UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Exploring Region-based Deep Learning to Understand Objects in Real-world Scenarios

by

Ruiheng Zhang
Supervisor: A/Prof. Min Xu

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2021

Certificate of Authorship/Originality

I, Ruiheng Zhang declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

Also, I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with Beijing Institute of Technology, China.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

To my loving parents, my dear grandparents, my beloved wife and daughter.

ABSTRACT

Exploring Region-based Deep Learning to Understand Objects in Real-world Scenarios

by

Ruiheng Zhang

Supervisor: A/Prof. Min Xu

One way to infer about the real scenes is by understanding the object that presents in it, involving object localization, object recognition, object tracking, etc. Despite many advances in computer vision techniques, object understanding in real-world scenarios still remains many challenging tasks. There is no universal algorithm that can solve all of the scenarios with their own practical difficulties. This dissertation focuses on exploring region-based deep learning to understand objects in three typical real-world scenarios.

The first part of the dissertation studies facial landmark detection in the condition of lack of finely labeled training data. We generate weakly labeled training data to replace finely labeled data using generative adversarial networks. Then, we propose a region-based convolutional neural network to detect facial components and landmarks simultaneously. Notably, our approach can handle the situation when large occlusion areas occur, as we localize visible facial components before predicting corresponding landmarks. Extensive evaluations on several datasets indicate the effectiveness of the proposed approach.

In the second part, multi-player identification and tracking tasks in sports video are discussed. We build a robust multi-camera multi-player tracking with identification framework, from player detection, to identification, to tracking. To handle the identity switches, we design a distinguishable deep representation for player identity, considering pose-guided partial features, team class, and jersey number. For

data association, a robust multi-player tracker incorporating with player identity is further developed to produce identity-coherent trajectories. Experiment results illustrate that our framework handles the identity switches effectively, and outperforms state-of-the-art trackers on the sports video benchmarks.

Finally, we study vehicle detection in infrared images with poor texture information, low resolution and high noise levels. To deal with these difficulties, we propose a backbone network to exploit discriminative features, composing of a frequency feature extractor, a spatial feature extractor and a dual-domain feature resource allocation model. Hypercomplex Infrared Fourier Transform is developed to calculate the infrared intensity saliency, while a convolutional neural network is used to extract feature maps in the spatial domain. To efficiently integrate and recalibrate the frequency and spatial features, we propose a Resource Allocation model for Features based on the well-designed attention blocks. The experiments substantiate the merits of the proposed method through comparisons with state-of-the-art methods.

Dissertation directed by Associate Professor Min Xu
School of Electrical and Data Engineering

Acknowledgements

First and foremost, I want to extend my heartfelt gratitude to my supervisor, A/ Prof. Min Xu. Her kind supervision, valuable suggestions and warm encouragement helped me to overcome difficulties and to successfully complete this thesis. She had empowered and inspired me to work on multiple research projects which led to publications in top journals and conferences. She will always be my most beloved supervisor. I have learned so much from her, not only how to be a successful scholar but also how to be a kind person in life. I also would like to appreciate my co-supervisor Dr. Xiaoying Kong for providing me with continuous support throughout my study and research.

Second, I wish to express my sincere appreciation to Prof. Lixin Xu and A/ Prof. Chengpo Mu, my Dual-PhD supervisors at Beijing Institute of Technology. Although we are far away from each other, they gave me many suggestions and much help.

Third, I would like to pay tribute to Prof. Dacheng Tao, Prof. Dayong Jin, Prof. Linlin Ge, Prof. Sean He, Prof. Qiang Wu, Prof. Shenghong Hu, Dr. Yu Peng, Dr. Wenguan Wang, Dr. Yang Yu, and Dr. Zhengyu Yu for their inspiring words, continued support, and elaborate instructions about my presentations and academic reports. They had a significantly positive influence on me during my Ph.D. career.

Forth, I thank my fellow lab mates and my excellent friends: Tianrong Rao, Haimin Zhang, Lingxiang Wu, Wanneng Wu, Yukun Yang, Lei Sang, Zhongqin Wang, Xiaoxu Li, Qiyu Liao, Shuo Yang, Jiahao Xia, Caoyuan Li, Yang He, Yaxin Shi, Shilei Zhou, Huan Li, Wei Wei, Xiaofeng Xu, Qi Zhang, Zhibin Li, Yan Huang, Huidong Xu, Deyang Liu, Yuexi Zhang, Jiayan Qiu, Ye Shi, Yulong Sun and others that I cannot list them all for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the funs we have had.

I thank, School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney for providing the infrastructures, computing power, and financial support.

Finally, I am grateful for my dear family, especially my father Bin Zhang, my mother Juan Li, my grandpa Wenhua Li, my grandma Yachun Liu, my father in law Shengchun Shi, my mother in law Conghua Lin, my uncle Donghai Zhang and Yunhai Ma, my brother Zhe Yuan and Zichao Ma, who had been supportive during my PhD candidature stages, researching and writing my thesis. I am sincerely grateful for everything and especially the opportunities given in this life, from being born to the completion of this Doctor of Philosophy thesis.

Special thanks to my wife Dr. Qiaolin Shi, who is always being with me. She was the one who encouraged me to grow as a real man and gave the pretty birth to my beloved daughter Yuxi Zhang.

Ruiheng Zhang

Sydney, Australia, 2020.

List of Publications

Journal Papers

J-1. **R. Zhang**, L. Wu, Y. Yang, W. Wu, Y. Chen, M. Xu, 'Multi-camera Multi-player Tracking with Deep Identification in Sports Video', *Pattern Recognition*, Volume 102, 2020, 107260, ISSN 0031-3203.

J-2. **R. Zhang**, C. Mu, M. Xu, L. Xu, Q. Shi, J. Wang, 'Synthetic IR Image Refinement using Bidirectional Mappings with Adversarial Learning', *IEEE Access*, 2019, 7: 153734-153750.

J-3. **R. Zhang**, C. Mu, M. Xu, L. Xu, X. Xu, 'Facial Component-Landmark Detection with Weakly-supervised LR-CNN', *IEEE Access*, 2019, 7: 10263-10277.

J-4. **R. Zhang**, L. Xu, Z. Yu, Y. Shi, C. Mu, M. Xu, 'Deep-IRTarget: An Automatic Target Detector in Infrared Imagery using Dual-domain Feature Extraction and Allocation', *IEEE Transactions on Multimedia*, 2021.

J-5. Q. Liang, W. Wu, Y. Yang, **R. Zhang**, Y. Peng, M. Xu, 'Multi-Player Tracking for Multi-View Sports Videos with Improved K-Shortest Path Algorithm', *Applied Sciences*, 2020, 10, 864.

J-6. Q. Zhang, **R. Zhang**, L. Ge, M. Xu, 'A High Accuracy Burned Area Detection Framework Based on The Joint Processing of Sentinel-1&2 Data', *Remote Sensing of Environment*. (under review)

J-7. Y. Yang, **R. Zhang**, W. Wu, M. Xu, '3D Localization for Multiple Players in Multiview Sport Videos with Deep Identification Reasoning', *Pattern Recognition*. (under review)

Conference Papers

C-1. **R. Zhang**, M. Xu, Y. Shi, J. Fan, C. Mu, L. Xu, 'Infrared Target Detection using Intensity Saliency and Self-Attention', *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020.

C-2. Y. Yang, **R. Zhang**, W. Wu, Y. Peng, M. Xu, 'Multi-camera sports players 3D localization with identification reasoning', *2020 25 th IEEE International Conference on Pattern Recognition(ICPR)*. IEEE, 2020.

C-3. Y. Yang, M. Xu, W. Wu, **R. Zhang**, and Y. Peng, '3D Multiview Basketball Players Detection and Localization Based on Probabilistic Occupancy', *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Vision is the primary source of information for humans to perceive the external objective world, and computer vision technology aims to allow machines to perceive and understand the physical world as accurately as humans. The development and progress of computer vision technology is the cornerstone of many human-computer interaction technologies. It is also a crucial step for human society to move towards the era of real artificial intelligence. With the rapid development of Internet technology and social media technology, various forms of visual media data such as images, dynamic pictures, and videos have shown 'explosive' growth. As shown in Figure 1.1, as of 2020, the video sharing website YouTube uploads about 500 hours of video data every minute, and the image sharing community Instagram uploads about 138,889 images every minute. Faced with the ever-increasing mass of visual media data, using computer vision technology to perceive and understand it, so as to realize the rapid utilization of mass visual media data, is of great significance to promote the progress of human society and application value.

The process of human visual cognition of the real world is subject to a certain logical way of observation. In order to better understand an image, when people observe the image, their eyes will move along the region of interest, and they will carefully observe the local details of the objects presented in the image. Dana H. Ballard and Christopher M. Brown [1] in 1982 defined computer vision in their published book, 'computer vision is the construction of explicit, meaningful descrip-

tions of physical objects from images'. Hence, object understanding is an important and fundamental topic in computer vision, which involves localizing, recognizing, tracking, and reasoning about objects.

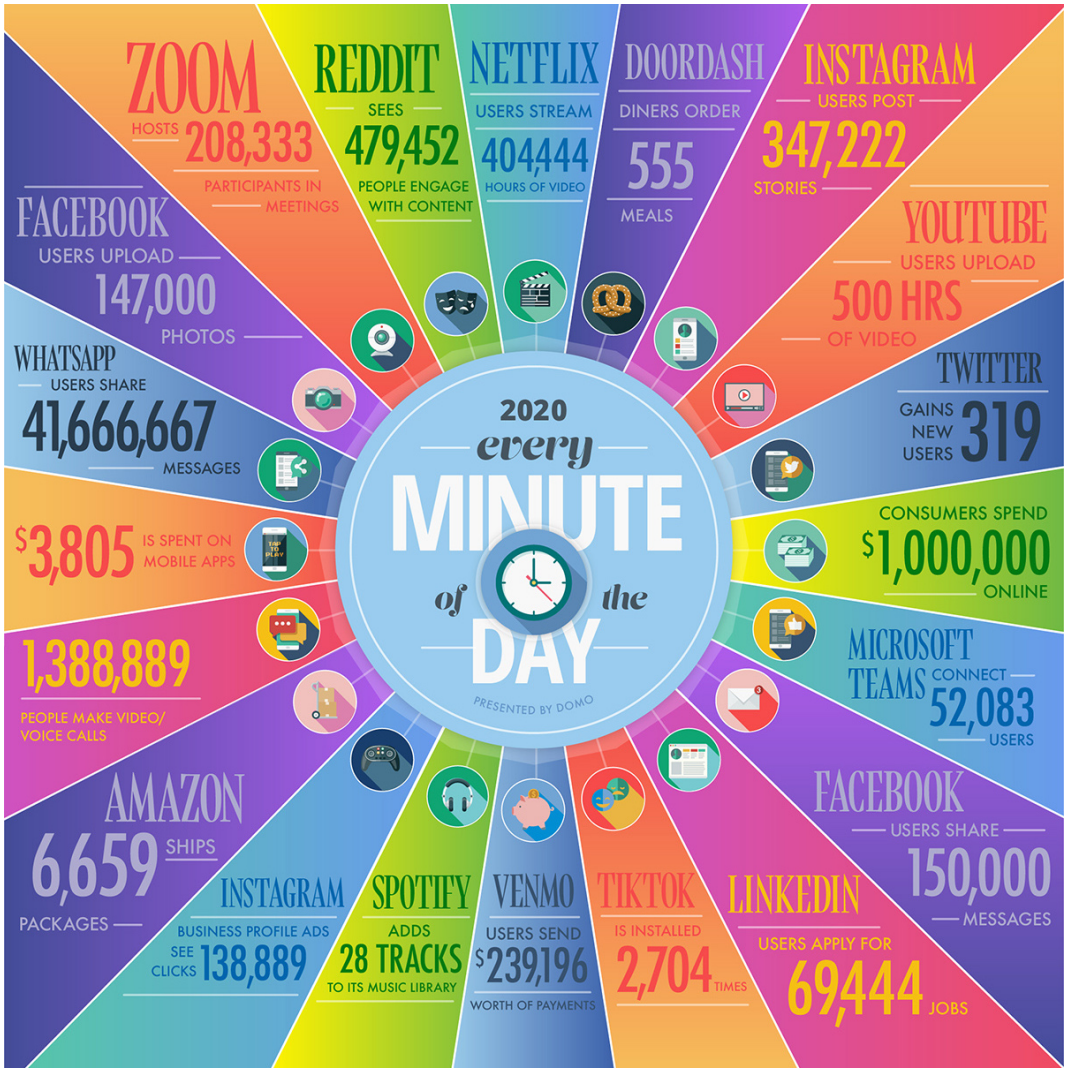


Figure 1.1 : In the era of big data, image, video and other media data show 'explosive' growth.

Recently, computer vision technology based on deep learning has made significant breakthroughs. It reaches or exceeds human performance in multiple visual tasks. For example, on the large-scale ImageNet dataset, a state-of-the-art deep learning model [2] has a classification accuracy of 88.4% in the Top-1 category and 98.7%

in the Top-5 category, while the accuracy of human is only 94.9% [3] in the Top-5 category. More recently, many region-based deep learning methods are rapidly developed to reason about objects. Compared with traditional methods, deep learning methods achieve a state-of-the-art performance by their strong feature extraction ability.

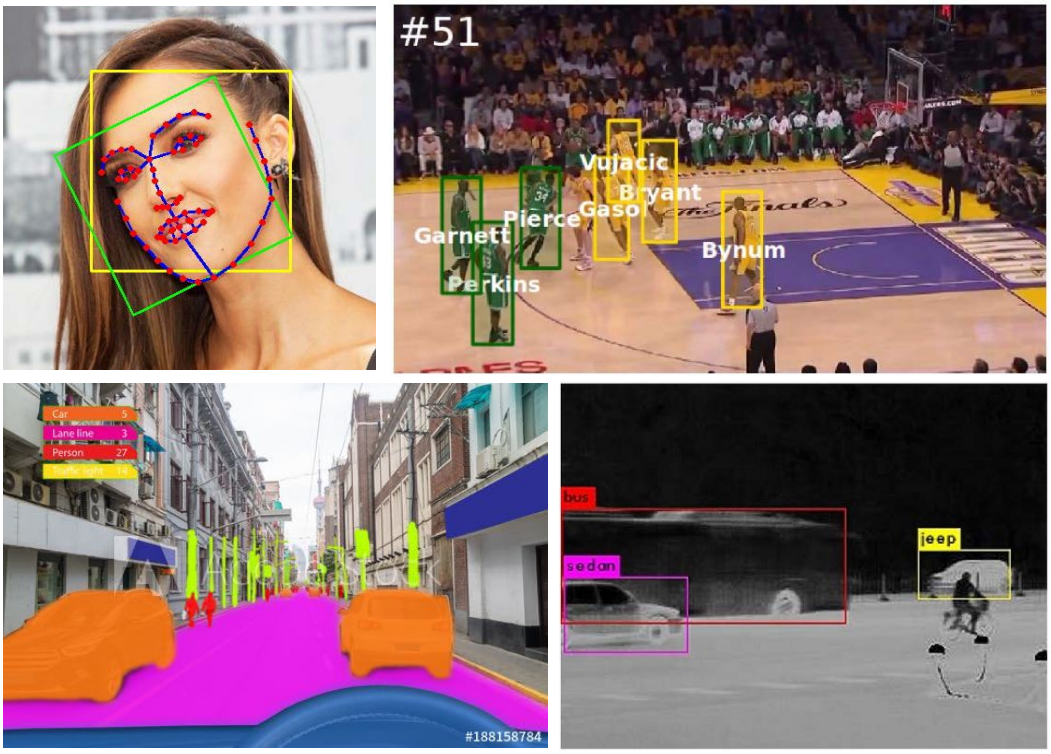


Figure 1.2 : Deep learning technology promotes the development of vision application.

However, for the perception and understanding of objects in real-world scenes, the performance of current models is still far from that of humans. It is far from reaching the level of large-scale popularization and landing applications on some object understanding tasks [4, 5, 6], as shown in Figure 1.2. The main reason for this is that complex visual scenes usually contain many objects and the interaction between objects. Meanwhile, there are also various occlusions and different scales between objects, which greatly increase the difficulty. On the other hand, different

real-world scenarios have their different challenges. It is impossible to develop a universal algorithm that can solve all of the scenarios with their own practical difficulties. For object understanding, one might try to answer questions such as, what types of objects exist in the image, where are they, how are they related to each other, what are their exact contour, etc. By finding the answers to these questions, it is necessary to infer about the real-world scenarios.

In this thesis, we study region-based deep learning for object understanding in different real-world scenarios. These typical scenarios include rigid objects and non-rigid objects, visible light scenes and infrared scenes, such as (1) facial component and landmark detection from face images in the condition of limited training data; (2) multi-object detection, identification and tracking in multi-camera sports video; (3) vehicle localization and recognition in forward looking infrared imagery. These different scenarios have their own features, which need to be considered and resolved. Meanwhile, these tasks are based on object detection. For instance, facial landmark detection is based on component detection, multi-player tracking is based on player detection, and vehicle recognition is based on vehicle detection. Furthermore, our methods for these tasks use the same region proposal backbone, and have different contributions to these tasks. In other words, we revisit region-based deep learning on these typical real-world scenarios.

1.2 Summary of Contributions

This thesis explores region-based deep learning to understand objects, and handle the challenges in three typical real-world scenarios. The expected contributions of the thesis are:

- For face understanding, we propose a weakly-supervised region-based CNN model to detect facial components and landmarks simultaneously, and tackle

the large occlusion problem. To cope with the lack of training data with detailed annotations, we generate weakly labeled data to replace pixel-level annotated.

- For player understanding, we build a robust multi-camera multi-player tracking with identification framework, from player detection, to identification, to tracking. In particular, a player identification model with region-based deep learning is developed to extract the distinguishable player ID, considering pose-guided partial features, team class, and jersey number.

- For vehicle understanding, we design a target detection and recognition method with region-based deep learning in infrared imagery. To exploit discriminative features, we propose a frequency and spatial feature extraction model. Furthermore, a resource allocation model is developed to recalibrate the features.

1.3 Organization

This thesis is organised as follows:

- *Chapter 1*: The first chapter explores the background on understanding objects in real-world scenarios with a focus on object detection. The main research contributions are highlighted as well.

- *Chapter 2*: In this chapter, a survey of the works most related to our object understanding methods is presented, including deep learning-based object detection, facial component and landmark detection, multi-object identification and tracking in sports video, and infrared target detection and recognition.

- *Chapter 3*: The third chapter studies facial component and landmark detection in the case of limited finely labeled training data. A weakly labeled data

generation model and a weakly-supervised facial component and landmark detection method are derived.

- *Chapter 4*: This chapter focuses on the scenario of multi-player tracking by detection and identification under multiple cameras. A robust multi-camera multi-player tracker with player identification is developed to resolve the identity switch problem, consisting of a player identification model, a players' 3D localization model and an ID-enhanced multi-player tracker.

- *Chapter 5*: This chapter presents an infrared target detection method. In particular, a dual-domain feature extraction model and a resource allocation model are designed to exploit discriminative features from infrared images with poor texture information.

- *Chapter 6*: A brief summary of the thesis contents and our recommendation for future works are given in the final chapter.

Chapter 2

Literature Review

This chapter provides a review of existing deep learning-based object detection methods. Meanwhile, we also include a brief review of research areas related to the content of the proposed methods in different real-world scenarios, like facial component and landmark detection, multi-object identification and tracking in sports video, and infrared target detection and recognition.

2.1 Deep Learning-based Object Detection

In recent years, CNN has made a breakthrough in the field of object detection with the advantages of high-level features in the extraction of images.

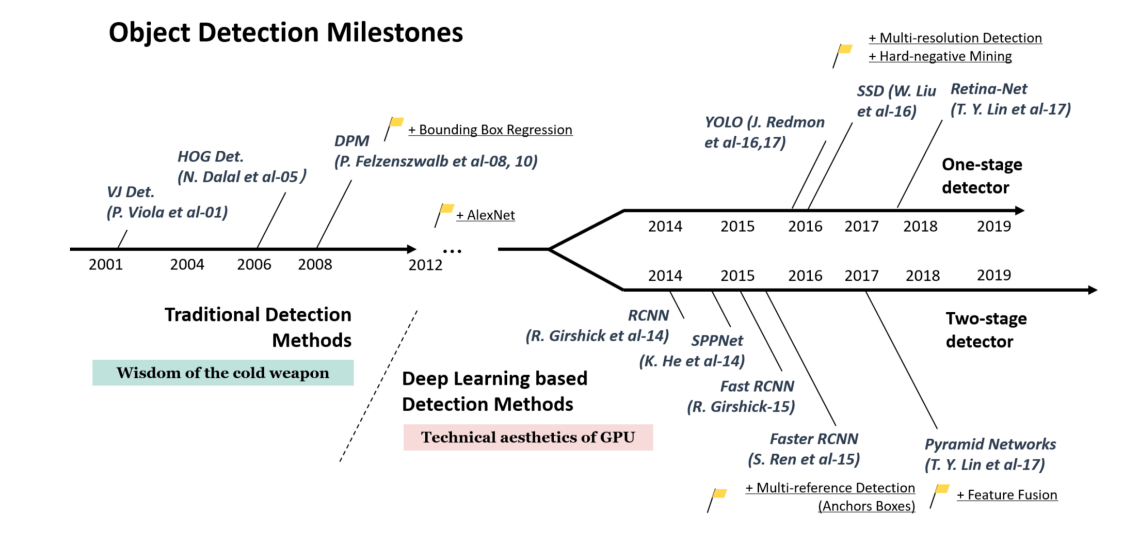
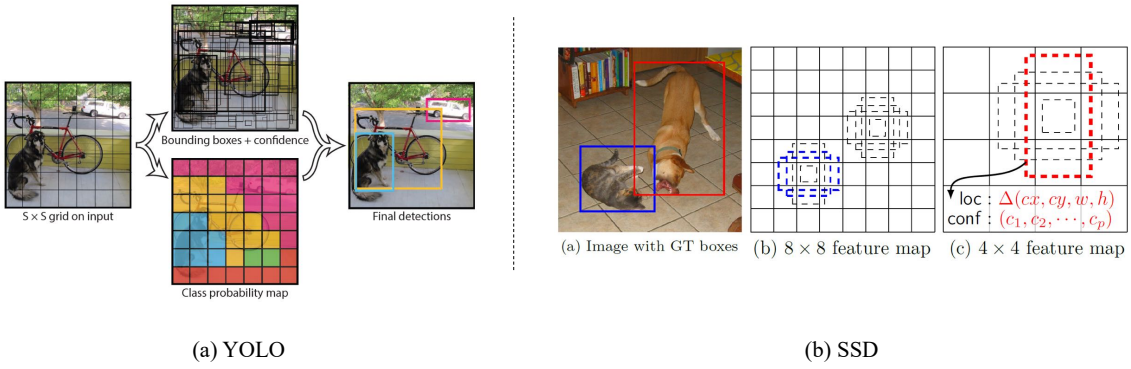


Figure 2.1 : Object detection milestones.

From the perspective of development time, object detection algorithms can be roughly categorized into two periods, one is based on traditional handcrafted features

(before 2013), the other is based on deep learning (2013 to present) [7]. Deep learning-based object detection algorithms can be roughly divided into two genres: "one-stage detector" based on integrated convolutional neural networks like YOLO series (as shown in Fig. 2.2) [8, 9, 4] , and "two-stage detector" based on region proposal such as RCNN series (as shown in Fig. 2.3) [10, 11, 12].



(a) YOLO (b) SSD

Figure 2.2 : The architecture of YOLO and SSD.

YOLO [8] was the first one-stage detector. The advantage of this algorithm was its fast speed, but its accuracy was lower than Faster-RCNN [12]. Especially for small target detection, the localization performance of YOLO was slightly insufficient. To tackle this problem, Redmon proposed YOLOv2 [9], introducing the "Anchor" to make a compromise between accuracy and speed. YOLOv3 [4] made some upgrades to further improve the detection accuracy, using multiscale training and data augmentation methods. SSD [13] was proposed by Liu *et al.*, which combined the advantages of YOLO's fast speed and RPN's precise localiztion, and further performed detection on feature maps with multiple resolutions. The lastest milestone is RetinaNet developed by Lin and Goyal *et al.*[14]. In the RetinaNet, the proposed Focal Loss achieved comparable accuracy of two-stage detectors and maintained high speed.

Girshick *et al.* [10] took the lead to propose the first two-stage detector—RCNN.

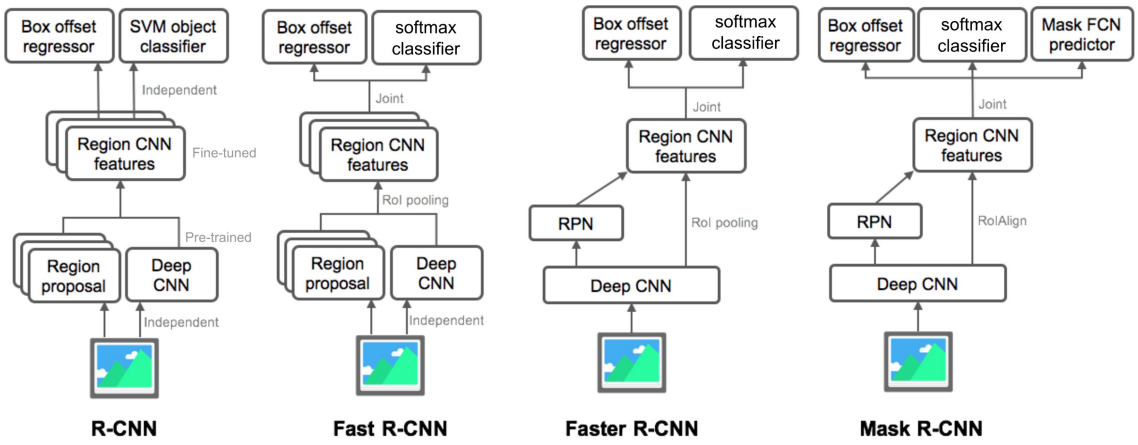


Figure 2.3 : The architecture of the RCNN series.

It firstly used selective search to find region proposals, and then conducted CNN with SVM to classify the proposals. To tackle the input size restriction of previous CNN, He *et al.* developed Spatial Pyramid Pooling (SPP) layer [15]. Fast RCNN [11] with a speed beyond 200 times faster than RCNN was a further improvement of RCNN and SPP, which could train a detector and a bounding box regressor simultaneously. But Fast RCNN also implemented selective search for region proposals. This restricts the speed of the region proposal stage. Ren *et al.* [12] proposed a Region Proposal Network (RPN) to estimate region proposals by sharing the CNN with backbone in the Faster RCNN method. On basis of Faster RCNN, Lin *et al.* developed Feature Pyramid Network (FPN) [16] that showed great progress in detecting objects with a wide variety of scales. Mask RCNN [17] proposed by He *et al.* was a more efficient framework for object detection and instance segmentation.

Recently, some researchers have begun to study Anchor-Free Detectors. The milestone work was CornerNet [18] proposed by Law, which detected objects as paired keypoints by a new type of pooling layer named Corner pooling. Another important study was from Zhao *et al.* [19]. ExtremeNet was a bottom-up approach, using a standard keypoint estimation network to detect one center point and four

extreme points of an object.

2.2 Facial Component and Landmark Detection

Facial component detection: Detection of facial components as a significant step of face analysis is aimed at detecting facial components like eyebrows, eyes, mouth, nose, either given a known face detection or under the assumption that there is only a single face in the image. Yihu *et al.* [20] used mapping-based localization to detect eyes and lip region with a fixed face structure. This method failed to facial deformation or expression change. YiMartin Urschler *et al.* [21] present an algorithm for detecting face and facial component candidates, and for robustly voting for the best face and eyes. But it could only detect eye and mouth regions. Jacek Naruniec *et al.* [22] employed Discrete area filters for face detection and facial feature detection, which focused on fiducial point detection of facial components with complex computing. K. Sudhakar *et al.* [23] used Gabor filter to detect facial components including left eye, right eye, nose, mouth, and also detect facial points in each component area. But this method was unable to distinguish eyebrow and eye region. These algorithms could only detect and guess all the components if some components are obscured, by using shallow models with fixed component structural relation.

