
Zhu et al. 2017a[75]	ResNet-101(DCN) + FlowNet[84]	None	78.6
STMN[76]	ResNet-101 + DeepMask	Seq-NMS, Ensemble	80.5
D&T [73]	ResNet-101	Viterbi [111]	79.8
STSN[77]	ResNet-101(6*DCN)	Seq-NMS	80.4
PSLA [78]	ResNet-101(6*DCN)	Seq-NMS	81.4
Deng et al.[79]	ResNet-101(6*DCN)	Seq-NMS	81.6
SELSA[80]	ResNet-101(6*DCN)	Seq-NMS	80.54
RDN [112]	ResNet-101	None	81.8
IFF-Net-E(ours)	ResNet-101	Seq-NMS+	79.7
IFF-Net-F(ours)	ResNet-101(3*DCN)	Seq-NMS+	80.5
IFF-Net-G(ours)	ResNet-101(3*DCN)	Seq-NMS+	81.2
IFF-Net-H(ours)	ResNet-101(6*DCN)	Seq-NMS+	82.1

TABLE 3.4: Comparison with state-of-the-art methods under **different** settings. It shows that our IFF-Net-H achieves the best performance among other methods with less training dataset and smaller network on ImageNet VID validation set.

In the Table 3.4, FGFA and Zhu et al. 2017a explores temporal information based on the optical flow from FlowNet. Although they have fine-tuned FlowNet on the target dataset, their performances are still inferior to our IFF-Net. STMN and PSLA are designed to align frames across frames. However, their similarity based feature alignment is too simple to align temporal features accurately. This results their performances are not the best.

For the STSN method, it only adopts DCN layers for capturing temporal information. Motion estimation and feature alignment across frames are not involved

in it. This simple temporal aggregation scheme results that it only achieves 80.4 mAP, which is much lower than our IFF-Net methods. D&T utilizes a post-process method, Viterbi [111], to improve its detection performance. But its performance is still lower than ours.

SELSA[80] and RDN [112] are proposed to fuse feature across frames on proposal level. From the Table 3.4, it can be seen that these methods are not better than our IFF-Net. In RDN [112], it utilizes the post-processing method Viterbi [111] in D&T, providing a better performance. Our IFF-Net can also uses this method. We set it as our future work.

Recently, some methods uses ResNeXt as their backbone. Network backbone greatly influences the detection performance. For fair comparison, we only compare their performances with ResNet-101 in this part. From Table 3.4, our model, IFF-Net-H, achieves the best performance (82.1% mAP) among all listed methods, surpassing all other methods. Some detection results of IFF-Net are visualized in Figure 3.5.

3.3.5 Speed Analysis

While achieving high detection accuracy, our method is also efficient. To show the efficiency of our IFF-Net, we discuss the time cost of our method and compare it with optical flow based methods in the Table 3.5.

Method	runtime
FGFA	20ms
IFF-Net(ours)	8.8ms

TABLE 3.5: Comparison with optical flow based methods on feature alignment per frame. It shows that our IFF-Net aligns feature maps much faster than FGFA.

As our IFF-Net share a similar detection pipeline with optical flow based methods, like FGFA, we only compare time cost on the feature alignment. The time cost of Flownet for optical flow prediction of one frame on our server is 20ms on average, which is similar to that reported in STMN. In comparison, our advanced IFF only costs 8.8ms. As the number of warped images is 20 in optical flow based methods, time cost on Flownet is 400ms on average. Our IFF only costs 176ms, which is

FIGURE 3.5: Examples of detection results based on IFF-Net.

less than half of FlowNet. In summary, our framework, IFF-Net, achieves better performance than other state-of-the-art methods in the sense that it yields a fast prediction speed with less training datasets.

3.4 Conclusion

We have proposed a novel end-to-end network, IFF-Net for video object detection. Our IFF-Net is able to efficiently perform feature flow estimation for effective and efficient feature alignment. Experimental results show that our IFF-Net achieves the best result with less training datasets and a smaller network in a fast speed compared to other state-of-the-art methods.

Chapter 4

Motion Context Network for Weakly Supervised Object Detection in Videos

4.1 Introduction

Object detection [41, 45, 46] has been substantially improved due to the remarkable progress of Convolutional Neural Network (CNN) [21–23]. However, this significant improvement is achieved on the cost of constructing labor-intensive and time-consuming box-level annotations in large scale datasets [3, 4]. To overcome this issue, various approaches [88–91] have been proposed for Weakly Supervised Object Detection (WSOD) which only requires image-level annotations. Image-level annotations require much less labor cost than box-level annotations. They also can be obtained by simple web search on image search engines without resorting to human labeling.

In weakly supervised object detection, most existing approaches are proposed for still images. Despite substantial progress has been made, they still perform significantly lower than the fully supervised methods. Existing weakly supervised approaches tend to regard the most discriminative region as an object, which cannot accurately identify the object region especially in cluttered scenes. They often locate either only a part of an object, like the head of a person, or an overlarge region including both the object and its neighborhood. This drawback greatly

affects their performances. Compared with images, videos contain much more information, i.e., not only appearance information, but also motion information. Considering objects may show different motion from their surrounding objects or background, we propose to leverage motion information from videos to alleviate this issue.

FIGURE 4.1: Illustration of optical flows for two objects. RGB images and optical flows are shown on the left and right, respectively. The airplane is clearly highlighted in the optical flow. In contrast, as a non-rigid object, this bear’s motion pattern is complex and the region of the whole object cannot be directly obtained from its corresponding optical flow.

A simple approach to utilise motion information is directly using optic flow to locate the object regions. As the airplane shown in Fig. 4.1, its corresponding optical flow provides precise information for the object location, clearly highlighting the region of this object. However, the motion pattern of an object is generally complex, which is usually different from the airplane scenarios. As the bear shown in Fig. 4.1, different parts of this non-rigid object show different motion pattern. It shows that the body and the legs of this bear are nearly stationary relative to background, whereas its head shows obvious movement. This indicates that different part of an object may show different motion pattern and some parts even show very similar motion pattern (stationary) to the background. As this type of non-rigid object motion widely exists in real-world scenarios, directly using optical flow may not be suitable or even degrades the detection performance.

FIGURE 4.2: Detection comparison between the baseline method OICR (left) and our MC-Net (right). One optical flow image is shown in the middle, where two moving regions of this hamster are marked in dashed boxes. The baseline method fails to locate the object. By incorporating motion context information and appearance information from images, our MC-Net accurately detect this hamster.

To effectively explore motion information, we propose a **Motion Context Network** (MC-Net) which utilizes motion context features derived from the motion information of moving regions to improve the detection accuracy. We first illustrate the effectiveness of our MC-Net by comparing with a image based weakly supervised method OICR [89] which has shown remarkable performance on weakly supervised object detection. In Fig. 4.2, the detection results of OICR (baseline) and our MC-Net are shown on the left and the right part of the figure, respectively. MC-Net uses our proposed motion context component to capture motion information from optical flow. One optical flow image is visualized in the middle of Fig. 4.2, where two moving regions of this hamster are highlighted by boxes. As illustrated in the figure, OICR cannot accurately locate this hamster. Its prediction box regards the hamster, the shadow and the hand object as one object. In contrast, our MC-Net accurately detects this hamster by incorporating motion context features derived from these moving regions.

In our MC-Net, we propose a **Motion Correlation Module** (MCM) which explicitly computes motion correlation between a region and its neighborhood to provide motion context information. The information is then fused with appearance information from images and forwarded to OICR. Benefited from MCM, our MC-Net is able to incorporate appearance information from images and motion context information in a single neural network to produce accurate detection. Our MC-Net can be trained end-to-end and generalizes well on both rigid and non-rigid objects with complex motion states.

Object detectors may suffer from deteriorated object appearances in videos, like motion blur and objects under unusual viewpoints. To alleviate this issue, we develop a **Temporal Aggregation Module** (TAM) according to FGFA [74], which leverages neighboring frames (RGB images) to enhance the feature representation of the current frame.

TAM consists of two components: feature alignment and feature aggregation. In feature alignment, neighboring frames are aligned according to optical flow by a bi-linear warping function. After that, these aligned features are aggregated according to the similarities between them. By employing TAM, the issue of deteriorated object appearances is effectively alleviated and the performance of our MC-Net is further improved.

Our main contributions are summarized as follows:

- We propose a novel method, MC-Net, for weakly supervised object detection in videos. A Motion Correlation Module (MCM) is proposed to produce motion context information based on neighborhood motion correlation. Our MC-Net effectively incorporates both appearance information and motion context information together to achieve accurate object detection. It can be trained end-to-end and generalize well on both rigid and non-rigid objects with complex motion pattern.
- We develop a Temporal Aggregation Module (TAM) according to FGFA [74], which leverages temporal information and image features of neighboring frames to enhance the feature representation of one frame. Our TAM effectively mitigates the issue caused by deteriorated object appearances in videos.
- Extensive experiments are carried out on ImageNet VID [108]. Our MC-Net outperforms the baseline image based method by a large margin (37.4% mAP v.s. 29.8% mAP).

4.2 Motion Context Network

As illustrated in Fig. 4.3, our MC-Net mainly consists of two modules: Motion Correlation Module (MCM) and Temporal Aggregation Module (TAM). Given L

FIGURE 4.3: Architecture of MC-Net (Motion Context Network). It illustrates the case of using two neighboring frames (I_{t-5} and I_{t+5}) as inputs. Frame, I_t , and its neighboring frames, I_{t-5} and I_{t+5} , are forwarded to MC-Net. The Network backbone and the FLOWNet produce image feature maps and optical flows, respectively. The concatenated optical flows are passed to Motion Correlation Module (MCM). The feature maps from neighboring frames and optical flows are passed to Temporal Aggregation Module (TAM) to generate enhanced features. Finally, the aggregated appearance features and the motion context features are fused and forwarded to OICR to generate the final results.

consecutive frames of a video, i.e. I_t , $t = 1, \dots, L$, the current frame I_t and its neighboring frames are forwarded to our proposed MC-Net. For convenience, we discuss our MC-Net in the case of using 2 neighboring frames (I_{t-5} and I_{t+5}) in the following parts. The network backbone, N_m and the FlowNet [84], N_{flow} , first produce image feature maps (X_t , X_{t-5} , and X_{t+5}) and optical flows ($\mathcal{F}_{t-5}$, and $\mathcal{F}_{t+5}$), respectively. Then, Motion Correlation Module (MCM) is employed to provide the motion context information. Optical flows are concatenated together as an original motion matrix and passed to MCM. In the meanwhile, the Temporal Aggregation Module (TAM) is applied to aggregate neighboring features and improve the feature representation of the current frame. After that, both output feature maps from MCM and TAM are fused to incorporate image appearance and motion context information together. The OICR [89] is then employed to predict scores of each instance.

