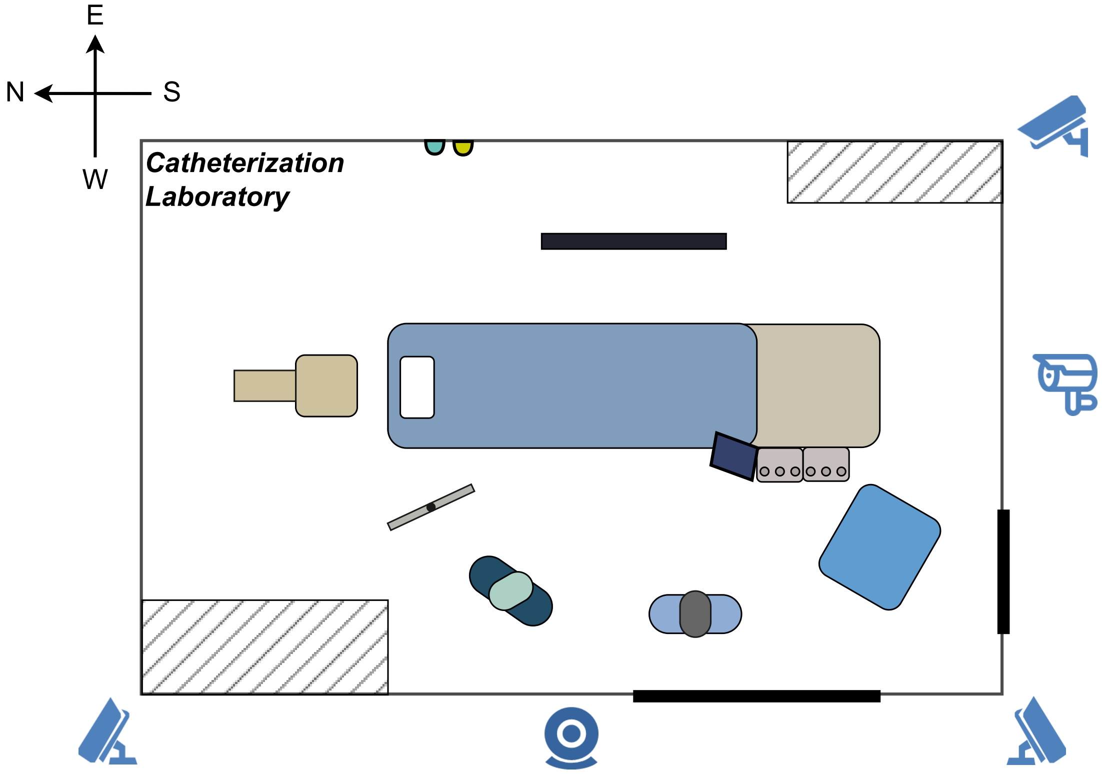
Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://cas.tudelft.nl/

CAS-2022-5369533

M.Sc. Thesis

Detecting Medical Equipment in the Catheterization Laboratory using Computer Vision

Renjie Dai B.Sc.



Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Detecting Medical Equipment in the Catheterization Laboratory using Computer Vision

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Renjie Dai B.Sc.
born in Jiaozuo, China

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Detecting Medical Equipment in the Catheterization Laboratory using Computer Vision"** by **Renjie Dai B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: Spetember 30th

Chairman: ______________________________
prof.dr.ir. Justin Dauwels

Advisor: ______________________________
ir. Rick M. Butler

Committee Members: ______________________________
dr. Jan van Gemert

____
dr. John van den Dobbelsteen

Abstract

Workflow analysis aims to improve the efficiency and safety in operating rooms by analysing surgical processes and providing feedback or support, where observations can be made and evaluated by algorithms rather than human experts. For our study, we mount five calibrated cameras from different angles in a Catheterization Laboratory (Cath Lab) to observe and analyse Cardiac Angiogram procedures.

To automate the classification of workflow and personnel activities, we propose an object detection algorithm based on Scaled-YOLOv4 with a filter to improve bounding box prediction. Scaled-YOLOv4, as a state-of-the-art technique, is featured with extremely fast processing speed and decent precision. We improve the Scaled-YOLOv4 network by using different IoU losses and an additional transformer layer.

In addition, we find that Scaled-YOLOv4 still suffers the object occlusion problem, especially in Cath Lab with limited room space but massive medical equipment. This can result in the inaccurate prediction of bounding boxes. In this work, we also develop a filter following Scaled-YOLOv4 to improve the prediction of bounding box by matching the features detected from different cameras. With the keypoints detected by SuperPoint and matched by SuperGLue, the filter adjusts the boundaries of bounding box to include all the matched keypoints and exclude unmatched points.

The proposed algorithm achieves 95.1% mAP in detecting medical equipment in Catheterization Laboratory and a real-time speed, 58 FPS on RTX 3090.

Acknowledgments

This thesis not only concludes the ten-month-long research on this project but also symbolizes the end of my two-year master study in Electrical Engineering at the Delft University of Technology. This work is inspired and supported by many people.

First, I would like to express my sincerest gratitude to my supervisor *Justin Dauwels* for his academic guidance on my thesis study in computer vision area. Your valuable expertise helped me discover interesting research ideas and develop good habits in academic research. Under you supervision, I got my first chance to present my work to others on SITB 2022 conference.

I would also like to thank my daily supervisor, *Rick Butler*, who help me solve innumerable puzzling problems in my thesis work. Rick is not only a patient teacher for me, but also an enthusiastic friend who often shares happiness and ideas with me. I'm very grateful that he took long journeys to supervise me every time.

I want to thank my teammates, Jinchen Zeng and Yingfeng Jiang. Though our thesis went in different direction, I'm still benefited from the teamwork and learning from them.

Special thanks to *Dr. Jan van Gemert* and *Dr. John van den Dobbelsteen* for being members for my thesis committee. I also want to express my gratitude to the *Circuit and System* group and Renier de Graaf Gasthuis hospital for supporting my thesis study.

I would also like to thank my friends - Yidi, Wenyu, Jinlin, Yang and others - for bringing me joy and accompany me through the bad times. Finally, I want to thank my parents who support me in economy and emotion. Without you, I couldn't have a great time during my two years of master study.

Renjie Dai B.Sc.
Delft, The Netherlands
Spetember 30th

Contents

Abstract v

Acknowledgments vii

1 Introduction 1
1.1 Background & Problem Statement 1
1.1.1 Catheterization Laboratory 1
1.1.2 Workflow Analysis 2
1.1.3 Camera System 2
1.1.4 Problem Statement 3
1.2 Pipeline 3
1.2.1 Object Detection with Multiple Cameras 4
1.2.2 The Whole Pipeline 4
1.3 Outline 5

2 Related Work 7
2.1 Clinical Workflow Analysis 7
2.2 Object Detection Algorithm 8
2.2.1 Two-stage Object Detectors 8
2.2.2 One-stage Object Detectors 9
2.3 Feature Extraction and Matching 11
2.3.1 Feature Detection and Description 12
2.3.2 Feature Matching 12

3 Methodology 15
3.1 Scaled-YOLOv4 15
3.1.1 Data Augmentation 16
3.1.2 Vision Transformer Layers 16
3.1.3 Losses 19
3.2 Keypoint-matching-based Filter 24
3.2.1 Keypoints Detection 24
3.2.2 Keypoints Matching 25
3.2.3 Clustering Filter 26
3.2.4 Matching across viewpoints and frames 28

4 Experiments and Results 31
4.1 Dataset and Evaluation matrices 31
4.1.1 Datasets 31
4.1.2 Evaluation Metrics 33
4.2 Results 36
4.2.1 Training Schedule 36
4.2.2 Fine-tuned Scaled-YOLOv4 36

4.2.3 Network Architectures 39
4.2.4 KM-filter 40

5 Conclusion 43
5.1 Summary 43
5.2 Future Work 44

List of Figures

1.1 The diagram of the Cath Lab. The medical equipment and personnel are labeled colorfully and the cameras are surrounding the laboratory. . 3
1.2 The whole pipeline of large project [1]. The pipeline consists of three parts: object detection to detect medical equipment, personnel and fixed objects; camera calibration to automatically calibrate the cameras; person tracking to estimate the 3D human poses and track personnel. . . . 5

2.1 Workflow analysis for patient management at trauma unit [2]. Prepresents the patient while N and R represent nurse and resident nurse. . . 7
2.2 The pipeline of RFID-RTLS system [3]. 8
2.3 You Only Look Once model [4]. YOLO system first divides the images into grids and predicts finite bounding boxes and corresponding confidences score and class probabilities for each grid cell. By combining these probabilities, object detection is consequently a regression problem. 10

3.1 Examples of data augmented by four popular cut-and-mix methods: (a) MixUp; (b) CutOut; (c) CutMix; (d) Mosaic 17
3.2 Overview of transformer layer. 18
3.3 The network achitecture of Scaled-YOLOv4-csp network with transformer layer at the end of backbone network. 19
3.4 The scheme and parameters for calculating the loss functions. 22
3.5 The mechanism of KM-filter to adjust boundaries by keypoint detection, matching and clustering. 25
3.6 Keypoints detected and matched with neighboring viewpoints for operating table. The keypoints filtered by left neighboring camera are mainly located at the left side of object, vice versa. The keypionts located at all the target object can be obtained by combining matched keypoints filtered by both left and right neighboring cameras. 27
3.7 The matched keypoints from different objects are seperated due to camera view shift and can be refined by the clustering algorithm. 28
3.8 The images are matched with images from neighboring cameras at the same frame or last frame. This depends on the order of updates. 29

4.1 Example images of labels in Cath Lab dataset: (a) cardiologist; (b) lab assistant; (c) patient; (d) operating table; (e) instrument table; (f) control panel display; (g) control panel button; (h) x-ray detector; (i) x-ray source; (j) lead shield; (k) radiation light; (l) display (TV). 32
4.2 Example images in Cath Lab Dataset taken from different viewpoints. . 33
4.3 The PR curve obtained in this thesis. 34
4.4 Medical equipment detected by fine-tuned Scaled-YOLOv4. 38

List of Tables

4.1 Distribution of labels in each dataset. . . . 32

4.2 Examples of TP, TN, FP, FN. . . . 34

4.3 Hyperparameters 37

4.4 Precision and recall rates of Scaled-YOLOv4 model with a range of confidence score thresholds. . . . 37

4.5 Precision, recall and mAP results for each label of fine-tuning pretrained ScaledYOLOv4-p5 network with SIoU loss function and confidence threshold 0.4. . . . 38

4.6 mAP@.5:.95 results of fine-tuned Scaled-YOLOv4 with different IoU losses for each label. . . . 39

4.7 mAP@.5:.95 results of fine-tuned Scaled-YOLOv4 with different network architectures. . . . 40

4.8 mAP@.5:.95 results of Scaled-YOLOv4 models trained from scratch with/without transformer layers for each label. . . . 41

Introduction 1

As a key task in Computer Vision, object detection is highly developed and widely used in many scenarios. For example, object detection algorithms are very powerful at detecting vehicles on the road [5], pedestrians in the room [6] and industrial equipment in factories [7], etc. In this project, five cameras are mounted from different angle in the Catheterization Laboratory (Cath Lab) to capture information for workflow analysis. We developed an object detection pipeline to detect medical equipment in the operating room through the multi-view camera footage. The developed algorithm will be performed in the Cath Lab at the Reinier de Graaf Hospital (RdGG) and provide support for the workflow analysis done by medical researchers. Although the algorithm is designed specifically for Cath Lab, our pipeline has the potential to be transplant to other scenarios and demonstrate powerful performance after retraining on the new dataset.

In this chapter, an introduction and overview of the pipeline of the thesis project are introduced. The basic introduction and problem statement to this thesis is presented in Section 1.1. In Section 1.2, the object detection algorithm based on multiple cameras and the whole pipeline developed for the CathLab are explained. Furthermore, the organization of the rest of this paper is shown in Section 1.3.