Facial landmark detection: Traditional landmark detection approaches with shallow models can be divided into two main categories, which was named as template fitting approaches and regression-based algorithms. (1) The former methods aim to learn a shape model from training set and to fit input pictures during testing. The pioneering works of template fitting algorithms are ASM [24] and AAM [25]. As for ASM, the shape of face was represented by the linear combination of basic shapes learning via PCA and appearance of face was modeled by different pretrained templates. In AAM, the shape representation was similar with ASM while

the appearance is modeled by PCA in regular coordinate system that eliminates shape changes. (2) Regression-based methods estimated landmark locations explicitly by regression using image features. Burgos-Artizzu *et al.* [26] and Cao *et al.* [27] used cascaded and random fern regression with pixel-difference features. Yang *et al.* [28] employed random regression forest to cast votes for landmark location based on local image patches using Haar-like features.

Recently, deep learning techniques are being widely applied to facial landmark detection, so that the accuracy is promoted undoubtedly. These methods usually regard landmark localization task as a regression problem. The common methods could be also divided into two types: one is given the initial position estimation, network learns the error between the true value and the estimation, and reduces the error between the output value and the real value through the iterative operation. The other is to predict the location of the key points directly. The most representative algorithm of the former method was proposed by Haoqiang Fan *et al.* [29]. They build an accurate and robust facial landmark localizer using deep learning tools, which includes two levels of convolutional neural network for course-to-fine prediction. The representation of the second approach is a multi-task learning algorithm for both facial attribute estimation and five-points landmark detection [30]. Another algorithm presented by Yue Wu *et al.* [31] was for simultaneous facial action unit recognition and facial landmark detection.

2.3 Multi-object Identification and Tracking in Sports Video

Player identification: A body of player identification algorithms is concentrated on a close-up single camera view and multi-camera views. Under the close-up camera views, players can be identified by jersey number recognition [32, 33, 34] and face recognition [35]. For jersey number recognition, these approaches directly tried to recognize the jersey numbers followed by characters recognition without detecting

the number regions. Ye *et al.* [33] employed a K-NN (K nearest neighbor) classifier with the Zernike moment features, to detect jersey numbers. Gerke *et al.* [32] first introduced CNNs into soccer jersey number recognition. Without any character detectors, Li *et al.* [34] employed CNN model to classify jersey number on the detected player's images. The method in [36] combined the textual cues with the visual face information to identify players.

Somewhat against the trend, in term of multiple cameras, the face information is limited. Lu *et al.* [37] first attempted to track and identify basketball players by recognizing the entire body instead of the face or jersey number. They designed a player's appearance representation by low-level hand-crafted features, including scale-invariant feature transform, maximally stable extremal regions and color histograms. Differently, we leverage deep convolutional features guided by pose estimation to model the player's representation. Recently, Gerke *et al.* [32] treated jersey number recognition as an image classification task using deep convolutional neural networks. They directly cropped the top half of the player images as the jersey number regions. In [38], player number and group information are used to associates tracklets of the same player. In this paper, we develop a coarse-to-fine-grained deep convolutional network to automatically detect and recognize the jersey numbers and team classes. Senocak *et al.* [39] use convolutional neural networks features to represent the player regions, and formulate the identification as a classification problem. Compared with this method, we pay more attention to how to find the difference among players. Some other works perform player identification by using the position of players. In [40], the player's location information is utilized as spatial constellation besides jersey number recognition. As players' trajectories are known, the problem is formulated as an assignment problem. Lu *et al.* [37] leverage both detection and tracking to build a conditional random field model for all the players.

Multi-object tracking with identity association: Existing approaches in multiple object tracking are mostly based on a tracking-by-detection framework. With this paradigm, objects are first detected and then linked into trajectories. K-Shortest Path (KSP) optimization [41] was one of the most feasible approaches for multi-object tracking. In this method, the data association is formulated as a constrained flow optimization problem, which can then be solved using KSP algorithm. Nevertheless, the neglect of the appearance model would easily lead to identity switches. To address this identity switch problem, some appearance or motion modules have been developed. Shitrit *et al.*[5] presented some appearance features for tracking. Although the identity switch problem was tackled to some extent, it cannot be avoided when some players are wrongly detected or when very little appearance information is available. Shitrit *et al.* [42] also considered image appearances based on KSP tracking. Liu *et al.* [43] designed a set of Game Context Features to describe the current state of the match. The context conditioned motion model implicitly incorporates complex inter-object correlations when remaining tractable by using cost flow networks. In the [44], an articulation-based detection selecting method with a stitching strategy was proposed to screen out detections unqualified for multiple people tracking. Shen *et al.* [45] presented a fast online MOT approach through introducing the minimum output sum of squared error filter. These improved methods can preserve identities well over long sequences, only if most of the appearance cues or motion cues are clear. However, this was rare in team sports because of the serious frequent occlusions. Moreover, these hand-crafted features failed to associate the player's trajectories accurately.

More recently, deep learning techniques begin to be applied on tracking. Tang *et al.* [46] designed an ID-Net combining the human part features for modeling the appearance, and also employ a network flow-based algorithm for data association to yield trajectories. In [47], Lu *et al.* proposed a deep regression tracking algorithm

with a shrinkage loss to penalize the importance of easy training data, and use residual connections to fuse multiple CNNs to preform favorably against state-of-the-art trackers. Philipp *et al.* [48] proposed a new tracking paradigm and point out promising future research directions. However, these methods cannot tackle the identity switch in the sports video scenario.

2.4 Infrared Target Detection and Recognition

In general, infrared target detection can be divided into four stages, including feature extraction, target localization, clutter filter, and target recognition. Many researchers studied infrared target detection algorithms during the last few decades, which can be roughly summarized into two groups.

One is to model the target detection as a segmentation task, considering the infrared characteristic of targets. The classic methods contain local threshold methods and mathematical morphology methods. Common local threshold method was OTSU thresholding [49], partial differential equation [50], and fuzzy thresholding [51]. Among the morphology methods, one efficient method was proposed by Braga-Neto *et al.* [52], using a morphological connected operator to extract and track infrared targets of interest. Later, Lu *et al.* [53] combined the morphology method with the sharp frequency localized contourlet transform for infrared image segmentation.

The other group is to use the sliding window or the saliency map to find the RoIs, and then conducts a classifier on the RoIs. Bassem *et al.* [54] proposed an RoIs detector that combines SURF-based local and global features, and then used an SVM classifier to recognize the RoIs. Wang *et al.* [55] combined OTSU with HOG feature to generate candidate regions, followed by a deep belief network as the classifier. Cai *et al.* [56] proposed a fusion saliency-based method on infrared pedestrians, followed by a histogram of local intensity difference feature to make the

final decision. A saliency map obtained using the discrete cosine transform [57] was designed to generate the region candidates, after which employed a local descriptor with an SVM classifier. More recently, some researchers have started to use CNN directly on this task. Zheng *et al.* [58] implemented YOLOv3 with a new training strategy on the infrared land battlefield dataset to reduce overfitting, and obtained a good result on low-resolution images. Nasser *et al.* [59] developed an ATR for infrared image named DeepTarget. It firstly utilized a fully convolutional network (FCN) for region proposal, and then used VGGNet for classification. However, either segmentation-based or RoI-based approaches work undesirably in the real scenarios, due to lack of extracting the discriminative feature.

Chapter 3

Facial Component and Landmark Detection With Weakly-Supervised LR-CNN

3.1 Introduction

The first object we study is the human face. In this chapter, understanding the human face involves facial component and landmark detection, which are important procedures in a multitude of face analysis tasks including face recognition [60],[61], facial expression analysis [62], face reconstruction [63], and face enhancement. With the enormous advancement of deep learning, the performance of many computer vision tasks, e.g. facial component and landmark detection, have been improved significantly. Generally, the success of applying deep learning to facial component and landmark detection relies on a reliable deep architecture with optimal parameters, which are trained and finely tuned using a large amount of training data with accurate and detailed annotations. Without enough quantity and quality of fully labeled training samples, however, detecting facial components and landmarks in images with severe occlusions is a formidable challenge.

Most of the traditional facial component detection algorithms rely on shallow models, such as SVM [64], Gabor Wavelet [23], and Bag-of-Words [65], which may fail to combine with facial alignment methods effectively. On the other hand, facial landmark detection approaches can be categorized into three types, the template fitting approach [24],[25], cascaded shape regression based methods[26],[27] and deep-learning-based models[66, 67]. Traditional methods, such as template fitting approaches and regression-based models, heavily rely on prior knowledge and

artificial design feature, which might not be able to extract essential features for face alignment. Recently, deep-learning-based facial landmark detection methods [66, 67, 28, 68, 69, 70, 71] have achieved remarkable results. Sun *et al.* [69] propose a cascaded convolutional neural network model with 23 layers, which requires a huge amount of computational power during training and testing. Kumar *et al.* [70] design a coarse-to-fine framework, of which the input is not only raw pixels but a set of given landmarks. The model has been trained four times with input images at different scales. Marek *et al.* [71] present a Deep Alignment Network trained by entire face images, which is robust to large variations in difficult initializations and head poses.

Although these algorithms have good performance in laboratory environment, they fail in some cases. First, lack of enough training data with detailed annotations will lead to poor generalization of deep learning models. In the case of fully supervised learning for locating facial landmarks, training images with corresponding pixel-level landmark annotations are highly demanded. However, it is often difficult to obtain pixel-level annotations, which are expensive and time-consuming. Second, existing face detectors fail to localize facial landmark in the real-world conditions owing to severe occlusions. When large occlusion areas occur, existing face detectors may fail to detect faces and miss the responding landmarks. In addition, since these methods detect landmarks in a whole face image, occlusion areas as the uninformative pixels effect the detection results of unblocked areas. Third, it lacks an end-to-end deep learning framework for facial component and landmark detection.

In order to tackle these challenges, the main contributions of this chapter are three-fold:

- To cope with lack of training data with detailed annotations, we consider to re-

place pixel-level annotated data with easily generated weakly labeled data. We propose a DCGAN-based data preprocessing and augmentation to generate facial component samples with weak labels effectively. After weakly-supervised learning on above data, our landmark-region-based CNN (LR-CNN) has a better landmark detection result, compared to just with fully-supervised learning.

- The proposed LR-CNN method based on region-based deep learning can tackle the large occlusion problem through detection of the visible facial components instead of a whole face in an image. AnchorAlign, RoIAlign, and a two-branch landmark detection model are presented in LR-CNN architecture, so that our method can detect facial components and landmarks simultaneously. The two-branch framework includes pixel-level classification, and landmark regression.

- This work is the first attempt to propose a comprehensive end-to-end framework, which firstly locates facial components and then infers corresponding landmark coordinates. The experimental results indicate that our algorithm outperforms state-of-the-art methods.

The whole pipeline mainly consists of two parts: data preprocessing and augmentation, and weakly-supervised LR-CNN model, as shown in Fig. 3.1. In the first part, we utilize DCGANs to generate facial component images with weak labels and convert facial landmark into a landmark vector. Then, a weakly-supervised LR-CNN (landmark-region-based CNN) is proposed for facial component and landmark detection, which firstly detects visible facial components (i.e. eyebrow, eye, nose and mouth), followed by estimation of landmarks based on the component location and classification results.

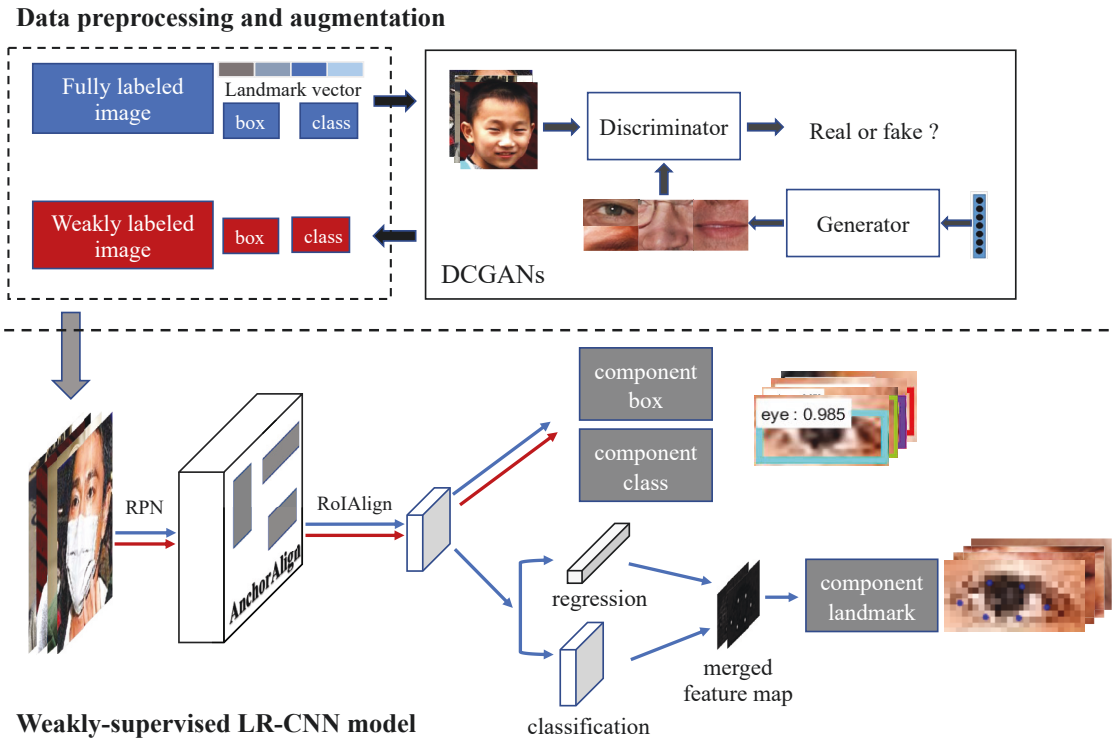


Figure 3.1 : The whole pipeline. Blue boxes and lines represent fully labeled data and fully-supervised learning processing, while red boxes and lines show weakly labeled data and weakly-supervised learning.

3.2 Data Preprocessing and Augmentation

Most facial landmark detectors require a large amount of training data with pixel-level annotations, such as 68-point landmarks. Since lack of training data with landmark-level labels, we consider to replace fully labeled data with weakly labeled data, so that our LR-CNN model can be trained by a small amount of fully labeled data and a large amount of weakly labeled data. In this section, we firstly generate massive weakly labeled data as training data preparation. Then, we design a landmark vector as the ground truth of fully labeled data, in order to achieve back propagation of fully-supervised part of LR-CNN model. The ground truth of weakly labeled data is component bounding-box coordinates and component class, while the

ground truth of fully labeled data includes component bounding-box coordinates, component class, and landmark coordinates.

The proposed data preprocessing and augmentation is shown in Fig. 3.2. One of the standard facial landmark benchmark has 68 points in one face, including jawline, eyebrow, eye, nose and mouth. We cut 68-landmark to 51-landmark by removing jawline. On one hand, each component region can be calculated by facial landmark coordinates. Maximum and minimum coordinates of landmarks in each component form a rectangular region, followed by proper amplification (1.25 times). Then, we put these real component images into 4 DCGAN models of different components to generate a large amount of 'fake' component images. After assign pseudo label to every 'fake' component image, weakly labeled data are ready for training. On the other hand, 51-landmark is converted into a 40-dimensional landmark vector which is regarded as fully-supervised labels, according to Table 3.1.

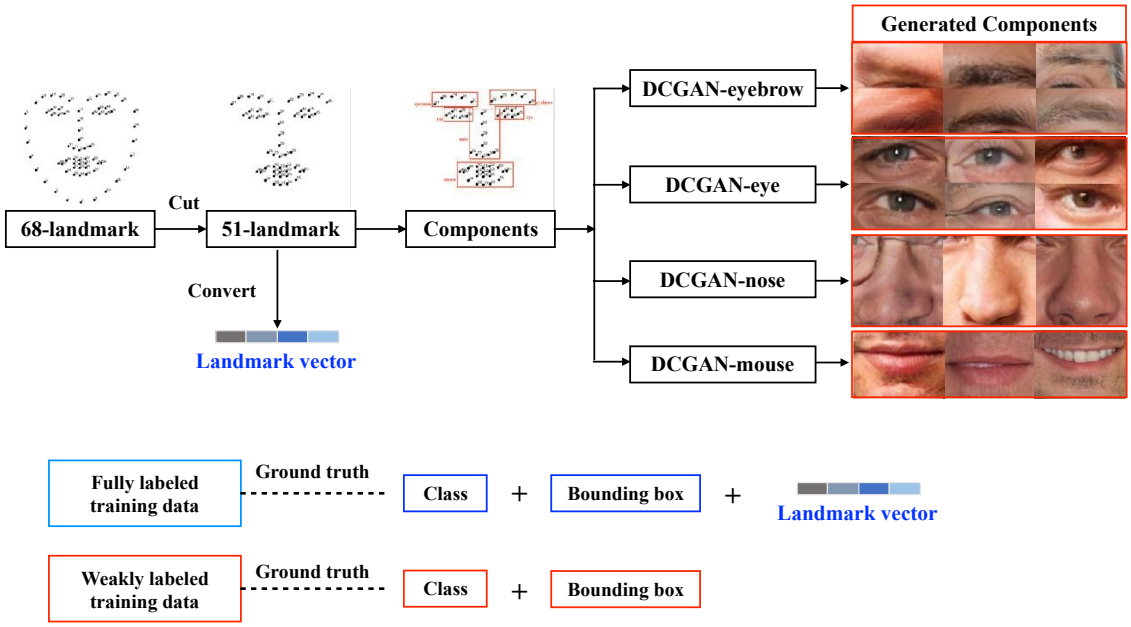


Figure 3.2 : Data preprocessing and augmentation.

Table 3.1 : The landmark indexes and the vector length of each facial component in the 68-landmark annotation.

	jawline	**eyebrow**	**eye**	**nose**	**mouth**
number	1-17	18-22, 23-27	37-42, 42-48	28-36	49-68
length	—	10	12	18	40

3.2.1 Weakly-supervised Training Data

Weakly-supervised training data are the main training set for LR-CNN model. Existing GAN-based data augmentation are directly generating face images, and they still require manually labeling. In addition, since the size of face image is larger than that of component image, GAN model for face images is often unable to converge, and GAN training is not well controlled. Therefore, we decide to generate different components respectively and automatically marking with weak labels. Considering the features and sizes of face components are different, in Fig. 3.2, four DCGANs [72] to generate four categories of 'fake' images containing four different components, i.e. eyebow, eye, mouth and nose. Each DCGAN model is independent of each other and has different hyper-parameters to generate different facial components. When training DCGANs, we update the generator G three times when updating the discriminator D once, other than original settings of DCGAN. After several experiments, we train the generator G to perform better than the discriminator D. In order to learn the generator's distribution p_g over each type of components, we define a prior on input random noise variables $p_z(z)$. Variable z obey the standard normal distribution $N(0, 1)$. Then we represent a mapping to data space as $G(z; \theta_g)$, where G is a differentiable function represented by a full convolutional neural network with parameters θ_g. We also define a second full convolutional neural network $D(x; \theta_d)$ that outputs a single scalar. $D(x)$ represents

the probability that x came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $log(1 - D(G(z)))$. In other words, D and G play the following two-player minmax game with value function $V(G, D)$:

$$\begin{aligned} minmaxV(D,G) = E_{x\sim p_{data}(x)}[logD(x)] \\ + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \end{aligned} \tag{3.1}$$

Four DCGANs are trained with SGD (stochastic gradient descent) in a mini-batch size of 64. All of weights are initialized from a zero-centered normal distribution with standard deviation 0.02. In the LeakyReLU, the slope of the leak was set to 0.2 in all models. We leverage the Adam optimizer with tuned hyper-parameters to accelerate training, and momentum is 0.5. The learning rate is 0.0002. For training DCGANs, the training sets of real face images include Helen [73], IBUG [74], AFW [75], and LFPW [76] dataset. When 'fake' components are generated, we replace real components with 'fake' components in real face images ,and mark them with weak labels (component class and bounding box) automatically. For class label, it is obvious that four DCGANs generate the images of eyebrow, eye, nose and mouth respectively, i.e. the DCGAN-eye can just output eye images. For bounding box label, we directly replace real component images with 'fake' component images in real face images. Thus, the ground truth of 'fake' components are the ground truth of real components. Finally, we generate 60,000 weakly-labeled data (ground truth: bounding box and category).

In the weakly-supervised training processing on LR-CNN model, we incorporate the proposals generated by RPN [12] into LR-CNN network. In the 2,000 candidate bounding boxes of each training sample, we randomly select 64 candidate boxes as a batch, which contains 16 positive samples (IoU to ground truth larger than 0.5, IoU is Intersection over Union as shown in Eq.(3.2)) and 48 negative samples

(IoU to ground truth larger than 0.1 and smaller than 0.5). For positive samples, their coordinates are converted into a vector (x, y, w, h) in relation to ground truth bounding box which each sample belongs to, in Eq.(3.3). The subscript s indicates the center coordinates of bounding box, and subscript g and t indicate the ground truth and training sample respectively. As for negative samples, we drop out of negative training samples and mark them with a background label for component classification.

$$IoU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth} \tag{3.2}$$

$$(x, y, w, h) = (\frac{x_{gs} - x_{ts}}{w_s}, \frac{y_{gs} - y_{ts}}{h_t}, log\frac{w_g}{w_t}, log\frac{h_g}{h_t}) \tag{3.3}$$

3.2.2 Fully-supervised Training Data

Besides weakly labeled training data, LR-CNN model also need a small amount of fully supervised for guidance, including Helen [73], IBUG [74], AFW [75], and LFPW [76] datasets. Fully labeled data extra contain landmark-level annotation compared to weakly-labeled data. 68-landmark is cut to 51-landmark, followed by converting into a landmark vector $(x_{1s}, y_{1s}, x_{2s}, y_{2s}, ..., x_{ns}, y_{ns})$, as shown in Eq.(3.4). (x_i, y_i) is the coordinate of i-th landmark. In Table 3.1, length is set according to different number of facial components. In addition, each sample has four weight vectors determining validity of coordinates. As for positive samples, all related coordinates of the components are available, and other components coordinates are unavailable. The negative landmarks coordinates are only valid inside the component bounding boxes.

$$(x_{ti}, y_{ti}) = (\frac{x_i - x_{ts}}{w_t}, \frac{y_i - y_{ts}}{h_t}) \tag{3.4}$$

Therefore, the preparation of training data has been completed. The sum of preprocessed data for LR-CNN model is about 66,000, including around 6,000 fully-labeled data (ground truth: landmark, bounding box and category) and 60,000 weakly-labeled data (ground truth: bounding box and category).

3.3 Weakly-supervised LR-CNN Framework

After data preparation, a novel architecture called weakly-supervised LR-CNN is presented in Fig. 3.3, which mainly consists of region-based component detection and two-branch landmark detection. When training, the input of network includes two parts: weakly labeled data, and fully labeled data. When testing, the input of network is just face images to be predicted. Firstly, we leverage ResNet-50[77] model to extract convolutional features of input image and share them to our RPN with AnchorAlign model , for calculating RoIs (Region of Interest). After RoIAlign layer, fixed size feature map is put into component detection model and landmark detection model simultaneously. The component detection model predicts component bounding box and category. For landmark detection, a two-branch landmark detection model is proposed, which consists of a landmark classification branch and a landmark regression branch. Finally, the whole framework outputs three parts: component bounding box coordinates indicating the offsets between ground truth and the RPN proposal; component category showing the category of the proposal region, i.e. eyebrow, eye, nose and mouth; component landmark demonstrating the landmarks distribution with the proposal belongs to corresponding category.

3.3.1 Region-based Component Detection

In the initial part of the proposed framework, we adapt ResNet-50 model to extract feature maps of input images and share them with RPN(region proposal network)[12] to generate RoIs by using AnchorAlign, followed by fixing feature size

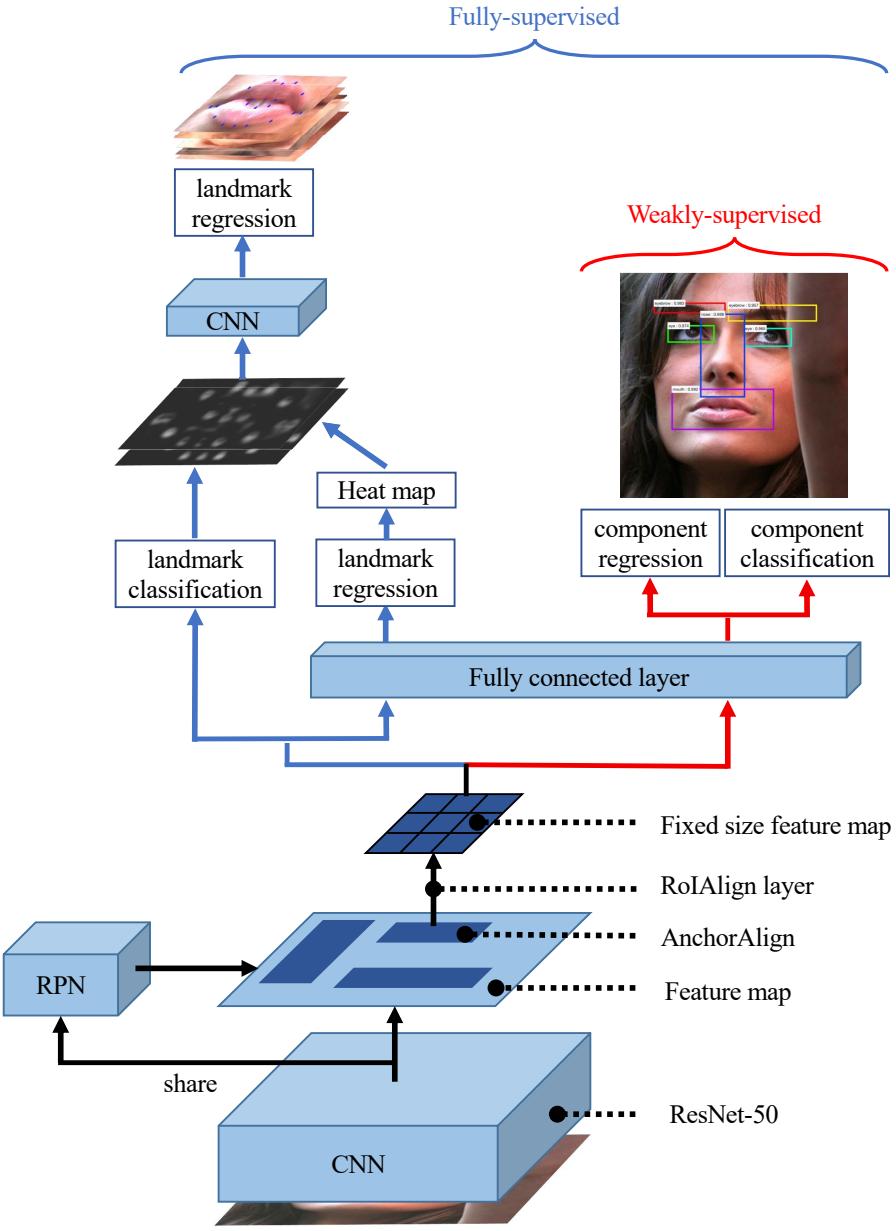


Figure 3.3 : The proposed LR-CNN architecture. Blue lines represent fully-supervised process, and red lines show weakly-supervised process.

through RoIAlign. Finally, the fixed feature map is put into fully connected layer to predict component class and bounding box. Next, we introduce the proposed AnchorAlign and RoIAlign model, for accurately localizing components and landmarks.