Our MC-Net incorporates both motion context information and appearance information in a single network. It generalizes well on rigid and non-rigid objects in any motion and can be trained end-to-end.

4.2.1 Motion Correlation Module

The Motion Correlation Module (MCM) is used to provide motion context information to improve object localization in the weakly supervised setting. In weak supervision, lacking of full supervision, a off-the-shelf decoder, like CNN, cannot be trained adequately. It is challenging to directly adopt an additional network for capturing motion context information. In this subsection, we focus on how to design the construction of the decoder. We propose a self-correlation layer, which does not introduce extra parameters and is able to effectively capture the object location information. MCM explicitly calculates motion correlation between a region and its neighborhood. It is composed of two parts: self-correlation layer and instance pooling layer.

4.2.1.1 Self-correlation layer

To efficiently compute the motion context information, we firstly propose a self-correlation layer to compute this information for each location. Then, a instance pooling layer is used to compute this information for a region. As illustrated in Fig. 4.4, the self-correlation layer explicitly calculates the motion correlation between a location and its neighborhood.

FIGURE 4.4: Illustration of self-correlation layer. Given a spatial location and a predefined neighborhood region, self-correlation layer computes the motion correlation between this location and its neighborhood. The central location, its neighborhood region and the motion correlation are visualized in purple, blue, and gray colors, respectively. The results are then reshaped into a vector.

Given an optical flow (original motion) map $\mathcal{M}_t$ and a location $\mathbf{p}_i$, the correlation computation can be written as:

$$o(\mathbf{p}_i, \delta) = \frac{\mathcal{M}_t(\mathbf{p}_i) \cdot \mathcal{M}_t(\mathbf{p}_i + \delta)}{C}, \quad (4.1)$$

where $\delta \in [-d, d] \times [-d, d]$ are offsets from the center $\mathbf{p}_i$, and C denotes the channel number of the input optical flow map. Let $D = 2d + 1$. The neighborhood region size for this operation is $D \times D$. Note that we need to perform C computation of $o(\mathbf{p}_i, \delta)$ for every neighborhood position. Given the input optical flow map: $\mathcal{M}_t \in R^{w \times h \times C}$, computing all motion correlation values needs $w^2 \cdot h^2 \cdot C \cdot D^2$ multiplications. For efficient computation, a stride s is applied to sub-sample the neighborhood positions δ . With this stride setting, this computation decreases to $w^2 \cdot h^2 \cdot C \cdot D^2 / s^2$. Finally, we rearrange the correlation values for a particular location into a vector, resulting a matrix $O \in w \times h \times D^2 / s^2$.

As the computation of motion correlation is performed on the *same* optical flow matrix, we denote this layer as **self-correlation layer**. Our self-correlation layer computes the correlation between two flow vectors, which does not introduce any additionally trainable parameters. Furthermore, it is implemented in a parallel way and can be run on GPUs. Benefited from the parallel computation, our motion context information can be efficiently calculated.

On the other hand, simple motion correlation information is not enough for MCM to identify moving regions. To enrich this information representation, the original flow matrix is concatenated with the output of this self-correlation layer.

4.2.1.2 Instance pooling layer

Self-correlation layer produces the motion context features for each spatial location. With the motion context feature map, we apply an instance pooling layer to produce fixed-size motion context vector for each proposal region. The instance pooling layer converts an arbitrary size feature map into a fixed size feature vector, while keeping relative spatial information. ROI pooling layer is widely used in object detection to generate fixed-size feature vectors. ROI pooling firstly divides the region of a proposal into several sub-windows and performs quantization of the window coordinates. Then, it applies max-pooling on each sub-window. However,

the max-pooling operation only keeps the max value in a sub-window, which cannot accurately express the motion context information. The quantization in ROI pooling further degrades this information. To mitigate these negative effects, we adopt ROI align in our instance pooling layer. ROI align uses average pooling and bilinear interpolation [113] to generate features for each window. In the Sec. 4.3, we show that using ROI align in our MC-Net achieves better performance than using ROI pooling.

4.2.2 Temporal Aggregation Module

Objects in video sometimes show deteriorated appearance like motion blur or object under strange viewpoints. To mitigate this negative effect, we develop a Temporal Aggregation Module (TAM) based on FGFA [74]. It leverages temporal information to align and aggregates the features of neighboring frames, enhancing the feature map of the current frame. TAM is composed of two components: feature alignment and feature aggregation.

4.2.2.1 Feature alignment

Due to the movement of objects or camera, there are displacement across frames. To alleviate this issue, feature alignment across frames are firstly performed via a warping process. As shown is Fig. 4.3, a current frame, I_t and its neighboring frames (I_{t-5} , I_{t+5}), are forwarded to the network backbone which produces the corresponding feature maps (X_t, X_{t-5}, X_{t+5}). At the same time, two corresponding optical flows ($\mathcal{F}_{t-5}$, $\mathcal{F}_{t+5}$) for I_{t-5} and I_{t+5} are generated respectively by FlowNet [84]. After that, these two feature maps, X_{t-5} and X_{t+5} , are warped to the feature map, X_t , according to optical flows, $\mathcal{F}_{t-5}$ and $\mathcal{F}_{t+5}$. The processing of warping a feature map X_j to the feature map X_i can be formulated as

$$X_{j \rightarrow i} = \mathcal{T}(X_j, \mathcal{F}_j), \quad (4.2)$$

where $X_{j \rightarrow i}$ denotes the warped feature map, $\mathcal{T}$ is a bilinear warping function, and $\mathcal{F}_j$ is the optical flow for X_j .

4.2.2.2 Feature aggregation

After warping neighboring features across frames, we aggregate these warped features together to enhance the feature at the current frame. It is defined as

$$\tilde{X}_i = \sum_{t=-\tau}^{\tau} \mathbf{W}_t \cdot X_t, \quad (4.3)$$

where $\mathbf{W}_t \in R^{w \times h}$ is the weight matrix. Note that it may not be effective if we simply adopt equal weights in (4.3), especially for some degraded feature maps. Instead of using equal weights, we compute weights according to the similarity between the current frame and its neighboring frames.

We firstly compute the correlation between the current feature map and its neighboring feature map according to (4.4) below.

$$\overline{\mathbf{W}}_t(x, y) = \frac{X_{t \rightarrow i}(x, y) \cdot X_i(x, y)}{|X_{t \rightarrow i}(x, y)| \cdot |X_i(x, y)|}, \quad (4.4)$$

where $X_{t \rightarrow i}$ is the warped feature map. Then, we normalize $\overline{\mathbf{W}}_t$ along the temporal axis according to (4.5) as follows.

$$\mathbf{W}_t(x, y) = \frac{\overline{\mathbf{W}}_t(x, y)}{\sum_{k=-\tau}^{\tau} \overline{\mathbf{W}}_k(x, y)}, \quad (4.5)$$

After that, we perform the feature aggregation based on these weight matrix $\mathbf{W}_t$ and performance ROI-pooling on this aggregated feature map. After applying TAM, the quality of feature map at the current frame is improved. The influence of objects in deteriorated appearances is alleviated.

4.2.3 Fusion Module

To incorporate both appearance information and motion context information effectively, we propose a fusion module illustrated in Fig. 4.5.

Fully connected (FC) layers are applied in this fusion module. As shown in Fig. 4.5, we use the two FC layers in a pre-trained VGG16 to encode features from the appearance information. Each FC layers has 4096 hidden nodes. For motion

FIGURE 4.5: The structure of fusion module. The features of appearance information are passed to two fully connected (FC) layers. The features from motion context information are passed to one FC layer. Then, both outputs are concatenated together and forwarded to OICR. With this module, our MC-Net is able to effectively incorporate appearance and motion context information together.

context branch, we add a new FC layer with 64 hidden nodes to extract the corresponding features. Then, both outputs of FC layers are concatenated together and forwarded to OICR. Each FC layer is followed by a ReLU layer and a dropout layer. By using this fusion module, our MC-Net is able to incorporate both appearance information and motion context information in a single neural network.

4.3 Experiments

4.3.1 Experimental Setup

Datasets and evaluation matrix. Our MC-Net is proposed for videos, so we evaluate it on ImageNet VID [108] which is a widely-used dataset for video object detection. It includes 3862 training videos and 555 validation videos with 30 classes. Following the protocols of existing methods in [73, 74, 83], we train our MC-Net on the training set of ImageNet VID and a training subset of ImageNet DET (only include the 30 classes in ImageNet VID). Since we aim to evaluate the performance our method under WSOD, only image-level annotations are kept in training.

In performance evaluation, we choose two evaluation metrics: mean average precision (mAP) and the CorLoc [109]. For mAP, we follow the protocols of methods [73, 74, 83] in video object detection and calculate mAP on the validation set of ImageNet VID. For CorLoc, it is widely used in weakly supervised object detection and is evaluated on the training datasets. Only the detection results with highest

scores in each class per image are recorded. CorLoc expresses the percentage of images which include at least one instance correctly detected among these results.

Motion Categories. Following the protocol of existing methods [74] in video object detection, we divide objects in ImageNet VID into three categories according to object speed which is measured based on the average IOU of object ground truth among neighboring frames $[t-10, t+10]$. The average IOU for slow motion category which occupies 37.9% in testing dataset, is $\in [0.9, 1.0]$. The average IOU of middle motion category is $\in [0.7, 0.9)$, occupying 35.9%. For the fast motion category, its average IOU is $\in [0.0, 0.7)$ and occupies 26.2%.