1.1 Background & Problem Statement

As mentioned before, the output of this project is very specific to the medical application and data from Cath Lab. Before we define the object detection task for this project , the motivation of implementing object detection pipeline in Cath Lab and related background is explained. In section 1.1.1, Cath Lab and some important medical equipment insides are introduced. The reason why we are interested in localizing these medical equipment is explained in section section 1.1.2, then the camera system used in Cath Lab is explained in section section 1.1.3. Finally, the problem statement of this project is presented in section section 1.1.4.

1.1.1 Catheterization Laboratory

The Cath Lab is a kind of examination room using minimally-invasive coronary physiology in a hospital. Patients with cardiovascular diseases, e.g. coronary artery disease (CAD) and myocardial ischemia [8], get examined to diagnose the cause of symptoms. For this purpose, cardiologists routinely employ pharmacologic stimulation to the coronary blood flow and use sensor-tipped angioplasty guidewires to measure the blood flow and pressure across the artery [9]. Cath Lab equipped with C-arm imaging equipment like angiographic computed tomography (ACT) and X-Ray machine is able to employ rotational angiography to get 3-dimensional (3D) angiographic images so that doctors can diagnose more artery diseases and heart disease [10].

Although Cath Lab provides a promising method to diagnose artery diseases, there are also sequrity risks in Cath Lab. The radioactive imaging equipment working continuously throughout the whole procedure poses threat to people inside the room, especially to the cardiologists, who work in Cath Lab every day. Researches show that long-term exposure to radiation from X-ray imaging would accumulate significantly high radiation dose and lead to telangiectasia and skin breakdown [11]. To protect cardiologists and lab assistants from radiation, Cath Lab also equips a transparent lead shield to absorb radiation.

1.1.2 Workflow Analysis

While workflow describes a sequence of activities performed by the personnel in a system, it is always assessed in research for monitoring or improving operational efficiency. For purposes of improving patient care and safety, workflow analysis is very popular in clinical applications [2], e.g. patient management. Workflow analysis also provides an effective method for the research of medical errors which would cause nearly 100,000 death worldwide each year [12]. For these reasons, workflow analysis is also performed in Cath Lab to improve efficiency and safety in operating rooms (OR) by analyzing surgical processes and providing feedback or support.

In this thesis, we take a significant step forward in automating workflow analysis by obtaining and evaluating the observations by multi-camera footage and algorithms instead of human experts. The automatically recorded multi-camera footage can provide extensive and accurate data for workflow analysis. In the Cath Lab, workflow analysis can improve the efficiency of the procedure, and enhance the safety of patient and doctors during procedures. For example, the object detection algorithm enables automatic localization of lead shield, C-arm X-ray source, and medical personnel that can be used to estimate the personnel's exposure to radiation over time. Then the activities of medical personnel can be optimized to minimize the exposure to radiation through the monitoring and assessment of the workflow data.

1.1.3 Camera System

For the purposes of examination and protection, the medical equipment integrated into the Cath Lab is made movable. In this thesis, we mount five finely calibrated cameras to observe and analyze the procedures made in Cath Lab. As shown in Figure 1.1, the cameras are fixed on the corners and walls of Cath Lab from different viewpoints, including northwest corner, west wall, southwest corner, south wall and southeast corner. The cameras capture the locations of medical equipment, which is labeled colorfully in Figure 1.1, during procedures in Cath Lab by videos in 25 FPS speed. It is worth noting that there is object occlusion issue between the medical equipment in some viewpoints due to the high volume of equipment and personnel in Cath Lab. However, the occluded medical equipment can be separated clearly by our camera system that capture information from different viewpoints.

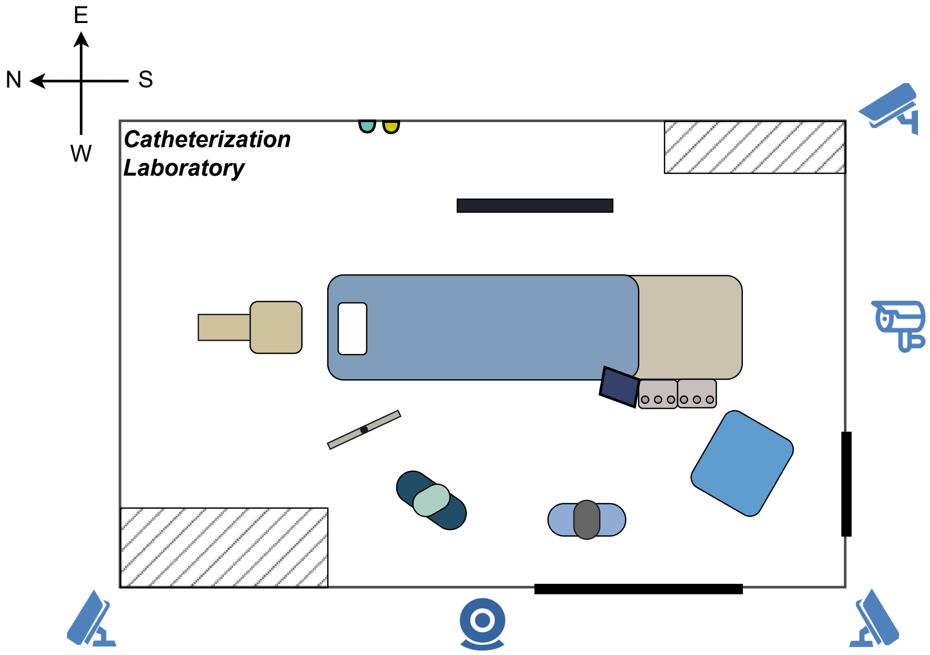


Figure 1.1: The diagram of the Cath Lab. The medical equipment and personnel are labeled colorfully and the cameras are surrounding the laboratory.

1.1.4 Problem Statement

In this work, three problems are studied to implement and improve the object detection algorithm on Cath Lab dataset:

- How to train and fine-tune Scaled-YOLOv4 based object detection model [13] for accurately and rapidly detecting medical equipment in Cath Lab, i.e. predicting the coordinates of medical objects and corresponding labels with large overlap area and real-time speed (25 FPS).
- How to improve the object detection network with e.g. self-attention mechanism and losses, for better precision.
- How to use multiple perspectives fixed in Cath Lab to improve the prediction of bounding boxes.

1.2 Pipeline

As mentioned in the problem statement, the objective of this thesis is to implement an object detector in Cath Lab to localize the medical equipment. With five cameras fixed in Cath Lab and surrounding the medical equipment, the correspondence of the target object is studied in this project and enables the object detector to improve the accuracy of prediction. Moreover, this project is a part of a larg project, which not only investigates the position of medical equipment but also the human pose estimation and camera calibration. Section 1.2.1 briefly introduces the object detector based on multiple cameras, followed by an overview of the large pipeline in section 1.2.2.

1.2.1 Object Detection with Multiple Cameras

The object detection algorithm consists of two modules. The first one is a fine-tuned object detection model based on the Scaled-YOLOv4 model. Due to the rapid development of computer vision and deep learning, there are lots of mature and excellent object detection models. We apply the state-of-the-art model Scaled-YOLOv4 since it has extremely fast processing speed and decent precision. We use the CSPDarknet with 31 layers (Scaled-YOLOv4-csp) as backbone with CSP-ized PAN to aggregate parameters from different depths. However, we find Scaled-YOLOv4 still has difficulties detecting some objects in Cath Lab. For instance, it performs unsatisfactorily when detecting the operating table occluded by the patient lying, and distinguishing the patient from doctors. In the first case, with occluded objects replacing part of the features of objects with unwanted information, the lack of key features will cause the objects not to be recognized or partly recognized. There are frequent object occlusion issues in Cath Lab due to the limited room space and massive equipment insides. In addition, the anchors with fixed scales sometimes also produce errors. When a scale of bounding box appears in the test set but missing or has few samples in the training set, the model may not accurately find the boundary of these boxes. For example, the object detector learns the vertical shape of patient due to numerous samples of patient walking (vertical boxes). However, it can only recognize part of patient, e.g. head, when the patient is lying in operating table (horizontal boxes) .

To solve these problems, we design a keypoints-matching-based filter (KM-filter) that can improve the prediction of boxes by combining information and predicted bounding boxes between different cameras. Since the Scaled-YOLOv4 performs reliably most times, the function of this filter is to fix incorrect predictions and complement inaccurate bounding boxes. The filter detects keypoints using SuperPoint [14] in the predicted bounding boxes and performs matching by SuperGlue [15] between viewpoints so that some missing points can be included in bounding boxes. However, this method also suffers from incorrect keypoints when the target object is transparent or other objects appear in the same bounding boxes. To ensure the correct keypoints are detected, the keypoints are processed to find a cluster that only locates on the target object and matched with keypoints in images of neighboring viewpoints, which have been refined and are believed to be reliable.

1.2.2 The Whole Pipeline

Our object detection algorithm is a subsystem of larger pipeline. The project is aiming to automate workflow analysis in Cath Lab at RdGG. Besides detecting medical equipment, the larger pipeline also automatically calibrates cameras in Cath Lab and estimates human poses for cardiologists and lab assistants, as shown in Figure 1.2.

The object detection algorithm developed in this thesis also support the work of automatic camera calibration and human pose estimation. The camera calibration subsystem requires the detection fixed objects, e.g. doors and windows, then uses

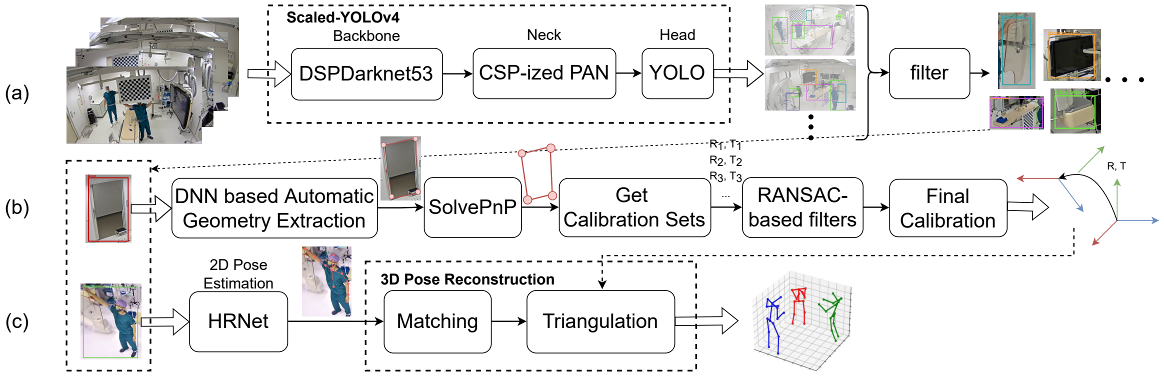


Figure 1.2: The whole pipeline of large project [1]. The pipeline consists of three parts: object detection to detect medical equipment, personnel and fixed objects; camera calibration to automatically calibrate the cameras; person tracking to estimate the 3D human poses and track personnel.

keypoints extraction and matching technique to find the matched keypoints. Due to single geometry shape of door and windows, their vertices can be found in the matched keypoints. Then the previously measured coordinates are assigned to the vertexes and calculate the calibration set. The calibration information of cameras is filtered by RANSAC algorithm and finally outputted. The pose estimation part also uses 2D bounding boxes of personnel detected by object detection algorithm to estimate 2D poses by HR-Net [16], then use keypoints matching techniques to reconstruct 3D poses. The camera calibration information obtained before is used in triangulation method to finally build 3D poses for personnel.

1.3 Outline

The rest of the thesis is organized as follow:

1. In Chapter 1, the basic introduction to the background, motivation and studied problem is presented.
2. In Chapter 2, some related works about workflow analysis in clinical application, object detection, keypoints detection and matching, multiple camera systems are discussed.
3. In Chapter 3, the fine-tuned Scaled-YOLOv4 with improved loss functions and attention layer is discussed. A well-designed keypoint-matching-based filter that compares matches of detected object to adjust bounding boxes are explained.
4. In Chapter 4, the Cath Lab datasets and other experiment setups are illustrated, and the evaluation methods and the results of methods explained before are shown and discussed.
5. In Chapter 5, the work of this thesis project is concluded.