AnchorAlign: Anchor [12] is no longer a stranger in object detection area, which can address multiple scales and aspect ratios. Generally, we take 9 anchors for granted in detecting object of Faster R-CNN. However, for our facial component-landmark detection task, the situation has become different. Since the scales and ratios of components are different from normal object, we design AnchorAlign by changing multiple scales and aspect ratios to adapt to facial components, so that AnchorAlign model can improve the localization accuracy. Comparing to Anchor, AnchorAlign can be used in a specific application. Unlike Anchor settings of Faster-RCNN, the scales and ratios of AnchorAlign is not selected through manual experiments. It reasonably relies on component structure, since the bounding boxes of components are approximately {2:1, 1:2, 1:3} ratios for {eye, nose, eyebrow/mouse}. Additionally, the scales of components are also different from the object detection task. The areas of components occupying the entire face has a certain regularity below about 256x256, while ordinary objects randomly appear on the image and have unfixed sizes. Therefore, in the RPN, we design a specified AnchorAlign model as shown in Table 3.6 and gain a better result, compared with Anchor of Faster R-CNN and Mask R-CNN.

RoIAlign: RoIPool [12] is a standard operation for extracting a small feature map from each RoI, but it misalignments between the RoI and the extracted features. To address predicting pixel-accurate landmarks, we employ RoIAlign that removes the harsh quantization of RoIPool, properly aligning the extracted features with the input. This operation greatly increases the accuracy of landmark detection, while it may be not beneficial for component detection and classification.

In the output step, we leverage standard regression and classification method for object detection, as the same as Mask R-CNN. $\mathcal{L}_{reg}$ is loss of component bounding box regression, and $\mathcal{L}_{cls}$ is loss of component classification.

$$\mathcal{L}_{reg} = \sum_{i} class_i \cdot Smooth_{L1}(box_i, \hat{box_i}) \tag{3.5}$$

$$\mathcal{L}_{cls} = \sum_{i} Softmax(class_i, \hat{class_i}) \tag{3.6}$$

where, i is the index of a proposal region in an image. box_i and $\hat{box_i}$ are the predicted offset and true offset value between the i-th proposal region and its corresponding ground truth bounding box. $\hat{class_i}$ and $class_i$ indicate ground-truth and predicted classification of the proposal region. SmoothL1 and Softmax are shown in Eq.(3.7) and Eq.(3.8) respectively. Compared with traditional Euclidean distance, SmoothL1 can reduce the outlier effect, and in that way our model converges faster.

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5. & others \end{cases} \tag{3.7}$$

$$P(i) = \frac{exp(\theta_i^T x)}{\sum_{k=1}^{K} exp(\theta_k^T x)} \tag{3.8}$$

3.3.2 Two Branch Landmark Detection

In Fig. 3.3, the two-branch landmark detection model consists of landmark classification and landmark regression model.

For the classification branch, each of the landmark of a component is a one-hot $m * m$ binary key-point where only a single pixel is labeled as foreground. As we know, Mask R-CNN is a framework for instance segmentation, through extending

Faster R-CNN detector with a mask branch. This mask branch motivates us to classify every pixel of the fixed size feature map as a one-hot mask. If the pixel is landmark, the model output 1 for this pixel. If not, the model outputs 0. $landmark_{icls}$ is pixel-to-pixel classification of i-th proposed component region, as shown in Eq.(3.9).

$$landmark_{icls} = \sum_{i} crossentropy(p_{i(cls)}, p_{i(c\hat{o}mp)}) \tag{3.9}$$

where $p_{i(c\hat{o}mp)}$ is the ground truth of landmark for i-th proposal. $p_{i(cls)}$ is the predicted landmark relative coordinates according to the i-th proposal, calculated by classification. In the cross-entropy function t and o represent $p_{i(cls)}$ and $p_{i(c\hat{o}mp)}$, in Eq.(3.10).

$$\begin{aligned} crossentropy(t, o) = -[t * log(o) \\ + (1 - t) * log(1 - o)] \end{aligned} \tag{3.10}$$

For the regression branch, fully connected layer is used to infer component landmark as a regression task. The landmark vector derived from Table 3.1 can be regressed along with bounding boxes regression. Then, for every component, we use Gaussian model to generate the heat map on the basis of key-point regression results. $landmark_{ireg}$ is landmark vector regression of i-th proposed component region, as defined in Eq.(3.11).

$$landmark_{ireg} = \sum_{i} Smooth_{L1}(p_{i(reg)}, p_{i(c\hat{o}mp)}) \tag{3.11}$$

where $p_{i(reg)}$ is the predicted landmark relative coordinates according to the i-th proposal, calculated by regression. SmoothL1 function is shown in Eq.(3.7).

Finally, since each branch of facial component feature map is local spatial related, the feature maps of two branches are stacked over as a fused local receptive field. Then, the fused feature maps are fed into a CNN model for learning local spatial

structures. The final output of the two-branch model is also inferred by landmark regression model. All the components' landmarks are combined together to form a whole facial landmark.

Our loss for landmark detection is defined in Eq.(3.12):

$$\mathcal{L}_{land} = \sum_{i} Smooth_{L1}[f_{i(comp)}(landmark_{icls}, landmark_{ireg}), p_{i(c\hat{o}mp)}] \tag{3.12}$$

$f_{i(comp)}(landmark_{icls}, landmark_{ireg})$ and $p_{i(c\hat{o}mp)}$ are the predicted and ground truth landmark relative coordinates according to i-th proposed component region. $landmark_{icls}$ is calculated in cross-entropy, as shown in Eq.(3.9). $landmark_{ireg}$ regression function is in Eq.(3.11).

3.3.3 Weakly-supervised Learning and Loss

Our learning system consists of two parts, a main part of weakly-supervised learning and a small amount of fully supervised learning, as shown in Fig. 3.2. For weakly-supervised learning, we only utilizes our weakly labeled training data generated by DCGANs. The ground truth of weakly labeled data is component bounding box and class. For fully-supervised learning, a small amount of fully labeled data are also used for guiding back-propagation of neural network while training. Comparing to weakly-supervised data, the ground truth of fully-supervised data extra includes landmark vectors. In total, the number of weakly-supervised training data is much larger than that of fully-supervised data. Though weakly-labeled data without landmark coordinates ground truth, the result of landmark detection is also enhanced greatly. This is because weakly-supervised learning make component localization and recognition results more accurate. Since landmark detection results strongly rely on the predicted component detection results, the improvement of component detection results have a positive influence on landmark detection results. This is the

core idea of our landmark detection strategy based on the facial component regions.

We use a multi-task loss $\mathcal{L}$ on each weakly labeled data and fully labeled data to jointly train. Our loss function for an image is defined as:

$$\mathcal{L} = \mu(\lambda_1 \mathcal{L}_{reg} + \lambda_2 \mathcal{L}_{cls}) + (1 - \mu)\mathcal{L}_{land} \tag{3.13}$$

The hyper-parameter μ, λ_1, λ_1 in Eq.(3.13) control the balance among the three task losses. μ represents weakly-supervised weight, and it is determined by the number of weakly-supervised samples. λ_1 and λ_1 represent the loss weight of component bounding box regression $\mathcal{L}_{reg}$ and component classification $\mathcal{L}_{cls}$. Each of three terms has a loss weight indicated to adjust the affect of each loss part. All experiments use $\lambda_1 = \lambda_2 = 0.5$, $\mu = 1/3$, to make our network focus on landmark detection task.

3.3.4 Training

The LR-CNN framework can be trained end-to-end by back-propagation and SGD. We follow the "image-centric" sampling strategy from [11] to train our network. Each mini-batch arises from images that include positive and negative example anchors, which are defined in data preprocessing and augmentation.

The shared convolutional layers (ResNet-50) are initialized by pre-training a model for ImageNet 1000-class dataset [78], as is standard practice. We randomly initialize all other layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. We tune all layers with weakly labeled and fully labeled data. Each mini-batch has 2 images per GPU and each image has 64 sampled RoIs, with a ratio of 1:3 of positive to negative [17]. We train on 4 GPUs (so effective minibatch size is 16) for 160k iterations, with a learning rate of 0.02 which is decreased by 10 at the 120k iteration. The weight decay is 0.0001 and momentum is 0.9. Our framework is also fast to train. Training with ResNet-50 on takes

30 hours in the synchronized 4-GPU implementation (0.98s per mini-batch = 16 samples).

The training set for LR-CNN consists of two part: (1) Helen [73], IBUG [74], AFW [75], and LFPW [76] as fully-supervised training data; (2) our generated weakly-supervised training data.

3.4 Experiments and Discussions

In the experiment, several benchmark datasets are used to train and test on our model. Firstly, we introduce our datasets and evaluation measurements. Then, we compare our method with other state-of-the-art algorithms in both facial component detection task and landmark detection task. In the ablation experiments, we discuss the performance of each proposed method in detail.

We implement the Caffe [79] framework for all training, inference, and testing, in a regular PC (3.2-GHz 8-core CPU, 32G RAM, 4×12G GPU and Ubuntu 14.04). The whole training costs 30 hours on four NVIDIA TITAN X Pascals. Our algorithm reaches a speed at 0.21s per image while testing.

3.4.1 Dataset and Evaluation Metrics

The sum of training data for LR-CNN is about 66,000, including around 6,000 fully labeled data (ground truth: landmark, bounding box and category) and 60,000 weakly labeled data (ground truth: bounding box and category). Helen [73], IBUG [74], AFW [75], and LFPW [76] are fully-supervised training data, while our generated weakly labeled data are weakly-supervised training data. We evaluate our method on Helen, LFPW and 300-W test sets.

Our facial component and landmark detection algorithm is a multi-task method. The performance of methods is measured by two indexes, which is average precision (AP) for component detection and average error distance for landmark detection,

shown in Eq.(3.14) and Eq.(3.15) respectively.

$$AP = \int_0^1 p(x)dx \tag{3.14}$$

Average precision computes the average value of p(x) over the interval from x=0 to x=1 and is the area under the precision-recall curve.

For facial landmark detection, the normalized error rate is used to represent the good or bad of an algorithm in Eq.(3.15).

$$e = \frac{1}{N}\sum_{i=1}^{N}\frac{\frac{1}{M}\sum_{j=1}^{M}|p_{i,j} - g_{i,j}|_2}{|le_i - re_i|_2} \tag{3.15}$$

Here, N is the number of test samples and M is the number of landmarks ($M = 51$ in our experiment). $p_{i,j}$ and $g_{i,j}$ are the predicted coordinates and real coordinates of j-th landmark ground truth of i-th test sample respectively. le_i and re_i are the center coordinates of the left eye and the right eye of the i-th test sample, respectively.

3.4.2 Comparison with Other State-of-the-art Methods

To better understand the advantage of the proposed method, the experimental results of component detection and landmark detection are compared to other state-of-the-art algorithms separately, including both shallow models and deep models.

Performance of Facial Component Detection

In order to illustrate the result of component detection, we use average precision (AP) as a rule to compare each facial component detection precision and mean average precision (mAP) to all facial component detection precisions. We compare against several the-state-of-art object detection methods, such as Mask R-CNN [17], Faster R-CNN [12], SSD [13], YOLO [8] and YOLOv2 [9], shown in Table 3.2. All the methods are trained by generated weakly-supervised training set, and evaluated

Table 3.2 : The component detection result (AP) compared with other methods.

Algorithm	**mAP**	**eyebrow**	**eye**	**nose**	**mouth**
Faster R-CNN [12]	0.751	0.613	0.762	0.809	0.821
Mask R-CNN [17]	0.765	0.640	0.776	0.812	0.832
SSD [13]	0.764	0.631	0.784	0.808	0.834
YOLO [8]	0.646	0.487	0.674	0.727	0.696
YOLOv2 [9]	0.732	0.596	0.747	0.781	0.803
LR-CNN	**0.861**	**0.704**	**0.898**	**0.919**	**0.921**

on the test set mixed by Helen, LFPW, and 300-W test set. As we can see, our algorithm has the best performances in every category of facial component, especially by 0.919 and 0.921 in nose and mouse. Actually, for component detection task, our method is fully-supervised learning. However, for landmark detection, our method is weakly-supervised learning. The mainly reason why our method outperforms other algorithms is our data preprocessing, which auto-annotate and augment training data effectively. AnchorAlign also play an important role in component detection, because other models are not suitable for facial component detection. In addition, compared with other R-CNN methods, LR-CNN employs batch normalization after convolutional layers to avoid over-fitting problem. Comparing to Yolo, we use batch normalization in CNN model, and dropout in fully connected layers. In Fig. 3.5, we observe that the proposed method is robust to faces with large pose variation, lighting, and severe occlusion.

Performance of Facial Landmark Detection

As mentioned before, several benchmark test sets are used to evaluate performance of different methods, including Helen, LFPW, and 300-W. We compare to

Table 3.3 : The landmark detection results (average error distance) compared with other methods, on Helen, LFPW and 300-W test set separately.

	Algorithm	Helen	LFPW	300-W common	300-W challenge	300-W full
Non-deep models	RCPR [26]	5.93	6.56	6.18	17.26	8.35
	CFAN [66]	5.53	5.44	5.50	–	–
	SDM [80]	5.50	5.67	5.57	15.40	7.50
	CDM [81]	12.86	24.68	10.10	19.54	11.94
	GN-DPM [82]	5.69	5.92	5.78	–	–
	CFSS [83]	4.63	4.87	4.73	9.98	5.76
Deep models	RAR [67]	–	–	**4.12**	8.35	4.94
	LDDR [70]	4.76	4.67	–	–	–
	TCDCN [84]	4.60	–	4.80	8.60	5.54
	CFT [85]	4.75	–	4.82	10.06	5.85
	DAN[71]	–	–	4.42	**7.57**	5.03
	LR-CNN(51L)[1]	3.03	3.12	3.02	6.65	4.25
	LR-CNN(51L)[1]+RAR(17L)[2]	**3.71**	**4.07**	4.61	8.56	5.32
	LR-CNN(51L)[1]+RAR(17L)[2]+300W[3]	4.86	4.79	**4.12**	8.26	**4.92**

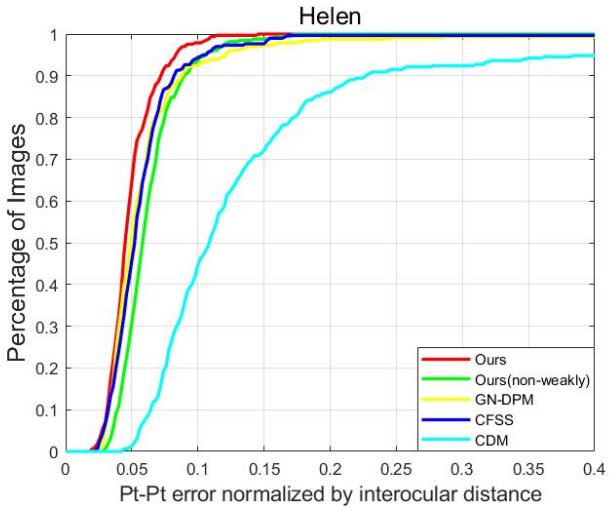
[1] LR-CNN(51L) mean that 51-landmark is the result of our method, trained on a small fully supervised and a weakly-supervised training set.

[2] RAR(17L) means that we use RAR method to detect another 17-landmark of jawline, for a clear and fair comparison with 68-landmark results.

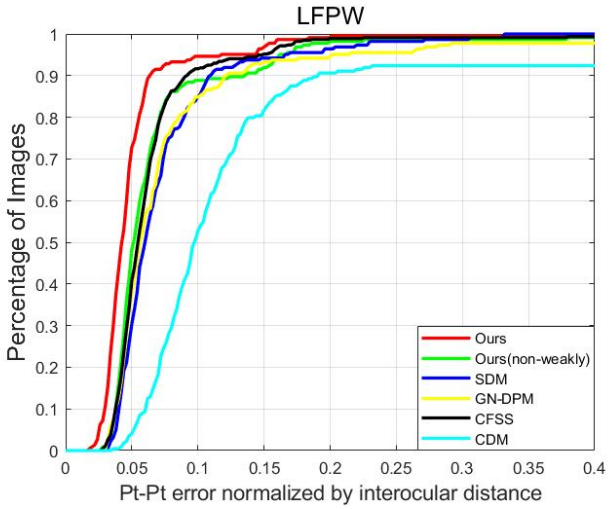
[3] 300W means that we additionally train on a training set with only 300-W, in order to compare with other methods which only focus on 300-W, like RAR and DAN.

non-deep models: (1) Robust Cascaded Pose Regression (RCPR) [26] using the publicly available implementation and parameter settings; (2) Coarse-to-Fine Auto-Encoder Networks (CFAN) [66], which focuses on real-time face alignment; (3) Supervised Descent Method (SDM) [80]; (4) Cascaded Deformable Shape Model (CDM) [81]; (5) Gauss-Newton Deformable Part Models (GN-DPM) [82]; (6) Coarse-to-fine shape searching (CFSS) [83]. And we also compare against deep models: (7) Recurrent Attentive-Refinement Networks (RAR) [67]; (8) Local Deep Descriptor Regression (LDDR) [70]; (9) Tasks-Constrained Deep Convolutional Network (TCDCN) [84]; (10) Coarse-to-fine training algorithm (CFT) [85]. (11) Deep alignment network (DAN)[71]. Given that our method is aimed at 51-point landmark, we combine LR-CNN of 51-point (eye, eyebrow, nose, mouse) with RAR algorithm of 17-point (jawline) for testing on 68-point landmark detection, for comparison to other approached listed above. As shown in Table 3.3, average error distances of all algorithms are measured by 68-point landmark detection result.

Evaluation on Helen: It is obvious that deep learning models produces a superior performance to shallow models on Helen test set, in Table 3.3. And LR-CNN(51-landmark)+RAR(17-landmark) outperforms all other state-of-the-art methods, far below CFSS and TCDCN. The proposed model perform best on Helen test set with average error distance less than 3. Fig. 3.4 (a) shows several algorithms' cumulative error curves. As we can see, our algorithm performs better than other state-of-art algorithms. Fig. 3.5 shows several examples of our detection, including component and landmark. We observe that the proposed method is robust to lighting and severe occlusion. It is worth pointing out that the size of input images is non-restricted, which means that LR-CNN can cope with both low-resolution and high-resolution images. We make a comparison between our method and LDDR tested on images with extreme illuminations and occlusions, as shown in Fig. 3.6. The results of other approaches are unreliable and rely on unfounded guesswork, while our method only



(a) evaluation on Helen test set



(b) evaluation on LFPW test set

Figure 3.4 : Cumulative error curves. The red line (Ours) is our weakly-supervised method and the green one (Ours(non-weakly)) is our LR-CNN without generated weakly labeled data.

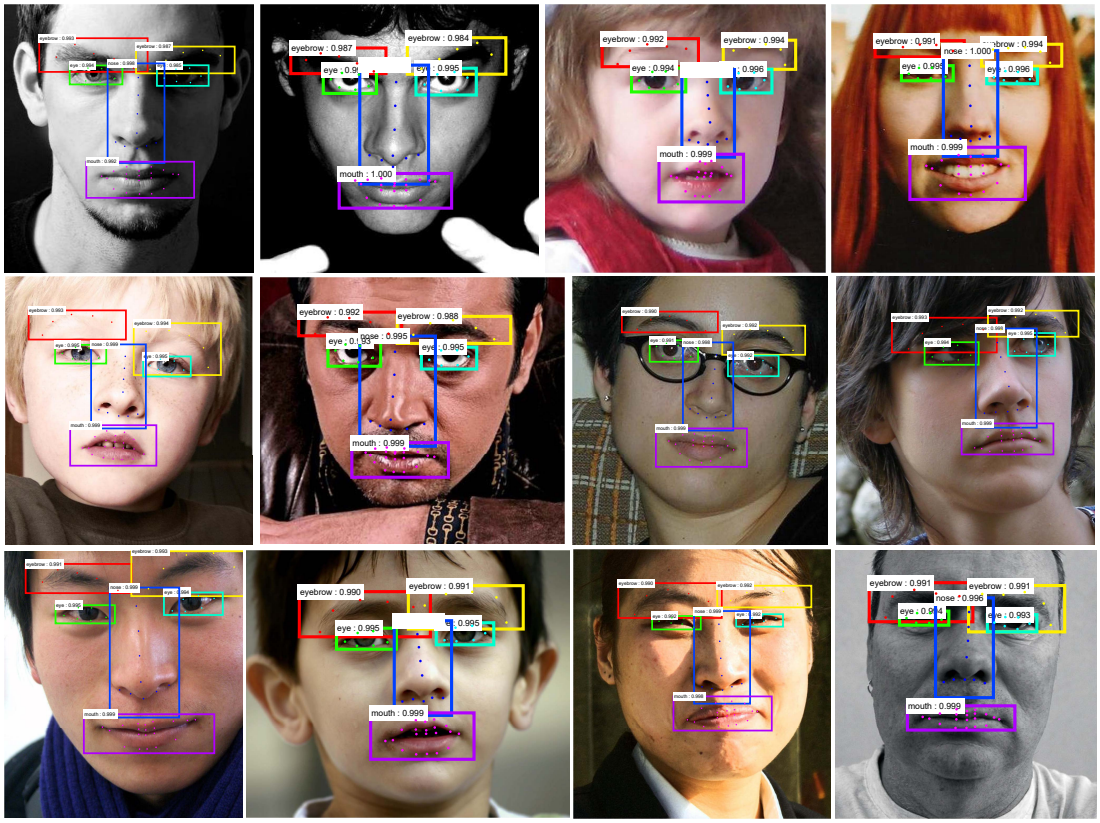


Figure 3.5 : Some detection results on Helen testset (the first row), LFPW testset (the second row) and 300-W testset (the rest row). The different color bounding boxed show facial component detection and the text and number pairs denote the probabilities of bounding boxes belong to the corresponding categories. The predicted landmark coordinates are plotted by different color points corresponding their component.

detects the landmarks in visible component regions. Therefore, the detection result of our method is more reasonable and intuitive.

Evaluation on LFPW: In addition to Helen, we also tested on LFPW test set and observe similar trend as on the Helen test set. Fig. 3.4 (b) also demonstrates the superiority of our method compared with some released code of other algorithm. Fig. 3.5 and Fig. 3.6 also indicate some detection examples using LR-CNN method.

Evaluation on 300-W: We report the landmark detection results of LR-CNN method as well as results of current state-of-the-art methods on the 300-W testing set. Compared with the performance on Helen and LFPW test set, LR-CNN(51-landmark)+RAR(17-landmark) result on 300-W is barely satisfactory but still outperforms other state-of-art algorithms, except RAR and DAN. Because both RAR and DAN are trained by 300-W training set, while our model is only trained by limited fully labeled data. TCDCN pre-trains their facial landmark detection model on the Multi-Attribute Facial Landmark database which consists of 19,000 face images with multiple facial attributes information, and tunes their model on 300-W. On the other hand, the training set of our model doesn't contain 300-W data set. What's more, RAR and DAN only focus on 300-W data set and has no test on other benchmark dataset, and our model has wider applicability than those. Therefore, for fair comparison with RAR, TCDAN and DAN, we only use 300-W training set as fully supervised part to train our model, and gain a improvement on 300-W test by about 0.5 on 300-W common set and by 0.4 on 300-W full test set, as shown in the last line of Table 3.3. Since the original training set including Helen and LFPW training set are replaced by 300-W, the results on Helen and LFPW decline reasonably, but are also better than many other Non-deep models. The reason why DAN outperforms ours in 300-W challenge test set is that DAN is also a deep-learning-based algorithm and it is a robust alignment method of which network input are entire face images. And cascading complexity of DAN is higher than our method, undoubtedly has better performance than our straightforward regression algorithm. DAN are trained sequentially while ours is an end-to-end architecture and easily trained.

In addition, our method only predicts visible components and landmarks, while other algorithms guess the facial landmarks of which components are occluded. It is obvious that guesswork is unreliable and useless, as shown in Fig. 3.6. In fact, this

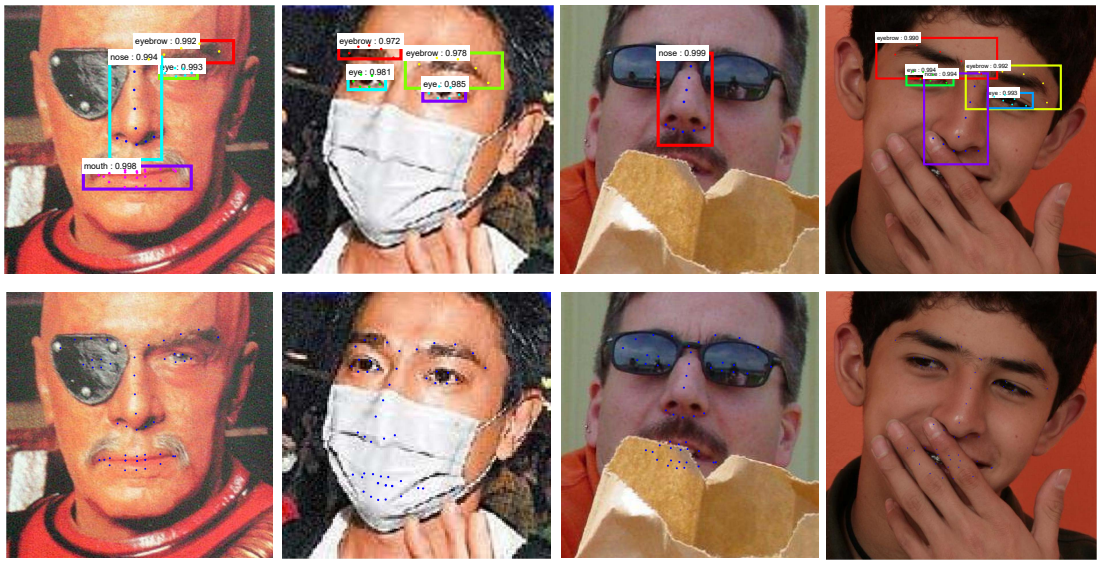


Figure 3.6 : Some detection results of our method on extreme illuminations and occlusions in first row. The second row shows the result of LDDR.

inaccurate estimation of landmarks is based on facial structure feature. Oppositely, our system is able to detect facial landmark precisely because our predicted landmark is based on our previous component detection results, which are trained effectively by weakly-supervised data.

3.4.3 Ablation Study

We run a number of ablations to analyze weakly-supervised LR-CNN. Results are shown in every subsections and discussed in detail next.

Weakly-supervised v.s. Non-weakly-supervised

To demonstrate the necessity of weakly-supervised learning in our architecture, we compare the results of two groups of experiments, which are with weakly-supervised method and without weakly-supervised method. In this controlled experiment, weakly-supervised method uses fully supervised data and weakly-supervised data to train our model while non-weakly-supervised method only uses fully super-

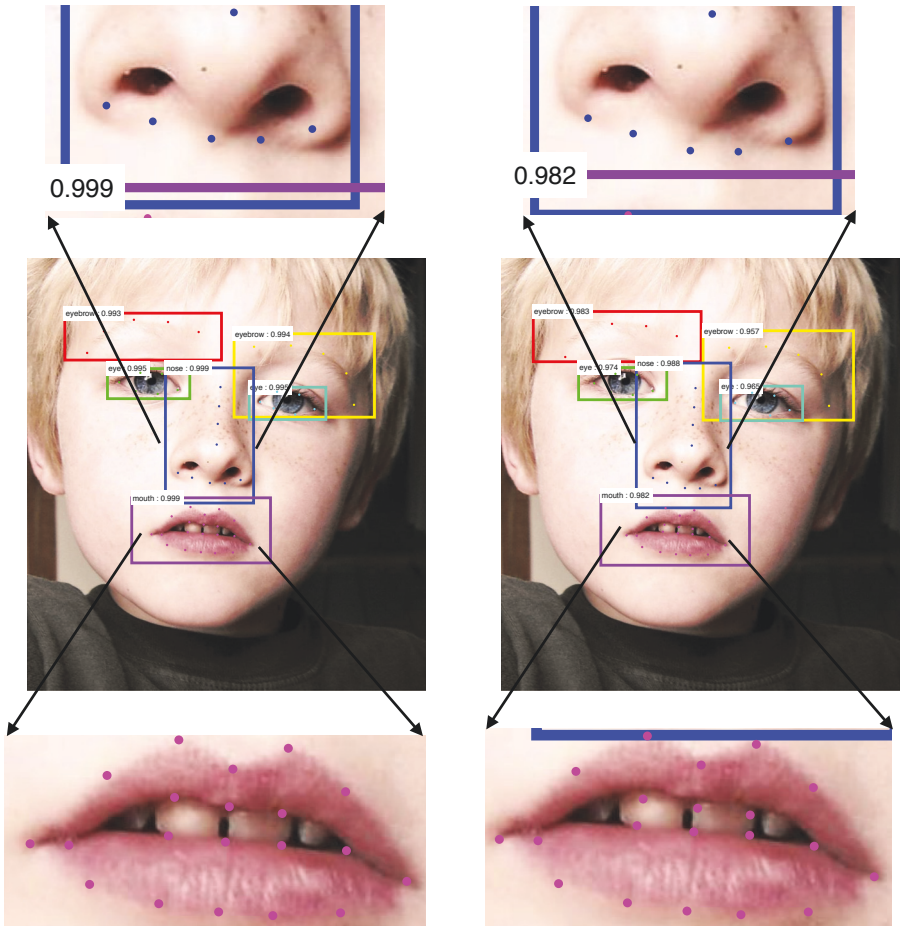


Figure 3.7 : Left image is weakly-supervised result, right image is non-weakly-supervised result.

vised data to train the same architecture. We test our method and ours(non-weakly) on facial component detection, and also on facial landmark detection with Helen, LFPW and 300-W as shown, as shown in Table 3.4 and Table 3.5 respectively. We can see that Ours outperforms Ours(non-weakly) in both facial component and landmark detection. For component detection, our generated weakly-supervised data can be regarded as data augmentation. As the number of training data increasing, our model improve a lot. Since landmark detection results strongly rely on the predicted component detection results, the improvement of component detection results have a positive influence on landmark detection results. In addition, Fig. 3.4 shows their

cumulative error curves. The red line (Ours) is our weakly-supervised method and the green one (Ours(non-weakly)) is our LR-CNN without generated weakly labeled data. Our algorithms performs better than other state-of-the-art algorithms, and weakly-supervised model performs much better than non-weakly-supervised model.