Implementation details. The following two networks are employed in our approach: VGG16 [21] and FlowNet [84]. The ImageNet pre-trained VGG16 is used as our network backbone which is to extract the feature map for detection. FlowNet is used to produce optical flows. To meet the requirement of our task, we make two modifications on VGG16. We firstly replace the final pooling layer, *pool5*, with ROI pooling layer. Secondly, we remove the penultimate pooling layer, *pool4*, and change the dilation rate on the *conv5* layers from 1 to 2. Two FC layers with 4096 hidden nodes are initialized by the FC layers in VGG16. The newly added layers are initialized randomly.

We pre-train OICR framework on PASCAL VOC 2007 according to OICR. Then, we train our MC-Net on ImageNet VID. SGD training is performed. We run 97.5K iterations on 4 GPUs and each GPU holds one images. The learning rate for the first 55K iterations is 0.001 and then decreases to 0.0001 for the rest iterations. The momentum is 0.9 and the weight decay is 0.0005. During training, we train MC-Net with four image scales, (480, 800), (576, 900), (688, 1000) and (800, 1000), where the first value is for shorter side of an image and the second value is to limit the maximum edge size of an image. The horizontal flip is applied in training. During testing, the image scale is (800, 1000). Following existing methods in WSOD, selective search is used for proposal generation, producing around 2,000 proposals per image. Limited by the memory of GPU, we only warp two neighboring frames at $t - 5$ and $t + 5$ in training and aggregate neighboring frames from $t - 5$ to $t + 5$ in testing. The original motion information is the concatenation of optical flows at frame $t - 3$ and $t + 3$. The stride s in the self-correlation layer is 2 and the parameter d in (4.1) is set to 10. The hyper parameters of OICR in our work follows its original settings in OICR. The IOU threshold value for NMS is 0.3.

Following the weakly supervised methods in [89, 114], we also perform pseudo ground training to improve the performance. We generate pseudo ground truth by selecting the top scored proposals predicted by MC-Net. We train a Fast RCNN [40] using the VGG16 model, horizontal flip (only in training), and image scale (800, 1000). Selective search is also used for proposal generation.

4.3.2 Ablation study

We show the effectiveness for each component in our MC-Net in this subsection. The results on the variants of MC-Net in mAP and CorLoc are presented in the Table 4.1 and the Table 4.2, respectively. We compute the overall CorLoc for the two training datasets. MCM (max) denotes MCM with ROI pooling and MCM (avg) is MCM with the ROI Align.

Methods	MCM (max)	MCM (avg)	TAM	mAP (%)	mAP (%) (slow)	mAP(%) (middle)	mAP (%) (fast)
Baseline (OICR)				29.8	37.5	29.3	18.0
MC-Net-A (ours)	✓			32.5 _{↑2.7}	40.7 _{↑3.2}	32.0 _{↑2.7}	19.2 _{↑1.2}
MC-Net-B (ours)		✓		32.9 _{↑3.1}	41.1 _{↑3.6}	32.3 _{↑3.0}	19.4_{↑1.4}
MC-Net-C (ours)		✓	✓	34.1_{↑4.3}	43.6_{↑6.1}	34.4_{↑5.1}	19.2 _{↑1.2}

TABLE 4.1: Ablation study in mAP for our MC-Net. MCM (max) is MCM with ROI pooling and MCM (avg) is MCM with ROI Align. It shows that each component of our MC-Net effectively improves the performance on all types of motion. In mAP, almost all MC-Net variants perform better than the baseline (OICR).

Methods	MCM (max)	MCM (avg)	TAM	VID CorLoc (%)	DET CorLoc(%)
Baseline (OICR)				58.5	63.3
MC-Net-A (ours)	✓			58.4 _{↓0.1}	62.9 _{↓0.4}
MC-Net-B (ours)		✓		58.8 _{↑0.3}	64.7 _{↑1.4}
MC-Net-C (ours)		✓	✓	59.9_{↑1.4}	69.1_{↑5.8}

TABLE 4.2: Ablation study in CorLoc of our MC-Net. From this table, It can be seen that each component of our MC-Net effectively improves the CorLoc performance. Almost all MC-Net variants perform better than the baseline (OICR).

To our best knowledge, our MC-Net is the first one applied to video based weakly supervised object detection. Since OICR shows remarkable performance in image

based WSOD, we select it as our baseline. We re-implement it and train it under the same setting of our MC-Net. As ImageNet VID includes many complex scenarios like multiple objects in various poses, it only shows 29.8 mAP on ImageVID validation set, 58.5% CorLoc in VID training set, and 63.3% CorLoc in DET training set.

Effectiveness of MCM. Two experiments (MC-Net-A and MC-Net-B) are conducted to illustrate the effectiveness of MCM in the Table 4.1 and the Table 4.2. For mAP, it is observed that MCM (max) improves the performance (32.5 v.s. 29.8) among all categories, especially for the slow motion category (3.2 points gain). It indicates that the motion context information provided by MCM effectively refines the detection results, improving the accuracy of detection. After applying the ROI Align (denoted as MC-Net-B), the performance of MC-Net is further improved, which shows that MCM (avg) provides a better motion context information for instance. For CorLoc, MC-Net-A performs similarly to the baseline, while MC-Net-B obtains a better performance.

Effectiveness of TAM. To validate the effectiveness of TAM, another experiment denoted as MC-Net-C is carried out. As shown in the Table 4.1 and the Table 4.2, TAM further improves the performance of our MC-Net on both mAP and CorLoc. For mAP, MC-Net-C obtains 2.5 mAP gain compared to MC-Net-B in slow motion category and 2.1 mAP gain in middle motion category. For CorLoc, MC-Net-C increases by 1.1 points relative to MC-Net-B in DET dataset and 4.4 points in VID dataset. We attribute this to the effectiveness of TAM which leverages temporal information from neighboring frames and enhances the feature map representation at the current frame. The problem caused by objects in deteriorated appearances is effectively alleviated.

4.3.3 Analysis on optical flow

In our MC-Net, we can directly extract features from optical flow to capture motion information, as an alternative to the motion context module for motion features generation. We denote this setting for motion feature generation as MC-Net-OptFlow. Directly using optical flow for motion features is also widely used in video action recognition tasks. We conduct the analysis experiment of using different motion features in Table 4.3.

Method	mAP(%)	VID CorLoc(%)	DET CorLoc(%)
Baseline (OICR)	29.8	58.5	63.3
MC-Net-OptFlow (ours)	31.8	58.9	62.7
MC-Net-MotionContext (ours)	32.9	58.8	64.7

TABLE 4.3: Analysis between optical flow and motion context. MC-Net-OptFlow is an alternative method to motion context, directly fusing optical flow features with image features for object detection. MC-Net-MotionContext denotes using motion correlation module to capture motion context information. MC-Net-OptFlow show better performance on mAP and CorLoc in VID than the baseline, while performs worse on CorLoc in DET. MC-Net-MotionContext performs better on all mAP, CorLoc in VID and DET. It shows that motion correlation module is more effective for object location compared to directly using optical flow.

In the variant of our method, MC-Net-OptFlow, we directly fuses optical flow features with image features for prediction. To better adapt optical flow to object detection, we insert an additional convolution layer ($3 \times 3 \times 128$ kernels) between optical flow and the fusion module. Compared to the baseline OICR, MC-Net-OptFlow shows better performances on mAP and CorLoc in VID, whereas its performance on CorLoc in DET is slightly lower than the baseline. Our method using the proposed motion correlation module (MCM) to capture motion context information is denoted by MC-Net-MotionContext in Table 4.3. Compared with baseline OICR, MC-Net-MotionContext shows better performance on all mAP (32.9 v.s. 29.8), CorLoc in VID (58.8 v.s. 58.5), and CorLoc in DET (64.7 v.s. 63.3), and it also outperforms MC-Net-OptFlow. This shows that the proposed motion correlation module is more effective for object location compared to directly using optical flow as motion features.

Among these experiments, we fine-tune FlowNet on the ImageNet VID dataset and use optical flow predicted from this fine-tuned FlowNet as motion information. Experiments show that our proposed MC-Net based on this fine-tuned FlowNet effectively improves the detection accuracy. This is also consistent to our conclusion in Chapter 3 that the fine-tuned FlowNet is able to warp features across frames. In theory, a more advanced optical flow method, like FlowNet 2.0 [115] and our proposed feature flow should provide better performances than the fine-tuned FlowNet. We will conduct these experiments in the future.

4.3.4 Comparison with other methods

We compare our MC-Net with other methods in Table 4.4. To our best knowledge, our approach is the first method for video based weakly supervised object detection. Existing WSOD methods are proposed for images only. To conduct the comparison, we re-implement the well-known WSOD method (OICR) to produce results on ImageNet VID. A Fast R-CNN with VGG16 is trained in fully supervised manner, which serves as a performance upper-bound for our method.

Method	Supervision	mAP(%)	VID CorLoc(%)	DET CorLoc(%)
Fast RCNN	Fully	51.1	89.8	93.9
OICR	Weakly	29.8	58.5	63.3
MC-Net-C (ours)	Weakly	34.1	59.9	69.1
MC-Net + PGT (ours)	Weakly	37.4	63.9	73.5

TABLE 4.4: Comparisons with other methods. Fast RCNN denotes a model trained in full supervision, serving as a performance upper-bound. MC-Net + PGT denotes Fast RCNN trained with the pseudo ground truth from our MC-Net. It shows that our MC-Net effectively improves the performance of OICR and shows the best performance under weak supervision on ImageNet VID.

As shown in Table 4.4, The fully supervised method, Fast RCNN with VGG16, obtains 51.1 mAP on ImageNet VID validation set. OICR performs much worse in both mAP and CorLoc, only 29.8 mAP, 58.5 CorLoc in VID and 63.3 CorLoc in DET. Compared to OICR, our MC-Net shows significant better performances on both mAP (34.1%) and CorLoc (59.9% for VID and 69.1% for DET). Furthermore, as discussed in the implementation details, we also train a Fast RCNN with pseudo ground truth generated from our MC-Net, which is denoted as MC-Net + PGT. After using the pseudo ground truth, MC-Net + PGT shows the best performance in the weakly supervised setting Overall, our MC-Net effectively improves the existing image based method by incorporating motion context information and appearance information.