Related Work 2

A state-of-the-art object detection model is selected from lots of deep learning (DL) methods to realize the medical equipment detection task. In addition, the KM-filter designed to improve the performance of bounding box prediction is based on feature detection, description, and matching. In this chapter, related research on clinical workflow analysis, object detection and feature extraction, and matching are discussed. In Section 2.1, the researches of workflow analysis in clinical application are reviewed. Popular object detection models are shown in Section 2.2. Section 2.3 presents the state-of-the-art feature extraction and detection models.

2.1 Clinical Workflow Analysis

With the applications to protecting personnel and improving efficiency, workflow analysis always plays a significant role in clinical medical research. Researchers in surgical workflow analysis observe and record the activities occurring in operating rooms. For example, a research aiming to improve the efficiency of patient management record the movement of nurses in the trauma unit and their healthcare operations to patients by activity lists [2]. These movements and operations are then visualized by graphs, as shown in Figure 2.1. In addition, the network graph and radar graphs that visualize the probability of location of medical equipment is also used to describe the workflow [3]. However, with application of computer science, there is an growing trend to automatically obtain and process data for workflow analysis research.

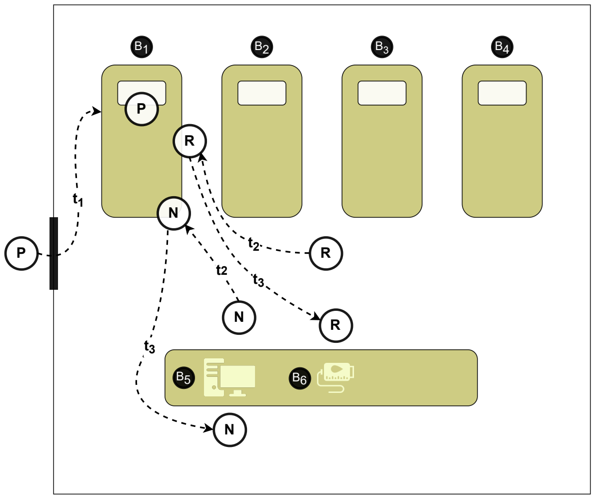


Figure 2.1: Workflow analysis for patient management at trauma unit [2]. Prepresents the patient while N and R represent nurse and resident nurse.

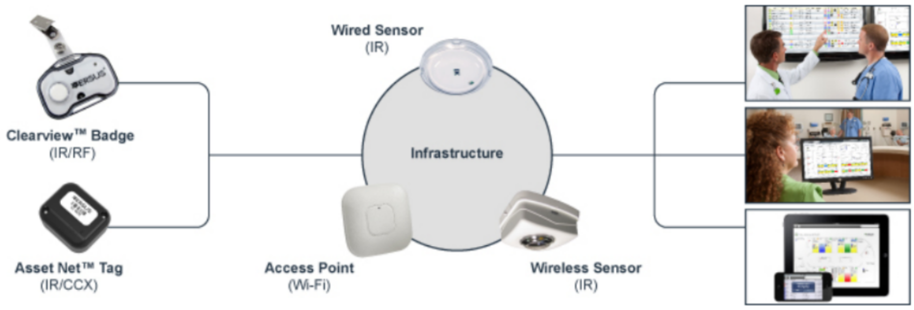


Figure 2.2: The pipeline of RFID-RTLS system [3].

Nowadays, procedures always take a long time and involve lots of large-scale medical equipment, making data collection costly by manual observation and tracking [17]. there is an growing trend to automatically obtain and process data by sensors and computers for workflow analysis research. For instance, the workflow steps can be represented by the electronic health record that contains large amount of data [17]. In addition, some researchers developed a system using portable radio-frequency identification (RFID) tags that measure the location of medical personnel and communicate data with computer, called Versus RFID-RTLS system [3], as shown in Figure 2.2

2.2 Object Detection Algorithm

Due to its wide application in the real world, object detection has gained great attention in the last decade. More and more object detection algorithms were proposed to improve the performance and speed of inference. The conventional methods in this field use handcrafted features to represent images [18], e.g. object detection system [19] based on keypoints extracted by SIFT [20] and clustered by mean shift clustering [21]. Benefit from deep learning, today's object detection algorithms use a convolutional neural network (CNN) to predict the position of objects, which is represented as rectangular bounding boxes in output. The bounding boxes are then classified to the corresponding labels. According to the combination or separation of these two stages, popular object detectors can be divided into two kinds, one-stage object detectors, and two-stage object detectors. Due to more intuitive structure, two-stage object detectors are first discussed in section 2.2.1, followed by one stage object detectors in section 2.2.2.

2.2.1 Two-stage Object Detectors

Two-stage object detectors divide the detection of objects into two stages. In the first stage, a CNN is used to obtain the candidate object locations, which are also known as region proposals. Then these region proposals are refined by another CNN and classified for the labels, just as people focus on a region of interest after a coarse scan [22]. In this section, we discuss the R-CNN series object detectors [23, 24, 25, 26, 27], which are the most representative two-stage object detectors.

The initial R-CNN [23] algorithm applies a selective search on the input images to extract 2000 region proposals which are classified by a CNN-based support vector machine (SVM) [28, 29]. However, by randomly proposing possible region proposals, the selective search method produces too many region proposals, which dramatically slow down the detection speed. To prevent the CNN wasting time processing the redundant information, spatial pyramid pooling networks (SPPnet) [24] speed up R-CNN by applying CNN to an entire input image and select the feature within the region proposals. As a result, the feature maps in different region proposals share the computation in CNN and then are converted to a fixed-length feature vector by max pooling. Inspired by SPPnet, Fast R-CNN [25] also make the computation of feature maps shared between region proposals and obtains fixed-length feature vectors by region of interest pooling layer (RoI layer). Then it employs softmax probabilities for label prediction and bounding box regression, which yeilds a more precise location than the previously mentioned SVM classifier. Faster R-CNN [26] further improves the performance and speed by adding Region Proposal Network (RPN) before RoI layer. RPN applies a sliding window to predict region proposals with different scales and combines them into a fixed-length vector. After the success of Faster R-CNN, there were more algorithms extending the R-CNN series, e.g. mask R-CNN [27], which not only predicts bounding boxes, but also object masks.

2.2.2 One-stage Object Detectors

By comparing with two-stage object detectors, one-stage detectors obtain outputs only by one single neural network. Nowadays, most of the state-of-the-art one-stage object detectors are based on You Only Look Once (YOLO) object detectors [22]. YOLO shows a extremely fast speed and decent precision through combining the two stages of class prediction and bounding box regression into a single regression problem [4]. It encodes the contextual information about labels into the positional information so it can regress not only the coordinates of bounding boxes, but also the class probabilities. YOLO divides the input images into $S \cdot S$ grid cells, which predict bounding boxes of objects whose centers are located in this cell, as shown in Figure 2.3. For each grid cell, YOLO predicts B bounding boxes with corresponding confidence scores that simultaneously consider the the bounding box coordinates and the label classification. With represented by the offset to the responsible grid cell, the predicted coordinates are optimized by intersection over the union (IoU), which is defined as

$$IoU_{pred}^{gt} = \frac{bbox^{pred} \bigcap bbox^{gt}}{bbox^{pred} \bigcup bbox^{gt}}, \tag{2.1}$$

where $bbox^{(Pred,GT)}$ represents the bounding boxes in prediction and ground truth. The IoU can be understood as the proportion of overlap area of the predicted box and ground truth box. To predict the coordinates only when an object appears in the grid cell, the confidence score is defined as follows:

$$C = Pr(Object) \cdot IOU_{pred}^{gt}, \tag{2.2}$$

where C is the confidence score and $Pr(\text{Object})$ is the probability of appearance of any object, which is represented by the objectness score. Then the conditional class probabilities $Pr(\text{Class}_i \mid \text{Object})$ to class i is introduced to combine coordinate information and label probabilities. Then the class-specific confidence score is shown as follows:

$$\begin{aligned} C_{class} &= Pr(Class_i \mid Object) \cdot Pr(Object) \cdot IOU^{gt}_{pred} \\ &= Pr(\text{Class}_i) \cdot IOU^{gt}_{pred}. \end{aligned} \tag{2.3}$$

With confidence scores encoding both class probabilities and the fitting performance of predicted bounding boxes, YOLO combines the prediction of bounding boxes and classification of the corresponding labels into a regression of class-specific confidence scores. As a consequence, the inference made by YOLO is faster (155 FPS) than most two-stage object detectors. However, the fixed grid cells and combined coordinate information in confidence score also contribute to worse detection precision (52.7% mAP on Pascal VOC 2007 dataset [30]) than most two-stage detectors (73.2% mAP achieved by Faster R-CNN [26]).

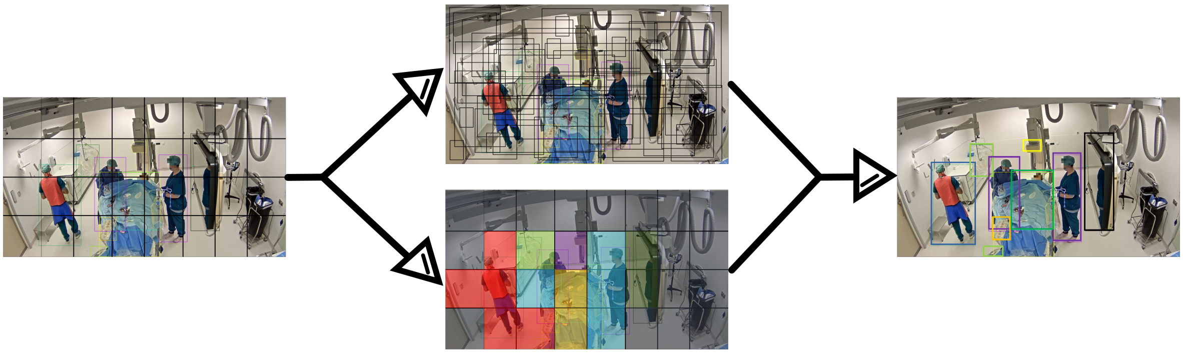


Figure 2.3: You Only Look Once model [4]. YOLO system first divides the images into grids and predicts finite bounding boxes and corresponding confidences score and class probabilities for each grid cell. By combining these probabilities, object detection is consequently a regression problem.

Based on the idea of YOLO, more regression-based one-stage object detectors are developed, e.g. Single Shot MultiBox Detector (SSD) [31], Fully Convolutional one-stage object detector (FCOS) [32] and subsequent YOLO versions [4, 33, 34, 35, 13, 36, 37]. Inspired by RPN in Faster R-CNN [26], SSD [31] uses anchor boxes with different scales in the grids cells formed in YOLO to improve the detection for objects with different scales. The FCOS [32], on the other head, get rid of the limitation of anchor boxes by the regression of distances of each point inside target boxes to the boundaries of boxes. Instead of objectness score, FCOS uses center-ness score to focus on the points closer to the center point of target box to prevent too many prediction are made for each target box. Although SSD (74.3 % mAP on VOC 2007 [30]) and FCOS (41.5% mAP on MS COCO [38]) outperform the performance of YOLOv1, they are defeated by the subsequent versions in YOLO series that dramatically improve the precision and speed.

YOLO9000 [33] is the improved version of YOLO. To solve the problem that YOLO is trained on the dataset with limited object labels, YOLO9000 is jointly trained on the

MS COCO detection dataset [38] and the ImageNet classification dataset [39] with more than 9000 classes and achieves decent performance (76.8% mAP on VOC 2007 [30] and 21.6 mAP on COCO [38]) and real-time speed (67 FPS). In addition, YOLO9000 uses some techniques to improve precision and speed on the basis of YOLO, including batch normalization [40] and anchor boxes inspired by RPN. It is worth noting that the anchor boxes selected by k-means clustering of the bounding box shapes significantly improve the precision, making them popular in many later algorithms in YOLO series. Based on YOLO9000, YOLOv3 [34] integrates more techniques to improve the performance (28.2 % mAP on COCO [38], with speed 45 FPS), e.g. anchor boxes collected by clustering and multilabel classification.