As shown in Fig. 3.7, the left image is the detection result of our weakly-supervised method while the right one shows our pipeline with only fully supervised learning. Both ours and ours(non-weakly) can detect six parts of facial components, but localization result of ours(non-weakly) is much worse than that of weakly-supervised LR-CNN. For instance, weakly-supervised LR-CNN can locate landmarks of the nose accurately while fully supervised method provides an unsatisfied result, especially at the teeth position of the mouth.

Table 3.4 : The component detection result (AP) compared with other methods.

Algorithm	**mAP**	**eyebrow**	**eye**	**nose**	**mouth**
Ours	**0.861**	**0.704**	**0.898**	**0.919**	**0.921**
Ours(non-weakly)	0.589	0.471	0.724	0.557	0.602
Ours(VGG [86])	0.846	0.680	0.889	0.909	0.907
Ours(NIN [87])	0.799	0.641	0.832	0.868	0.855
Ours(ZF [88])	0.833	0.664	0.868	0.901	0.898

Comparision with Other Cascaded CNNs

We also list detection results of several cascaded CNN models including VGG [86], NIN [87], ZF [88] and ResNet[17] as shown in Table 3.4 and Table 3.5. In Table 3.4, there is no doubt that ours with ResNet model outperforms other CNN models.

Table 3.5 : The 51-landmark detection result (average error distance) compared with our methods with variants on different test set.

Algorithm	Helen	LFPW	300-W full
Ours	**3.03**	**3.12**	**4.25**
Ours(non-weakly)	3.77	4.25	6.68
Ours(VGG [86])	3.10	3.25	4.69
Ours(NIN [87])	3.22	3.27	4.85
Ours(ZF [88])	3.18	3.24	4.76

The Roles of AnchorAlign and ROIAlign

To investigate the behavior of AnchorAlign and RoIAlign, we conducted several ablation studies. First, we show the effect of different Anchors for component detection and landmark detection results. In this experiment, we use the ResNet-50 model with weakly-supervised learning, which is our standard settings. As shown in Table 3.6, our proposed AnchorAlign is compared with Anchor which is presented by Ren [12]. Anchor changes in 4 kinds of settings, and AnchorAlign changes in 3 kinds of settings. As for the result of both component detection and landmark detection, the best performance of Anchor is still below the worst performance of AnchorAlign. By default we use $\{64^2, 128^2, 256^2\}$ scales and {2:1, 1:2, 1:3} ratios (0.961 mAP and 3.03 average error distance on Helen test set). The mAP is higher if using this kind setting of specified scales or ratios, The landmark detection has the same trend. What's more, the effect of ratio is larger than that of scale, The mAP and average error distance is 0.836 and 3.31 when we only change the ratios. But when we only change the scales, the result become much better, suggesting that scales and aspect ratios are not disentangled dimensions for the detection accuracy.

Next, we evaluate three kinds of RoI layer to demonstrate which operation is the best for our system. A comparison experiment of RoIPool, RoIWarp and RoIAlign layer is shown in Table 3.7. RoIAlign improves component detection mAP by about 2 points over RoIWarp. RoIAlign reduces Helen and LFPW landmark detection by about 0.7 below RoIWarp, with much of the gain coming at 300-W benchmark. RoIPool performs on par with RoIWarp and also much worse than RoIAlign. This also highlights that proper alignment is the key.

Table 3.6 : Detection results of our algorithm on Helen test set using different settings of anchors. The network is ResNet-50.

settings	**scales**	**ratios**	**mAP**	**Helen**
Anchor	256^2	1:1	0.789	4.01
	256^2	{2:1, 1:1, 1:2}	0.806	3.88
	$\{128^2, 256^2, 521^2\}$	1:1	0.817	3.59
	$\{128^2, 256^2, 521^2\}$	{2:1, 1:1, 1:2}	0.829	3.35
Anchor Align	$\{64^2, 128^2, 256^2\}$	{2:1, 1:1, 1:2}	0.836	3.31
	$\{128^2, 256^2, 521^2\}$	{2:1, 1:2, 1:3}	0.857	3.10
	$\{64^2, 128^2, 256^2\}$	**{2:1, 1:2, 1:3}**	**0.861**	**3.03**

Table 3.7 : Detection results with various RoI layers.

	mAP	**Helen**	**LFPW**	**300-W full**
RoIPool[12]	0.841	3.77	3.98	5.68
RoIWarp[89]	0.843	3.80	3.75	5.69
RoIAlign[17]	**0.861**	**3.03**	**3.12**	**4.25**

Table 3.8 : Landmark detection results with various architectures.

	Helen	**LFPW**	**300-W full**
classification branch	4.31	4.58	7.17
regression branch	3.84	3.76	5.92
two-branch	**3.03**	**3.12**	**4.25**

Two-branch v.s. One-branch

In the architecture, we propose a two-branch model for landmark detection. This ablation experiment demonstrates the superiority of two branches, as shown in Table 3.8. We compare our two-branch model with two single-branch landmark detection models respectively. For the classification detection model, this branch directly outputs landmark result when we remove the regression branch. For the regression detection model, we eliminate the classification branch. And we train and test them separately. All these architectures are trained via weakly-supervised learning and standard settings. In Table 3.8, it is obvious that two-branch architecture outperforms other two models in three datasets. The regression model has better performance than the classification model by about 1.2 average error distance in 300-W full test set, which is much larger the gap between two-branch model and the regression branch. This illustrates that the two branches complement each other. It is also illustrates the regression branch is more suitable for difficult task than the classification branch, as 330-W is more challenging than Helen and LFPW. This is the reason why we also use regression method on the last layer after merging feature maps. The improvement of combination of two branches illustrates that these two method overlap and complement each other.

3.5 Conclusion

In this chapter, we propose an end-to-end weakly-supervised LR-CNN framework based on region-based deep learning for facial component and landmark detection to tackle large occlusion and limited training data problems. Our presented method use DCGANs and automatic labeling to generate weakly-supervised training data, which solve the problem of small training set. Moreover, we design a two-branch architecture that makes it possible to detect facial components and predict facial landmarks simultaneously. For large area occluded faces, many existing face detectors are failed to detect any faces in the picture while ours could detect visible facial components and predict corresponding landmarks without any no sense guesswork. Experiments on benchmark datasets reveal that our method outperforms most of the state-of-art algorithms. One of the reason may be that our weakly-supervised framework is able to predict more accurate box coordinates, which thanks to weakly-supervised augmentation and data preprocessing by using generative models. This is also because that our two-branch architecture can extract more discriminative features by using classification and regression branch. We also discuss the advantages of our weakly-supervised learning compared with fully supervised learning. In addition, a comparison experiment among different AnchorAlign, RoIAlign and cascaded CNN models demonstrates the feasibility of our weakly-supervised algorithm successfully.

Chapter 4

Multi-camera Multi-player Tracking with Deep Player Identification

4.1 Introduction

After exploring facial object with a strong pattern, we intend to study some object with more variations and challenges. Hence, the second object of our research is the sports player in the real-world scenario. In this chapter, the task is multi-player tracking by detection and identification under multiple cameras. Multi-camera sports video analysis has received increasing interest in recent years. The resulting analysis enable various applications including enhancing sports videos broadcast[90, 91, 92], reconstructing 3D match [93, 94], providing interactive contents for audiences[95, 96, 97], and collecting game data to support coaches to make tactical analysis[98]. For the ultimate sports game understanding, player tracking is a fundamental task. Player tracking in sports videos can be regarded as a multi-target, multi-camera tracking (MTMCT) task, which aims to determine the position of every target at all times, and to ensemble multi-camera trajectories from multi-view video streams.

A critical influencing factor on MTMCT accuracy is the identity switch, which is caused by the players interchanging frequently in the camera view. This problem often occurs in sports video, such as the mistracking in Figure 4.1. Therefore, it is necessary to handle the identity switches for the reliable tracking.

Many methods attempt to resolve this problem by designing features. Possegger *et al.* [99] defined a color model to the drifting problem. Shen *et al.* [43] presented

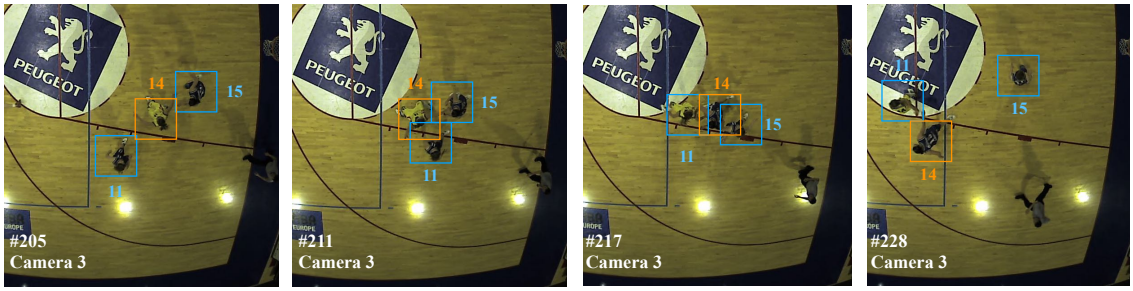


Figure 4.1 : An example of identity switch in player tracking. The identities of Player 11, Player 14 and Player 15 exchange when players interchange.

a set of game context features and decomposed the players' motion likelihood into independent per-player models. Liu *et al.* [100] introduced a patch-based appearance model and the spatial–temporal similarity measurement for adaptive pedestrian tracking. However, these approaches are only based on the hand-crafted appearance models, which fail to distinguish between different players due to the similar appearance of teammates. Therefore, we explore the distinguishable representation for the player identity by deep learning to tackle the identity switches.

Player identification is even more challenging in the real-world sports video. Unlike pedestrians[47] and vehicles[101, 102] that have relatively predictable motion patterns, players tend to confuse their opponents with abrupt moves in directions and unexpected changes in velocity. Meanwhile, compared with person re-identification, commonly used features for re-identification, *e.g.*, color and gait, become invalid in the scenario of player identification. To address the player tracking incorporating with identification, we figure out the following challenges:

- Jersey numbers encounter serious deformation due to player's movement. Low resolution and variant image size caused by players' distance to cameras also make the jersey number difficult to read.

- Players' similar appearance due to the similar uniform, body-shape variation,

erratic motion, spectator interference, and the illumination variation make it difficult to track and identify players reliably.

- The player's 3D localization is seriously affected by heavy occlusions and foreground noise.
- The inaccurate player identity leads to mis-tracking and mis-association when players frequently interchange.

These challenges cannot be comprehensively solved by existing studies, which makes it desirable to propose a framework that learns players' identities with deep representation and improves the tracker by the identity information. In addition, most existing algorithms of multiple object tracking require the manually labeled or automatically detected bounding boxes as input. Nevertheless, current off-the-shelf detectors fail to achieve the optimal identification results due to the similar appearance of teammates. In both academia and industry, there is a lack of an integral and reliable solution from multi-target multi-camera detection, to identification, to tracking.

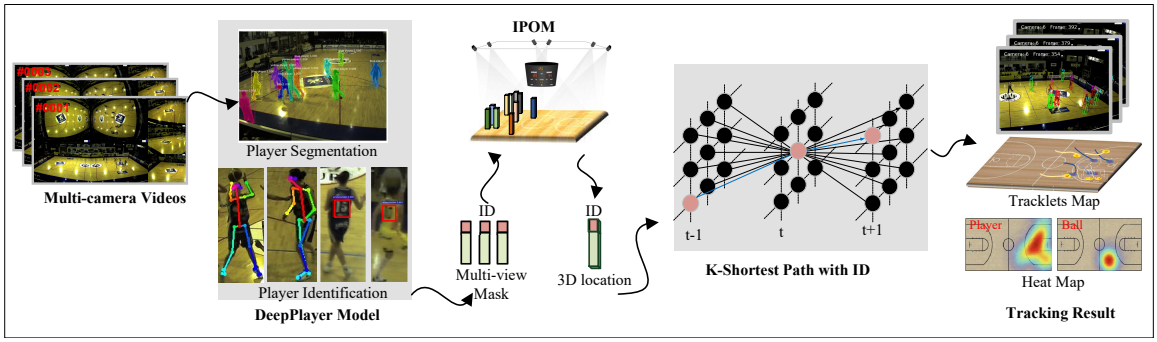


Figure 4.2 : The overall framework consists of three modules: (1) the DeepPlayer model for players' 2D localization, instance segmentation and identification, (2) the IPOM model to localize players' 3D coordinates with ID, (3) the KSP-ID model for the ID-enhanced multi-player tracking.

Considering the aforementioned problems and the limit of previous approaches, we propose a robust multi-camera multi-player tracking with identification framework (MCMPTI). To our knowledge, this is the first attempt to introduce deep-learning-based player identification into the tracking. As shown in Figure 4.2, the framework consists of three parts: (1) a Deep Player Identification (DeepPlayer) model for players' identification, 2D localization and segmentation; (2) an Individual Probability Occupancy Map (IPOM) model for players' 3D localization with ID; and (3) a K-Shortest Path with ID (KSP-ID) model for the ID-enhanced multi-player tracking.

DeepPlayer: In the proposed DeepPlayer model, we develop a Cascade Mask RCNN model and a PoseID model to jointly identity players. Unlike the conventional jersey number recognition, the Cascade Mask RCNN model recognizes team classes and jersey numbers with a coarse-to-fine region-based CNN model. Different from the original Mask RCNN method, the Cascade Mask RCNN model shares the CNNs to two Region Proposal Networks (P-RPN and J-RPN) and outputs team class and jersey number. In addition, we propose a pose-guided partial feature embedding to represent the distinguishable information in the PoseID model.

IPOM: To solve the occlusions and noises problems, on the basis of the POM algorithm, we design a new strategy to add the player IDs in each location in the proposed IPOM model. The IPOM model can perform on both the identified and ambiguous IDs, and output the 3D location and player ID of each player, with the corresponding probability.

KSP-ID: For the identity switch problem, we introduce player ID into the original KSP algorithm, and propose a player ID correlation coefficient to measure the identity similarity between two nodes in consecutive frames. The edge cost contains both the marginal posterior probability of the presence of the player and the proposed

player ID correlation coefficient of the node pairs.

To summarize, this chapter provides a reliable solution for the real-world multi-player tracking. The main contributions is five-fold:

- We propose a robust multi-camera multi-player tracking with identification framework, from detection, to identification, to tracking.
- We propose a DeepPlayer model to extract the distinguishable player ID, considering pose-guided partial features, team class, and jersey number.
- We propose a IPOM model for 3D localization with ID.
- We propose a KSP-ID model with a player ID correlation coefficient.
- Experiments on the APIDIS and STU datasets illustrate that our method achieves compelling performance compared with the state-of-the-arts.

In Figure 4.2, we propose a robust MCMPTI framework which can identify players and improve the multi-player tracking by the players' ID. The input is the multi-camera videos and the output is the multi-player tracking result. The framework mainly consists of three models: (1) the DeepPlayer model for player 2D localization, instance segmentation and identification in each frame each view, (2) the IPOM model for localizing the players' 3D coordinates, and (3) the KSP-ID model for the multi-player tracking according to the players' 3D location and ID. In the following, we will introduce each model in detail.

4.2 DeepPlayer Model for Player Identification

The DeepPlayer model is proposed to obtain each player's ID. This model contains two parts: (1) the Cascade Mask RCNN for coarse-grained player segmentation and fine-grained jersey number recognition; (2) the player segmentation embedding

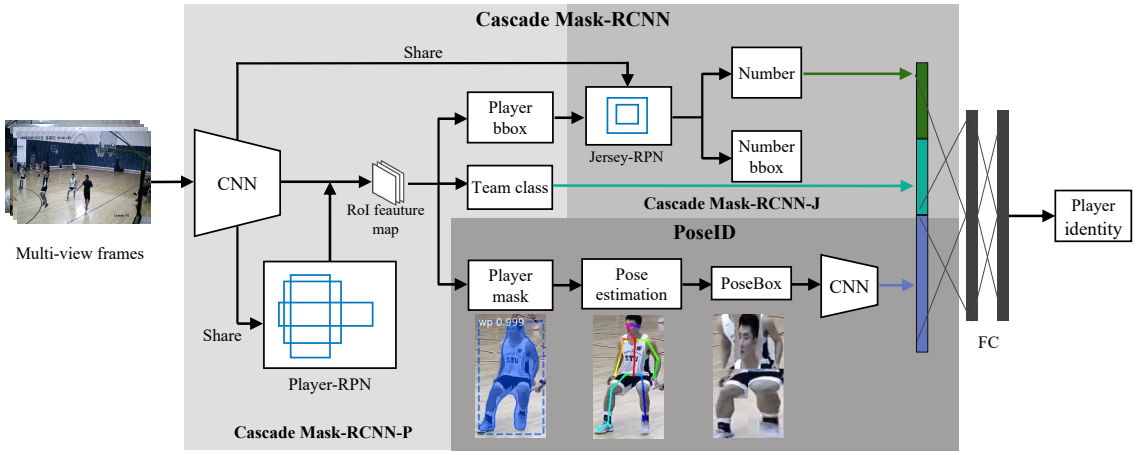


Figure 4.3 : The architecture of the DeepPlayer model. This model consists of two part: (1) the Cascade Mask RCNN for coarse-grained player detection(Cascade Mask-RCNN-P) and fine-grained jersey number recognition(Cascade Mask-RCNN-J); (2) the player mask embedding into the deep representation using PoseID. Finally, the player identity is decided by the jersey number class, the team class and the deep representation.

into a deep representation through posed-guided partial feature embedding (PoseID). As shown in Figure 4.3, the Cascade Mask RCNN model firstly detects each player and classifies the player by team, and segments the player's instance mask. Then the model recognizes the jersey number from the detected player bounding box. In a nutshell, we obtain the team class and number class of the detected player if the jersey number can be detected. Otherwise, we extract the deep representation of the detected player by PoseID. Finally, we concat the jersey number class, the team class and the pose-guided partial feature embedding to infer the player ID after fully connected layers.

4.2.1 Cascade Mask RCNN

In terms of a player with readable jersey number, we formulate player identification as player jersey number detection and classification, since the jersey num-

ber/class can provide unique ID information. If a rough detector is employed to detect players and jersey numbers directly, it will produce inaccurate region proposals and mis-association of player and jersey number. Therefore, we extend and modify the Mask RCNN[103], which is a CNN-based detector for detection and instance segmentation. We propose a Cascade Mask RCNN model, which includes two parts: (1) Cascade Mask RCNN-P for player detection and instance segmentation under coarse granularity, (2) Cascade Mask RCNN-J for jersey number detection and recognition under fine granularity. Firstly, we detect all players from multi-view images to obtain the bounding box, the team category and the instance segmentation of each player. Then, the player bounding boxes are put into a jersey number localization model to detect number location, followed by a number classification model to recognize the jersey number. To reduce the duplicate calculation, both RPN of player and RPN of jersey number share the CNN feature map of the input image. Finally, we save the team class (a 3-dimensional vector) and the jersey number (a 24-dimensional vector) for subsequent processes.

Player Detection and Instance Segmentation

First, we leverage a ResNet-50 [104] model to extract CNN feature of input frames and share the feature to Player region proposal network (P-RPN) and Jersey number region proposal network (J-RPN), to generate Region of Interest (RoI) feature map of the player by Anchor. Then, the player bounding box (bbox) is predicted by regression, and the team class is predicted by classification. The player mask is a one-hot $m * m$ binary key-point where the pixels that belong to the mask are labeled as foreground. The team class contains two teams and a referee. The background and audience are defined as background. $\mathcal{L}_{cls}(p_i^c, g_i^c)$ is the loss of team classification, and $g_i^c \cdot \mathcal{L}_{loc}(p_i^l, g_i^l)$ is the loss of player bounding box regression, and $\mathcal{L}_{mask}(p_i^m, g_i^m)$ is the loss of player mask. The loss of player $\mathcal{L}_{ply}$ is defined as:

$$\mathcal{L}_{ply} = \sum_i \mathcal{L}_{cls}(p_i^c, g_i^c) + \sum_i g_i^c \cdot \mathcal{L}_{loc}(p_i^l, g_i^l) + \sum_i \mathcal{L}_{mask}(p_i^m, g_i^m). \tag{4.1}$$

g_i^c and p_i^c indicate ground-truth and predicted classification of the proposal region. p_i^l is the predicted vector representing the offset between the i-th proposal and its corresponding ground-truth bounding box, and g_i^l is the true offset value between them. g_i^m and p_i^m represent ground-truth and predicted mask of the proposal region. We use Softmax as the loss function of $\mathcal{L}_{cls}$ and SmoothL1 as the loss function of $\mathcal{L}_{loc}$, respectively. Compared with Euclidean distance, SmoothL1 can reduce the outlier effect, and make the model converge faster.

$$Softmax(i) = \frac{exp(\theta_i^T y)}{\sum_{k=1}^{K} exp(\theta_k^T y)}, \tag{4.2}$$

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & others \end{cases}. \tag{4.3}$$

We adopt the cross-entropy as the loss function of $\mathcal{L}_{mask}$. In the cross-entropy function t and o represent p_i^c and g_i^c:

$$\begin{aligned} crossentropy(t, o) = -[t * log(o) \\ + (1 - t) * log(1 - o)]. \end{aligned} \tag{4.4}$$

Jersey Number Recognition

After the player detection, the J-RPN calculates the jersey number bbox from the detected player bbox, and classifies the jersey number bbox. In this work, we treat jersey number recognition as a detection problem. We model all occurring jersey numbers as a separate class. In this case, this is a 24-class classification problem, as not all numbers appear in the dataset. For positive samples, there is a

restriction that the jersey number bounding box must be included in the responding player bounding box. The loss of jersey number $\mathcal{L}_{jrs}$ is defined as:

$$\mathcal{L}_{jrs} = \sum_i \mathcal{L}_{cls}(j_i^c, h_i^c) + \sum_i h_i^c \cdot \mathcal{L}_{loc}(j_i^l, h_i^l). \tag{4.5}$$

h_i^c and j_i^c indicate ground-truth and predicted the classification of the proposal region. j_i^l and h_i^l are the predicted offset and true offset value between the i-th proposal region and its corresponding ground truth bounding box.

Full Objective

Our full objective for an image is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ply} + \lambda_2 \mathcal{L}_{jrs}. \tag{4.6}$$

The hyper-parameter λ_1, λ_2 control the balance between the two losses. Note that, we train the P-RPN module only on fully labled data, while we train the J-RPN module on both weakly labeled data and fully labeled data. The weak labels are player bounding boxes, and the full labels are jersey number bounding boxes. For the APIDIS dataset, the training set includes 2000 fully ground-truth labels and 10000 weakly ground-truth labels. This kind of weakly supervised learning[35] can improve the efficiency of the network training compared to supervised learning.

4.2.2 Pose-guided Partial Feature Embedding

Besides the jersey number and team class, we develop a posed-guided partial feature to represent a specific player to assist player identification.

What Makes a Player Distinguishable?

It would be interesting to find what makes a player different from others. To find the regions that distinguish a player from his teammates, we implement GRAD-

CAM[105] on the player bounding box classification. We directly train an Inception V4 model and the class is the identity of the player. We compute the gradient of the class output value with respect to the feature map. Then, we weigh the output feature map with the computed gradient values, and average the weighed feature map along the channel dimension resulting in a heat map. Through observing the heat map, we reach a conclusion similar to [39]. Discernible details always appear in similar positions for each player. Similarly, the head part, sleeves, socks, and shoes look distinctive to the players. This is interpretable. Therefore, we propose the pose-guided partial feature embedding (PoseID) for the player identification.

Pose Estimation

Player occlusion may cause multiple players appearing in one bounding box detected by the body keypoint detector. This may lead to incorrect pose estimation owing to the player bounding boxes with impurities. Different from others' pose estimation by using detected player bounding box [39, 106], this paper localizes the keypionts from the pure player mask generated by our Cascade Mask RCNN. This will avoid the incorrect pose estimation because the player mask is instance segmentation, which contains only one object. We adopt the off-the-shelf model of OpenPose [107], which is an effective tool to detect the 2D pose of people in an image. We leverage 25-keypoint body/foot keypoint estimation. A set of 25 body joints are detected, *i.e.*, face, neck, left and right shoulders, left and right elbows, left and right wrists, left and right hips, left and right knees, left and right ankles, and left and right feet, as shown in Figure 4.3.

PoseBox Construction and Embedding

According to aforementioned player mask and pose estimation, we build a set of PoseBoxs, as shown in Figure 4.4. The PoseBox can eliminate background noise and correct the pose variations.

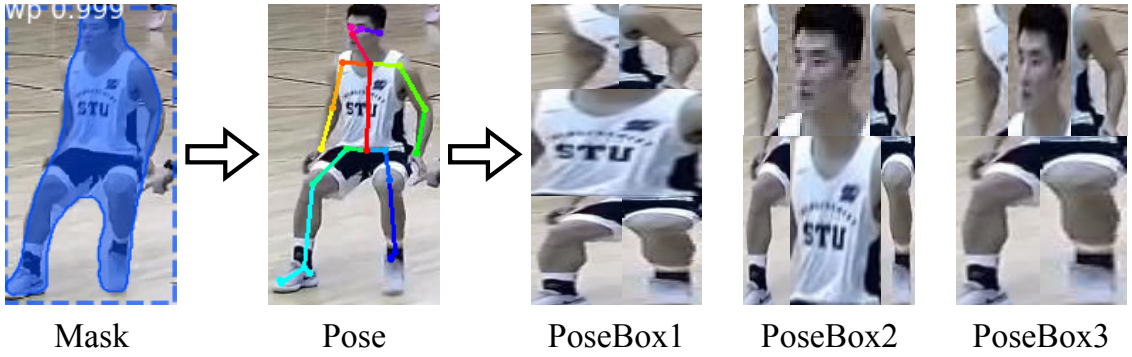


Figure 4.4 : PoseBox construction. Given a mask, the player pose is estimated by OpenPose. PoseBox1 = torso + arms + legs; PoseBox2 = head + torso + arms + legs; PoseBox3 = head + arms + legs.

- PoseBox 1. This type is designed by [106]. It includes the torso, two arms and two legs. An arm consists of the upper and lower arms. A leg is comprised of the upper and the lower leg sub-modules.
- PoseBox 2. On the basis of PoseBox 1, we add the face. In our experiment, we show that PoseBox 2 is superior to PoseBox 1 thanks to the enriched information brought by the face.
- PoseBox 3. Based on PoseBox 2, we put subtract the torso box. We find that the subtraction of the torso brings performance increase. In our case, this increase is explicable because of the same jersey color.