4.3.5 Effectiveness of Motion Context

In this subsection, we aim to improve the accuracy of a detector by incorporating motion context information and appearance information together. To show the

effectiveness of this incorporated information, we visualize some detection results (illustrated in Fig. 4.6), including both rigid and non-rigid objects, and analyze them in details.

In Fig. 4.6, the first to the third columns show the detection results from OICR, the improved results from our MC-Net, and those corresponding optical flows, respectively. In the first row on the left side, OICR wrongly predicts a bounding box including a hamster and a region on its left. In comparison, our MC-Net detects this hamster accurately according to the motion context on the head part of this hamster. Similarly, our MC-Net detects a lizard accurately, since there exists difference between the background and the upper part of this lizard on the optical flow. These examples show that our MC-Net effectively improve the detection results based on motion context. Apart from it, our MC-Net can also incorporate the appearance information in object recognition. In the image including a horse, although the person shows a similar motion to that of the horse, our MC-Net detects the horse accurately by incorporating the appearance information. Overall, all these examples show that our MC-Net effectively incorporates motion context information and appearance information together, yielding a better performance. This also demonstrate that our MC-Net is able to generalize well on both rigid and non-rigid objects.

4.4 Conclusion

We have proposed a novel network, MC-Net for weakly supervised object detection in videos. Under weak supervision, it effectively incorporates appearance information and motion context information from videos, yielding a better performance. Our MC-Net can be trained end-to-end and generalizes well on any object in any motion. Various experimental results demonstrate that our MC-Net considerably improves the performance of existing methods proposed for still images and shows the best results on ImageNet VID.

FIGURE 4.6: Visualization of some detection results. The detection results of OICR (1st row), our MC-Net (2nd column), and the corresponding optical flows (3rd column) are shown in the figure. With the proposed motion context component, our MC-Net effectively improves the detection results of OICR. It shows that our MC-Net effectively incorporates motion context and appearance information to improve the detection accuracy. It generalizes well on both rigid and non-rigid objects.

Chapter 5

Online Active Proposal Set Generation for Weakly Supervised Object Detection

5.1 Introduction

Object detection [40, 41, 45, 46] achieves significant advance benefited from the remarkable progress of Convolutional Neural Network (CNN) [21–23]. However, these object detection approaches which are proposed based on fully supervised learning, requires labor-intensive work on bounding box-level annotations. To alleviate this issue, weakly supervised object detection which only requires image-level annotations easily obtained from image tags, recently receives much attention.

Based on off-the-shelf approaches in full supervision, people propose many approaches [1, 88, 89, 91, 97, 100, 102, 104, 106] for weakly supervised object detection. They formulate the weakly supervised object detection task as a Multiple Instance Learning (MIL) problem, where each image is regarded as a bag and the original proposals in fully supervised methods are regarded as instances. All proposals are involved in the training process and a detector is trained to predict each proposal score.

However, the original proposals in fully supervised approaches are dominated by overwhelming negative samples, which causes the detection methods directly trained

Methods		Supervised Setting	Ratio
Fast R-CNN		Full	1:3
Faster R-CNN	RPN	Full	1:1
	R-CNN	Full	1:3
SSD		Full	1:3
WSOD		Weak	$\approx$ 1:100

TABLE 5.1: The ratio between positive and negative samples in training proposal sets. WSOD is short for weakly supervised object detection methods. Since nearly all weakly supervised object detection use a fixed training proposal set including all proposals, we use WSOD to indicate them. The table shows that this ratio for fully supervised methods are 1:3 or 1:1, while the ratio for weakly supervised methods is unreasonable (around 1:100). This unreasonable ratio may degrade the detection accuracy of weakly supervised detection methods.

on these proposals are easily biased towards negative samples [41]. To mitigate the issue of the imbalance between positive and negative samples, fully supervised approaches like Fast R-CNN[40], Faster R-CNN [41] and SSD [46], utilize box-level ground truths to generate a fixed training proposal set which maintains a reasonable ration (1:1 or 1:3) between positive and negative samples, as listed in Table 5.1. Focal loss [49] also utilizes the box-level annotations to filter out redundant samples and reshapes the loss function for the rest proposals. In contrast, weakly supervised methods lacks box-level annotations. It is challenging for weakly supervised methods to identify positive and negative samples. Nearly all existing methods in weakly supervised learning directly involves all proposals in their training proposal sets, causing this ratio around 1:100, as listed in Table 5.1. This unreasonable ratio (1:100) in the training proposal sets for weakly supervised detection methods may degrade their performances.

To evaluate the impact of this unreasonable ratio on weakly supervised detection accuracy, we firstly design analysis experiments on PASCAL VOC dataset. In Sec. 5.2, we study the ratio influence on the performance of a weakly supervised detection method. Given only image-level annotations, we change the ratio in a training proposal set and evaluate the trained model. These experiments reveal that the ratio between positive and negative samples significantly affects the detection accuracy in weakly supervised detection methods.

To address this problem, we propose an **Online Active Proposal Set Generation** (OPG) algorithm for weakly supervised object detection. Different from the fixed

training proposal set in existing methods, our proposed OPG is able to online generate an active training proposal set according to the training model predictions which are byproducts of the training procedure. To present our OPG clearly, a diagram for active proposal sets generated from our OPG during training process is illustrated in Fig.5.1, where circle points and rectangles points indicate negative and positive proposals, respectively.

FIGURE 5.1: Diagram of the active training proposal set from our **Online Active Proposal Set Generation (OPG)** during a training process. Circle and rectangle points indicate negative and positive samples, respectively. Our active training proposal set is initialized with the fixed set in weakly supervised detection methods. During the training process, our OPG online generates the active proposal set according to the predictions of the training model. The ratio between positive and negative samples in our active set gradually becomes similar to that of fully supervised methods, which mitigate the issue arised by the imbalance between positive and negative samples, improving the detection accuracy.

Our active training proposal set is initialized with the original training proposal set which is the same as the fixed training proposal set in weakly supervised object detection methods. During the training procedure, our OPG utilize the predictions from the training model to online generate an active training proposal set. The issue caused by overwhelming negative samples is gradually alleviated. Finally, the positive and negative samples in our generated active set is balanced, which is similar to that of fully supervised methods. Experimental results show that our proposed OPG is able to train a high-accuracy detector which outperforms existing weakly supervised methods, showing new state-of-the-arts performances.

Recently, methods in [1, 106] proposes various region proposal networks (RPNs) to generate proposals with high quality. However, these methods generally requires a complex training procedures or an external dataset. In comparison, our proposed OPG does not need any extra operations. It can be easily applied to any

weakly supervised object detection method and shows better improvement than these methods [1, 106].

Some people may mix up our proposed method with other weakly supervised object detection methods [1, 100, 103, 116]. Although [116] and [100] try to filter out background samples, they need the segmentation results to filter out these samples. This makes these methods computationally expensive. Additionally, segmentation results in [100, 116] are semantic level, not instance level. However, object detection requires instance-level results. Training based on these segmentation results may limit their performances. Methods [100, 116] are more complex and inefficient than our method.

Our OPG is also different from methods [1, 103]. Although methods [1, 103] assign 0 weight for samples with low IOU value, they do not limit the number of training proposals. They simply set a fixed threshold value to filter out some samples. However, the distribution of IOU among samples depends on images. It is difficult to use a fixed threshold to filter out most negative samples. To training stably, training process in [1, 103] still involves many negative samples. By contrast, our methods limit the number of proposals according to different training phases (DPC). Additionally, our method filter out samples by proposal ranking (PR). These enable our method to filter out samples effectively and achieves a better performance than them.

Compared with weakly supervised object detection, there are some existing methods [49, 62, 65] aiming to optimize the training process under full supervision. OHEM [65] sorts all proposal samples by computing loss based on box-level annotations and selects a certain number of samples as the hard examples for training. In RetinaNet [49], it firstly filters out some samples according to their overlaps with the box-level annotations and focuses on training the retained samples with large loss. In the method [62], it tries to solve this problem by computing the overlaps between these samples and the box-level ground truths. Among these methods, the box-level annotations are necessary components. Since box-level annotations are not available in weakly supervised object detection, it is challenging to apply these methods to the weakly supervised object detection task. To overcome this issue, we propose an online active proposal set generation algorithm without the needing of box-level annotations.

Overall, our contribution can be summarized as follows:

- We conduct a series of analysis experiments on the influence caused by the imbalance between positive and negative samples on detection performance. These experiments shows that this imbalance affects the performance of weakly supervised object detection methods.
- We propose an Online Active Proposal Set Generation (OPG) algorithm for weakly supervised object detection. It is able to generate an active propose set according to the prediction of the training model and prevents the training model from the overwhelming negative samples. It effectively alleviates the issue caused by the imbalance between positive and negative samples, significantly improving the detection accuracy.
- Extensive experiments are carried out on PASCAL VOC 2007 and 2012. Our OPG shows effective improvement on the performance of our baseline, setting a performance comparable to the state-of-the-arts.

5.2 Influence of Proposal Set

Without box-level annotations, it is difficult for weakly supervised object detection methods to remove the overwhelming negative samples from the training proposal set. Training on this proposal set dominated by negative samples, the performance of these methods may be degenerated. In this section, we investigate the impact of the ratio between positive and negative samples on detection accuracy.

Our controlled experiments are conducted based on a basic detection framework [88] as listed in Fig. 5.2. Following WSDDN [88], the proposal features are connected with two layers named FC-8d and FC-8c. The outputs of these two layers, $\mathbf{x}^c$ and $\mathbf{x}^d$ are forwarded to two softmax functions, respectively: $\delta_c(\mathbf{x}^c)(i, j) = \frac{e^{x_{i,j}^c}}{\sum_{m=1}^C e^{x_{m,j}^c}}$, and $\delta_d(\mathbf{x}^d)(i, j) = \frac{e^{x_{i,j}^d}}{\sum_{m=1}^{|R_r|} e^{x_{i,m}^d}}$, where $|R_r|$ expresses the total number of the actually forwarded proposals and C denotes the number of classes. The final score for each proposal is computed by an element-wise production: $\mathbf{x}^R = \delta_c(\mathbf{x}^c) \odot \delta_d(\mathbf{x}^d)$. Finally, the proposal score metric at c th class are converted to image score at c th class: $p_c = \sum_{k=1}^{|R_r|} x_{ck}^R$. We train this network using a

FIGURE 5.2: Illustration of the basic detection framework under weak supervision. To investigate the influence of the ratio between positive and negative samples, we firstly train a network with only blue parts under weak supervision to generate pseudo ground truth. Then, we train the whole network including blue and yellow parts without sharing the parameters of the trained network above. During the second training process, we control the ratio between positive and negative samples according to pseudo ground truths. The original proposal set is denoted as R and the filtered set is indicated by R_f .

multi-class cross entropy loss L_{base} , as listed in Eq. 5.1.

$$L_{base} = -\frac{1}{|C|} \sum_{c=1}^C [y_c \ln p_c + (1 - y_c) \ln(1 - p_c)], \quad (5.1)$$

where y_c is 1 or 0 indicates whether the image includes class c and C is the number of classes.