By dividing the whole network into backbone, neck, and head in terms of operations and functions, more attempts at network architecture and techniques are made to improve the performance of a network. Backbones are usually complex CNN that extract feature maps from input images. Based on feature maps obtained from backbones, necks process the feature maps to refine contextual information and restructure data, and finally, the head performs the inference of object detection. YOLOv4 [35] discussed the combination of techniques and different parts in network architecture and found the solution reaching the optimal performance (43.5 % mAP on COCO [38]) and speed (65 FPS). Scaled-YOLOv4 [13] utilizes the model scaling approach to modify the depth, width, resolution, and structure of the network so that YOLO can make better prediction (51.4 % mAP on COCO [38]) and also faster (41 FPS).

There are more object detectors in YOLO series with better accuracy and speed, e.g. YOLOX [36], YOLOv5, YOLOv6 and YOLOv7 [37]. Like FCOS, YOLOX is also an anchor-free object detector so it consequently performs better than other YOLO detectors on datasets of objects with various shapes. YOLOv7 outperforms almost all object detectors by applying some bag of freebies techniques and proposing to extend and compound scaling methods.

2.3 Feature Extraction and Matching

In this thesis, we improve the prediction by comparing information from different images. To establish similarity between images, the state-of-the-art image matching method are based on feature detection and description, while conventional method use optimized or learnable transformation model to estimate the common area between two images [41].Since the feature-based methods generate local features represented as keypoints with coordinate information, we use a feature extraction model to detect keypoints, around which algorithms extract descriptors for further matching stage. Then the coordinate correspondence is built by matching the keypoints between the keypoints in two images. The methods detecting and describing keypoints are reviewed in section 2.3.1, then the methods to match these keypoints are discussed in section 2.3.2

2.3.1 Feature Detection and Description

There are many algorithms detecting keypoints and extracting feature descriptors. According to the process of detection and description, the feature extractors can be classified into detect-then-describe methods and detect-and-describe methods [42]. Most conventional methods are detect-then-describe methods, first detecting keypoints and then performing feature descriptions from a patch around each keypoints, e.g. SIFT [20] and ORB [43]. On the contrary, the detect-and-describe approaches simultaneously detect keypoints and perform feature description in one neural network. Some approaches, like SuperPoint [14] and R2D2 [44], share neural networks to learn feature detectors and dense descriptors, which are trained separately with different losses. D2-Net algorithm [45], on the contrary, shares all parameters of feature detector and descriptor which are optimized by a joint loss function simultaneously.

SuperPoint [14] approach is a self-supervised deep learning method using a detector model pre-trained on a synthetic dataset. The dataset composed of synthetic basic shapes with artificially marked keypoints used to extract the keypoints of simple polygons. With warping the original image, the descriptor predictor is optimized for minimum distance between descriptors at keypoints of original and wrapped images. This model is also trained based on the MS COCO dataset [38] to get a general detector that can extract keypoints from general images. The matching performance of SuperPoint is shown to be better than traditional methods like ORB and SIFT.

Reliable and Repeatable Detector and Descriptor (R2D2) [44] is another self-supervised approach trained with losses to learn salient regions containing frequently useless points for keypoint detection and generate descriptors at the detected keypoints. With salient regions reducing the efficiency of the algorithm on detection and description, therefore, R2D2 algorithm builds a network to learn descriptors in regions that are more likely to perform matching. As a predictor of the degree of distinctiveness, the R2D2 approach trains reliable keypoints detector and descriptor jointly on image points without ambiguous regions. The R2D2 approach is found to be better than the SuperPoint method and former traditional methods but needs more hyperparameters so detecting keypoints slower.

D2-Net [45] trains a neural network which is both a dense feature descriptor and a feature detector. Similar to SuperPoint, the D2-net method also trains CNN with a deep representation shared between the detector and the descriptor. However, with sharing all weights between detection and description, D2-net relies on a joint formulation to optimize them simultaneously, as opposed to the SuperPoint approach using different decoder branches that are trained separately. The descriptors of D2-Net, which is a dense set of 3D tensors, are obtained by collecting the tensor in vertical dimension after applying a neural network to the input image. As for detection, the 2D responses of the tensor are calculated first and then post-processed by hard feature detection and soft feature detection.

2.3.2 Feature Matching

Most image matching techniques are based on feature descriptors at keypoints detected by detection and description methods, like SIFT and D2-Net. Traditional feature matching, like Brute-Force [46] and Hashing methods [47], is done by nearest-neighbor search and filtering. For the nearest-neighbor search, the Euclidean distance of each feature descriptor is calculated to find the closest feature keypoint pairs with the corresponding shortest distance. Some evaluation methods like Random Sample Consensus (RANSAC) [48] can be performed based on the nearest-neighbor search to decide the matching result. Furthermore, the filtering technique can be implemented to refine the image data for faster matching speed. Besides, there are also some feature matching approaches based on neural networks which would reduce the cost and matching time. The feature matching models with neural networks always have a more complex structure and better matching accuracy.

Order-Aware Network (OA-Net) [49] builds a network to estimate the probabilities corresponding inliers between two images, and regress the matrix of relative poses. OA-Net uses the local information of the neighborhood of the keypoints to reject the outliers for keypoints correspondences. By using both local and global features, OA-Net shows better performance than traditional approaches like the brute-force method. However, since it trains a network rejecting outliers based on the neighborhood of keypoints, OA-Net is tremendously dependent on the descriptors.

Adaptive Locally-Affine Matching (AdaLAM) [50] is a hierarchical pipeline for an effective and efficient outlier filter that uses a sample-adaptive threshold to verify local affine motion. Since the outliers within descriptors can contribute to low efficiency and performance of image matching, AdaLAM increases the accuracy and execution speed by outlier filtering to reject contamination from outliers. AdaLAM first compares the descriptors for each keypoint to select the seed points and then applies RANSAC to verify the local affine consistency in the neighborhood of seed points. Thus, the task of AdaLAM is highly parallelizable, and it is designed to be accelerated by GPU implementation. AdaLAM performs more competitively than existing methods in both indoor and outdoor images.

SuperGlue [15] matches local features by jointly finding correspondences and rejecting non-matching points. It uses a graph neural network to predict cost for solving a differential optimal transport problem. Attention mechanism is used to aggregate contextual information is introduced to allow SuperGlue to jointly infer the underlying 3D scene and feature assignments. The priors over geometric transformations and regularities of the 3D world are learned by SuperGlue through end-to-end training from image pairs.

3 Methodology

As shown in Chapter 2, there are many kinds of object detector series with different versions. Though the latest object detectors are obviously faster and more powerful in object detection than old version algorithms in the same series, the performance of object detectors partly depends on the dataset. In the Cath Lab data, the object classes are mainly medical equipment and personnel. Although the appearance of the objects is very common to most normal object detection tasks, there are some difficulties with several labels and in some situations:

- Cardiologist, Lab Assistant, and Patient who wear similar colored clothes and consequently are difficult to distinguish;
- Objects occluded by other things, e.g. operating table obscured by the patient who is also covered by operating blue sheet;
- Medical equipment in similar off-white color;
- Lead Shield that is transparent;
- Radiation Lights that is very small.

The object detector used should be able to detect the objects listed above. In addition, the algorithm should be able to work in real-time, i.e., faster than 25 FPS. In order to implement the object detection task with the requirements listed above, we tested different object detectors and finally select Scaled-YOLOv4 as the baseline of our object detection pipeline due to its stable precision and extremely fast speed. It is notable that the YOLOv7 is not used in this thesis because it was released towards the end of this project. However, the precision of Scaled-YOLOv4 still cannot meet the requirement for detecting occluded objects. To improve the performance of object detection, we designed a filter that uses feature matching to compare objects in different cameras and adjust predicted bounding boxes. The implementation and optimization of the Scaled-YOLOv4 model are explained in Section 3.1. In Section 3.2, the design of KM-filter to adjust bounding boxes is presented.

3.1 Scaled-YOLOv4

Scaled-YOLOv4 is an improved version of YOLO. The mechanism of YOLO has been explained in section 2.2.2. The data augmentation methods used by Scaled-YOLOv4 is presented in section 3.1.1. Section 3.1.2 and section 3.1.3 discuss the optimization of attention layers and losses for better performance respectively.

3.1.1 Data Augmentation

When training Scaled-YOLOv4, input images are processed by data augmentation techinques. Data augmentation can effectively solve the lack of unique features caused by fixed camera views and limited dataset size and variety. The basic data augmentation methods, which change the shape, scale and color of images, include random affine, HSV random augmentation and random flip. Images are randomly rotated, translated and scaled through random affine, while HSV augmentation stochastically modifies the color of images by hue (H), saturation (S) and value (V). The random flip operation is done on vertical or horizontal axes.

While all methods mentioned above augment data on image level, cut-and-mix methods change the context of image and augment data in pixel level such that the objects outside their normal context can be detected [35]. Some examples of data augmented by cut-and-mix methods are shown in fig. 3.1. MixUp [51] is used to mix two images together with a certain probability, while CutOut [52] randomly selects and discards an area in images. By integrating two methods, CutMix [53] mixes images with a certain probability, where pixels in an area of input image are discarded and filled with another image. Mosaic [35], borrowing the idea of CutMix enhancement, applies four images with random cropping, scaling, rotation and other operations, then combine them into one image. Mosaic method can increase simultaneously expand the size of whole dataset and increase the number of small samples by transforming large objects into small samples in a image. Since more samples are concentrated in single image, more features are processed by the network in a batch, leading to smaller batch size required in network. In addition, cut-and-mix methods, as a simulation to the object occlusion help the trained model perform better in detecting occluded objects.

3.1.2 Vision Transformer Layers

The attention mechanism was proposed to simulate the process of human selectively focusing on interesting parts in an image [54]. The self-attention (SA), as the most widely used attention mechanism, relates different words (image patches) in a single sequence (image) to compute the representation of this sequence [55]. To improve the performance of computer vision algorithms, transformers are designed to combine self-attention mechanism with convolution or recurrence network [56]. In this project, we also use a transformer layer to improve the ability of our network to focus on regions where objects are presented.

The basic attention mechanism is achieved by adding weights to the correlation between different words (image patches). To find the attention of a word $\boldsymbol{z}$ to another word $\boldsymbol{z}^*$, each word is projected into three matrices: query $\boldsymbol{q}$, key $\boldsymbol{k}$ and value $\boldsymbol{v}$, which are given as:

$$[\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}] = \boldsymbol{z}\boldsymbol{U}_{qkv}, \tag{3.1}$$

where $\boldsymbol{U}_{qkv}$ are the learnable projection matrices. It is worth noting that word $\boldsymbol{z}^*$ doesn't necessarily pay much attention (large weight) to $\boldsymbol{z}$, even if $\boldsymbol{z}$ pay much attention to $\boldsymbol{z}$. The attention from word $\boldsymbol{z}$ to $\boldsymbol{z}^*$ can be represented by the dot-product between