After constructing the PoseBox, we adopt the ResNet-50 [104] to extract the convolutional feature and then flat it to a 2048-dimensional vector.

4.2.3 Player Identification

To the end, we obtain the team class, jersey number class and pose-guided partial feature embedding. The team class can be described as a 3-dimensional vector $\mathbf{z_1}$, containing each class with its probability. The jersey number class is a 24-

dimensional vector $\mathbf{z_2}$, containing each class with its probability. The pose-guided partial feature can be described as a 2048-dimensional vector $\mathbf{z_3}$, an embedding of the PoseBox.

We concat these three vectors as the input, and construct a Softmax classifier with two fully connected layers to predict player identity. Since the confidence of the three vectors is different, the input vector is defined below:

$$\mathbf{z} = \mu_1 \mathbf{z_1} + \mu_2 \mathbf{z_2} + \mu_3 \mathbf{z_3}, \tag{4.7}$$

where μ_1, μ_2, μ_3 control the weight of team class $\mathbf{z_1}$, jersey number class $\mathbf{z_2}$, pose-guided partial feature $\mathbf{z_3}$ respectively. In our case, we set $\mu_1 = 1$, $\mu_2 = 1/2$, $\mu_3 = 1/4$, as the error increases progressively.

4.3 Individual POM with ID

In order to provide player's 3D location with ID, we develop an effective module, Individual Probabilistic Occupancy Maps (IPOM), by applying object segmentation and identification jointly on multiple cameras. This module not only provides object's 3D location, but also gives distinguishable ID information for every player.

As the aforementioned DeepPlayer model predicts the probability of player ID, it may happen that some ID is ambiguous when the probability of the player ID is low. Thus, we develop the IPOM model as two parts: (1) POM with identified ID for the individuals with a high confidence of ID; (2) POM with ambiguous ID for those individuals with ambiguous IDs. An overview of the IPOM module can be seen in Figure 4.5. We develop the IPOM module by taking multi-camera segmentation and multi-target identification as inputs.

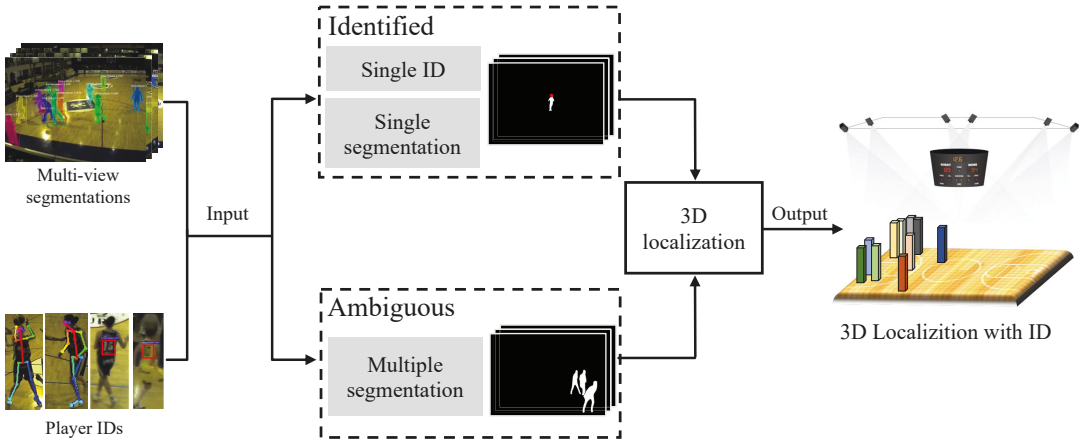


Figure 4.5 : Overview of the IPOM model. The input includes player's segmentations and IDs. The 3D localization model processes the players with identified ID and the players with ambiguous ID, followed by a post-process with a threshold. The output is the final 3D localization results.

4.3.1 3D Localization Formulation

We denote a set of discrete random variable $\mathbf{X} = \{X_k | k \in G\}$, where $X_k \in \{0, 1\}$. Let variable X_k represent the presence and absence of an object at location k, $X_k = 1$ represents for presence, while $X_k = 0$ represents for absence.

Let $\mathbf{B}_i = \{B_i^1, B_i^2, ..., B_i^c\}$ represent the segmentation images for the player with identification i in a number of camera views c, where the player is visible. Thus, the conditional probability of an individual with identification i standing on location k can be written as:

$$P\{X_k = 1 | \mathbf{B}_i\}. \tag{4.8}$$

However, the conditional probability $P\{X_k | \mathbf{B}_i\}$ is intractable. By providing the prior probability $P(X_k)$ and the likelihood probability $P(\mathbf{B}_i | X_k)$, tracking the

posterior probability $P\{X_k|\mathbf{B}_i\}$ becomes an Bayesian problem.

$$P\{X_k|\mathbf{B}_i\} = P(X_k)P(\mathbf{B}_i|X_k), \tag{4.9}$$

where the probabilities for all the location ks can be presented as:

$$P(X_1, X_2, ..., X_G|\mathbf{B}_i) = \prod_{k=1}^{G} P(X_k|\mathbf{B}_i). \tag{4.10}$$

The IPOM module firstly takes multiple 2D monocular segmentation and identification as inputs. It models the unique individual segmentations from each view using a synthetic average image related to the 3D coordinates of pre-defined locations [108]. To iteratively compute the probabilities of occupancy of an individual, we minimize the $K-L$ divergence between an estimated distribution and the posterior probability that we are after. Then, the likelihood probabilities are approximated by the normalized image distances between the synthetic average image and the individual's segmentations. Finally, an iterative process is designed to obtain optimal probability with the individual's ID.

4.3.2 IPOM with Identified ID

The player segmentations can be presented as $\mathbf{B}_i|c \in \mathbf{C} - \mathbf{C}^{am}$. By inputting these segmentations $\mathbf{B}_i$, with identification i, we implement the POM algorithm to calculate the probability of player i standing at location k, which can be presented as $P_k(X, Y, Z, i)$, where (X, Y, Z) denotes player's 3D coordinate and i denotes player's identification. Note that we use the ground plane to define the 3D world coordinate system, so the 3D coordinate Z is typically set to be zero.

4.3.3 IPOM with Ambiguous ID

In the case of the extreme occlusion, the accuracy player identification i for the occluded player may be unavailable in some views $c \in \mathbf{C}^{am}$. For those who are ambiguous, we use $\mathbf{B}_{am} = \{B^1_{am}, B^2_{am}, ..., B^c_{am}\}$ to represent their segmentations, where

$c \in \mathbf{C}^{am}$. Then, we implement the POM algorithms based on these segmentations to extract the probability of occupancy of the player with ambiguous ID standing on location k, which is $P_k(X, Y, Z, am)$.

Finally, we post-process those probabilities by setting a threshold (experimentally 0.85) to extract the certain number of probabilities of occupancy. To the end, the IPOM module outputs the 3D location and player ID of each player, with the corresponding probability.

4.4 KSP-ID for Tracking

After IPOM model processing, the 3D localization with ID of each player is put into our KSP-ID model to optimize the trajectories of each player. The original KSP [41] is a multi-player tracking algorithm for searching the K-th shortest path among all paths from the start point to the end point of a graph. It is assumed that each people moves one or more grid cells between successive frames in the tracking space of the world coordinate system which is divided into grids. However, as the link weights from nodes only depend on its human existence probability, the weights from one node to any other node are set as the same even they are not the same person.

We propose a player ID correlation coefficient in the KSP-ID model to correct the identity switch. As shown in Figure 4.6, a node in location i at time t can be represented as m_i^t where i denotes the location index and t is the time stamp. The linking edge $e_{i,j}^t$ connects from node m_i^t to m_j^{t+1}, which corresponds to admissible player motions between consecutive frames. The weight $e_{i,j}^t$ of edge depend on not only the marginal posterior probability ρ_i^t of the presence of the player, but also the proposed player ID correlation coefficient $\varrho_{i,j}^t$ of the node pairs. It is also noted that two virtual nodes, v_{source} and v_{sink}, in the graph are connected to all the possible nodes that player may enter and exit.

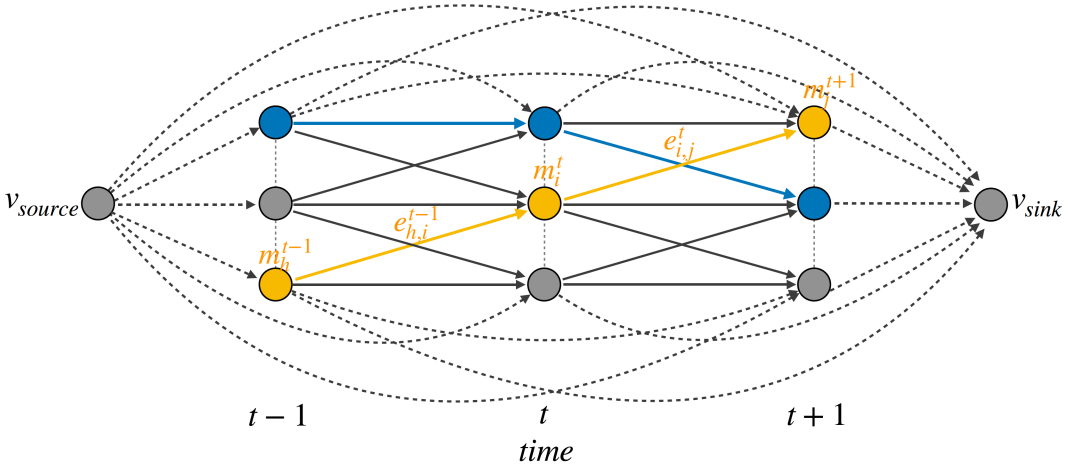


Figure 4.6 : A simplified flow system for a directed graph. The yellow and blue color represent two different players. Given a depth of one, a player occupying the location i at time t can arrive at one of the three neighbor locations at time $t+1$. The weight $e_{i,j}^t$ of edge depend on not only the marginal posterior probability ρ_i^t of the presence of the player, but also the proposed player ID correlation coefficient $\varrho_{i,j}^t$ of the node pairs.

Firstly, we discretize the court into G grid cells. If the frames number of one batch is represented as F, the total number of nodes except the two virtual nodes is $n = G \cdot F$. The nodes represent the player location in the court, and the edges denote the possible move for players in consecutive frames. The weights between the pairs of nodes are the player occupied the probability of each node. After the construction of the directed acyclic graph (DAG), K paths between these nodes can be optimized according to their minimum total costs of the paths.

We have estimated the probability of the presence of some player at every location i through aforementioned the DeepPlayer and IPOM models. Then the marginal posterior probability of the presence of the player can be a formula as:

$$\rho_i^t = P(M_i^t = 1|\mathbf{I}^t), \tag{4.11}$$

where p_i^t is the probability of location i at time t. X_i^t is the random variable standing the true occupancy of location i at time t, and $\mathbf{I}^t$ is the input image at time t. Here, we propose a player ID correlation coefficient to measure the identity similarity between two nodes i, j in consecutive frames :

$$\begin{aligned} \varrho_{i,j}^t &= P(D_i^t = D_j^{t+1}|\mathbf{I}^t, M_i^t = 1) \\ &= \begin{cases} (p_{idi} + p_{idj})/2, & if\ id_i^t = id_j^t \\ 1 - (p_{idi} + p_{idj})/2, & if\ id_i^t \neq id_j^t \\ 0.5, & if\ nonexistent\ id_i^t\ or\ id_j^t \end{cases}, \end{aligned} \tag{4.12}$$

where $j \in \mathbb{N}(i)$ is the neighborhood of location i. D_i^t and D_j^{t+1} are the player identity of location i at time t and location j at time $t+1$, respectively. id_i^t and id_j^t are the identities of location i and location j. p_{idi} and p_{idj} are the corresponding confidence of the identities. Thus, $\varrho_{i,j}^t$ obeys a multinoulli distribution in three different situations.

After that, we aim to find a set of trajectories, and each trajectory corresponds to one player. Thus, we need to connect the trajectory as long as possible, and the identity distance between the connected nodes should be as small as possible. We formulate it as:

$$\mathbf{m}^* = \arg\max_{z \in \mathcal{F}} P(M_i^t = m_i^t, D_i^t = D_{j \in \mathbb{N}(i)}^{t+1}|\mathbf{I}^t), \tag{4.13}$$

where $\mathcal{F}$ is the space of occupancy maps in the condition of Eq. (4.16). Assuming the conditional independence of the occupancy map for a given $\mathbf{I}^t$, our objective function can be redefined as follows:

$$\begin{aligned}\mathbf{m}^* &= \arg\max_{m\in\mathcal{F}} \log \prod_{t,i} P(M_i^t = m_i^t, D_i^t = D_{j\in\mathbb{N}(i)}^{t+1} | \mathbf{I}^t) \\ &= \arg\max_{m\in\mathcal{F}} \sum_{t=1}^{T}\sum_{i=1}^{n} (\log \frac{\rho_i^t \cdot \varrho_{i,j}^t}{1-\rho_i^t}) \cdot m_i^t,\end{aligned} \tag{4.14}$$

where x_i^t represents the actual occupancy of location i at time t, and it is equal to the sum of flows leaving from the same location $\sum_{j\in\mathbb{N}(i)} f_{i,j}^t$. Therefore, our goal is to solve the following linear program optimization problem:

$$\text{Maximize} \sum_{t=1}^{T}\sum_{i=1}^{n} (\log \frac{\rho_i^t \cdot \varrho_{i,j}^t}{1-\rho_i^t}) \cdot \sum_{j\in\mathbb{N}(i)} f_{i,j}^t, \tag{4.15}$$

subject to

$$\begin{aligned} &\forall t, i, f_{i,j}^t \geq 0, \\ &\forall t, i, \sum_{j\in\mathbb{N}(i)} f_{i,j}^t \leq 1, \\ &\forall t, i, \sum_{j\in\mathbb{N}(i)} f_{i,j}^t - \sum_{k\in\mathbb{N}(i)} f_{k,j}^t \geq 0, \\ &\sum_{i\in\mathbb{N}(v_{source})} f_{v_{source},i} - \sum_{k,v_{sink}\in\mathbb{N}(k)} f_{k,v_{sink}} \leq 0. \end{aligned} \tag{4.16}$$

Finally, we leverage a k-shortest node-disjoint paths method on a DAG to solve the linear program optimization problem, as illustrated in Figure 4.6. A directed edge $e_{i,j}^t$ from location i at time t to location j at time $t+1$ is assigned the cost value:

$$\mathcal{L}(e_{i,j}^t) = -(\log \frac{\rho_i^t \cdot \varrho_{i,j}^t}{1-\rho_i^t}). \tag{4.17}$$

The cost value of the edges emanating from the source node is set to zero to allow players to appear at any entrance position and at any time instant at no cost. We formulate the problem as a minimization problem by negating the objective function of Eq.(4.16) and Eq.(4.17), and solve it according to [41].

4.5 Experiments and Discussions

4.5.1 Dataset

APIDIS dataset[109] is a publicly available dataset with 7 cameras. The video files are recorded at 25 fps in 800 × 600 resolution in MPEG-4, including 1,500 frames. Additionally, the dataset contains 16 unlabeled periods. The basketball court is 2,797cm × 1,499cm. There are 12 people on the court, 2 referees and two 5-player teams. The publicly available dataset is challenging due to the difficult lighting conditions, reflections and shadows.

STU dataset is a new dataset that we collected at Shantou University. It is a timing synchronised dataset with 8 cameras. The video files are recorded at 24 fps in 1,280 × 720 resolution in MPEG-4. We have implemented our experiments on 2 periods, including 2,500 frames. The basketball court is 2,800cm × 1,500cm. There are 16 people on the court, 2 referees and two 8-player teams. The dataset contains 11 periods. We will publish it in the future.

4.5.2 Experiments Settings

DeepPlayer. We train the DeepPlayer model with SGD solver [110] in three steps, and follow the "image-centric" sampling strategy in [12]. As shown in Table 4.1, the training sequence starts with Cascade Mask RCNN-P, followed by Cascade Mask RCNN-J, after which is PoseID. We use the ResNet-50 pre-trained by the ImageNet 1000-class dataset. Other layers are randomly initialized by a Gaussian distribution with standard deviation as 0.01 and mean as 0. The ratio of positive and negative anchors in each image is set as 1:3. In Eq. (4.6), we set $\lambda_1 = 0.55, \lambda_2 = 0.45$ experimentally. On each GPU, the mini-batch size is 2. The whole training takes 20 hours on four NVIDIA 1080Ti Pascals under the Caffe framework.

IPOM and KSP-ID. We use two sets of parameters for the two datasets

Table 4.1 : Training parameters of the DeepPlayer model in three steps.

	Step size	Learning rate	γ	Momentum	Weght decay
Cascade Mask RCNN-P	80,000	0.002	0.1	0.9	0.001
Cascade Mask RCNN-J	60,000	0.001	0.1	0.9	0.0005
PoseID	40,000	0.05	0.1	0.9	0.0005

Table 4.2 : The characteristics of the datasets and the corresponding parameters.

Dataset	Camera	People	Frame	Resolution	Location	Grid
APIDIS	7	12	1,500	800×600	9,216	128×72
STU	8	17	2,500	1,280×720	6,720	112×60

respectively. For the APIDIS dataset, we divide the basketball court into rectangle grid cells with a size of 128 × 72, each of which is named as location i (from 0 to 9215). For each grid, the corresponding cube is designed to be 50cm × 50cm × 185cm, with the head plane as 1.85m height. For the STU dataset, we divide the basketball court into rectangle grid cells with a size of 112 × 60, each of which is named as location i (from 0 to 6720). For each grid, the corresponding cube is designed to be 40cm × 40cm × 185cm, with the head plane as 1.85m height. The parameters are summarized in Table 4.2.

4.5.3 Baselines

KSP tracker [41]. The original KSP algorithm ignores appearance. KSP directly use the POM detection results within 4 grid cells in time or space.

C-KSP tracker [111]. This modified KSP incorporates frame-to-frame appearance cues by modifying the edge costs. But the cues are extracted empirically and

manually.

DP tracker [112]. Dynamic Program (DP) model iteratively discovers and removes the shortest path from our tracklets graph until the path can no longer be found from source to sink. DP model is greedy and, unlike our algorithm, it cannot be traced back if it goes wrong.

VMD tracker [99]. VMD algorithm exploits the full geometry and the objects' center of mass for 3D object tracking.

VMDc tracker [99]. On the basis of the VMD tracker, VMDc model adds color information to avoid identity switch.

TMOSA tracker [113]. TMOSA algorithm uses geometric information regarding 3D scene structure rather than appearance information.

TMAP-EM tracker [114]. TMAP-EM method employs geometric information regarding 3D scene structure, by an MAP-EM model.

MSPT tracker [115]. MSPT tracker is a framework that combines off-the-shelf YOLO detector[9], POM model[112] and KSP tracker[41]. This framework is as similar as ours, but without player identification.

HybirdFull tracker [116]. HybirdFull tracker is a hybrid stochastic/deterministic optimization scheme that performs a stochastic search over the space of detection configurations, interleaved with deterministic computation of the optimal multi-frame data association for each proposed detection hypothesis.

4.5.4 Evaluation Metrics

Multi-object tracking (MOT) performance is typically measured by the Multiple Object Tracking Accuracy (MOTA) [117], which includes three sources of errors:

$$\text{MOTA} = 1 - \frac{\sum_t (c_m \cdot m_t + c_m \cdot fp_t + c_s \cdot mme_t)}{\sum_t g_t}, \tag{4.18}$$

where m_t is the number of missing or false negatives, and fp_t is the number of false positive, and mme_t is the number of instantaneous identity switch, and g_t is the number of detection ground truth. With reference to [117], the weight factors are set as $c_m = c_f = 1$, $c_s = \ln 10$.

In order to better evaluate the influence of identity switch, [5] introduced a new term *gmme* instead of *mme* to measure the proportion of identity switch in a global manner. *gmme* can count the proportion of a trajectory that is correctly labeled over a whole sequence while *mme* only counts the number of instantaneous identity switch. Global Multiple Object Tracking Accuracy (GMOTA) is defined as:

$$\text{GMOTA} = 1 - \frac{\sum_t (c_m \cdot m_t + c_m \cdot fp_t + c_s \cdot gmme_t)}{\sum_t g_t}, \tag{4.19}$$

where $c_m = c_f = c_s = 1$ to guarantee *gmme* greater than *mme*.

Overall, for the quantitative performance, we report the precision metric MOTA (the higher the better) and GMOTA (the higher the better), as well as the number of true positives (TP), false positives (FP), false negatives (FN), and identity switches (IDS).

4.5.5 Results

In Table 4.3, we list the quantitative performance on APIDIS dataset. As can be seen from the MOTA and GMOTA scores, our proposed framework achieves the state-of-the-art performance at general visual scenarios. Additionally, our method outperforms others significantly on the IDS scores, and only 4 identity switches occur in our result. Specifically, our model achieves 0.723 at the GMOTA score and outperforms others to a large margin, which demonstrate the efficiency of our Deep-

Figure 4.7 : Illustrative tracking results on the APIDIS dataset. Different color indicates different team, and the number indicates the identity across frames.

based model. We can see that our model results in fewer identity switches compared with the others. These outstanding results are partly attributed to the strength of the DeepPlayer model with CNN features, which have a much better ability to describe a player than low-level hand-crafted features. Compared with two KSP-based methods KSP [41] and C-KSP[111] trackers, our KSP-based method achieves a great improvement that the MOTA and GMOTA scores are as twice as that from [41][111]. The reason is that, the existing KSP does not consider the player ID, and our IPOM algorithm initializes the parameters of the court to avoid 3D misdetection. C-KSP only employs the color histograms to measures the similarity of two nodes in successive frames, while our DeepPlayer and IPOM model can precisely identify each player and provide the corresponding confidence to obtain the more accuracy edge costs, which considering both the player occupied probability and the proposed player ID correlation coefficient. The MOTA score of TMOSA [113] is higher than ours. Note that, it is based on a closed-world assumption and the location of player is manually labeled. Although we detect player automatically by our detector, we perform a promising result. Our detector for players named Cascade Mask RCNN-P achieves high accuracy at 90.3%. Furthermore, the GMOTA and IDS score of ours is better than TMOSA algorithm, and this further indicates that our method can solve the problem of identity switches better.

Table 4.3 : Quantitative comparison results of the proposed method with state-of-the-art trackers on the APIDIS dataset.

Method	**MOTA↑**	**GMOTA↑**	TP↑	FP↓	FN↓	**IDS↓**
KSP [41]	0.490	0.380	607	156	220	46
C-KSP[111]	0.589	0.433	620	148	158	32
DP [112]	0.495	0.391	-	-	-	-
VMD [99]	0.597	-	625	121	202	10
VMDc [99]	0.675	-	656	88	172	9
TMOSA [113]	0.855	0.687	**738**	**18**	**95**	8
TMAP-EM [114]	0.799	-	719	47	108	11
MSPT [115]	0.786	-	698	72	156	30
HybirdFull [116]	0.626	-	-	-	-	-
MCMPTI	**0.811**	**0.723**	**730**	**44**	**102**	**4**

We show illustrative results on the APIDIS dataset in Figure 4.7. It can be seen that our model achieves accurate tracking across multiple views. The illustrative player association result on the period 3 of the APIDIS dataset is shown in Figure 4.8. The results of camera 6 from frame 354 to 394, indicate that the proposed algorithm is able to avoid the identity switch between the two players. For the STU dataset, the illustrative results are shown in Figure 4.9. The STU dataset is a new dataset, on which no results of the state-of-the-art algorithm have been reported. Thus, we implement the ablation baselines and report quantitative results on Sec. 4.5.6.

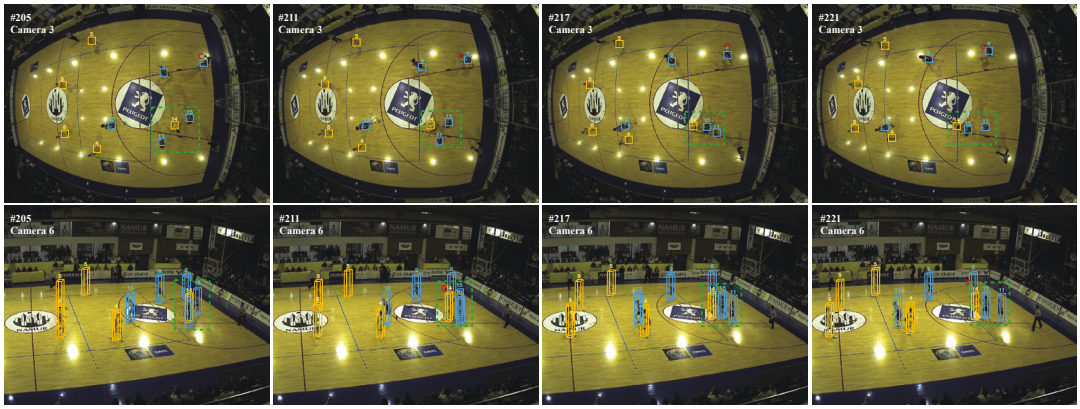


Figure 4.8 : Illustration of the APIDIS dataset in Camera 3 and 6 indicate that the proposed method is able to avoid identity switch among Player 11, Player 14, and Player 15 in the dashed box.

4.5.6 Ablation Study

To evaluate the effect of some essential components of our MCMPTI, we implement and test several variants of our model. The ablation baselines include the following: (i) POM + KSP: the model without the proposed DeepPlayer component, (ii) Cascade Mask RCNN-P + POM + KSP: the model without Cascade Mask RCNN-J and PoseID components, (iii) Ours w/o Cascade Mask RCNN-J: the model without Cascade Mask RCNN-P(jersey number feature), (iv) Ours w/o PoseID: the model without PoseID(pose-guided feature), and (v) Ours w/o IPOM: the model with POM instead of the proposed IPOM. The comparison are presented in Table 4.4 and Table 4.5 respectively.

From Table 4.4 and Table 4.5, it is clear that the DeepPlayer model is significant for the multi-player tracking. Specifically, the IDS will occur frequently if we do not leverage the DeepPlayer component, where the IDS increases from 4 to 46 on the APIDIS dataset and increases from 10 to 95 on the STU dataset. Meanwhile, the scores of MOTA and GMOTA decrease dramatically without the DeepPlayer.

Figure 4.9 : Illustration of our tracking results on the STU dataset in Camera 1, 4 and 6. Red and yellow color boxes indicate black and white teams respectively. The blue dotted boxes show that our method avoids identity switch between Player 6 of the white team and Player 11 of the black team.

Comparison between the Cascade Mask RCNN-P + POM +KSP and the POM + KSP can illustrate that the Cascade Mask RCNN-P achieves a better result as it brings a useful result of player detection and segmentation in each view. However, compared with our full model, the Cascade Mask RCNN-P + POM +KSP model encounters more identity switches as it has no jersey number recognition and pose-based identification. Our full model also outperforms the model without the IPOM component.

To evaluate the effectiveness of different components of DeepPlayer, ablation experiments are conducted on the APIDIS and STU datasets, by removing Cascade Mask RCNN-J and PoseID from DeepPlayer respectively. From Table 4.4 and Table 4.5, we can see that Ours w/o PoseID outperforms Ours w/o Cascade Mask RCNN-J. In addition, the player identification results on the STU dataset are shown in

Table 4.4 : Ablation experiments on the APIDIS dataset.

Method	**APIDIS dataset**					
	MOTA↑	GMOTA↑	TP↑	FP↓	FN↓	IDS↓
POM + KSP	0.490	0.380	607	156	220	46
Cascade Mask RCNN-P + POM +KSP	0.613	0.514	651	102	189	33
Ours w/o Cascade Mask RCNN-J	0.668	0.565	683	89	167	18
Ours w/o PoseID	0.799	0.711	709	47	112	10
Ours w/o IPOM	0.802	0.717	718	55	109	5
Ours (full)	**0.811**	**0.723**	**730**	**44**	**102**	**4**

Table 4.5 : Ablation experiments on the STU dataset.