To control the ratio between positive and negative samples in the proposal set, we firstly train the network with only blue parts as shown in Fig. 5.2 under weak supervision, obtaining the pseudo ground truths. After that, without sharing parameters with the trained model above, we train new networks including yellow parts and change the ratio according to the generated pseudo ground truths for following experiments. We apply the proposal sampling strategy of Fast R-CNN [40] to generate training proposal sets $\mathcal{P}_r$:

- *Step1*: Divide the original proposal set $\mathcal{P}$ into a positive set $\mathcal{P}_f$ and a negative set $\mathcal{P}_b$ according to the generated pseudo ground truths. The proposals with $\text{IOU} \geq 0.5$ are regarded positive samples, and others are negative.
- *Step2*: Compute the number of selected positive and negative samples according to total selected samples number N_r and the selected positive sample fraction in the total selected sample r_p : $N_p = r_p * N_r$ and $N_n = (1 - r_p) * N_r$.
- *Step3*: Randomly select N_p positive samples and N_n negative samples from $\mathcal{P}_f$ and $\mathcal{P}_b$, respectively.

- *Step4*: Generally, $|\mathcal{P}_f| < N_p$. Therefore, we randomly select $N_p - |\mathcal{P}_f|$ negative samples from $\mathcal{P}_b$ and add these negative samples to $\mathcal{P}_r$.

Based on this strategy, when N_r is small, the ratio between positive and negative samples is equal to $r_p : 1 - r_p$. However, when N_r becomes large, like 512, $|\mathcal{P}_f|$ will be much smaller than N_p . Affected by the step 4 of this strategy, $N_p - |\mathcal{P}_f|$ is very large and the training proposal set $\mathcal{P}_r$ will be dominated by negative samples.

In this section, we conduct seven experiments under different N_r values: 128, 256, 384, 512, 768, 1024, and *all*. When N_r is small, like 128, 256 and 384, the ratio between N_p and N_n in the generated fixed training proposal set is equal to $r_p : 1 - r_p$. With N_r increasing, the training proposal set $\mathcal{P}_r$ will be dominated by negative samples. When we select *all* samples from the set $\mathcal{P}$, the fixed training proposal set is the same as that of existing weakly supervised object detection methods. The performances of the trained models under different N_r values are plot in Fig. 5.3.

FIGURE 5.3: The performances of the trained models under different N_r values. It shows that the performance of the trained model is significantly affected by the ratio between positive and negative samples.

From Fig. 5.3, it can be seen that the performance of the trained model is improved when N_r increases from 128 to 384. When N_r continues to increase, the performances is gradually degraded. It is because the ratio between positive and negative samples in the fixed training proposal set is around $r_p : 1 - r_p$, when N_r

is between 128 and 384. For larger N_p , $|\mathcal{P}_f| < N_p$ and $\mathcal{P}_r$ is dominated by negative samples, degenerating the detection accuracy. This experiment reveals that the ratio between positive and negative samples in a fixed training proposal set significantly affects the training process and hinders the performances of weakly supervised object detection methods.

5.3 Online Active Proposal Set Generation

According to the experiments in Sec. 5.2, we find that the ratio between positive and negative samples hinders the detection performance for weakly supervised object detection methods. To address this problem, we propose an **Online Active Proposal Set Generation** (OPG) algorithm. Different from the fixed training proposal set in most approaches, our OPG is able to online generates an active training proposal set according to the prediction of the training network.

During the training process, a simple method, which generates training proposal set, is to use the top-scoring proposal as a pseudo ground truth to separate other proposals into positive and negative samples. The proposals which overlaps with this pseudo ground truth than a threshold value, are regarded as positive samples and the others are regarded as negative samples. However, the prediction of the training network is not accurate. This simple method will filter out many false negative samples, damaging the training process.

The detection accuracy of a training network is gradually improved during the training process. According to detection performance of the training network, we divide the training process into three stages: warm up stage, transition stage and stable stage. Instead of using the simple method, we propose a strategy of proposal set generation which adapts well to different training stages. During the training process, we expect our active proposal set involves enough proposals at the warm up stage, providing a solid foundation for the performance improvement at the following stages. At the transition stage, our OPG gradually decreases the size of our active proposal set. For the stable stage, our OPG algorithm effectively removes the overwhelming negative proposals and prevent the training model from biasing towards negative samples. Our OPG consists of two components: dynamic proposal constraint (DPC) and proposal ranking (PR). DPC is able to limit the proposal

number in our active proposal set for different training stages. PR is responsible to measure the ranking confidence for each proposal and add the proposals with high confidences to our active proposal set.

5.3.1 Dynamic Proposal Constraint

In our OPG algorithm, we expect the active training proposal set includes different number of proposals in different training stages. To control the number of the proposals in our active set, we propose dynamic proposal constraint (DPC), which is defined as:

$$\gamma(i) = \frac{1}{1 + e^{\alpha*(i/T-\beta)}}, \quad (5.2)$$

where i is the iteration number, T serves as a scaling parameter, α and β are constant value. Then, the proposal number in our active training proposal set at i th iteration, $N_v(i)$ is defined as:

$$N_v(i) = \max(\gamma(i) * |\mathcal{P}|, N_{min}), \quad (5.3)$$

where $|\mathcal{P}|$ denotes the total number of original proposals, N_{min} is a constant value.

After employing this DPC, the $N_v(i)$ during training process is visualized in Fig. 5.4. From the Fig. 5.4, we can find that the number of proposal varies during the training process. At the warp up stage, nearly all proposal are involved in the training proposal set. The number of proposal at the transition stage gradually decreases. Finally, the number of proposals at the stable proposal decreases the minimum value, removing the overwhelming negative samples.

5.3.2 Proposal Ranking

Given a pre-defined proposal number by our DPC, a proposal ranking (PR) method is proposed to evaluate each proposal ranking confidence and add the proposals with high confidence to our active proposal set.

Suppose the image-level annotation for an image is $\mathbf{y} = [y_1, y_2, \dots, y_C]^T \in \mathbb{R}^{C \times 1}$, where y_c is 1 or 0 denotes that whether this image includes the class c . For each training iteration, we firstly choose the prediction with highest score at class c as a

FIGURE 5.4: Visualization of the proposal number in our active training proposal set during the training process. It can be seen that nearly all proposals are involved at the warm up stage, which provides a solid foundation for the performance improvement as the following stage. At the transition stage, the number of proposals gradually decreases. At the stable stage, only few proposals are involved, which prevents the training model from biasing to negative samples.

pseudo ground truth g_c , if its corresponding annotation y_c is 1. Then, we compute the compute the ranking confidence rc_i for each proposal b_i according to Eq 5.4:

$$rc_i = \arg \max_c \text{IOU}(g_c, b_i), \quad (5.4)$$

where we choose the overlap values with pseudo ground truths as the ranking confidences. After obtaining the ranking confidence, we generate the active proposal set $\mathcal{P}_a$ according to the pseudo ground truth and the ranking confidences:

- *Step1*: Separate original proposal set $\mathcal{P}$ into a positive set $\mathcal{P}_f$, a negative set $\mathcal{P}_b$ and an unknown set $\mathcal{P}_u$ according to Eq. 5.5:

$$l_i = \begin{cases} 1 & rc_i \geq \theta_f \\ 0 & \theta_{b1} > rc_i \geq \theta_{b2}, \\ -1 & \text{others} \end{cases} \quad (5.5)$$

where l_i denotes the proposal b_i label, -1 indicates unknown. θ_f , θ_{b1} and θ_{b2} are constants.

- *Step2*: Sort positive and negative samples in $\mathcal{P}_f$ and $\mathcal{P}_b$ according to ranking confidence, respectively.
- *Step3*: Compute the number of positive and negative samples according to N_v and the positive sample fraction r_p in $\mathcal{P}_a$: $N_p = r_p * N_v$ and $N_n = (1 - r_p) * N_v$.
- *Step4*: Choose top N_p positive and top N_n negative samples from $\mathcal{P}_f$ and $\mathcal{P}_b$.
- *Step5*: If $|\mathcal{P}_f| + |\mathcal{P}_b| < N_v$, randomly choose $N_v - (|\mathcal{P}_f| + |\mathcal{P}_b|)$ proposals from the left samples.

After applying our OPG algorithm, the imbalance between positive and negative samples is effectively solved. The performances of weakly supervised object detection methods are significantly improved.

5.3.3 Detection Framework

In this work, we adopt the OICR [89] architecture as our detection framework which is illustrated in Fig. 5.5.

Our detection framework consists of four branches: one basic branch and three OICR branches. The basic branch is the same as FC layers in Fig. 5.2, which has been presented in Sec. 5.2. For the OICR branches, it regards the predictions with highest score as pseudo ground truths and divide proposal into positive and negative samples, which is similar the step 1 in our proposal ranking (PR). The only different point is that the OICR branch merges the unknown proposals in Eq. 5.5 with the negative samples together. Given the predicted scores of the pseudo ground truths w_{ik} for the i th proposal at the k th OICR branch, the loss function L_r^k at the k th OICR branch is defined as:

$$L_r^k = -\frac{1}{|R|} \sum_{i=1}^R \sum_{c=1}^{C+1} w_{ik} y_{ik}^c \ln x_{ik}^c, \quad (5.6)$$

where y_{ik}^c is the class c label for the i th proposal at the k th OICR branch.