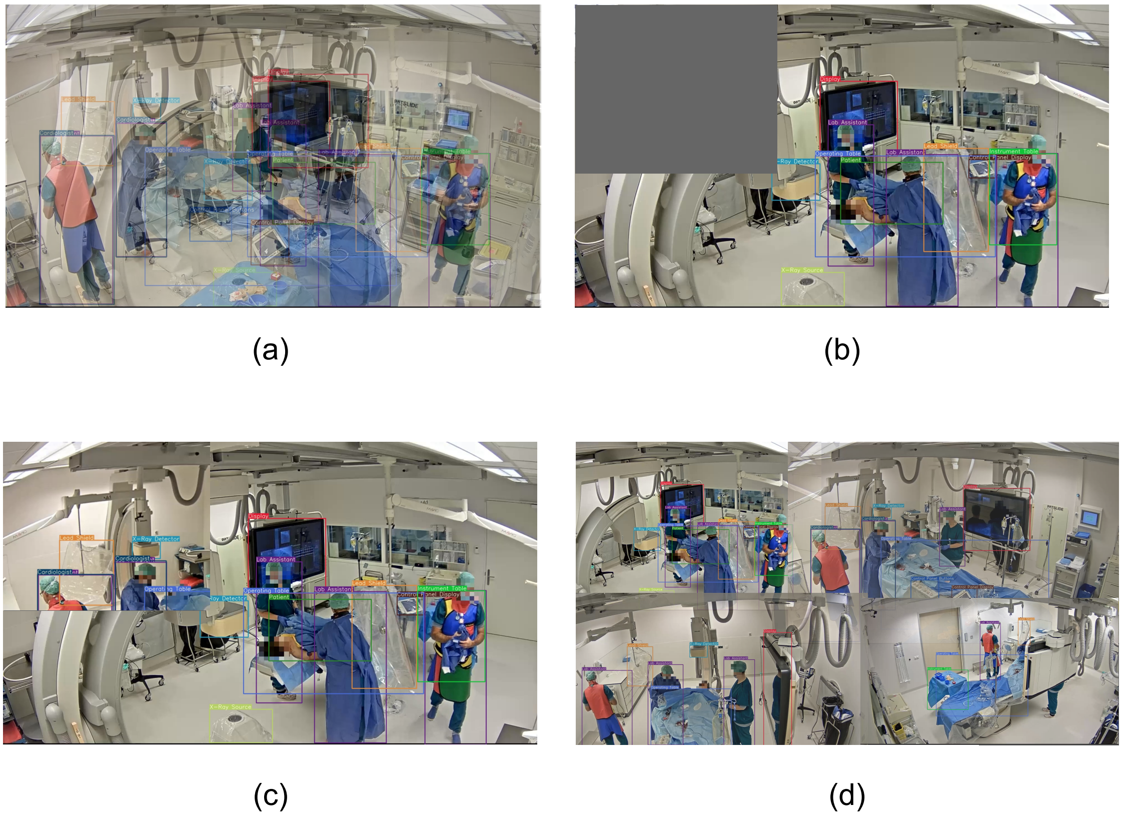


Figure 3.1: Examples of data augmented by four popular cut-and-mix methods: (a) MixUp; (b) CutOut; (c) CutMix; (d) Mosaic

the $\boldsymbol{q}$ of $\boldsymbol{z}$ and the $\boldsymbol{k}^*$ of $\boldsymbol{z}^*$. Then the self-attention function can be defined as:

$$\mathrm{SA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = \mathrm{softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d_k}})\boldsymbol{v}. \tag{3.2}$$

While single self-attention operation can be called SA head, multi-head self-attention (MSA), as an extension of standard SA, is implemented through h parallel self-attention operations. Each SA head uses different queries $\boldsymbol{q}$ by applying different projection matrix $\boldsymbol{U}$. Then the output of MSA is the projection of concatenated SA head output:

$$\mathrm{MSA}(\boldsymbol{Z}) = [SA_1(\boldsymbol{z}); SA_2(\boldsymbol{z}); \ldots; SA_h(\boldsymbol{z})]\boldsymbol{U}_{msa}, \tag{3.3}$$

where $\boldsymbol{U}_{msa}$ is the projection matrix.

An overview of the architecture of the transformer layer is shown in fig. 3.2. Since the transformer layer is placed at the end of backbone network network, as shown in fig. 3.3, it applies the self-attention mechanism on input feature maps extracted by other convolution layers in backbone network instead of entire images. It first divides the input feature map $\boldsymbol{f}$ of size $(H \times W \times C)$ into N flattened 2D patches $\boldsymbol{f}_p$ of size $(P \times P \times C$, where $(H \times W)$ is the resolution of feature map [57], P is the length of patches and C is the number of channels. Then the feature map patches are embedded

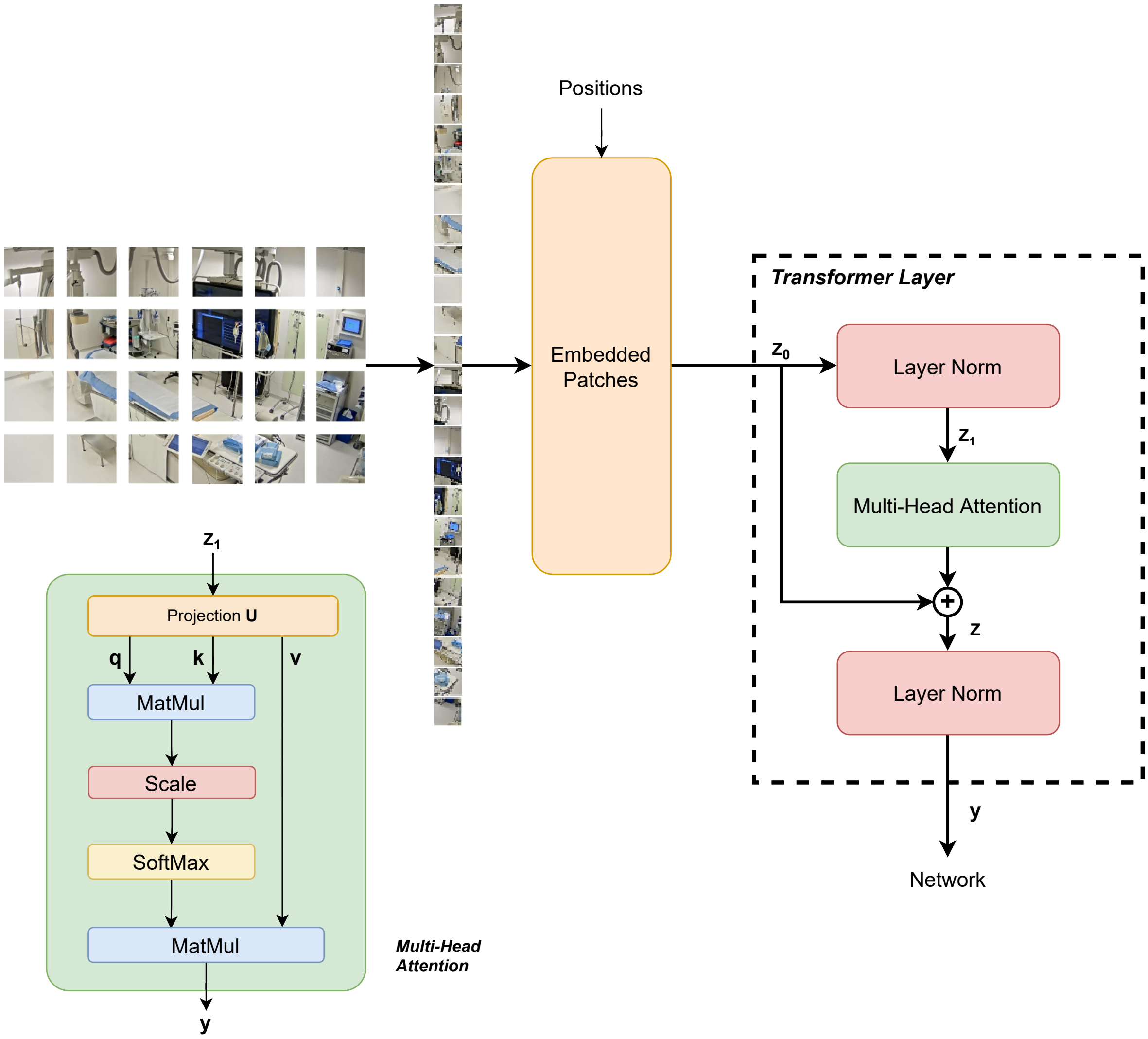


Figure 3.2: Overview of transformer layer.

with positional information using a learnable embedding:

$$\boldsymbol{z}_0 = [\boldsymbol{f}_p^1\boldsymbol{E}; \boldsymbol{f}_p^2\boldsymbol{E}; \ldots; \boldsymbol{f}_p^N\boldsymbol{E}] + \boldsymbol{E}_{pos}, \tag{3.4}$$

where $\boldsymbol{E}$ and $\boldsymbol{E}_{pos}$ are embedding matrices which are learned during training. As mentioned before, the embedded patches are then projected into queries, keys and values, and processed by MSA block. Before the weighted values are outputted, we also employ residual connection [58] and layer normalization [59]:

$$\boldsymbol{z} = \mathrm{MSA}(\mathrm{LN}(\boldsymbol{z}_0)) + \boldsymbol{z}_0, \tag{3.5}$$

$$\boldsymbol{y} = \mathrm{LN}(\boldsymbol{z}), \tag{3.6}$$

where MSA is the multi-head self-attention layer and LN is the layer normalization. The transformer layer is put in the end of backbone network to enhance the regions

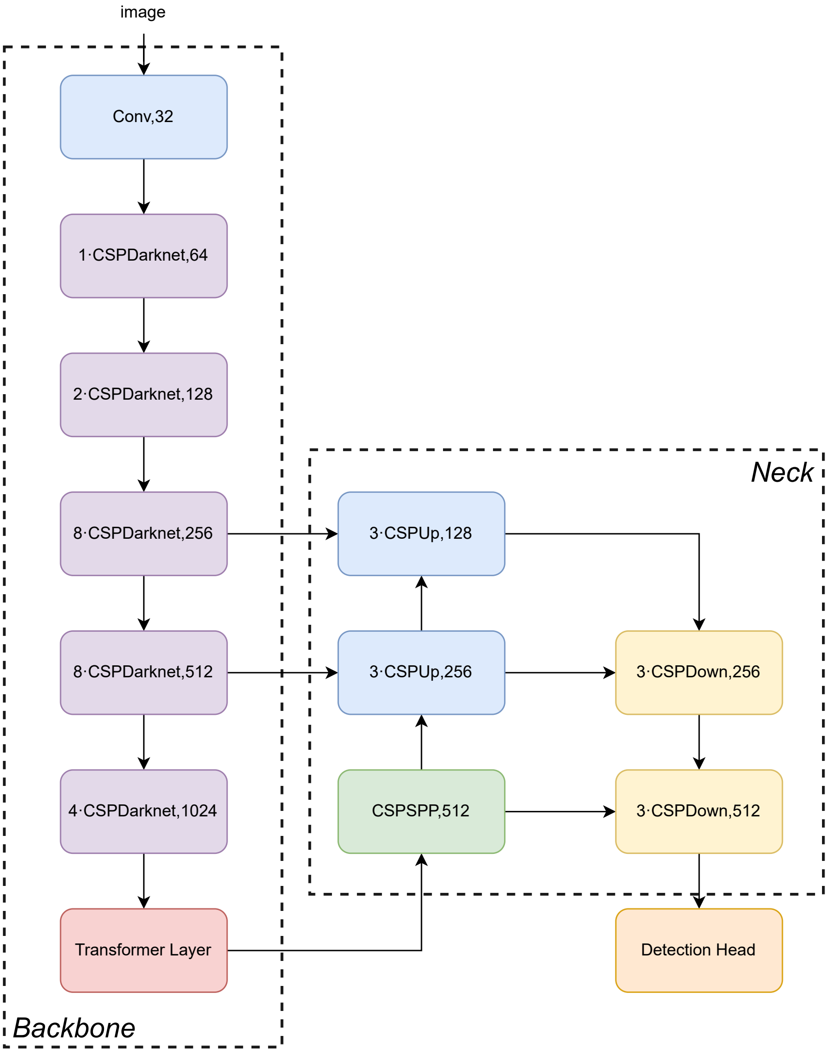


Figure 3.3: The network achitecture of Scaled-YOLOv4-csp network with transformer layer at the end of backbone network.

with objects and generate more distinguishable features for later SPP network and detection head.

3.1.3 Losses

The loss in Scaled-YOLOv4 consists of three parts: classification loss, regression loss, and objectness loss. Each loss learns different information for prediction. The classification loss and regression loss provide the objective functions for optimizing the predictions of bounding box coordinates and labels respectively. The prediction of objectness, providing the probability that an object appears, is learned by objectness loss. Classification loss, regression loss, and objectness loss jointly contribute to predicting the confidence scores of proposed bounding boxes, as shown in eq. (2.3).