Method	**STU dataset**					
	MOTA↑	GMOTA↑	TP↑	FP↓	FN↓	IDS↓
POM + KSP	0.414	0.349	545	179	253	95
Cascade Mask RCNN-P + POM +KSP	0.551	0.425	607	156	220	69
Ours w/o Cascade Mask RCNN-J	0.653	0.505	813	118	184	36
Ours w/o PoseID	0.762	0.672	1188	62	128	15
Ours w/o IPOM	0.767	0.678	1186	65	139	11
Ours (full)	**0.787**	**0.692**	**1255**	**48**	**121**	**10**

Table 4.6. We report the accuracy of top n ranks in the rank list, and the mean Average Precision (mAP). It is clear that Cascade Mask RCNN-J performs better than PoseID, which means that jersey number features are more effective than pose-guided features. Meanwhile, we observe that the performance of DeepPlayer is higher by +4.4% and + 4.7% in rank-1 accuracy and mAP, respectively. Thus, each component is necessary for the DeepPlayer model.

To study the effect of different PoseBox introduced in Sec. 4.2.2, we compare three PoseBoxes in Figure 4.4. From Table 4.10, we can observe that PoseBox3 outperforms the other two. PoseBox2 is superior to PoseBox1 on MOTA score by 0.5%, with the inclusion of face. PoseBox3 outperforms PoseBox2 by less than 0.5%, and the torso has no distinctive features among teammates. From the results, we

Table 4.6 : Ablation studies on player identification on the STU dataset.

Method	**top-1↑**	**top-3↑**	**mAP↑**
Cascade Mask RCNN-P + PoseID	58.4	68.1	39.8
Cascade Mask RCNN	85.9	90.9	75.0
DeepPlayer	**90.3**	**93.4**	**79.7**

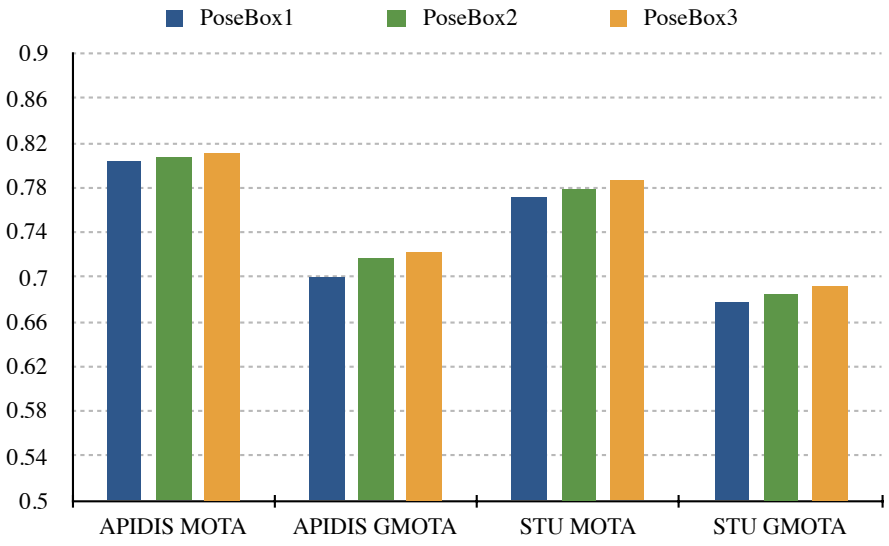


Figure 4.10 : Results of the three types of PoseBoxes on the APIDIS and STU dataset. PoseBox1 = torso + arms + legs; PoseBox2 = head + torso + arms + legs; PoseBox3 = head + arms + legs.

think that the face contributes more than the torso, and that the torso confuses the system among the teammates.

4.6 Conclusion

This chapter presents a robust multi-camera multi-player tracking framework to understand players in a sports scene. In this system, we specifically consider player identity, the most distinguishable information, which is commonly ignored in existing methods. We tailor-make the DeepPlayer model based on region-based deep learning in the scenario of sports video, by studying the patterns of jersey number, team class, and pose-guided partial feature. Furthermore, we propose a player ID correlation coefficient in the KSP-ID model for linking nodes in the DAG, in order to handle the identity switches. The qualitative, quantitative and ablation experiments verify that effectively tackle the identity switch problem and the proposed framework achieves state-of-the-art performance. The drawback of the proposed framework is that the computation cost is higher compared with those existing methods.

Chapter 5

Infrared Target Detection using Dual-domain Feature Extraction and Allocation

5.1 Introduction

Chapter 3 and 4 discuss non-rigid objects in visible light scenes, we study rigid objects in infrared scenes in Chapter 5. The third object we focus on is the vehicle target of infrared images. The task is aiming to locate vehicles and identify vehicle categories in infrared images. Automatic target detection (ATD) in forward looking infrared imagery (FLIR) has attracted significant attention as the rapid developments of FLIR imaging [118] and computer vision technologies [119]. The advantages of infrared images are that they are not affected by the shadows and illumination variations, and targets could be distinguished from background since the background is usually colder than targets. Additionally, infrared target detection also handles the total darkness environment, where there is no signal in the visual camera. Therefore, infrared target detector is important and useful in various real-world applications, such as face recognition [120], pedestrian detection [121], intelligent monitoring [122], and self-driving vehicle[123].

Despite many superiorities, infrared target detection is facing many difficulties. Compared with RGB images, IR images captured by FLIR sensors, are deeply influenced by weather and atmospheric conditions as well as various background clutters [119]. The poor texture information, low resolution, high noise, and unique physical characteristics of IR images still remain to be the serious factors that make infrared target detection an open problem. These unwanted attributes make it challenging

to obtain the discriminative features of the target and hence degrade the detection performance. In addition, without color patterns, the detector needs more efficient feature representation capabilities to distinguish objects with similar outlines. Among these challenges, how to extract discriminative features is the basic and the most significant one.

The prevailing infrared target detection methods follow a traditional procedure of potential IR target search and decision learning. These infrared target detection models can be roughly divided into two categories. One is the model-based approach, including template matching [124], Hausdorff metric [125], thermal characteristics modelling [126], geometric hashing, and contour matching [127]. The other is based on feature extraction to find the Region of Interests (RoIs), such as HOG features [128], thermal characteristics features [129], and SVM [130]. Though these studies perform effectively in some practical applications, they cannot extract the discriminative features of infrared imagery as they strongly rely on prior knowledge and handcrafted features.

Recent years have witnessed much progress achieved using convolutional neural networks for feature extraction in the object detection task. With CNN models, one-stage methods and two-stage methods handle the object detection task effectively. One-stage methods such as YOLO [4] are faster, while two-stage methods such as R-CNN are more accurate. Thanks to the proposed region proposal network (RPN), Faster R-CNN[12] has become a successful two-stage baseline for object detection in RGB images. And many variants of R-CNN perform well in different scenarios. However, these off-the-shelf detectors fail to detect IR targets due to the low resolution, high noise of the infrared imagery. Therefore, finding a discriminative representation is the key to the problem.

Considering the aforementioned problems and the limitations of previous meth-

ods, we propose a deep learning-based infrared target detection method, with a dual-domain feature extraction model and a feature resource allocation model. (1) In general, the infrared radiation intensity of the target is more than the infrared radiation intensity of background, which means that the target and background are different in the frequency domain. Considering this characteristic of IR images, we consider the frequency features and spatial features simultaneously in the dual-domain feature extraction model. (2) We regard features as a kind of resource, and allocate this kind of resource reasonably, in favour of the dual-domain feature integration and re-extraction. Here, a novel Resource Allocation model for Features (RAF) is proposed to reduce redundant information and highlight distinguishable message by developing two new self-attention modules.

To summarize, this chapter presents a novel automatic target detection method in infrared imagery, named Deep-IRTarget. The main contributions are three-fold:

- For the challenges of infrared feature extraction, we propose a Dual-domain Feature Extraction model (DFE) to obtain the informative representation for the infrared imagery. Besides using a CNN for spatial feature extraction, a Hypercomplex Infrared Fourier Transform (HIFT) is developed to extract the infrared intensity saliency in the frequency domain, which dramatically increases the target detection accuracy.

- To integrate and re-extract the dual-domain feature, we develop a Resource Allocation model for Features (RAF), in which a Channel Squeeze-Excitation Self-Attention block (Channel SE-Attention) and a Position Squeeze-Excitation Self-Attention block (Position SE-Attention) are designed to recalibrate channel-wise feature and position-wise feature responses by explicitly modelling inter-dependencies between channels and positions.

- Experiments on the MWIR, BITIR, and WCIR datasets demonstrate that our

method achieves compelling performance compared with the state-of-the-arts. The ablation studies prove the effectiveness of each model.

5.2 Overall of The Deep-IRTarget Framework

The imaging principle of infrared image is to visualize the thermal radiation energy emitted by the object, so that the image can reflect the temperature characteristics of the object. This imaging feature of infrared image makes it very different from RGB visible light images. How to extract the infrared feature plays a key role in the infrared target detection.

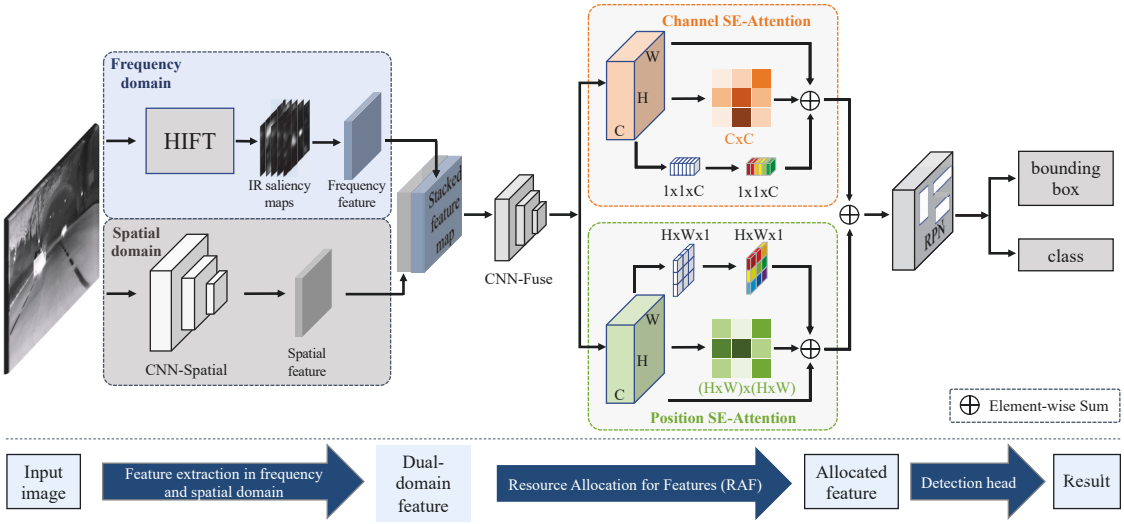


Figure 5.1 : The architecture of the Deep-IRTarget framework mainly consists three parts, (1) Dual-domain Feature Extraction(DFE) in frequency and spatial domain; (2)Resource Allocation for Feature(RAF) with channel-wise and position-wise attention; (3) Detection head based on the Region Proposal Network (RPN).

In this chapter, we focus on designing an effective backbone to extract infrared features. The Deep-IRTarget framework is proposed, based on the Dual-domain Feature Extraction model (DFE) and the Resource Allocation for Feature (RAF), as illustrated in Fig. 5.1.

Given an input IR image, both the frequency feature and spatial feature are considered in the respective domains. The discriminative frequency feature with infrared intensity saliency is extracted by the proposed Hypercomplex Infrared Fourier Transform model (HIFT). The spatial feature with visual charateristc is extracted by the designed CNN network. After that, these two features are concatenated together and step into the RAF process. The dual-domain features are firstly fed to the CNN-Fuze for feature fusion. Next, in order to maximize the mining of dual-domain features, a Channel Squeeze-Excitation Self-Attention block (Channel SE-Attention) and a Position Squeeze-Excitation Self-Attention block (Position SE-Attention) are proposed. The Channel SE-Attention can generate the feature of channel contextual information with a channel self-attention matrix which models the relationship between any two channels of the features. Meanwhile, the position-wise attention can extract features of pixel contextual information with the similar process in position-level dimension. The use of these two blocks to perform feature integration and resource allocation for dual-domain features in two dimensions can greatly reduce information redundancy while enhancing recognizable features. Then, we aggregate the two dimensional features by weights to obtain allocated feature representations. To the end, our detection head with Region Proposal Network (RPN)[12] can easily search the region proposals from the features and predict the locations and classes of targets. The details of the detection head is illustrated in the Appendix section.

5.3 Dual-domain feature extraction

In this section, the Dual-domain Feature Extraction model (DFE) is described in detail. We first emphatically introduce the Infrared Intensity Saliency in the frequency domain, and then briefly describe the CNN-Spatial network for spatial feature extraction.

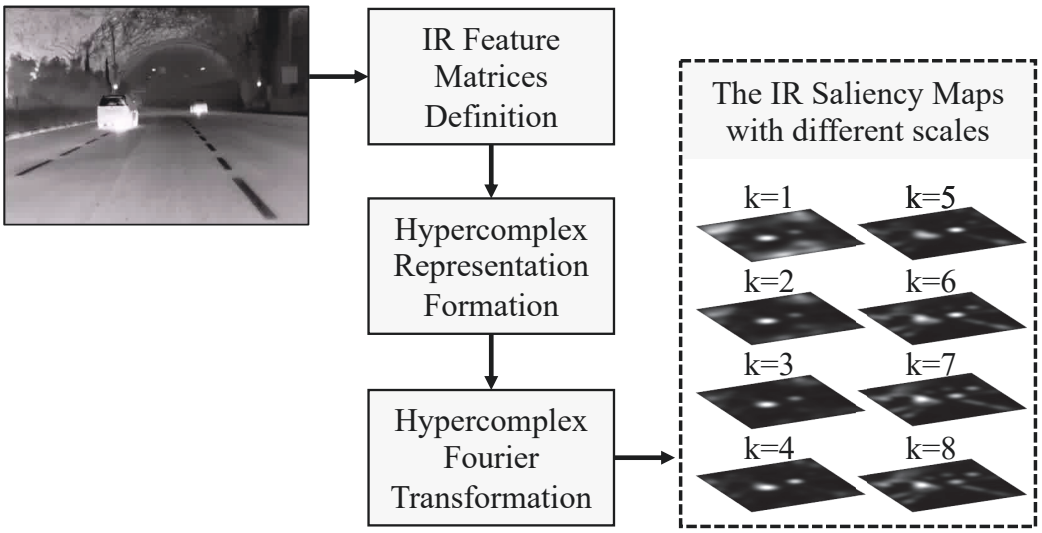


Figure 5.2 : The pipeline of the Hypercomplex Infrared Fourier Transform.

5.3.1 Infrared Saliency using Hypercomplex Infrared Fourier Transform in the Frequency Domain

Motivated by the Hypercomplex Fourier Transform (HFT) that has been successfully applied in RGB color image processing[131], we propose a Hypercomplex Infrared Fourier Transform (HIFT) to extract infrared features in the frequency domain, as shown in Fig. 5.2. The proposed HIFT redefines the feature matrices for IR data and forms the hypercomplex representation of IR images. After Hypercomplex Fourier Transform, we create the Spectrum Scale Space[131] by a series of Gaussian kernels. Finally, based on the calculated amplitude and phase spectra, a series of IR saliency maps at different scales can be performed by the inverse transformation. The IR saliency maps can be regarded as the frequency features.

Hypercomplex Fourier Transform

The Hypercomplex Fourier transform [132] is a development based on the Fourier transform. The hypercomplex input is specified to be a quaternion formula, which can be seen as an extension of the concept of complex numbers. A hypercomplex

matrix is defined as follows:

$$f(n,m) = a + bi + cj + dk, \tag{5.1}$$

where i, j, k are imaginary unit, and satisfy $i^2 = j^2 = k^2 = ijk = -1, ij = k, ik = j, jk = i$. The discrete version of the HFT is:

$$F_H[s,t] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathrm{e}^{\mu 2\pi(ms/M+nt/N)} f(n,m), \tag{5.2}$$

where μ is a unit pure quaternion, $\mu^2 = -1$. And the inverse Hypercomplex Fourier Transform is given by:

$$f(n,m) = \frac{1}{\sqrt{MN}} \sum_{v=0}^{M-1} \sum_{u=0}^{N-1} \mathrm{e}^{\mu 2\pi(ms/M+nt/N)} F_H[s,t]. \tag{5.3}$$

IR Feature Matrices Definition

Since the RGB color image can be represented as a quaternion image by color features, the Hypercomplex Fourier transform can be easily used in RGB color image processing. However, the infrared image is a single-channel grey image which is different from the RGB color image due to the different imaging principles. Here, we define the IR feature matrices using three features. Consequently, The quaternion representation can be used to conveniently represent the IR image, and the quaternion-based HFT can be used to achieve frequency domain processing of the IR image.

Grey value: The grey value reflects both the structure and temperature of the target and environment. Thus, this single channel can be seen as the grey value features directly.

Infrared radiation: The infrared radiation values[102] reflect 'heat' of the

target and environment. The infrared radiation value metric:

$$\Gamma = \Gamma_{min} + \frac{(g/255 - r) \times (\Gamma_{max} - \Gamma_{min})}{1 - r}, \tag{5.4}$$

where g is the grey level of each pixel of an IR image. The minimum and maximum radiation intensity of the target is Γ_{min} and Γ_{max} respectively. r is constant and $r \in [0, 1]$, which depends on the environment and the IR thermal sensor.

Local thermal variation: The local thermal variation feature was chosen because manmade objects often show greater variation in temperature than natural objects. This feature merely determines the average absolute difference between each pixel and the mean of the internal region and compares them to the same measurements for a local background region. The feature is calculated as:

$$LV_{i,j} = \frac{L_{\text{in}}\,(i,j)}{n_{\text{in}}} - \frac{L_{\text{out}}\,(i,j)}{n_{\text{out}}}, \tag{5.5}$$

where

$$L_{\text{in}}(i,j) = \sum_{(k,l)\in N_{\text{in}}(i,j)} \left| f(k,l) - \mu_{\text{in}}(i,j) \right|, \tag{5.6}$$

$$\mu_{\text{in}}(i,j) = \frac{1}{n_{\text{in}}} \sum_{(k,l)\in \text{N}_{\text{in}}(i,j)} f(k,l), \tag{5.7}$$

where $f(k,l)$ is the gray level value of the pixel in the k-th row and l-th column, $N_{in}(i,j)$is the neighborhood of the pixel i,j, defined as a rectangle whose width is the length of the longest vehicle in the target set and whose height is the height of the tallest vehicle in the target set. The neighborhood $N_{out}(i,j)$ contains all of the pixels in a larger rectangle around i,j except those pixels that are in $N_{in}(i,j)$. And $L_{\text{in}}(i,j)$, and $L_{\text{in}}(i,j)$ are defined similarly.

Hypercomplex Representation Formation

The Hypercomplex IR representation is employed to combine the aforementioned three features. We define the input hypercomplex matrix as follows:

$$f(n,m) = w_1 f_1 + w_2 f_2 i + w_3 f_3 j + w_4 f_4 k, \tag{5.8}$$

where w_1, w_2, w_3, w_4 are weights and f_1, f_2, f_3, f_4 are feature matrices. f_1 is the motion feature, hence, $f_1 = 0$ for static input images.

$$f_2 = Grey(x); f_3 = \Gamma(x); f_4 = LV(x). \tag{5.9}$$

After a large number of experiments, we select the weights so that $w_1 = 0, w_2 = w_3 = 0.4, w_4 = 0.2$.

Computing the IR Saliency Map by HFT

To construct the infrared intensity saliency map, both the amplitude and phase spectrum are utilized in the frequency domain. Generally, in a thermal image, the value of the amplitude spectrum at low frequencies is higher than that at high frequencies. Since the regions of targets are always the high-frequency component, the high-frequency region can be regarded as the potential target regions. Given an IR image $f(x, y)$, the input is defined as the Hypercomplex representation formation. The Hypercomplex Fourier Transform is rewritten in polar form:

$$\mathcal{F}_{\mathcal{H}}[s,t] = \|\mathcal{F}_{\mathcal{H}}[s,t]\| \, e^{\mu\Phi(s,t)}, \tag{5.10}$$

where $\|\cdot\|$ represents the modulus for every element of a hypercomplex matrix. $\mathcal{F}_{\mathcal{H}}[s,t]$ is the frequency domain representation of $f(x,y)$. The amplitude spectrum $\mathcal{A}(s,t)$, phase spectrum $\mathcal{P}(s,t)$ and the eigenaxis spectrum $\mathcal{X}(s,t)$ are calculated as:

$$\begin{aligned} \mathcal{A}(s,t) &= \|\mathcal{F}_{\mathcal{H}}(s,t)\| \\ \mathcal{P}(s,t) &= \Phi(s,t) = \tan^{-1}\frac{\|\mathcal{V}(\mathcal{F}(s,t))\|}{\mathcal{S}(\mathcal{F}(s,t))} \\ \mathcal{X}(s,t) &= \mu(s,t) = \frac{\mathcal{V}(\mathcal{F}(s,t))}{\|\mathcal{V}(\mathcal{F}(s,t))\|}, \end{aligned} \tag{5.11}$$

where $\mathcal{S}$ represents the real part of the quaternion image, and $\mathcal{V}$ represents its imaginary part.

Studies of HFT models show that the amplitude spectrum contains both significant information and non-HFT significant information. Therefore, the model employs different Gaussian kernel functions to smooth the amplitude spectrum to achieve the purpose of suppressing high-frequency information while enhancing low-frequency information. Different amplitude spectrum smoothing filters form a spectral scale space. A Gaussian kernel g is defined as:

$$g(s,t;k) = \frac{1}{\sqrt{2\pi}2^{k-1}v_0}e^{-\left(s^2+t^2\right)/2^{2k-1}v_0}, \tag{5.12}$$

where $v_0 = 0.5$, k is the scale parameter, $k = 1, ..., K$, and $K = [\log_2 \min\{H, W\}]+1$. H and W is the height and width of the original IR image.

Convolution using a Gaussian kernel in the frequency domain of the logarithmic amplitude spectrum can be regarded as an image saliency detector[133], which can suppress non-saliency parts such as the background and form a salient area. The Spectrum Scale-Space is a family of derived signals $\{\Lambda_k\} = \Lambda(s,t;k)$ that is defined as follows:

$$\Lambda(s,t;k) = (g(.,.;k) \star \mathcal{A})(s,t). \tag{5.13}$$

Given a single smooth amplitude spectrum Λ_k and the original phase and eigenaxis spectra, we can perform an inverse transform to infer an IR saliency map for each scale, which constitutes a series of IR saliency maps $\{\mathcal{S}_k\}$, as illustrated in Fig.

5.2.

$$\mathcal{S}_k = g \star \left\| \mathcal{F}_{\mathcal{H}}^{-1} \left\{ \Lambda_k(u,v) e^{\mathcal{XP}(s,t)} \right\} \right\|^2 \tag{5.14}$$

Where F^{-1} denotes the inverse HFT. The saliency maps $\mathcal{S}_k$ are selected from $\{\mathcal{S}_k\}$ in the condition of the best scale k_p, which is calculated by [131]. Different from all the other HFT-based methods, we employ convolutional layers to learn the latent frequency features, instead of selecting the proper saliency map by entropy analysis methods.

5.3.2 CNN-based Feature Extraction in the Spatial Domain

As we know, the convolutional layer, the pooling layer, and the fully connected layer are equal to transform the pixel space of the image. Hence, they can be understood as spatial feature extraction of the image.

In order to better extract spatial features, a convolutional neural network (CNN-Spatial) with several Residual Blocks (ResBlock) is designed. The details of the CNN-Spatial are shown in the Table 5.1. Drawing on the network structure of ResNet, the CNN-Spatial consists of a 7×7 convolutional layer, two maxpooling layers and two ResBlocks. The input image is standardized to ensure that the scale is unified to 448×448. The feature map of the conv5 layer is used as the spatial feature.

5.4 Resource Allocation for Feature

As aforementioned in Fig. 5.1, the frequency feature map and the spatial feature map are extracted in the separate domains, and stacked along the depth dimension. In this section, we introduce a novel Resource Allocation model for stacked Feature maps (RAF) to reduce redundant information and highlight distinguish-

Table 5.1 : The architecture of CNN-Spatial

layer name	**output size**	**architecture**
input	$448 \times 448 \times 1$	-
conv1	$224 \times 224 \times 32$	7×7, 32, stride = 2
conv2	$112 \times 112 \times 32$	3×3, maxpooling, stride = 2
conv3	$112 \times 112 \times 64$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$, ResBlock
conv4	$56 \times 56 \times 64$	3×3, maxpooling, stride = 2
conv5	$56 \times 56 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$, ResBlock

able message for conveniently emerging potential IR target regions. The proposed RAF model considers to integrate the feature maps in the channel dimension and the position dimension simultaneously. Inspired by the Self-attention model [134] which can be used to focus on the relevant region of the image, we propose a Channel Squeeze-Excitation Self-Attention block (Channel SE-Attention) and a Position Squeeze-Excitation Self-Attention block (Position SE-Attention) in the RAF model, through referring the Dual-Attention [135] and SE Network [136]. The Channel SE-Attention block can capture long-range channel-wise contextual information in channel dimension, and the Position SE-Attention block can capture long-range position-wise contextual information in the position dimension.

5.4.1 CNN-based Feature Integration

The frequency feature and spatial feature are extracted in separate domains and stacked along the channel dimension. The size of the frequency domain feature map is $56 \times 56 \times 8$, and the size of the spatial feature map is $56 \times 56 \times 128$. The two

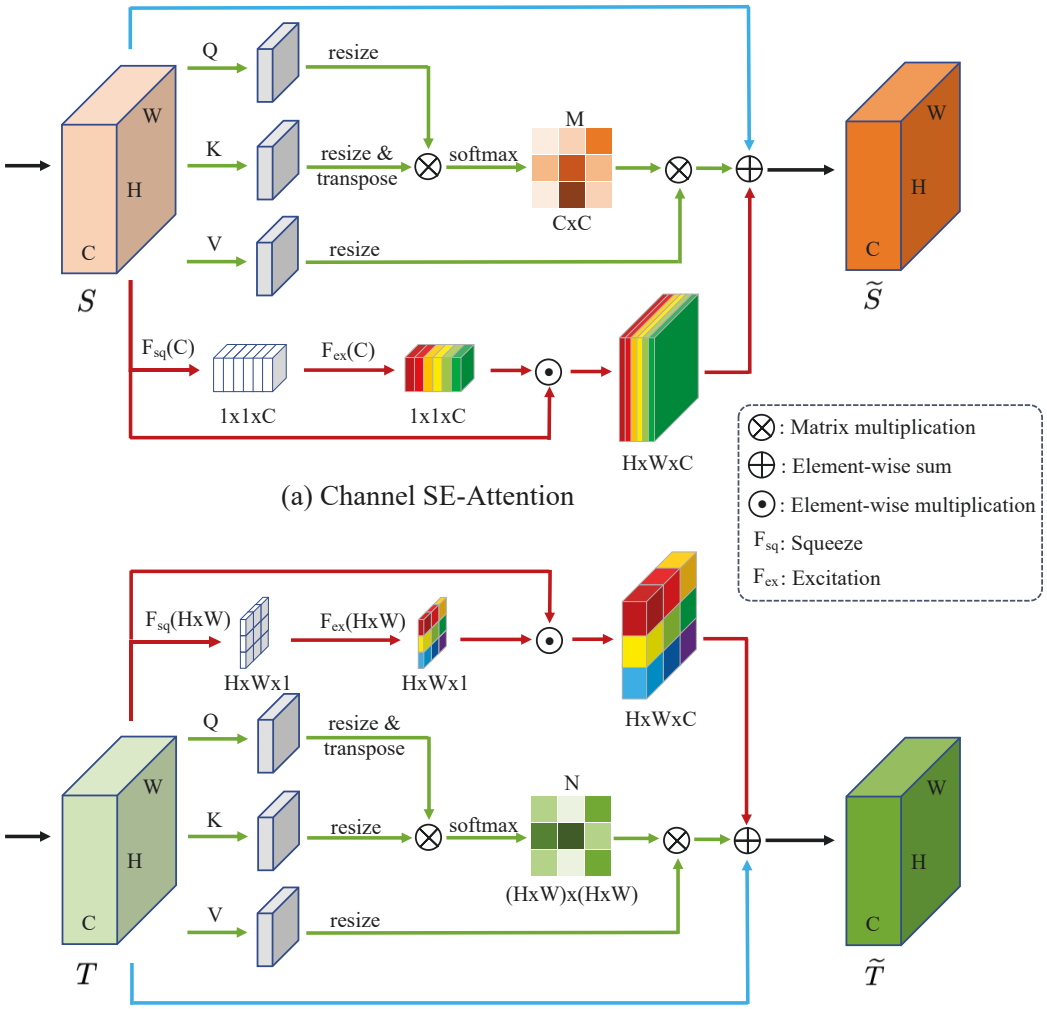
Table 5.2 : The architecture of CNN-Fuse

layer name	**output size**	**architecture**
input	$56 \times 56 \times 136$	-
conv6	$56 \times 56 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$, ResBlock
conv7	$28 \times 28 \times 128$	3×3, maxpooling, stride $= 2$
conv8	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$, ResBlock

features are stacked to form a $56 \times 56 \times 136$ dual-domain feature map. Here, we design the CNN-Fuse to fuse these feature maps. The model parameters are shown in Table 5.2. The feature map of the conv8 layer is obtained, and two novel attention models will be performed on this feature later to re-extract the more discriminative feature.