In the Figure 5.5, we firstly generate the pseudo ground truths, PGT-1 according to the prediction of the basic branch and pass PGT-1 to OICR-1. Simultaneously, our OPG-1 generates the active training proposal set for the current training stage

FIGURE 5.5: Illustration of our weakly supervised object detection framework. The basic branch is the same as the FC layers of the network in Sec. 5.2. We firstly forward images and corresponding proposals to the basic branch. Then, our OPG-1 generates an active proposal set according to the predictions from the basic branch. The pseudo ground truth, PGT-1 and the active proposal set from OPG-1 are passed to the OICR branch, OICR-1. We repeat this process by 2 iterations. The predictions of these three OICR branches are averaged to obtained the final results.

and pass this proposal set the OICR-1. We repeat these operations iteratively for the following two OICR branches. Finally, the prediction from three OICR branches are averaged to obtain the final results. The loss function for our detection framework is defined as:

$$L = L_{base} + \sum_{m=1}^M L_r^m, \quad (5.7)$$

where M is 3.

5.4 Experiments

In this section, we present experiments to verify the effectiveness of our method by comparing with the baseline and other state-of-the-art methods.

5.4.1 Experiment Settings

Datasets and evaluation metrics Following the protocol of previous methods [89, 97], our methods are evaluated on the benchmark PASCAL VOC 2007 and 2012 datasets [3]. There are 9,962 and 22,531 images in PASCAL VOC 2007 and 2012, respectively. We choose the trainval set as our training dataset and use the testing set as our testing dataset. There are 5,011 and 11,540 images in PASCAL VOC 2007 and 2012 trainval set, respectively.

Two metrics mean average precision (mAP) and correct localization (CorLoc) metrics are used to measure the performance of our method. Following the protocol of PASCAL VOC [3], mAP is used to measure our method performance on the testing set. CorLoc is to evaluate our method performance on the training set. The threshold value for IOU in both evaluation metrics is 0.5.

Implementation Details The network backbone in our experiments is VGG16 [21] which is pre-trained on ImageNet [87] dataset. We replace the last max-pooling layer, *pool5* with a ROI-pooling layer [40]. To improve the resolution of feature map, we remove the penultimate pooling layer, *pool4* and increase the dilation rate on the conv5 layers from 1 to 2. The FC layers with 4096 hidden nodes are initialized by the FC layers in VGG16 network. We initialize the newly added layers in random. For training, SGD optimization method is performed and the mini-batch size is 2. The learning rate for the first 40K iterations is set to 0.001 and then decreases to 0.0001 for the following 55K iterations. The momentum is set to 0.9 and the weight decay is 0.0005.

According to previous methods [89, 102], multi-scale training and testing are adopted in our experiments. we use five images scales and resize the shorter side of an image to one of these sizes: 480, 576, 688, 864, and 1200. The horizontal flip is also used for data argumentation. Selective search [38] is applied to generate the original proposals. For the hyper-parameters in OICR, we follow the original setting in

OICR [89]. For hyper-parameters in OPG, r_p is 0.25, T is 140,000 and N_{min} is set to 384. θ_f and θ_{b1} are 0.5 and θ_{b2} is 0.1.

5.4.2 Analysis of OPG Algorithm

In this subsection, we discuss the experiments for various α and β in 5.2 for our proposed online active proposal set generation (OPG) algorithm. We evaluate mAP on PASCAL VOC 2007 test set and CorLoc on PASCAL VOC 2007 trainval set. For convenience, we only apply multi-scale in training and resize the shorter size of an image to a single size, 1200 in testing.

FIGURE 5.6: Visualization of the proposal number in our active training proposal set under different α values. It shows that α controls the occupation of the transition stage in the training process.

The $N_v(t)$ for different α values are visualized in Fig. 5.6. In this chapter, we divide the training process into three stages: warm up, transition and stable stage. From the Fig. 5.6, it can be seen that α controls the occupation of the transition stage in the training process. When α is small, the number of proposals decreases slowly and many epochs are occupied by the transition stage in the training process. When α becomes large, the occupation of warm up and stable stage in the training process increase. Results for various α are listed in Table 5.2, where β is set to 0.8 for all experiments. It shows that nearly all models trained with OPG perform better than our baseline. When α is 10.0, the model trained with OPG yields the best performance.

Method	α	mAP	CorLoc
Baseline (OICR)	-	36.7	61.4
Baseline+OPG (ours)	8.0	45.9	63.7
Baseline+OPG (ours)	9.0	45.8	63.4
Baseline+OPG (ours)	10.0	46.0	64.1
Baseline+OPG (ours)	11.0	44.6	61.2
Baseline+OPG (ours)	12.0	45.5	63.1

TABLE 5.2: Experiments for various α on PASCAL VOC 2007 dataset, where β is fixed at 0.8. Multi-scale training is only applied in training process. It shows that nearly all model trained with our OPG show better performances than our baseline. The best performance is obtained when α is 10.0.

FIGURE 5.7: Visualization of the proposal number in our active training proposal set under different β values. It can be seen that β changes the ratio between the epochs in the warm up stage and that in the stable stage.

We also conduct another series of experiments for various β in our proposed OPG and the corresponding $N_v(t)$ is plot in Fig. 5.7. In Fig. 5.7, it shows that the β changes the ratio between the epochs in the warm up stage and that in the stable stage. When β is small, only few epochs are involved in the warm up stage and many epochs are assigned to the stable stage. However, with β increasing, more epochs are assigned to the warm up stage and less epochs are at the stable epochs. The results for different β are shown in Table 5.3, where α is fixed to 10.0. From the Table 5.3, it demonstrates that our OPG effectively improves the performance of the baseline. When β is equal to 0.8, the model trained with our OPG shows the best performance.

Method	β	mAP	CorLoc
Baseline (OICR)	-	36.7	61.4
Baseline+OPG (ours)	0.6	43.9	63.9
Baseline+OPG (ours)	0.7	45.0	63.0
Baseline+OPG (ours)	0.8	46.0	64.1
Baseline+OPG (ours)	0.9	45.4	63.7
Baseline+OPG (ours)	1.0	42.6	62.5

TABLE 5.3: Experiments for various β on PASCAL VOC 2007 dataset, where α is fixed at 10.0. It shows that all model trained with our OPG show better performances than our baseline. The best performance is obtained when β is 0.8.

According to the analysis experiments, it shows that our OPG can effectively improve the performance of our baseline by online generating an active proposal set for training. The performance of our OPG is robust to the variation of α and β .

Actually, our OPG algorithm is proposed based on the proposal selection schedule of Faster R-CNN. Compared with Faster R-CNN, there are only 4 more hyper-parameters: α , β , T and N_{min} . Our proposed method depends on the curve property in the Figure 5.4. These hyper-parameters are used to formulate a curve with a certain property. Through experiments, we find that if this curve makes the warm up stage occupying around 40% in the training process and the stable stage occupying around 50% in the training process, our proposed method will significantly improves the performance. This curve property is easily realized by adjusting these parameters. In the following experiments, α and β in 5.2 are 10.0 and 0.8, respectively.

5.4.3 Effectiveness of OPG Algorithm

In this subsection, we conduct experiments to verify the effectiveness of our proposed OPG algorithm. As discussed in Sec. 5.3.3, we directly apply our OPG algorithm to OICR [89] and use OICR as our baseline. We firstly re-implement OICR in MX-Net. The OICR implemented by ourselves is trained under the same hyper-parameter of our OPG+OICR and performs a little better than the original OICR. We apply our OPG on this re-implemented OICR. Multi-scale testing is applied in the following experiments. We compare the performance of OICR and

our detection method OICR + OPG with mAP and CorLoc in Table 5.4 and Table 5.5, respectively.

Method	aero	bike	bird	boat	bottl	bus	car	cat	chair	cow	mean
Baseline (OICR)	55.7	63.1	37.4	19.4	17.2	64.7	65.9	43.9	27.7	48.7	
Baseline+OPG (ours)	63.0	65.3	49.2	31.7	25.3	70.9	70.9	58.1	27.4	58.6	
Method	table	dog	horse	mbike	per	plant	sheep	sofa	train	tv	
Baseline (OICR)	35.5	35.8	40.6	61.4	8.9	24.2	40.6	46.9	62.3	64.2	43.2
Baseline+OPG (ours)	44.7	47.0	47.2	69.8	13.1	26.1	49.9	51.8	61.7	68.2	50.0

TABLE 5.4: Compare our OPG method with the baseline in *mAP* on PASCAL VOC 2007 *test* set. After employing our OPG, the performance of our method surpasses the baseline in nearly all classes. It shows that our OPG algorithm significantly improves the performance of the baseline in mAP.

Method	aero	bike	bird	boat	bottl	bus	car	cat	chair	cow	mean
Baseline (OICR)	82.1	83.9	64.9	49.5	43.5	81.2	85.9	56.4	44.1	75.3	
Baseline+OPG (ours)	83.3	77.7	64.0	50.0	43.1	80.7	87.5	61.6	40.7	79.5	
Method	table	dog	horse	mbike	per	plant	sheep	sofa	train	tv	
Baseline (OICR)	43.7	53.3	61.6	88.4	14.7	57.9	78.4	57.5	77.6	79.6	64.0
Baseline+OPG (ours)	43.4	60.0	71.4	90.8	15.9	58.6	76.3	62.6	74.9	83.2	65.3

TABLE 5.5: Compare our OPG method with the baseline in *CorLoc* on PASCAL VOC 2007 *trainval* set. After employing our OPG, the performance is improved. It shows that our OPG algorithm effectively improves the performance of the baseline in CorLoc.

In Table 5.4, it shows that our method (OICR+OPG) performs significantly better than the baseline OICR (50.0 v.s 43.2) in mAP. Trained with our proposed OPG algorithm, OICR+OPG surpasses OICR in nearly all classes. Similarly, in the Table 5.5, our OPG algorithm effectively improves the performance of the baseline (65.3 v.s. 64.0) in CorLoc. This demonstrates that our OPG effectively mitigates the issue which is caused by the imbalance of positive and negative samples in the training proposal set.