Since the estimation of objectness is actually a binary classification problem, both classification loss and objectness loss can use a well-researched loss function in classification tasks. In contrast, the loss for bounding box regression attracts more attention from researchers and gets updates for better convergence speed and performance, due to the direct influence on the precision of bounding box prediction. In section 3.1.3.1, the Focal loss used by classification and objectness losses is explained [4]. A brief discussion of IoU loss [60] and its variants including GIoU [61], DIoU and CIoU [62] losses are given in section 3.1.3.2, followed by state-of-the-art variants EIoU [63] loss in section 3.1.3.3 and SIoU loss [64] in section 3.1.3.4.

3.1.3.1 Focal loss

As the most basic loss function in classification task, the cross-entropy loss always suffers from the class imbalance problem that makes classifiers perform poorly on classes with few samples. When some classes have too many samples compared to other classes, they would dominate the components of cross-entropy loss and lead to the omission of categories with small samples. A solution to the class imbalance is a balanced cross-entropy loss function, which balances the distribution of loss by increasing the weights for classes with few samples. The weights in balanced cross-entropy loss are thereby determined by the distribution of classes. However, to address the same problem, Focal loss [65] focus on determining weights in terms of the classification results. In other words, Focal loss keeps the losses for classes classified as bad but decrease the impact of well-classified labels by modulating factor, as shown below:

$$L_{Focal} = -(1 - p_t)^{\gamma} log(p_t), \tag{3.7}$$

where p_t is the estimated probability for a class, $(1 - p_t)^{\gamma}$ is the modulating factor and γ is the tunable focusing parameter that is always set to 2. When a class is classified accurately, i.e. $p_t \to 1$, the modulating factor is approaching zero and the samples of this label have no impact on the loss function. Therefore, the modulating factors are reduced according to the correctness of the classifier prediction and keep the scale of loss if the classifier cannot make the correct prediction. As aforementioned, both objectness loss and classification loss use Focal loss to in the binary and multi-category classification tasks.

3.1.3.2 IoU, GIoU, DIoU and CIoU losses

To optimize bounding box prediction, the YOLOv1 regresses the center coordinates and side lengths of the predicted box by Mean Square Error (MSE) or $ln - norm$ as the objective function. For anchor-based object detectors, IoU-based losses are used to regress the offsets of bounding boxes corresponding to anchors [60]. As explained in section 2.2.2, the IoU depicts the coverage of the predicted bounding box and ground truth box area, involving the prediction of coordinate points of bounding boxes. The simple IoU loss [60] is given by:

$$\mathrm{IoU} = -\ln \frac{\mathrm{bbox}^{\mathrm{pred}} \bigcap \mathrm{bbox}^{\mathrm{gt}}}{\mathrm{bbox}^{\mathrm{pred}} \bigcup \mathrm{bbox}^{\mathrm{gt}}}, \tag{3.8}$$

where $bbox^{(Pred,GT)}$ represents the bounding boxes in prediction and ground truth respectively. Prediction with larger IoU overlaps better with ground truth box. As a scale-invariant representation, IoU can solve the increasing loss caused by boxes with large scales when using ln-norm losses [35]. However, simple IoU loss only considers the overlap of boxes and cannot make accurate predictions of the coordinate points of boxes. To further include more geometric information into IoU losses, the penalty term $\mathcal{R}(\mathrm{bbox}^{\mathrm{pred}}, \mathrm{bbox}^{\mathrm{gt}})$ for predicted and ground truth boxes are introduced [62], which is given by:

$$L_{\mathrm{IoU}} = 1 - \mathrm{IoU} + \mathcal{R}(\mathrm{bbox}^{\mathrm{pred}}, \mathrm{bbox}^{\mathrm{gt}}). \tag{3.9}$$

Equation 3.9 shows the general form for IoU-based losses in the optimization of bounding box regression. Improved IoU losses are proposed by designing different penalty terms to considering more spatial factors, which are shown in Figure 3.4. For example, generalized intersection over union (GIoU) loss [61] outperforms simple IoU loss shown in eq. (3.8) by including the shape and orientation of the object in addition to the coverage area. The penalty used in GIoU loss function is given as:

$$\mathcal{R}_{\mathrm{GIoU}} = \frac{C - \mathrm{bbox}^{\mathrm{pred}} \bigcup \mathrm{bbox}^{\mathrm{gt}}}{C}, \tag{3.10}$$

where C is the smallest box covering two bounding boxes. As for Distance-IoU (DIoU) loss [62], it additionally considers the distance of the center points of two bounding boxes, with penalty form given by:

$$\mathcal{R}_{\mathrm{DIoU}} = \frac{\rho^2(p^{\mathrm{pred}}, p^{\mathrm{gt}})}{c^2}, \tag{3.11}$$

where $\rho(p^{\mathrm{pred}}, p^{\mathrm{gt}})$ is the Euclidean distance between the center points of the predicted box and ground truth box and c is the diagonal length of the smallest enclosing box. Based on DIoU loss, Complete IoU (CIoU) loss [62] further takes the aspect ratio into consideration. The consistency of aspect ratio is measured as

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}} - \frac{w^{\mathrm{pred}}}{h^{\mathrm{pred}}})^2, \tag{3.12}$$

where $(w, h)^{\mathrm{pred}}, p^{\mathrm{gt}}$ is the width and height of predicted box and ground truth box respectively. To focus on the predicted boxes with small overlap area, a trade-off weights is employed as a weight to aspect ratio consistency, which is given as

$$\alpha = \frac{v}{1 - \mathrm{IoU} + v}. \tag{3.13}$$

Finally, penalty term of CIoU is constructed by adding weighted consistency of aspect ratio to the penalty term of DIoU, given as:

$$\mathcal{R}_{\mathrm{CIoU}} = \mathcal{R}_{\mathrm{DIoU}} + \alpha v. \tag{3.14}$$

Since CIoU loss considers more spatial factors, it can achieve better convergence speed and accuracy on the bounding box regression problem than other variants mentioned above [35].

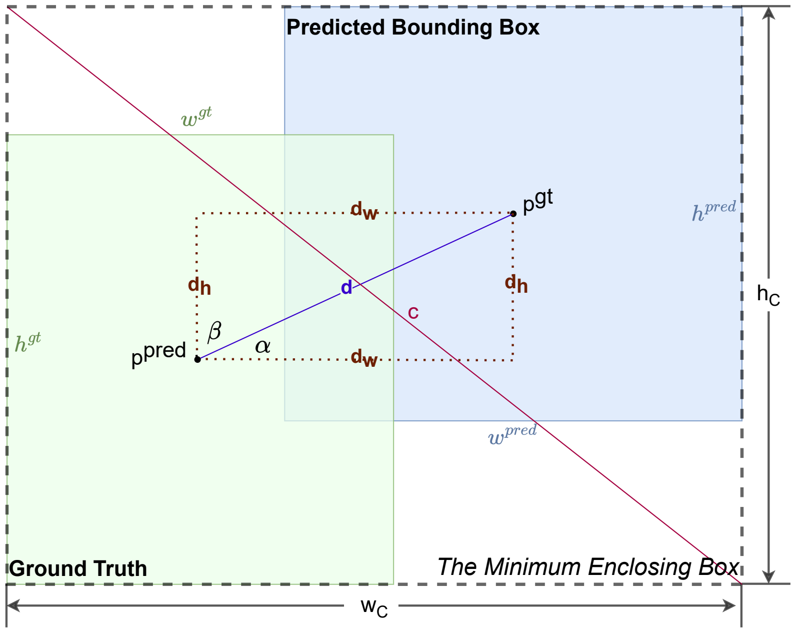


Figure 3.4: The scheme and parameters for calculating the loss functions.

3.1.3.3 EIoU-based losses

The DIoU and CIoU losses outperform the traditional ln-norm method and IoU method by considering more geometric information. In order to depict the object that appears in bounding boxes more efficiently, Efficient Intersection over Union (EIoU) loss [63] takes the side length information into account based on the DIoU loss. The penalty term considering side length information is given as:

$$\mathcal{R}_{\text{EIoU}} = \frac{\rho^2(p^{\text{pred}}, p^{\text{gt}})}{c^2} + \frac{\rho^2(w^{\text{pred}}, w^{\text{gt}})}{w_C^2} + \frac{\rho^2(h^{\text{pred}}, h^{\text{gt}})}{h_C^2}, \tag{3.15}$$

where w_C and h_C are the width and height of the smallest box enclosing the ground truth and predicted boxes. Though the aspect ratio information has already been implicitly included in side length information, we can still add the aspect ratio term to EIoU to further improve the depiction of objects. Similar to CIoU, the Efficient-Complete-IoU (ECIoU) is established as:

$$\mathcal{R}_{\text{ECIoU}} = \mathcal{R}_{\text{EIoU}} + \alpha v. \tag{3.16}$$

With the development of IoU losses, α-IoU loss is proposed as a new family of IoU-based losses. It is obtained by employing power transformations to existing IoU-based losses to improve the performance [66], which is calculated as:

$$L_{\alpha\text{-IoU}} = 1 - \text{IoU}^{\alpha} + \mathcal{R}^{\alpha}(\text{bbox}^{\text{pred}}, \text{bbox}^{\text{gt}}). \tag{3.17}$$

With α larger then one, the α-IoU loss improves the prediction accuracy by giving more priority to the predicted boxes with high IoU values. By applying this technique, the

EIoU is upgraded to α-IoU version by the power transformation to each term in $\mathcal{R}$:

$$\mathcal{R}^{\alpha}_{\alpha\text{EIoU}} = \frac{\rho^{2\alpha}(p^{\text{pred}}, p^{\text{gt}})}{c^{2\alpha}} + \frac{\rho^{2\alpha}(w^{\text{pred}}, w^{\text{gt}})}{w_C^{2\alpha}} + \frac{\rho^{2\alpha}(h^{\text{pred}}, h^{\text{gt}})}{h_C^{2\alpha}}. \tag{3.18}$$

3.1.3.4 SIoU Loss

In addition to central distance, overlap area, and shapes, the SCYLLA-IoU (SIoU) loss [64] also considers the direction of reducing the distance between center points of the predicted box and the target box. The penalty terms used in other IoU-based losses are redefined as costs that considering geometric information. The costs represent the gaps in different geometric dimension between the ground truth bounding box and predicted box. With the angle cost works as a phasor in distance cost, the penalty term consisting of distance cost and shape cost is given as:

$$\mathcal{R}_{\text{SIoU}} = \frac{\mathcal{C}_d + \mathcal{C}_s}{2}, \tag{3.19}$$

where $\mathcal{C}_{a,d,s}$ represent the costs for angle, distance and shape respectively. The angle cost measures the angle between the closest axe and the line connecting the predicted box and the ground truth, shown as α or β in fig. 3.4. It is worth noting that both α or β are positive acute angles, which are measured by the inverse sine function of distances, given as:

$$\alpha = \arcsin\left(\frac{d_h}{d}\right). \tag{3.20}$$

To find the closest axis, the angle α and its complementary angle (β) are compared and the smaller one is used. Assume that α used in later equations is the smaller angle. The predicted box will move to the horizontal axe if the sine value is positive and move to the vertical axe if negative. The cost for the angle is given by:

$$\mathcal{C}_a = 1 - 2 \cdot \sin^2\left(\alpha - \frac{\pi}{4}\right). \tag{3.21}$$

The consideration of angle can minimize the number of variables used to consider the distance between centers by reducing the loss of angle information first [64]. In other words, predicted bounding boxes in SIoU model first move to the closest horizontal or vertical line and then approach the target along the line. The distance cost guided by the angle cost is given by:

$$\mathcal{C}_d = \sum_{t=w,h} (1 - e^{-(2-\mathcal{C}_a)(d_t/t_C)^2}), \tag{3.22}$$

where $d_{w,h}$ are the distance projected onto x and y axes. It is shown by Equation 3.21 and 3.22 that the distance cost reaches the largest magnitude when the angle α equals to $\frac{\pi}{4}$. It is also worth noting that the distance cost would be reduced in magnitude and similar to penalty terms of distance in other losses when α is close to zero or $\frac{\pi}{2}$. The shape cost is defined as:

$$\mathcal{C}_s = \sum_{t=w,h} (1 - e^{-\omega_t})^{\theta} \tag{3.23}$$

$$\text{where } \omega_w = \frac{\mid w^{pred} - w^{\text{gt}} \mid}{max(w^{pred}, w^{\text{gt}})}, \ \omega_h = \frac{\mid h^{pred} - h^{\text{gt}} \mid}{max(h^{pred}, h^{\text{gt}})}. \tag{3.24}$$

The value of θ defines the weights of shape cost on SIoU loss and the impact of bounding box shapes on prediction. If θ is set to be 1, the shape of predicted boxes would be optimized rapidly but leave its location unchanged.