5.4.2 Channel SE-Attention Block

As far as we know, each CNN layer acts as a pattern detector, and each channel of a feature map can be regarded as a response to the corresponding CNN layer. It is worth noting that different channels of the same layer associate with each other by different semantic responses. In our case, the two-domain feature map consists of the frequency channels and the spatial channels. By exploiting the interdependence and latent relationship between different channels, we can emphasize interdependent feature maps and improve feature representations for resource allocation on the frequency and spatial channels. Therefore, applying the channel-wise attention mechanism is equivalent to the process of reallocating two-domain infrared and semantic information.



(a) Channel SE-Attention

(b) Position SE-Attention

Figure 5.3 : The details of the Channel SE-Attention block and the Position SE-Attention block are illustrated in (a) and (b). Different colored lines represent different streams.

Fig. 5.3(a) shows the architecture of the Channel SE-Attention block. Given a feature map $S = [s_1, s_2, ..., s_C] \in \mathbb{R}^{C \times H \times W}$ that is the two-domain feature maps, we apply three streams for attention, as shown in the different colors.

The green stream

As shown in green lines of Fig. 5.3(a), we feed the feature map to a convolutional layer (kernel=1×1, maintaining the number of channels) to generate three feature maps Q, K, and V, where $Q, K, V \in \mathbb{R}^{C \times H \times W}$. Q, K, and V mean query, key and value respectively. After that, we resize Q and K to $\mathbb{R}^{C \times (H \times W)}$, and transpose K to $\mathbb{R}^{(H \times W) \times C}$. Then, we apply a matrix multiplication between Q and K. Then, we perform a softmax operation to infer the channel-wise attention matrix $M \in \mathbb{R}^{C \times C}$:

$$m_{ji} = \frac{\exp\left(Q_i \cdot K_j\right)}{\sum_{i=1}^{H \times W} \exp\left(Q_i \cdot K_j\right)} \tag{5.15}$$

where m_{ji} represents the impact of the i-th channel on the j-th channel. More similar the feature representations of the two channels mean that they have a higher the correlation.

Meanwhile, we also resize V to $\mathbb{R}^{C \times (H \times W)}$, and apply a matrix multiplication between the transpose of M and V. Then, we obtain the result and reshape it to $\mathbb{R}^{C \times H \times W}$. The output feature map of the green stream is defined as:

$$S'_j = \sum_{i=1}^{C} \left(m_{ji} V_i\right) \tag{5.16}$$

The red stream

As shown in red lines, we squeeze the global spatial information into a channel descriptor as [136]. The squeeze operation is achieved by global average pooling to calculate the statistic $\mathbf{z} \in \mathbb{R}^C$ of channel-wise distribution. Considering the feature map $S = [s_1, s_2, ..., s_C] \in \mathbb{R}^{C \times H \times W}$, $\mathbf{z}$ is generated by shrinking S, such that its the k-th element:

$$z_k = \mathbf{F}_{sq}(s_k) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} s_k(i,j). \tag{5.17}$$

The channel-wise global spatial information is encoded in $\mathbf{z}$. Then, $\mathbf{z}$ is transformed to $\mathbf{z}'$:

$$\mathbf{z}' = \mathbf{F}_{ex}(\mathbf{z}) = \sigma(\mathbf{W}_1(\delta(\mathbf{W}_2\mathbf{z}))), \tag{5.18}$$

where $\mathbf{W}_1 \in \mathbb{R}^{C \times \frac{C}{r}}$, $\mathbf{W}_2 \in \mathbb{R}^{\frac{C}{r} \times C}$ are weights of two fully connected layers, with the sigmoid activation function $\sigma(\cdot)$ and the ReLU activation function $\delta(\cdot)$, r is the number of hidden layer routines in the middle layer in the channel excitation. Foreshadowing some of our results, we set $r = 16$ for the best performance. The output $S'' = [s_1'', s_2'', ...s_k''] \in \mathbb{R}^{C \times H \times W}$ of the red stream is obtained by rescaling S using element-wise multiplication:

$$s_k'' = z_k' s_k \tag{5.19}$$

The blue stream

This stream can be viewed as the long-range feature maintenance. The connection between S and $\widetilde{S}$ models the long-range channel-wise context dependencies. It helps to tackle the gradient vanishing problem and boost feature discriminability.

Finally, we plus S' and S'' on S to perform the final output feature map $\widetilde{S} \in \mathbb{R}^{C \times H \times W}$:

$$\widetilde{S} = \alpha S' + \beta S'' + S \tag{5.20}$$

where α and β are initialized to 0 and gradually learned to assign more weight. The resulting feature map $\widetilde{S}$ can be regarded as a weighted sum of the features across all

channels and the original input features. Hence, the channel-wise attention enables the network to selectively aggregate channels.

5.4.3 Position SE-Attention Block

Positional features in both the frequency domain and spatial domain are essential for target localization and classification. Many studies [137, 138] present that over-focusing on local features could lead to misclassification of targets. Hence, we consider both local and global features in the position dimension. Capturing the positional dependency between any two positions of the feature map is important to update features at all positions. The positional dependency performs as the feature similarity between the corresponding two positions. Therefore, the positions of any two existing similar features can contribute to each other, regardless of the distance between them.

Fig. 5.3(b) shows the architecture of the Position SE-Attention block. Given a feature map $T = [t_1, t_2, ..., t_C] \in \mathbb{R}^{C \times H \times W}$ that is the same to the feature map S, we perform three main streams, as shown in different colors.

The green stream

As shown in green lines, we feed the feature map to a convolutional layer (kernel=1×1, maintaining the number of channels) to generate three feature maps $Q, K, V \in \mathbb{R}^{C \times H \times W}$. After that, we resize Q and K to $\mathbb{R}^{C \times (H \times W)}$, and transpose Q to $\mathbb{R}^{(H \times W) \times C}$. Then, a matrix multiplication is applied to Q and K. Then, we apply a softmax operation to infer the position-wise attention matrix $N \in \mathbb{R}^{(H \times W) \times (H \times W)}$:

$$n_{ji} = \frac{\exp\left(Q_i \cdot K_j\right)}{\sum_{i=1}^{C} \exp\left(Q_i \cdot K_j\right)} \tag{5.21}$$

where n_{ji} represents the impact of the i-th position on the j-th position. The more similar feature representations of the two position contributes to greater correlation

between them. We also resize V to $\mathbb{R}^{C\times(H\times W)}$ and apply a matrix multiplication between the transpose of V and N. Then, we reshape the result to $\mathbb{R}^{C\times H\times W}$. The output feature map of this green stream is given by:

$$T'_j = \sum_{i=1}^{H\times W} (n_{ji} V_i) \tag{5.22}$$

The red stream

We propose to squeeze global positional information into a position descriptor. The squeeze on position is performed by applying a global average operation to each position of the feature map to generate the statistic $\mathbf{y} \in \mathbb{R}^{H\times W}$ of position-wise distribution. This is equivalent to average each pixel of the feature map along with the channel direction. Considering the feature map $T = [t_1, t_2, ..., t_{H\times W}] \in \mathbb{R}^{C\times H\times W}$, $\mathbf{y}$ is generated by shrinking T, such that the k-th element is :

$$y_k = \mathbf{F}_{sq}(t_k) = \frac{1}{C}\sum_{i=1}^{C} t_k(i, j). \tag{5.23}$$

The position-wise global positional information is embedded in $\mathbf{y}$. Then, $\mathbf{y}$ is transformed to $\mathbf{y}'$:

$$\mathbf{y}' = \mathbf{F}_{ex}(\mathbf{y}) = \delta(\mathbf{W}_3(\delta(\mathbf{W}_4\mathbf{y}))), \tag{5.24}$$

where $\mathbf{W}_3 \in \mathbb{R}^{H\times W\times \frac{(H\times W)}{d}}$, $\mathbf{W}_4 \in \mathbb{R}^{\frac{(H\times W)}{d}\times H\times W}$ are weights of two convolutional layers, with the ReLU activation function $\delta(\cdot)$, d is the depth number of the convolutional layers in the position excitation. Here, we set $d = 256$ for the best performance. The output $T'' = [t''_1, t''_2, ...t''_k] \in \mathbb{R}^{C\times H\times W}$ of the red stream is obtained by rescaling T using element-wise multiplication:

$$t''_k = y'_k t_k \tag{5.25}$$

The blue stream

As with in the Channel SE-Attention block, the final output feature map $\widetilde{T} \in \mathbb{R}^{C\times H\times W}$ is as follows:

$$\widetilde{T} = \epsilon T' + \zeta T'' + T \tag{5.26}$$

where ϵ and ζ are initialized to 0 and gradually learned to assign more weight. The resulting feature map $\widetilde{T}$ can be regarded as a weighted sum of the features across all channels and the original input features. Thus, this block has a global contextual view and selectively aggregates contexts in the position dimension. The aggregated positional features achieve to improve intra-class compact and semantic consistency.

5.4.4 Allocated Feature

To this end, we add the channel-wise and position-wise resulting feature maps to obtain the final allocated feature map $\widetilde{F} \in \mathbb{R}^{C\times H\times W}$, which contains the organized frequency and spatial information, as follows:

$$\widetilde{F} = \widetilde{S} + \widetilde{T}. \tag{5.27}$$

The allocated feature which emphasizes the potential target location and yields the distinctive representation, is fed to the detection head illustrated in the Appendix section.

5.5 Detection Head

In this section, we introduce the detection head of the Deep-IRTarget. As aforementioned, we produce an allocated feature map, which emphasizes the potential target location and yield the distinctive representation of the target. To this end, we use the Region-based CNN [12] as the detection head.

We feed the feature map $\widetilde{F}$ to the Region Proposal Network (RPN) to generate Region of Interests (RoIs). Then, the target location (bounding box) is predicted by regression after Non-maximum suppression (NMS), and the target class is predicted by classification. For training the RPN, we randomly select 256 candidate boxes as a batch, which includes 64 positive samples (IoU with ground truth larger than 0.7, IoU is Intersection over Union) and 192 negative samples (IoU with ground truth larger than 0.1 and smaller than 0.7).The loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = \sum_i \mathcal{L}_{cls}\left(p_i, p_i^*\right) + \sum_i p_i^* \cdot \mathcal{L}_{loc}\left(t_i, t_i^*\right), \tag{5.28}$$

where $\mathcal{L}_{cls}\left(p_i, p_i^*\right)$ is the loss of classification, and $p_i^* \cdot \mathcal{L}_{loc}\left(t_i, t_i^*\right)$ is the loss of bounding box regression. p_i^* and p_i indicate ground-truth and predicted classification of the proposal region i. t_i is the predicted vector representing the offset between the i-th proposal and its corresponding ground-truth bounding box, and t_i^* is the ground truth. We use the cross entropy loss as $\mathcal{L}_{cls}$ and SmoothL1 as the loss function of $\mathcal{L}_{loc}$.

5.6 Experiments and Discussions

5.6.1 Implementation Details

We adopt the synchronized Adam solver [139] for training on two GPUs with synchronized batch normalization, and follow the image-centric sampling strategy in [12]. A mini-batch has one image per GPU and 256 proposals per image for detector training. Note that, we pretrain the CNN-Spatial and CNN-Fuse on the grey ImageNet 1000-class dataset, in order to prevent the neural network learning color features. Other layers are randomly initialized by sampling from a Gaussian distribution with the mean of 0 and standard deviation of 0.001. The base learning rate is set to 0.02. Momentum and weight decay coefficients are set to 0.9 and

0.0005. The ratio of positive and negative anchors in each image is set to 1:3. The whole training procedure takes around 28 hours on two NVIDIA 1080Ti Pascals using the PyTorch framework [140].

5.6.2 Datasets

To evaluate the proposed method, we carry out comprehensive experiments on MWIR, BITIR and WCIR dataset.

MWIR dataset is a mid-wave infrared dataset collected by the US NVESD. It contains over 200GB of MWIR data which contains different non-human targets in the sensor range of 1000 metres. For our experiments, we consider all 10 target types, including the Ford F150, Sport Utility Vehicle (SUV), BTR70, BMP2, BRDM2, T72, ZSU23-4, 2S3, MTLB and D20.

BITIR dataset is collected by Beijing Institute of Technology in several scenarios, where longitude and latitude are $118°55'XX''E$ and $44°53'XX''N$. It contains five categories: sedan, jeep, van, tank and bus. In this dataset, there are 2500 training samples and 500 test samples, around 30% in near range (less than 30 m), 60% in medium range (30 m to 100 m), and 10% in far range (more than 100 m).

WCIR dataset is collected from the Internet and labeled by Beijing Institute of Technology, containing 2000 training images and 800 test images captured under natural conditions, such as different weather and complex backgrounds in the real world. It contains five categories: sedan, SUV, van, pickup and truck.

5.6.3 Baseline

The proposed Deep-IRTarget is compared with the following state-of-the-art methods:

Shallow models:

- DCT+PSO: This method is composed of a detector based on discrete cosine transform (DCT)[141] and a classifier based on particle swarm optimization (PSO)[142]. DCT is an approach of bottom-up saliency detection, and calculated in the frequency domain.

- GP-PSO: The gradient-boosted particle swarm optimization (GP-PSO)[142] improves the computational efficiency by taking advantage of the analytical nature of the objective function.

- DCT-LLC: DCT-LLC[54] is proposed to jointly localize and classify a target by DCT-based detection of the region of interest and local descriptor and the locality-constrained linear coding (LLC) for target recognition.

- VCRF: Vehicle detection with Centro Ricerche FIAT (CRF)[143] is a traditional algorithm for detecting preceding vehicles in IR images using the temperature characteristic.

- SURF-SVM: This algorithm combines the local and global features based on Speeded Up Robust Features (SURF) [144] for vehicle recognition in far infrared images [54] . The SURF-baed representation is invariant to the scale and the number of local features.

Deep models:

- DRSI: This network based on deep representation for searching infrared object (DRSI) [101] handles the different scales of IR target detection in a wild dataset.

- 2D-DBN: 2D-DBN[55] proposes a local adaptive threshold based on maximum distance (2D) for target searching and a vehicle candidate verification model based on a deep belief network (DBN).

- YOLOv3IR: YOLOv3IR[58] is a network for low-resolution infrared image target detection. This network is based on the YOLOv3 [4]. It tackles the limited data training problem and the low resolution problem. In our experiments, we implement the YOLOv3IR.

- Faster R-CNN: This is a famous framework for object detection. Faster RCNN[12] uses the RPN to reach real-time processing.

- ExtremeNet: This method[19] tackles the general object detection with a novel bottom-up idea. It firstly detects four extreme points and one center point of objects using a standard keypoint estimation network. Then, the bounding boxes can be calculated using these points.

5.6.4 Evaluation Metrics

To evaluate the proposed network, we use Average Precision (AP) and mean Average Precision (mAP) as the measurement metrics on the three datasets, as shown in Eq. 5.29 and Eq. 5.30.

$$AP = \int_0^1 p(x)dx, \tag{5.29}$$

$$mAP = \frac{1}{N}\sum_{q=1}^{N} AP(q). \tag{5.30}$$

Average precision computes the average value of $p(x)$ over the interval from $x = 0$ to $x = 1$ and is the area under the precision-recall curve. For mAP, N is the number of categories. For a candidate region, if the Intersection-over-Union (IoU) is larger than 0.7, we believe this region contains the target and the detecter performs well. IoU is defined by:

$$IoU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth}. \tag{5.31}$$

5.6.5 Evaluation on MWIR Dataset

We start the experimental evaluation from the MWIR dataset. The performance is computed by comparing the predictions on the MWIR validation set with the ground truth. In Table 5.3, it is clear that deep learning models achieves a superior performance compared to shallow models on the MWIR dataset. And our Deep-IRTarget outperforms all other state-of-the-art methods, achieving much higher performance than shallow models. The Deep-IRTarget scores of SUV, T72, ZSU23-4 and 2S3 are relatively high, because their features of the RoIs are more obvious when using our proposed feature extraction. Especially, BTR70 is hard to construct a distinguishable representation, because its appearance is so similar to other tanks, like BRDM2, BMP2. The reason why the scores of F150 and D20 are low is that they are difficult to distinguish from the background and hard to localize.

Furthermore, we show some illustrative results in Fig. 5.4, using different colored bounding boxes to present different classes. The first column is the ground truth, and the last column is the result of our method. Compared with other methods, DeepIR-Target can not only predict the accurate target location, but also the correct category. This dataset mainly contains the long-distance target, so it can effectively evaluate the feature extraction ability of different methods. In the second row, DCT-LLC and Faster RCNN detect the target, but label the wrong class to the target. The reason is they cannot learn the discriminative representation. In comparison with our Deep-IRTarget, the localization accuracy of YoloV3IR and ExtremeNet is lower, since the CNN cannot extract features as effective as the infrared frequency feature. In addition, from Fig. 5.4, we can observe that the proposed method is robust to lighting and different weather. For example, the first row shows the results

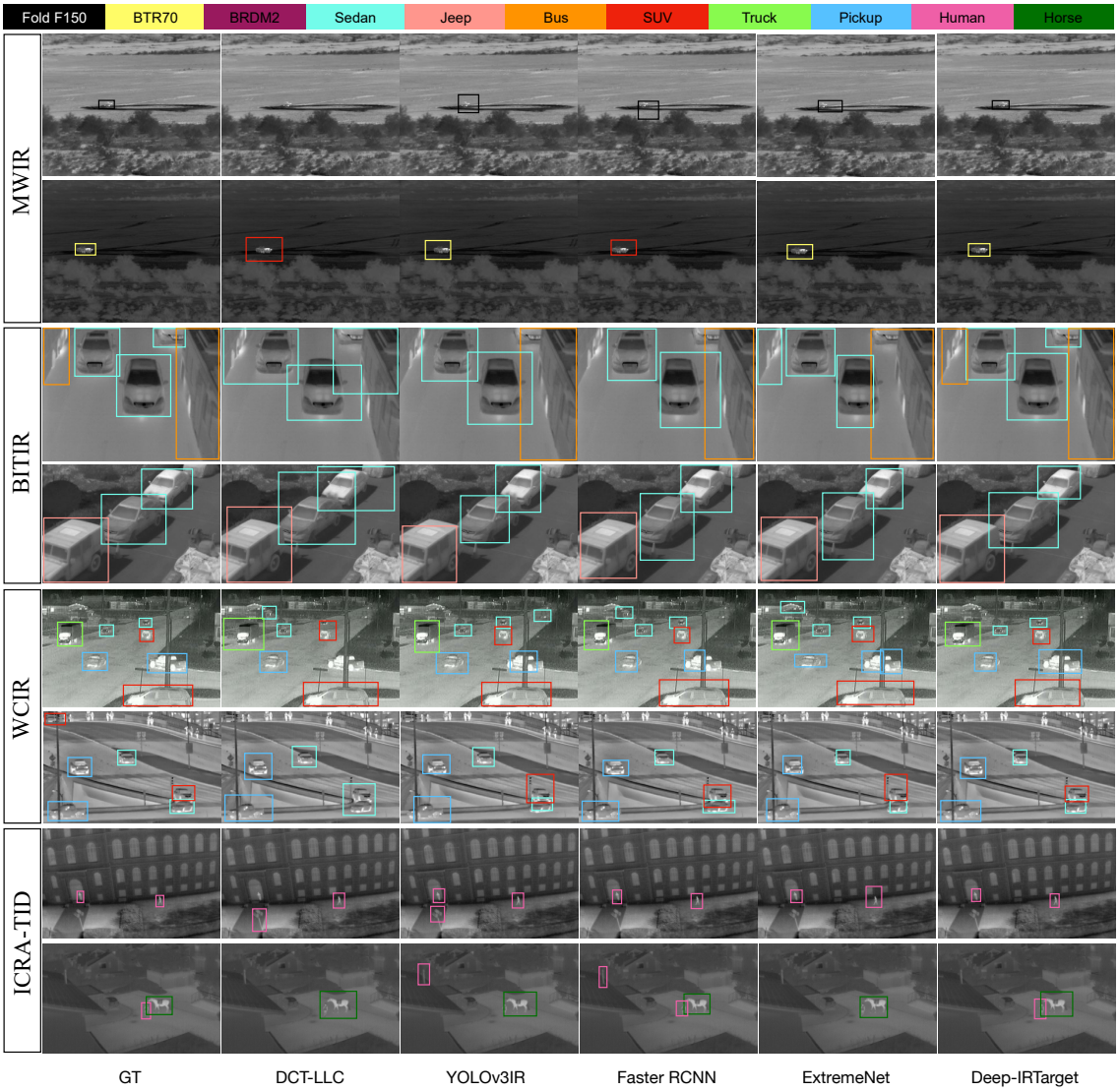


Figure 5.4 : An illustration of the detection results on MWIR, BITIR and WCIR dataset, comparing with other methods. The first group of two rows is related to the MWIR dataset, the second group of two rows is related to the BITIR dataset, the last six to the WCIR dataset. The first column is the ground truth of the samples. Different colored bounding boxes mean different classes.

Table 5.3 : The performance of our method and the baselines on the MWIR dataset.

	Method	MWIR dataset (unit:%) ↑										
		F150	**SUV**	**BTR70**	**BRDM2**	**BMP2**	**T72**	**ZSU23-4**	**2S3**	**MTLB**	**D20**	**mAP**
Shallow model	DCT+PSO[141]	52.49	53.85	50.61	55.77	58.01	62.48	58.61	50.34	53.08	49.92	54.52
	GP-PSO[142]	53.69	53.58	50.82	54.69	59.56	63.60	57.75	49.82	52.20	50.94	54.67
	DCT-LLC[54]	50.38	52.79	40.87	55.77	49.80	59.43	58.23	56.01	56.81	48.50	52.86
	VCRF[143]	44.97	43.86	34.76	44.47	40.98	46.12	46.67	45.10	44.61	40.20	43.17
	SURF-SVM[144]	60.51	63.82	55.93	63.89	60.46	67.47	65.72	63.03	62.27	58.95	62.21
Deep model	DRSI[101]	70.78	74.58	58.93	73.64	68.57	80.37	75.90	74.26	73.83	64.10	71.50
	2D-DBN[55]	61.96	65.47	53.72	64.18	59.47	70.38	66.80	66.40	60.98	58.41	62.78
	YOLOv3IR[58]	72.43	75.32	62.90	73.40	70.93	79.29	78.15	74.20	74.10	66.80	72.75
	Faster RCNN[12]	69.56	73.44	56.78	73.09	68.10	76.38	74.99	76.29	70.35	64.42	70.34
	ExtremeNet[19]	70.31	76.21	60.89	72.54	69.40	79.01	76.48	75.25	74.50	65.00	71.92
	Deep-IRTarget	**79.28**	**89.78**	**71.80**	**82.21**	**80.66**	**87.76**	**88.39**	**90.98**	**82.41**	**77.34**	**83.06**

in the strong sunlight, while the second row shows the results on a cloudy night.

5.6.6 Evaluation on BITIR Dataset

To ensure that our approach can generalize well to other real datasets, we performed the same experimental evaluation procedure also on the BITIR dataset. In Table 5.4, we list the quantitative performance on the BITIR dataset. As can be seen from the AP scores, the proposed method outperforms others. These outstanding results are attributed to the strength of the infrared frequency saliency extraction and the RAF model, which have a much better ability to extract the potential target than the use of single handcrafted or CNN features.

The second group of the Fig. 5.4 refers to the BITIR dataset, illustrating that our method can cope with different terrain and different angles, in comparison with the MWIR dataset. As we can see, only our method tackles the extreme occlusions, such as the top left bus and the top right sedan in the third row.

5.6.7 Evaluation on WCIR Dataset

We present the results of experiments on the WCIR dataset to demonstrate that the proposed detector achieves promising results. We also use Average Precision as the evaluation criteria. We report the target detection results of the Deep-IRTarget as well as results of current state-of-the-art methods on the WCIR test set in Table 5.5. Compared with the performance on MWIR and BITIR dataset, our result on WCIR is barely satisfactory but still outperforms other methods. The main factor is the WCIR dataset is more challenging due to the numerous targets in one image. Another factor is some targets are too small to recognize their outlines, because they are too far away from the sensor to reach the effective range of the sensor.

From the bottom two rows of the Fig. 5.4, we can see that the proposed method can detect all the targets of different scales and sizes on the WCIR dataset. In

Table 5.4 : The results on the BITIR dataset.

Method	BITIR dataset (unit:%) ↑					
	Sedan	**Jeep**	**Van**	**Tank**	**Bus**	**mAP**
DCT+PSO[141]	58.12	43.89	42.43	65.32	69.21	55.79
GP-PSO[142]	56.21	45.42	43.79	64.93	68.23	55.72
DCT-LLC[53]	50.32	48.32	46.65	53.29	60.22	51.76
VCRF[143]	42.12	36.65	30.45	44.47	47.34	40.20
SURF-SVM[144]	62.79	52.35	55.93	64.97	67.23	60.65
DRSI[101]	69.88	63.74	62.40	70.72	68.19	66.99
2D-DBN[55]	64.88	56.23	53.86	63.19	61.75	59.98
YOLOv3IR[58]	73.50	64.28	64.78	72.59	74.17	69.86
Faster RCNN[12]	71.58	62.66	64.95	70.43	73.88	68.70
ExtremeNet[19]	73.44	64.74	63.56	72.54	72.88	69.43
Deep-IRTarget	**83.29**	**72.33**	**73.76**	**82.13**	**83.31**	**78.96**

the fifth row, Deep-IRTarget detects the SUV, pickup, truck and sedan successfully, while other methods make some mistakes. DCT-LLC, YOLOv3IR, Faster RCNN and ExtremeNet wrongly detect background clutter chips as targets. In the sixth row, all the methods cannot detect the top left target, but our method still outperforms others.

5.6.8 Discussions

In summary, our method outperforms others on the MWIR, BITIR and WCIR dataset. As we can see in Table 5.3, 5.4, 5.5, and Fig. 5.4, it is convincing that the Deep-IRTarget correctly localizes and recognizes the IR target from various backgrounds by extracting and allocating both frequency and spatial features. It is

Table 5.5 : The results on the WCIR dataset.