The methods in [1, 106] propose RPNs to improve the quality of proposals. To show the effectiveness of our OPG algorithm, we compare our method performance with these two methods in Table 5.6. WeakRPN[1] requires training model three times in an alternative way, which is time-consuming and complex. W-RPN[106] needs an additional video dataset to train their RPN, which requires an heavy engineering process. Compared with them, our OPG only uses the prediction of the training network which is a byproduct in the training process. Our OPG can be directly applied to any weakly supervised object method without adding much extra computation. Our OPG is more efficient than other two methods.

Dataset	VOC 2007		VOC 2012	
Method	mAP	CorLoc	mAP	CorLoc
WeakRPN[1]	45.3	63.8	40.8	64.9
W-RPN [106]	46.9	66.5	43.2	67.5
OPG (ours)	50.0	65.3	46.2	65.8

TABLE 5.6: Comparison with other methods proposed for improve proposal quality in PASCAL VOC 2007 and VOC 2012 dataset. It shows that our OPG performs much better than other two methods in mAP. For CorLoc, our OPG shows a comparable performance to W-RPN and a better performance than WeakRPN[1].

The methods in [1, 106] propose RPNs to improve the quality of proposals. To show the effectiveness of our OPG algorithm, we compare our method performance with these two methods in Table 5.6. WeakRPN[1] requires training model three times in an alternative way, which is time-consuming and complex. W-RPN[106] needs an additional video dataset to train their RPN, which requires an heavy engineering process. Compared with them, our OPG only uses the prediction of the training network which is a byproduct in the training process. Our OPG can be directly applied to any weakly supervised object method without adding much extra computation. Our OPG is more efficient than other two methods.

From the Table 5.6, it can be seen that our OPG performs significantly better than other two methods in mAP. For CorLoc, our OPG also shows a comparable performance to W-RPN and a better performance than WeakRPN[1]. This demonstrates that our OPG is also more effective than these two methods in performance improvement. Overall, our OPG algorithm is more effective and more efficient then these two methods.

Both methods also apply the pseudo ground truths (PGT) which are generated by the trained model under weak supervision, to train a Fast R-CNN for better performances. We also carry out the Fast RCNN experiment with PGT, but we find that the improvement is limited. The reason for it may be that other weakly supervised methods do not filter out the redundant negative samples in their training procedure. However, they use the PGT to remove many negative samples when they train a Fast RCNN under full supervision. The reason why PGT is effective is that many negative samples are removed during the experiment with PGT.

Since our proposed method OPG has already removed these redundant negative samples during training, the PGT experiment cannot show much improvement. Actually, other methods are composed of two steps: train under weak supervision and train with PGT. In comparison, our proposed method is able to combine these two steps into one step, which shows that our proposed OPG algorithm is more efficient than other methods. Additionally, our proposed OPG algorithm is more effective than their two-stage methods. OICR + PGT show 47.0 mAP on VOC 2007 and 42.5 mAP on VOC 2012. Compared with it, we directly apply our OPG on OICR, which obtains 50.0 mAP on VOC 2007 and 46.2 mAP on VOC 2012.

Overall, our proposed method OPG not only simplifies the training procedure of weakly supervised object detection methods, but also improves their performances. The techniques proposed by other methods may further improve the performance. Since their source codes are not open, it is difficult to apply our OPG algorithm to their methods. We will try to re-implement these methods in the future.

5.4.4 Comparison with the state-of-the-art

In this subsection, we compare our experimental results with other state-of-the-arts in the Table 5.7.

Dataset	VOC 2007		VOC 2012	
Method	mAP	CorLoc	mAP	CorLoc
TS ² C[100]	44.3	61.0	40.0	64.4
WeakRPN[1]	45.3	63.8	40.8	64.9
W-RPN [106]	46.9	<u>66.5</u>	43.2	67.5
WS-JDS [104]	45.6	64.5	39.1	63.5
CL [117]	48.3	64.7	43.3	65.2
C-MIL[102]	50.5	65.0	46.7	<u>67.4</u>
Satoshi et al.[103]	47.6	66.7	43.4	66.7
OICR + OPG (ours)	<u>50.0</u>	65.3	<u>46.2</u>	65.8

TABLE 5.7: Comparison with other methods. It demonstrates that our method (OICR+OPG) shows a comparable performance to other state-of-the-art methods. Most listed methods are trained in a fixed training proposal set. Our OPG can be applied to these listed methods to further improve their performances.

In Table 5.7, our method OICR+OPG shows a comparable performance to the state-of-the-art methods. For mAP, our method shows a comparable performance,

FIGURE 5.8: Visualization of our detection results.

which is comparable to the best performance in VOC 2007 (50.0 v.s. 50.5) and VOC 2012 (46.2 v.s. 46.7). Our method even surpasses the approaches [100, 103, 104] which adopts the segmentation masks in training process.

Recently, C-MIDN [105] shows an excellent performance in both VOC 2007 (52.6 mAP and 68.7 CorLoc) and VOC 2012 (50.2 mAP and 71.2 CorLoc). But it benefits a very strong baseline which obtains 49.0 mAP and 46.8 CorLoc in VOC 2007. This method jointly trains an weakly segmentation network with their detection network and refines the detection results by iterations. Compared to its baseline, the effective improvement is limited. Our OPG aims to improve the detection accuracy by removing the redundant negative proposals, which is not conflict to C-MIDN [105] in terms of technique. By employing our OPG to C-MIDN, we expect to show a better performance. Some detection results are visualized in Fig. 5.8.

5.5 Conclusion

In this Chapter, we have conducted extensive experiments which reveals that the ratio between positive and negative samples substantially affects the detection performance. The overwhelming negative samples will bias the training process. To solve this problem, we propose an Online Active Proposal Set Generation (OPG) algorithm, which is able to online generate an active proposal set according to the predictions of the training network during the training process. Compared with previous methods which improves the quality of proposals, our OPG is more efficient and more effective. Our OPG algorithm can be directly applied to other weakly supervised detection methods, without adding additional computation. Experimental results show that our OPG algorithm significantly improves the performance of the baseline and obtains a performance comparable to the state-of-the-art on PASCAL VOC 2007 and VOC 2012 dataset.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, we present our work on video object detection. Video object detection aims to explore temporal information to improve detection accuracy in videos. Existing video object detection methods are composed of two parts: temporal feature aggregation and object detection. Most methods are proposed for full supervision. However, a large-scale video detection dataset with box-level annotations are labor-intensive and time consuming. To solve these problem, three works are presented in this thesis.

In Chapter 3, we propose a novel detection network (IFF-Net) with an in-network feature flow estimation module for video object detection. Existing methods in video object detection requires a dataset with pixel-level annotation to train a FlowNet to predict optical flow for temporal feature aggregation. Different from these methods, our IFF-Net is able to predict feature flow to align features across frames by self supervised training. Without the needing of any flow dataset, our IFF-Net outperforms those optical flow based methods, while remaining a fast inference speed. Among all video object detection methods, our IFF-Net yields a comparable performance to the state-of-the-arts methods.

In Chapter 4, a motion context network (MC-Net) is proposed under weak supervision. Existing video object detection methods are proposed under full supervision.

These methods require box-level annotations for each frame, which poses a challenge to construct such a large-scale video detection dataset. To solve this issue, the MC-Net is proposed under weak supervision. Our proposed MC-Net does not require box-level annotations. Furthermore, different from existing weakly supervised methods proposed for static images, our MC-Net is able to directly capture motion information of objects, which effectively improves the detection performance of static image based baseline in weak supervision.

In Chapter 5, an online active proposal set generation (OPG) algorithm is proposed to further improve the detection accuracy under weak supervision. Affected by the overwhelming negative samples in the fixed proposal set, most weakly supervised object detection performances are degenerated. To solve this problem, we propose an OPG algorithm, which online generate an active proposal set according to the predictions of the training network. With this method, the imbalance effect is effectively mitigated. The performance of the baseline is significantly improved, yielding a comparable performance to the state-of-the-arts performances.

6.2 Future Works

Three works are presented in this thesis for video object detection. Although effective improvements are achieved in these methods, there are still some limitations to solve in the future.

In the IFF-Net presented in Chapter 3, an IFF module is proposed to predict feature flow to align features across frames. However, IFF module is simply trained via a self-supervision manner (TRL function), where occlusion and large motion are not considered. In optical flow approximation, occlusion and large motion often affects the accuracy of optical flow estimation. Similar to optical flow, feature flow is also influenced by these two factors. In the future, we will propose a new IFF module with an improved TRL function to alleviate these issues, expecting to show a better performance.

Apart from it, in aligned feature fusion part, existing video object methods combine features based on similarity. However, the temporal offset of a fused frame relative to the current frame is neglected, which provides an important information for temporal feature aggregation. Temporal feature fusion should consider two

factors: feature similarity and temporal offset. In our future work, we will propose a more effective temporal feature aggregation method based on these two factors and improves the detection accuracy.

For weakly supervised object detection, our MC-Net presented in Chapter 4 aims to utilize motion context information to improve the detection performance. However, this method is limited by the quality of fused motion information. Our MC-Net leverages motion information in a fixed temporal internal. If the moving regions in objects are not captured in this fixed temporal internal, our MC-Net cannot effectively improve the detection accuracy. In the future, we will propose a module, which is able to utilize motion information in a dynamically temporal internal. According to the current object motion state, the proposed module estimates a temporal internal including moving regions of objects and uses the motion information in this predicted temporal internal to improve the detection performance. With this method, the detection performance is improved.

Author's Publications

Journal Papers

- **Ruibing Jin**, Guosheng Lin, Changyun Wen, Fayao Liu, and Jianliang Wang, “Feature Flow: Innetwork Feature Flow Estimation for Video Object Detection” in *IEEE Transactions on Multimedia* (submitted).
- **Ruibing Jin**, Guosheng Lin, Changyun Wen, and Jianliang Wang, “Motion Context Network for Weakly Supervised Object Detection in Videos” in *IEEE Signal Processing Letters*.
- **Ruibing Jin**, Guosheng Lin, and Changyun Wen, “Online Active Proposal Set Generation for Weakly Supervised Object Detection” in *IEEE Transactions on Circuits and Systems for Video Technology* (submitted).