3.2 Keypoint-matching-based Filter

In Cath Lab, object detection is much more difficult than in an open environment because medical equipment and personnel are frequently occluded by each other. When an object is entirely occluded by other equipment, it is impossible for an object detector to determine its exact boundaries from one single view. At the same time, the same objects may be separated and clearly visible in other views. Moreover, Scaled-YOLOv4 may make the wrong prediction to the bounding boxes occasionally. To correct the errors that may occasionally appear in some frames and views, we consequently design the keypoint-matching-based filter (KM-filter) to adjust the boundaries of bounding boxes. The idea of the KM-filter is to use information captured in neighboring views to improve the performance of the object detector, which is always performed for single-view images.

The mechanism of KM-filter is presented in fig. 3.5. The KM-filter first extracts keypoints for images and then matches the keypoints across viewpoints to compare the feature. Since camera view shift separates keypoints from the different objects, the density-based spatial clustering of applications with noise (DBSCAN) technique [67] is used to group the matched keypoints. The target keypoints are used to enlarge the bounding boxes by calculating new boundaries that include all keypoints. In section 3.2.1, the keypoints extracted for the medical equipment in Cath Lab are analyzed. Section 3.2.2 discusses the keypoints matching across viewpoints. The clustering technique to filter the target keypoints is explained in section 3.2.3, followed by the complete pipeline of the KM-filter to improve detection results in section 3.2.4.

3.2.1 Keypoints Detection

The first step of the KM-filter is to extract keypoints for the image that is processed by the object detector at this moment. As mentioned in section 2.3.1, there are many non-DL or DL methods in this area, e.g. SIFT [20] and SuperPoint [14]. Since the accuracy of the improvement by the filter depends on the number of keypoints extracted for each object. It is expected to obtain more keypoints and better descriptors for medical equipment.As a result, we use SuperPoint — which is very sensitive to geometry keypoints — to detect keypoints efficiently. The set of keypoints detected can be represented as $\boldsymbol{P}^v$, where v is the view and can be one of viewpoints: NW (northwest corner), w (west wall), sw (sorthwest corner), s (south wall) and se (southeast corner).It is worth mentioning that, the number and performance of detected keypoints also depend on the resolution of input images, i.e. size of images. However, a larger input image size leads to more processing time so we use downsampled images as input

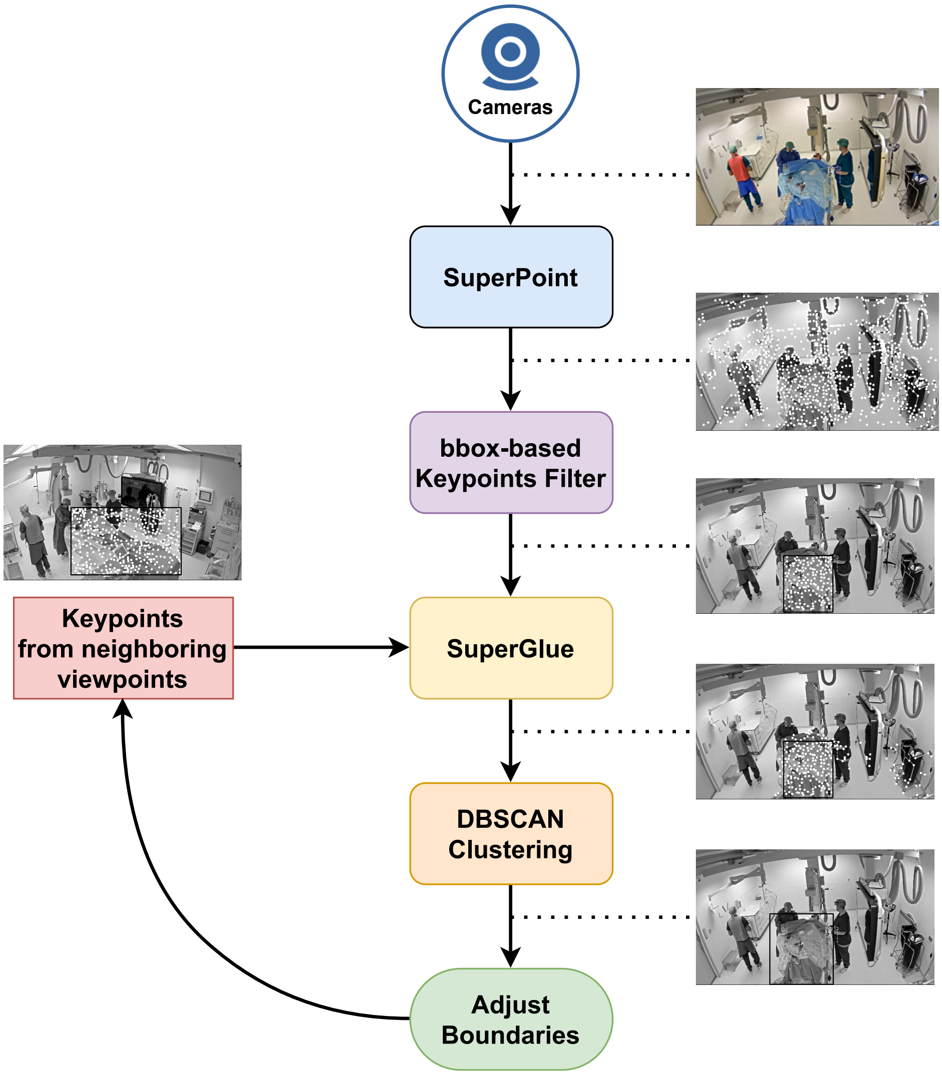

Cameras
SuperPoint
bbox-based Keypoints Filter
Keypoints from neighboring viewpoints
SuperGlue
DBSCAN Clustering
Adjust Boundaries

Figure 3.5: The mechanism of KM-filter to adjust boundaries by keypoint detection, matching and clustering.

to keypoints detector. Since the images in Scaled-YOLOv4 are down-sampled from 1920 × 1088 to 640 × 363 (maintain the aspect ratio), we also use this size in KM-filter.

3.2.2 Keypoints Matching

We use SuperGlue [15] to match the keypoints we extract in the previous stage since it is the fastest and also very accurate in matching keypoints by compared with other methods mentioned in section 2.3.2. As defined in most image matching problems, we call the input image (outputted by object detector and need to be improved) as query image and the matched images (from neighboring cameras) as gallery images. The target of matching keypoints is to build the correspondence between keypoints so that the label information can be transmitted from keypoints in the gallery images to keypoints in the query image. To match keypoints efficiently, only keypoints inside bounding boxes in the gallery image are matched with all keypoints in query image such that each bounding boxes lead to a match operation between two set of keypoints. Thus, the keypoints that fall into any predicted bounding boxes in gallery images are collected in a list labeled the object class, which is given as:

$$\boldsymbol{P}_b^{vn} = \{p^{vn} \mid p^{vn} \in bbox^{vn}b\}, \tag{3.25}$$

where $\boldsymbol{P}$ is a set of keypoints, p^{vn} represents the keypoints from the image taken by the neighboring views and $bbox$ is one of the bounding boxes predicted by the object detector (b is the box number). In contrast, all keypoints in the query image are used in matching to find the list of all possible corresponding keypoints of that object, given as

$$\boldsymbol{P}_b^{vn} \xrightarrow{SuperGlue} \boldsymbol{P}_b^{v}. \tag{3.26}$$

As each list represent an object, the lists are collected in a dictionary with keys as the object classes. By combining the lists of keypoints with predicted boxes in the query image, the new boundary for each box can be calculated. After the output of new bounding boxes, the query image becomes a gallery image and the list of keypoints will be used to match keypoints from next neighboring image, given as:

$$\boldsymbol{P}_b^{v} = \boldsymbol{P}_b^{v} \bigcup \{p^{v} \mid p^{v} \in bbox_b^{v}\} \tag{3.27}$$

Figure 3.6 shows the matching cases, providing evidence that boundaries can be determined by keypoints matching. Due to camera view shift, some features may be lost and new features may appear in different views. The clip angle between two neighboring cameras is about 45°, contributing to non-negligible changes in keypoints. Though still numerous matched keypoints left on target keypoints, the keypoints lost are mainly located at the edge of the side where the view rotates and resulting in unrecognized boundaries. For instance, the keypoints on the right edge of the operating table in the south view cannot be matched in the southwest view, and vice versa. However, KM-filter uses two neighboring views to obtain the information on the left and right sides of objects. In the previous case, the right edge can be found on the southeast camera, and the left edge can be matched on the southwest edge. As a result, almost all keypoints in the south view can be matched with keypoints in neighboring views. However, since there are only five cameras in Cath Lab, the northwest and southeast views only have one neighboring camera, resulting in less improvement than central cameras.

3.2.3 Clustering Filter

Since the bounding boxes of target objects are actually regular rectangles parallel to horizontal and vertical axes, some unrelated objects may fill the blank in the boxes. This would also happen when Scaled-YOLOv4 predicts boxes with errors. The previous parts of KM-filter may also generate keypoints for these unrelated objects and match them into another view. Because of the camera view shift, the matched keypoints of these unwanted objects may be located far away from the target object. For example, the black trash cans that appear in the box of the operating table in left image (southwest

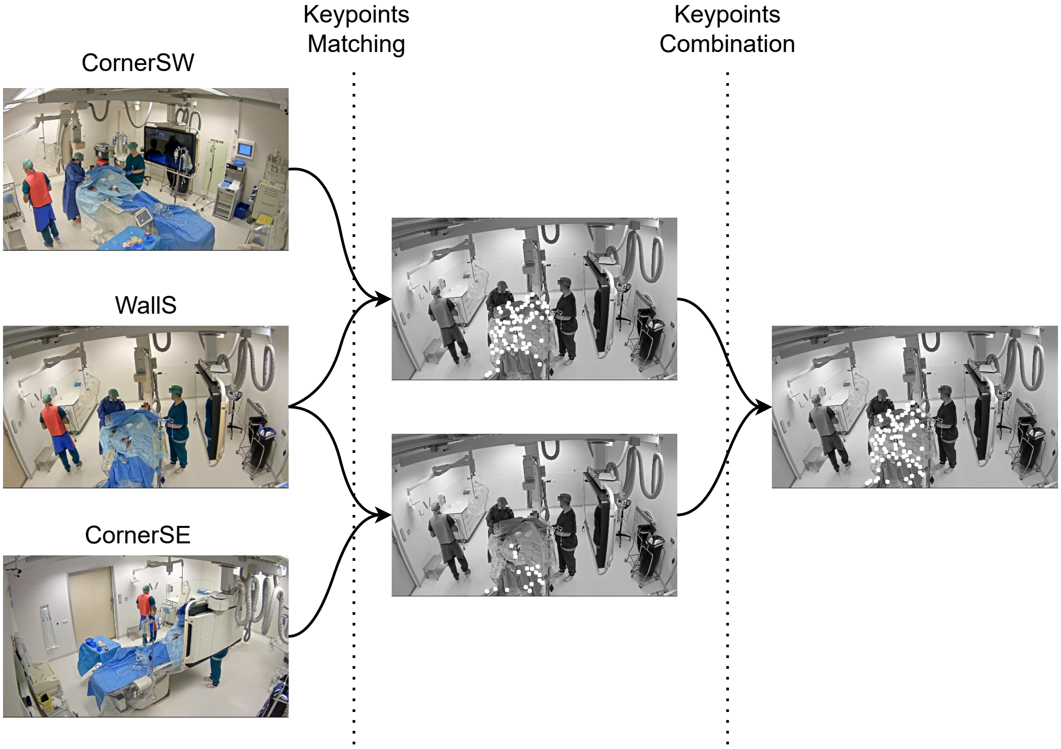


Figure 3.6: Keypoints detected and matched with neighboring viewpoints for operating table. The keypoints filtered by left neighboring camera are mainly located at the left side of object, vice versa. The keypionts located at all the target object can be obtained by combining matched keypoints filtered by both left and right neighboring cameras.

corner) of fig. 3.7 are far from the table in right image (south wall). To remove prevent the expansion of bounding boxes by mistakes, the DBSCAN clustering technique [67] is used in KM-filter to refine the matched keypoints. As a density-based clustering algorithm, DBSCAN can find all keypoints of an object by starting from a point located on the object and search for nearest points. Because the close objects in a box must move away from each other after switching camera views, the keypoints of different objects can be separated into different clusters. It is also worth noticing that the DBSCAN cluster may collect unwanted points when two objects are in physical contact. In this case, some keypoints from two objects are so close that two clusters are combined into a large cluster, which make the bounding boxes over-expand and produce errors.