Method	WCIR dataset (unit:%) ↑					
	Sedan	**SUV**	**Van**	**Pickup**	**Truck**	**mAP**
DCT+PSO[141]	47.32	46.98	50.90	45.19	51.04	48.28
GP-PSO[142]	46.58	46.16	49.06	43.75	48.28	46.77
DCT-LLC[54]	43.78	42.56	47.49	43.01	48.64	45.10
VCRF[143]	31.47	32.63	36.08	30.45	37.97	33.72
SURF-SVM[144]	51.38	50.92	55.77	48.80	54.17	52.21
DRSI[101]	58.78	59.73	63.29	58.03	62.10	60.39
2D-DBN[55]	55.43	52.87	58.67	51.01	57.46	55.08
YOLOv3IR[58]	65.32	64.85	70.43	63.78	71.45	67.17
Faster RCNN[12]	64.83	63.11	66.31	60.42	69.34	64.80
ExtremeNet[19]	64.73	64.74	69.38	64.04	70.93	66.76
Deep-IRTarget	**69.90**	**70.53**	**80.95**	**70.37**	**84.35**	**75.22**

worth pointing out that the size of input images is non-restricted, which means that Deep-IRTarget can cope with both low-resolution and high-resolution images.

Compared with two PSO-based methods DCT+PSO and GP-PSO, our method achieves a great improvement. The reason is that they are based on a shape generative model and shape matching, which are restricted from the quality of the IR image. Both DCT-LLC and our method consider the frequency information, but DCT-LLC ignore the spatial features and is a shallow model, so our method is better. VCRF performs the worst, because it only uses the temperature feature. SURF-SVM performs the best among the shallow models, because it combines several handcrafted features.

The deep learning-based methods perform better than the shallow models. Com-

pared with YOLO-based DRSI and YOLOv3IR, our method outperforms these methods by approximately 20%. This is because we design a frequency saliency extraction, and integrate the frequency and spatial features using the attention mechanism. 2D-DBN localizes targets using a local adaptive threshold based on maximum distance, so it may fail in some challenging scenarios. The detection head of our method is from Faster RCNN, which is a general object detector, but it ignores infrared features. ExtremeNet is an anchor-free method that can effectively detect objects based on key points estimation, but the backbone is not sufficient for infrared feature extraction. Hence, ExtremeNet also cannot achieve a good result. To summarize, compared with these deep learning methods, our advantages include two parts. (1) Besides the infrared CNN feature, we specify the infrared feature in the frequency domain. (2) Instead of simply superimposing features, we integrate feature maps using the proposed resource allocation strategy.

5.6.9 Ablation Study

To evaluate the effect of some essential components of our Deep-IRTarget, we implement and test several variants of our model. As aforementioned, the Deep-IRTarget mainly contributes two parts, the dual-domain feature extraction part and the feature resource allocation part. In the DFE module, we consider both frequency feature by HIFT and spatial feature by CNN-Spatial (CNN-S) in two domain. In the RAF module, we mainly design a Channel SE-Net (RAF-C) and a Position SE-Net (RAF-P).

The experimental results are shown in Table 5.6. It is clear that each proposed module is significant for the Deep-IRTarget. With only spatial features (CNN-S), the performance decreases dramatically by around 16% compared with the standard Deep-IRTarget framework. When we implement CNN-S and the full RAF module, the performance has a huge increase compared with the use of only spatial features,

Table 5.6 : The ablation study of our method.

Module				mAP (unit:%) ↑		
CNN-S	**HIFT**	**RAF-C**	**RAF-P**	**MWIR**	**BITIR**	**WCIR**
✓				69.34	64.70	62.80
✓		✓	✓	77.29	72.97	70.44
✓	✓			78.77	74.12	72.67
✓	✓		✓	79.94	77.62	74.19
✓	✓	✓		81.37	77.84	74.40
✓	✓	✓	✓	**83.06**	**78.96**	**75.22**

to 78.77% on MWIR, 74.12% on BITIR and 72.67% on WCIR respectively. A similar trend occurs, when the CNN-S and the HIFT module are used. Note that, we observe the HIFT contributes a little more than the RAF module. This trend can also be seen in the Fig. 5.5 and Fig. 5.6. Therefore, we believe that effective feature extraction is a little more important than feature post-processing, since the performance without frequency features will decrease more than that without resource allocation for features. In addition, we also explore the effect of Channel SE-Net and Position SE-Net. Form the second last and third last line of the Table 5.6, we find that the Deep-IRTarget without RAF-P outperforms that without RAF-C, meaning the channel attention contributes more than the position attention. We believe that the channel-wise feature selection and fusion between frequency feature maps and spatial feature maps is more effective.

HIFT v.s. HFT v.s. PFT

The proposed HIFT, Hypercomplex Fourier Transform (HFT) [131] and Phase spectrum of Fourier Transform (PFT) [145] are based on Fourier transforms for

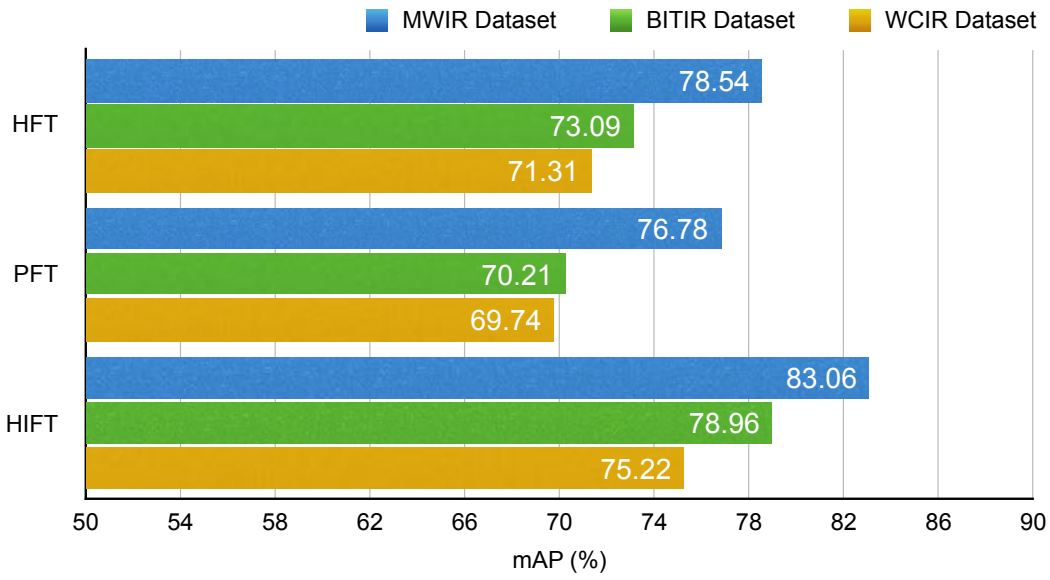


Figure 5.5 : The performance of the Deep-IRTarget with HFT, PFT, and HIFT on MWIR, BITIR and WCIR.

image saliency detection. HFT performs the analysis of RGB image in the frequency domain, and defines three-color channels for the hypercomplex feature matrices. By extracting the spectral residual of an image in spectral domain, PFT can quickly construct the corresponding saliency map in the spatial domain.

We validate the performance of the Deep-IRTarget method by replacing HIFT with HFT or PFT, as shown in Fig. 5.5. On all datasets, the Deep-IRTarget with HIFT outperforms that with others. With HFT, the performance is better than PFT, and also better than the Deep-IRTarget without HIFT which is shown in the fourth row of the Table 5.6. On the contrary, PFT reduces the performance, as it only considers the phase spectrum and fails to detect the infrared saliency.

RAF v.s. SENet v.s. DANet

The proposed RAF model is inspired from the self-attention mechanism [134]. Here, we conduct the experiment to compare RAF with two mainstream attention

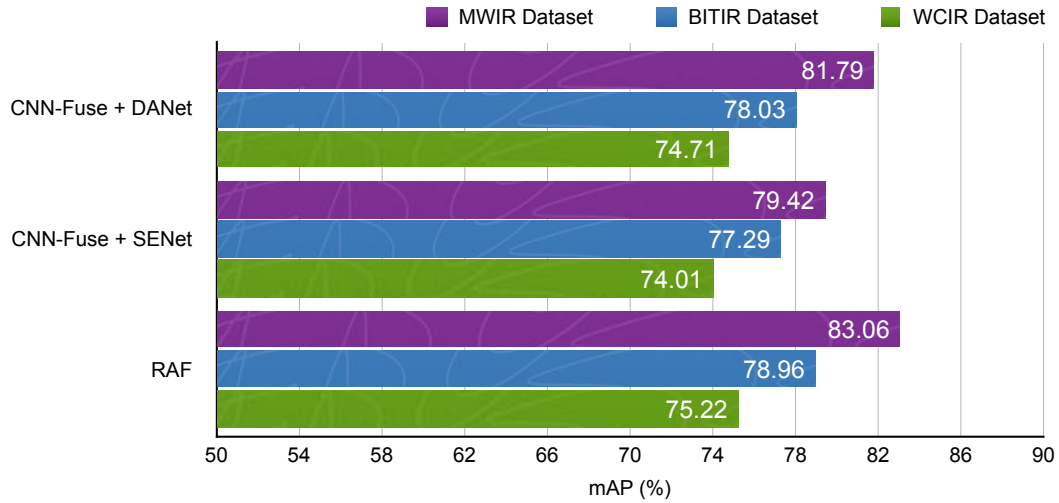


Figure 5.6 : The performance of the Deep-IRTarget with DANet, SENet, and RAF on MWIR, BITIR, and WCIR.

models, *i.e.* Squeeze-and-Excitation Network (SENet) [136] and Dual Attention Network (DANet) [135]. SENet presents an attention model for digging channel-wise features, which is the same as the red steam of our Channel SE-Attention block. DANet uses two types of attention modules, a position attention module and a channel attention module, to catch the inner-position and inner-channel relationship.

We evaluate the performance of the Deep-IRTarget method by replacing the RAF model with DANet or SENet. The result is illustrated in Fig. 5.6, showing that our RAF is superior to DANet and SENet, by 1.27% and 3.64% on the MWIR dataset. The gap among them on the BITIR and WCIR is relatively small. For DANet, its performance is better than SENet. From Table 5.6 and Fig. 5.6, it is interesting to see that the Deep-IRTarget with DANet outperforms that without RAF-P, and the Deep-IRTarget with SENet outperforms that without RAF-C and RAF-P. This verifies our hypothesis, "Attention can improve the dual-domain feature source allocation".

5.7 Conclusion

In this chapter, we present a Deep-IRTarget framework based on region-based deep learning for vehicle detection and recognition in the infrared imagery. The proposed framework jointly considers the discriminative feature in the frequency domain and spatial domain. In the dual-domain feature extraction, the HIFT module is proposed to estimate the infrared intensity saliency, and the CNN is invoked to extract the spatial feature. Specifically, we develop the RAF model for dual-domain feature fusion and resource allocation. It recalibrates channel-wise feature and position-wise feature responses by explicitly modelling interdependencies between channels and positions. The multiple experiments demonstrate substantial improvements in the infrared target detection in comparison with the sate-of-the-art on the MWIR, BITIR and WCIR datasets. The ablation studies show that the proposed HIFT module captures the infrared saliency, and the RAF can enhance the feature effectively.

Chapter 6

Conclusions

This thesis explores the problem of reasoning about the visual world by understanding the objects that exist in it. We study region-based deep learning to understand objects in different real-world scenarios. The objects include the human face, the player and the infrared vehicle. The corresponding tasks are facial component and landmark detection (Chapter 3), multi-player identification and tracking (Chapter 4), and infrared vehicle detection (Chapter 5). The main conclusions of this thesis are as follows.

- To tackle large occlusion and limited training data, we propose a weakly-supervised convolutional neural network to detect facial components and landmarks simultaneously. This method effectively learns the position of the component through the generated training data with weak labels, thereby improving the accuracy of landmark detection. Meanwhile, a two-branch landmark detection model based on classification and regression is proposed to further improve the performance. The experiments achieve state-of-the-art results on several benchmarks.

- To cope with identity switches in multi-player tracking in real-world sports video, we propose a distinguishable deep representation for player identity, considering pose-guided partial features, team class, and jersey number. A robust multi-player tracker incorporating with deep player identification is further developed to produce identity-coherent trajectories. Experiment results illustrate that our framework handles the identity switches effectively,

and outperforms state-of-the-art trackers on the sports video benchmarks.

- To handle the poor texture information, low resolution and high noise of infrared images, we propose a backbone network to exploit latent features. Particularly, the Hypercomplex Infrared Fourier Transform is developed to extract the infrared intensity saliency in the frequency domain. A Resource Allocation model for Features (RAF) is proposed to recalibrate features efficiently. The experiments on three challenging infrared imagery databases substantiate merits of the proposed method.

This thesis focuses on the research and development of object understanding based on region-based deep learning, and has achieved meaningful results in different real-world scenarios. Based on the research content and results, we have in-depth thinking about the next research work. Future directions for the work described in this dissertation include three parts.

- We intend to combine our weakly-supervised learning with semi-supervised learning to detect landmarks of jawline. Moreover, DCGAN and LR-CNN models could share features and be reformed to an end-to-end model to accelerate training and testing.

- We intend to trade off the accuracy of tracking against real-time performance, and try to consider the temporal information of the player to refine the 2D detection. And we will also transfer our framework to other real-world scenarios, such as football and baseball matches.

- We carry out research on infrared target tracking of continuous frames, which is expected to further improve the target detection results, combined with the LSTM network dedicated to sequence processing to achieve infrared target detection and tracking.

Bibliography

[1] D. H. Ballard and C. M. Brown, "Computer vision. englewood cliffs," *J: Prentice Hall*, 1982.

[2] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[5] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1614–1627, 2014.

[6] R. Zhang, M. Xu, Y. Shi, J. Fan, C. Mu, and L. Xu, "Infrared target detection using intensity saliency and self-attention," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1991–1995.

[7] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

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

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[14] P. Goyal, R. Girshick, K. He, P. Dollár, and T. Lin, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, pp. 2999–3007, 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[18] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[19] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *CVPR*, 2019.

[20] Y. Yi, D. Qu, and F. Xu, "Face detection method based on skin color segmentation and facial component localization," in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, vol. 1. IEEE, 2010, pp. 64–67.

[21] M. Urschler, M. Storer, H. Bischof, J. A. Birchbauer, and S. B. Center, "Robust facial component detection for face alignment applications," in *Proc. 33rd Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM)*, 2009, pp. 61–72.

[22] J. Naruniec, "Discrete area filters in accurate detection of faces and facial features," *Image and Vision Computing*, vol. 32, no. 12, pp. 979–993, 2014.

[23] K. Sudhakar and P. Nithyanandam, "An accurate facial component detection using gabor filter," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 3, pp. 287–294, 2017.

[24] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape

models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[25] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[26] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1513–1520.

[27] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 177–190, 2014.

[28] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 386–391.

[29] H. Fan and E. Zhou, "Approaching human level facial landmark localization by deep learning," *Image and Vision Computing*, vol. 47, pp. 27–35, 2016.

[30] Y. Wu and Q. Ji, "Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3400–3408.

[31] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 918–930, 2015.

[32] S. Gerke, K. Muller, and R. Schafer, "Soccer jersey number recognition using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 17–24.

[33] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao, "Jersey number detection in sports video for athlete identification," in *Visual Communications and Image Processing 2005*, vol. 5960. International Society for Optics and Photonics, 2005, p. 59604P.

[34] G. Li, S. Xu, X. Liu, L. Li, and C. Wang, "Jersey number recognition with semi-supervised spatial transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1783–1790.

[35] R. Zhang, C. Mu, M. Xu, L. Xu, and X. Xu, "Facial component-landmark detection with weakly-supervised lr-cnn," *IEEE Access*, vol. 7, pp. 10 263–10 277, 2019.

[36] M. Bertini, A. Del Bimbo, and W. Nunziati, "Matching faces with textual cues in soccer videos," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 537–540.

[37] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.

[38] T. Yamamoto, H. Kataoka, M. Hayashi, Y. Aoki, K. Oshima, and M. Tanabiki, "Multiple players tracking and identification using group detection and player number recognition in sports video," in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 2442–2446.

[39] A. Senocak, T.-H. Oh, J. Kim, and I. So Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1732–1739.

[40] S. Gerke, A. Linnemann, and K. Müller, "Soccer player recognition using spatial constellation features and jersey number recognition," *Computer Vision and Image Understanding*, vol. 159, pp. 105–115, 2017.

[41] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[42] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 137–144.

[43] J. Liu, P. Carr, R. T. Collins, and Y. Liu, "Tracking sports players with context-conditioned motion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1830–1837.

[44] Y. Liu, J. Yin, D. Yu, S. Zhao, and J. Shen, "Multiple people tracking with articulation detection and stitching strategy," *Neurocomputing*, 2019.

[45] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast online tracking with detection refinement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 162–173, Jan 2018.

[46] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[47] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 353–369.

[48] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[49] S. Zhou, P. Yang, and W. Xie, "Infrared image segmentation based on otsu and genetic algorithm," in *2011 International Conference on Multimedia Technology*. IEEE, 2011, pp. 5421–5424.

[50] B. Zhang, T. Zhang, Z. Cao, and K. Zhang, "Fast new small-target detection algorithm based on a modified partial differential equation in infrared clutter," *Optical engineering*, vol. 46, no. 10, p. 106401, 2007.

[51] S. Sun and H. W. Park, "Segmentation of forward-looking infrared image using fuzzy thresholding and edge detection," *Optical Engineering*, vol. 40, no. 11, pp. 2638–2646, 2001.

[52] U. M. Braga-Neto, M. Choudhury, and J. I. Goutsias, "Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators," *Journal of Electronic Imaging*, vol. 13, no. 4, pp. 802–814, 2004.

[53] H. Lu, L. Zhang, M. Zhang, X. Hu, and S. Serikawa, "A method for infrared image segment based on sharp frequency localized contourlet transform and morphology," in *2010 International Conference on Intelligent Control and Information Processing*. IEEE, 2010, pp. 79–82.

[54] B. Besbes, A. Apatean, A. Rogozan, and A. Bensrhair, "Combining surf-based local and global features for road obstacle recognition in far infrared images,"

in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1869–1874.

[55] H. Wang, Y. Cai, X. Chen, and L. Chen, "Night-time vehicle sensing in far infrared image with deep learning," *Journal of Sensors*, vol. 2016, 2016.

[56] Y. Cai, Z. Liu, H. Wang, and X. Sun, "Saliency-based pedestrian detection in far infrared images," *IEEE Access*, vol. 5, pp. 5013–5019, 2017.

[57] M. Ding, Z. Sun, L. Wei, Y. Cao, and Y. Yao, "Infrared target detection and recognition method in airborne photoelectric system," *Journal of Aerospace Information Systems*, vol. 16, no. 3, pp. 94–106, 2019.

[58] G. Zheng, X. Wu, Y. Hu, and X. Liu, "Object detection for low-resolution infrared image in land battlefield based on deep learning," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8649–8652.

[59] N. M. Nasrabadi, "Deeptarget: An automatic target recognition using deep convolutional neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2687–2697, Dec 2019.

[60] R. R. Atallah, A. Kamsin, M. A. Ismail, S. A. Abdelrahman, and S. Zerdoumi, "Face recognition and age estimation implications of changes in facial features: A critical review study," *IEEE Access*, vol. 6, pp. 28 290–28 304, 2018.

[61] Z. Xiang, H. Tan, and W. Ye, "The excellent properties of a dense grid-based hog feature on face recognition compared to gabor and lbp," *IEEE Access*, vol. 6, pp. 29 306–29 319, 2018.

[62] C. Qi, M. Li, Q. Wang, H. Zhang, J. Xing, Z. Gao, and H. Zhang, "Facial expressions recognition based on cognition and mapped binary patterns," *IEEE Access*, vol. 6, pp. 18 795–18 803, 2018.

[63] J. Roth, Y. Tong, and X. Liu, "Adaptive 3d face reconstruction from unconstrained photo collections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4197–4206.

[64] D. Xi and S.-W. Lee, "Face detection and facial component extraction by wavelet decomposition and support vector machines," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 199–207.

[65] B. A. Efraty, M. Papadakis, A. Profitt, S. Shah, and I. A. Kakadiaris, "Facial component-landmark detection," in *Face and Gesture 2011*. IEEE, 2011, pp. 278–285.

[66] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European conference on computer vision*. Springer, 2014, pp. 1–16.

[67] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim, "Robust facial landmark detection via recurrent attentive-refinement networks," in *European conference on computer vision*. Springer, 2016, pp. 57–72.

[68] X. Wu, J. Zhou, and Y. Pan, "Initial shape pool construction for facial landmark localization under occlusion," *IEEE Access*, vol. 5, pp. 16 649–16 655, 2017.

[69] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483.

[70] A. Kumar, R. Ranjan, V. Patel, and R. Chellappa, "Face alignment by local deep descriptor regression," *arXiv preprint arXiv:1601.07950*, 2016.

[71] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 88–97.

[72] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[73] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European conference on computer vision*. Springer, 2012, pp. 679–692.

[74] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2013, pp. 896–903.

[75] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2879–2886.

[76] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[77] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[78] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual

recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[79] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[80] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 532–539.

[81] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1944–1951.

[82] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1851–1858.

[83] S. Zhu, C. Li, C. Change Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4998–5006.

[84] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European conference on computer vision*. Springer, 2014, pp. 94–108.

[85] Z. Shao, S. Ding, Y. Zhao, Q. Zhang, and L. Ma, "Learning deep representation from coarse to fine for face alignment," *arXiv preprint arXiv:1608.00207*, 2016.

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[87] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[88] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[89] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[90] T. D'Orazio and M. Leo, "A review of vision-based systems for soccer video analysis," *Pattern recognition*, vol. 43, no. 8, pp. 2911–2926, 2010.

[91] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3623–3632.

[92] W. Wang, J. Shen, F. Porikli, and R. Yang, "Semi-supervised video object segmentation with super-trajectories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 985–998, 2019.

[93] J. Boisvert, M.-A. Drouin, and P.-M. Jodoin, "High-speed transition patterns for video projection, 3d reconstruction, and copyright protection," *Pattern Recognition*, vol. 48, no. 3, pp. 720–731, 2015.

[94] T. T. Pribanic, T. T. Petkovic, and M. Monlic, "3d registration based on the direction sensor measurements," *Pattern Recognition*, vol. 88, pp. 532–546, 2019.

[95] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[96] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3395–3402.

[97] W. Wang, J. Shen, J. Xie, and F. Porikli, "Super-trajectory for video segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1671–1679.

[98] Z. Niu, X. Gao, and Q. Tian, "Tactic analysis based on real-world ball trajectory in soccer video," *Pattern Recognition*, vol. 45, no. 5, pp. 1937–1947, 2012.

[99] H. Possegger, S. Sternig, T. Mauthner, P. M. Roth, and H. Bischof, "Robust real-time tracking of multiple objects by volumetric mass densities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2395–2402.

[100] X. Shen, X. Sui, K. Pan, and Y. Tao, "Adaptive pedestrian tracking via patch-based features and spatial–temporal similarity measurement," *Pattern Recognition*, vol. 53, pp. 163–173, 2016.

[101] R. Zhang, C. Mu, Y. Yang, and L. Xu, "Research on simulated infrared image utility evaluation using deep representation," *Journal of Electronic Imaging*, vol. 27, no. 1, p. 013012, 2018.

[102] R. Zhang, C. Mu, M. Xu, L. Xu, Q. Shi, and J. Wang, "Synthetic ir image refinement using adversarial learning with bidirectional mappings," *IEEE Access*, vol. 7, pp. 153 734–153 750, 2019.

[103] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[104] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[105] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[106] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose invariant embedding for deep person re-identification," *IEEE Transactions on Image Processing*, 2019.

[107] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[108] Y. Yang, M. Xu, W. Wu, R. Zhang, and Y. Peng, "3d multiview basketball players detection and localization based on probabilistic occupancy," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–8.

[109] C. De Vleeschouwer and D. Delannay, "Basket ball dataset from the european project apidis," 2009.

[110] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[111] A. Andriyenko and K. Schindler, "Globally optimal multi-target tracking on a hexagonal lattice," in *European Conference on Computer Vision*. Springer, 2010, pp. 466–479.

[112] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 267–282, 2007.

[113] T. Sekii, "Robust, real-time 3d tracking of multiple objects with similar appearances," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4275–4283.

[114] N. S. Ghedia, C. Vithalani, and A. Kothari, "A novel approach for monocular 3d object tracking in cluttered environment," *International Journal of Computational Intelligence Research*, vol. 13, no. 5, pp. 851–864, 2017.

[115] Y. Nishikawa, H. Sato, and J. Ozawa, "Multiple sports player tracking system based on graph optimization using low-cost cameras," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018, pp. 1–4.

[116] R. T. Collins and P. Carr, "Hybrid stochastic/deterministic optimization for tracking sports players and pedestrians," in *European Conference on Computer Vision*. Springer, 2014, pp. 298–313.

[117] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 17–35.

[118] Z. Li, H.-M. Hu, W. Zhang, S. Pu, and B. Li, "Spectrum characteristics preserved visible and near-infrared image fusion algorithm," *IEEE Transactions on Multimedia*, 2020.

[119] X. Li, V. Monga, and A. Mahalanobis, "Multiview automatic target recognition for infrared imagery using collaborative sparse priors," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[120] W. Hu and H. Hu, "Disentangled spectrum variations networks for nir–vis face recognition," *IEEE Transactions on Multimedia*, vol. 22, no. 5, pp. 1234–1248, 2019.

[121] Q. Liu, Z. He, X. Li, and Y. Zheng, "Ptb-tir: A thermal infrared pedestrian tracking benchmark," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 666–675, 2019.

[122] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 763–771, 2016.

[123] H. Yang, L. Liu, W. Min, X. Yang, and X. Xiong, "Driver yawning detection based on subtle facial action recognition," *IEEE Transactions on Multimedia*, 2020.

[124] R. Zhang, C. Mu, X. Gao, K. Liu, and Y. Ma, "A fusion algorithm of template matching based on infrared simulation image," in *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, vol. 10033. International Society for Optics and Photonics, 2016, p. 1003307.

[125] L. Wang, G. Leedham, and D. S.-Y. Cho, "Minutiae feature analysis for infrared hand vein pattern biometrics," *Pattern recognition*, vol. 41, no. 3, pp. 920–929, 2008.

[126] T. Maurer, R. G. Driggers, and D. L. Wilson, "Search and detection modeling of military imaging systems," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XVI*, vol. 5784. International Society for Optics and Photonics, 2005, pp. 201–215.

[127] J. Han and B. Bhanu, "Fusion of color and infrared video for moving human detection," *Pattern Recognition*, vol. 40, no. 6, pp. 1771–1784, 2007.

[128] M. N. A. Khan, G. Fan, D. R. Heisterkamp, and L. Yu, "Automatic target recognition in infrared imagery using dense hog features and relevance grouping of vocabulary," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 293–298.

[129] M. Bertozzi, A. Broggi, C. Caraffi, M. Del Rose, M. Felisa, and G. Vezzoni, "Pedestrian detection by means of far-infrared stereo vision," *Computer vision and image understanding*, vol. 106, no. 2-3, pp. 194–204, 2007.

[130] M. Popescu, A. Paino, K. Stone, and J. M. Keller, "Detection of buried objects in flir imaging using mathematical morphology and svm," in *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications*. IEEE, 2012, pp. 1–5.

[131] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 996–1010, 2012.

[132] T. A. Ell, "Quaternion-fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems," in *Proceedings of 32nd IEEE Conference on Decision and Control*. IEEE, 1993, pp. 1830–1841.

[133] Y. Chen, N. Sang, and Z. Dan, "A saliency-based approach to detection of infrared target," in *MIPPR 2013: Automatic Target Recognition and Navigation*, T. Zhang and N. Sang, Eds., vol. 8918, International Society for Optics and Photonics. SPIE, 2013, pp. 174 – 180.

[134] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[135] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[136] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[137] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[138] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognition*, vol. 102, p. 107260, 2020.

[139] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[140] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[141] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 194–201, 2011.

[142] L. Yu, G. Fan, J. Gong, and J. P. Havlicek, "Joint infrared target recognition and segmentation using a shape manifold-aware level set," *Sensors*, vol. 15, no. 5, pp. 10 118–10 145, 2015.

[143] L. Andreone, P. Antonello, M. Bertozzi, A. Broggi, A. Fascioli, and D. Ranzato, "Vehicle detection and localization in infra-red images," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems.* IEEE, 2002, pp. 141–146.

[144] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[145] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on computer vision and pattern recognition.* Ieee, 2007, pp. 1–8.