Conference Papers

- **Ruibing Jin**, and Jianliang Wang, “Two-stream Network for Online Quadrotor Detection without Dedicated Annotations” in *International Conference on Control and Automation (ICCA 2019)*.
- **Ruibing Jin**, Bojan Andonovski, Zhigang Tu, Jianliang Wang, Junsong Yuan, and Desmond Mark Tham, “A Framework Based on Deep Learning and Mathematical Morphology for Cabin Door Detection in an Automated Aerobridge Docking System” in *Asian Control Conference (AsCC 2017)*.

Bibliography

- [1] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 352–368, 2018. [xxix](#), [73](#), [75](#), [76](#), [89](#), [90](#), [91](#)
- [2] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. [7](#)
- [3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010. [8](#), [25](#), [26](#), [31](#), [55](#), [85](#)
- [4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. [8](#), [25](#), [26](#), [32](#), [55](#)
- [5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. [8](#)
- [6] Guangzheng Yang and Thomas S Huang. Human face detection in a complex background. *Pattern recognition*, 27(1):53–63, 1994. [8](#)
- [7] Ian Craw, David Tock, and Alan Bennett. Finding face features. In *European Conference on Computer Vision*, pages 92–96. Springer, 1992. [8](#)
- [8] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. [8](#)
- [9] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1(511-518):3, 2001. [8](#)
- [10] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. [8](#), [9](#)
- [11] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 577–584, 2003. [8](#)

- [12] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780): 1612, 1999. [8](#)
- [13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. [9](#)
- [14] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. [9](#)
- [15] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2241–2248. IEEE, 2010. [9](#)
- [16] Tomasz Malisiewicz, Abhinav Gupta, and Alexei Efros. Ensemble of exemplar-svms for object detection and beyond. 2011. [9](#)
- [17] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. [9](#)
- [18] Ross B Girshick, Pedro F Felzenszwalb, and David A Mcallester. Object detection with grammar models. In *Advances in Neural Information Processing Systems*, pages 442–450, 2011. [9](#)
- [19] Régis Vaillant, Christophe Monrocq, and Yann Le Cun. Original approach for the localisation of objects in images. *IEE Proceedings-Vision, Image and Signal Processing*, 141(4):245–250, 1994. [10](#)
- [20] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [10](#)
- [21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [10](#), [55](#), [65](#), [73](#), [85](#)
- [22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. [10](#)
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [11](#), [42](#), [45](#), [49](#), [55](#), [73](#)

- [24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. [10](#)
- [25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. [10](#), [11](#)
- [26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. [10](#)
- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [10](#)
- [28] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. [10](#)
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. [11](#)
- [30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. [11](#), [24](#)
- [31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019. [11](#)
- [33] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. [11](#)

- [34] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. [11](#)
- [35] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. [11](#)
- [36] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. [11](#), [16](#)
- [37] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014. [12](#), [25](#)
- [38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. [12](#), [26](#), [85](#)
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. [12](#)
- [40] R. Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision*, 2015. [12](#), [25](#), [66](#), [73](#), [74](#), [78](#), [85](#)
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. [12](#), [16](#), [25](#), [55](#), [73](#), [74](#)
- [42] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. [14](#)
- [43] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, 2016. [14](#), [25](#), [45](#), [49](#)
- [44] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations*, 2014. [15](#)
- [45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [15](#), [55](#), [73](#)

- [46] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016. [15](#), [55](#), [73](#), [74](#)
- [47] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [16](#)
- [48] Joseph Redmon and Ali Farhadi. Yolo3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. [16](#)
- [49] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *The IEEE International Conference on Computer Vision*, 2017. [16](#), [74](#), [76](#)
- [50] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. [16](#)
- [51] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [16](#), [17](#)
- [52] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. [16](#)
- [53] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. [16](#)
- [54] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019. [16](#)
- [55] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. [16](#)
- [56] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019. [16](#)
- [57] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Repoints: Point set representation for object detection. *arXiv preprint arXiv:1904.11490*, 2019. [17](#)
- [58] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019. [16](#), [17](#)

- [59] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2015. [18](#)
- [60] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *The IEEE International Conference on Computer Vision*, 2017. [18](#), [50](#)
- [61] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. [18](#)
- [62] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019. [18](#), [76](#)
- [63] Xingyu Zeng, Wanli Ouyang, Junjie Yan, Hongsheng Li, Tong Xiao, Kun Wang, Yu Liu, Yucong Zhou, Bin Yang, Zhe Wang, et al. Crafting gbd-net for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(9):2109–2123, 2017. [18](#)
- [64] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. [18](#)
- [65] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. [18](#), [76](#)
- [66] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3578–3587, 2018. [18](#)
- [67] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. [18](#)
- [68] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international conference on computer vision*, pages 1134–1142, 2015. [18](#)
- [69] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. [18](#)

- [70] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [19](#)
- [71] Peng Tang, Chunyu Wang, Xinggang Wang, Wenyu Liu, Wenjun Zeng, and Jingdong Wang. Object detection in videos by high quality object linking, 2018. [19](#)
- [72] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016. [19](#), [39](#), [44](#)
- [73] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [20](#), [50](#), [51](#), [64](#)
- [74] Xizhou Zhu, Yujie Wang, Jifeng Dai, Yuan Lu, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *The IEEE International Conference on Computer Vision*, 2017. [21](#), [22](#), [24](#), [25](#), [37](#), [40](#), [43](#), [44](#), [45](#), [46](#), [50](#), [51](#), [58](#), [62](#), [64](#), [65](#)
- [75] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [21](#), [22](#), [25](#), [37](#), [51](#)
- [76] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory. In *European Conference on Computer Vision*, 2018. [21](#), [22](#), [24](#), [44](#), [51](#)
- [77] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 331–346, 2018. [51](#)
- [78] Chaoxu Guo, Bin Fan, Jie Gu, Qian Zhang, Shiming Xiang, Veronique Prinet, and Chunhong Pan. Progressive sparse local attention for video object detection. *ICCV*, 2019. [21](#), [22](#), [23](#), [24](#), [51](#)
- [79] Hanming Deng, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Object guided external memory network for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6678–6687, 2019. [51](#)
- [80] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9217–9225, 2019. [20](#), [25](#), [51](#), [52](#)

- [81] Mykhailo Shvets, Wei Liu, and Alexander C Berg. Leveraging long-range temporal relationships between proposals for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9756–9764, 2019. [20](#), [25](#)
- [82] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [20](#), [46](#)
- [83] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [21](#), [22](#), [37](#), [43](#), [44](#), [45](#), [64](#)
- [84] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision*, 2015. [21](#), [22](#), [37](#), [38](#), [42](#), [51](#), [59](#), [62](#), [65](#)
- [85] Mason Liu and Menglong Zhu. Mobile video object detection with temporally-aware feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5686–5695, 2018. [24](#)
- [86] Xizhou Zhu, Jifeng Dai, Xingchi Zhu, Yichen Wei, and Lu Yuan. Towards high performance video object detection for mobiles. *arXiv preprint arXiv:1804.05830*, 2018. [24](#)
- [87] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2009. [25](#), [26](#), [85](#)
- [88] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. [26](#), [55](#), [73](#), [77](#)
- [89] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017. [28](#), [57](#), [59](#), [66](#), [73](#), [83](#), [85](#), [86](#), [88](#)
- [90] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1306, 2018. [30](#)
- [91] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4262–4270, 2018. [26](#), [30](#), [55](#), [73](#)

- [92] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014. [26](#)
- [93] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010. [26](#)
- [94] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *European conference on computer vision*, pages 452–466. Springer, 2010.
- [95] Hyun Oh Song, Ross Girshick, Stefanie Jegelka, Julien Mairal, Zaid Harchaoui, and Trevor Darrell. On learning to localize objects with minimal supervision. *arXiv preprint arXiv:1403.1024*, 2014.
- [96] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, 2016. [26](#)
- [97] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016. [28](#), [73](#), [85](#)
- [98] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8292–8300, 2019. [29](#)
- [99] Mingfei Gao, Ang Li, Ruichi Yu, Vlad I Morariu, and Larry S Davis. C-wsl: Count-guided weakly supervised localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 152–168, 2018. [29](#)
- [100] Yunchao Wei, Zhiqiang Shen, Bowen Cheng, Honghui Shi, Jinjun Xiong, Jiashi Feng, and Thomas Huang. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 434–450, 2018. [29](#), [30](#), [73](#), [76](#), [91](#), [92](#)
- [101] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2f: A weakly-supervised to fully-supervised framework for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–936, 2018. [30](#)
- [102] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2199–2208, 2019. [30](#), [73](#), [85](#), [91](#)

- [103] Satoshi Kosugi, Toshihiko Yamasaki, and Kiyoharu Aizawa. Object-aware instance labeling for weakly supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6064–6072, 2019. [30](#), [76](#), [91](#), [92](#)
- [104] Yunhang Shen, Rongrong Ji, Yan Wang, Yongjian Wu, and Liujuan Cao. Cyclic guidance for weakly supervised joint detection and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 697–707, 2019. [30](#), [73](#), [91](#), [92](#)
- [105] Yan Gao, Boxiao Liu, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, and Dongrui Fan. C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9834–9843, 2019. [30](#), [92](#)
- [106] Krishna Kumar Singh and Yong Jae Lee. You reap what you sow: Using videos to generate high precision object proposals for weakly-supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9414–9422, 2019. [30](#), [73](#), [75](#), [76](#), [89](#), [90](#), [91](#)
- [107] Vicky Kalogeiton, Vittorio Ferrari, and Cordelia Schmid. Analysing domain shift factors between videos and images for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2327–2334, 2016. [33](#)
- [108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. [33](#), [39](#), [45](#), [58](#), [64](#)
- [109] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012. [36](#), [64](#)
- [110] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, 2007. [44](#)
- [111] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015. [51](#), [52](#)
- [112] Jiajun Deng, Yingwei Pan, Ting Yao, Wengang Zhou, Houqiang Li, and Tao Mei. Relation distillation networks for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7023–7032, 2019. [51](#), [52](#)
- [113] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. [62](#)

-
- [114] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016. [66](#)
 - [115] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [68](#)
 - [116] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 914–922, 2017. [76](#)
 - [117] Jiajie Wang, Jiangchao Yao, Ya Zhang, and Rui Zhang. Collaborative learning for weakly supervised object detection. *IJCAI*, 2018. [91](#)