The DBSCAN algorithm used to refine keypoints is designed as follow:

1. Compute the center point coordinates of matched keypoints in query image;

2. If there are multiple predicted boxes with the same label, compare the distances between their center points and center point of keypoints;

3. Select the closest box as the target box and its center point as start point of clustering

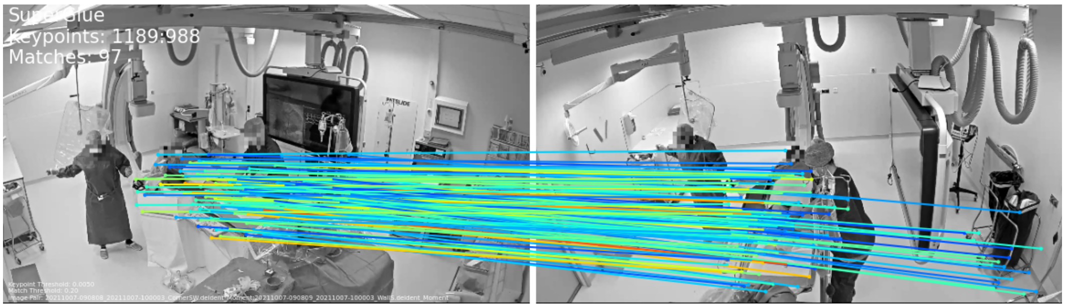

SuperGlue
Keypoints: 1189:988
Matches: 97
Keypoint Threshold: 0.0050
Match Threshold: 0.20
Image Pair: 20211007-090808_20211007-100003_CornerSW.deident_Moment:20211007-090809_20211007-100003_WallS.deident_Moment

Figure 3.7: The matched keypoints from different objects are seperated due to camera view shift and can be refined by the clustering algorithm.

4. For each newly added point in cluster (including start point), calculate their Euclidean distance from each unvisited point;
5. Compare the distance with threshold and add near points (distance smaller than threshold) to the cluster;
6. Repeat steps 4 and 5 until no point is close to any point in cluster (Euclidean distance between each unvisited point and each cluster point greater than threshold);
7. Output points in cluster.

3.2.4 Matching across viewpoints and frames

After obtaining the keypoints of an object, KM-filter can adjust the boundaries of bounding boxes. Since the bounding boxes are expressed by the top-left and bottom-right points of boxes, the KM-filter mix the vertexes with the keypoints and calculates the maximum and minimum values along two axes. The extreme values are then the new boundaries of boxes that can include all keypoints.

Figure 3.5 shows the images matched with images taken from neighboring cameras. Five predicted images with filtered keypoints are stored for matching with the new output image. After the new image is adjusted by the KM-filter, it replaces the stored image from the same perspective. Images coming from different cameras are interleaved to be detected by Scaled-YOLOv4 so that the stored images in KM-filter are updated in turn. To achieve this, image patches are built to contain 5 images from the same frame but from different perspectives. We cannot control whether the image is filtered by an image from the same batch (frame) or the last batch, because this depends on the order of images in batches. The image at the beginning of the batch updates the stored images first and has to be filtered by images from the last frame. However, this may also enable the transmission of keypoint information across frames.

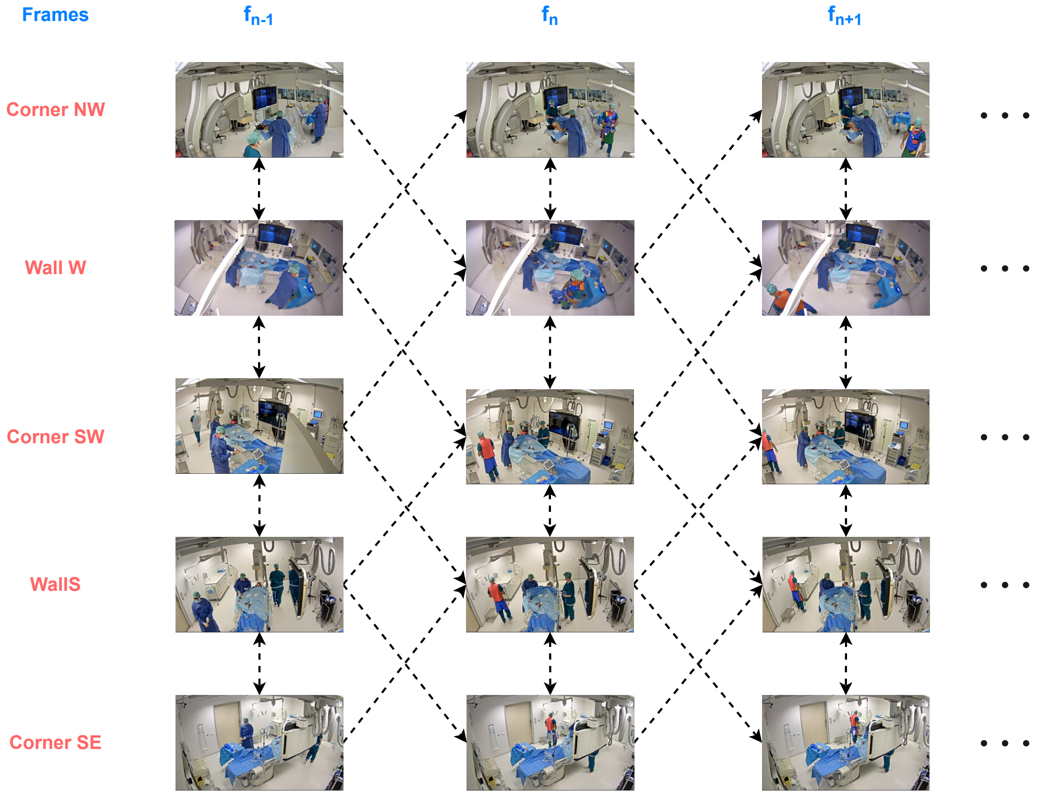


Figure 3.8: The images are matched with images from neighboring cameras at the same frame or last frame. This depends on the order of updates.

4 Experiments and Results

After explaining the theoretic design of our object detection pipeline, some experiments are carried out and discussed in this chapter to verify the feasibility of each module and design in the system. Before presenting the results, the dataset used for the experiments and evaluation matrices to assess the results are presented in Section 4.1. Section 4.2 shows results made by our pipeline and ablation study.

4.1 Dataset and Evaluation matrices

4.1.1 Datasets

The dataset used in this thesis is very specific to the Cath Lab operating room in RdGG hospital. There are 12 classes that appear in this dataset: Cardiologist, Lab Assistant, Patient, Operating Table, Instrument Table, Control Panel Display, Control Panel Buttons, X-ray Detector, X-ray Source,Lead Shield, Radiation Lights, Display (TV). The examples of these labels are given in fig. 4.1. Since the samples of these labels are collected in Cath Lab of RdGG, the same lable in other operating room may appear in different way and lead to slightly inferior performance. In addition, there are difficulties in detecting some specific labels due to the object occlusion and similar appearance to the background, e.g. patient in surgery and transparent lead shield, as mentioned in the beginning of Chapter 3. It is worth noting that, there are five cameras in Cath Lab, and the images taken by the same camera looks similar due to fixed positions and orientation. However, the shapes and characteristics of medical equipment in different cameras appear differently, e.g. Display and operating table. Some examples of images from Cath Lab dataset are shown in Figure 4.2.

The Cath Lab datasets consist of 7209 images, where a small 991-image dataset is obtained in empty operating room and the others are frames of videos of real surgeries. Since the 991-image dataset is initially designed for camera calibration task, the personnel in this dataset are only lab assistant, and the medical equipment inside is switched off and stationary. The objects in images taken in real procedures, in contrast, are more flexible and diverse in terms of poses, behavior and photographing light. Since each procedure lasts for more than half hour (almost 1 hour including the time for preparation and cleaning), each video with 25 FPS contains around 80,000 frames. Because there is only limited difference between consecutive frames, we sample the videos with frame step size of 125 (sample 1 image per 5 seconds) for variety of features and training efficiently. As a consequence, each video provides about 640 images. We use videos for two procedures, and each one is recorded as five videos (each camera generates one video). The training set consequently includes small 991-image dataset and images from videos taken in a procedure, while testing set consists of images taken in another procedure. The detailed components and division of the dataset are shown in

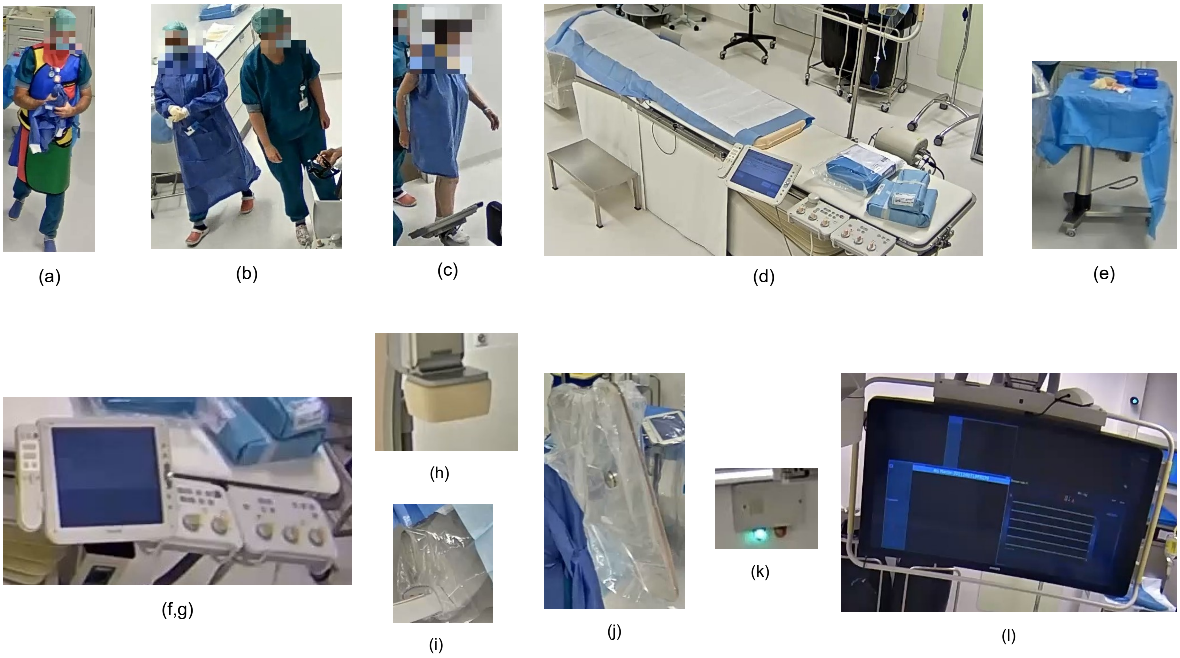


Figure 4.1: Example images of labels in Cath Lab dataset: (a) cardiologist; (b) lab assistant; (c) patient; (d) operating table; (e) instrument table; (f) control panel display; (g) control panel button; (h) x-ray detector; (i) x-ray source; (j) lead shield; (k) radiation light; (l) display (TV).

Table 4.1. In this thesis, the Cath Lab dataset is used to fine-tune the object detection model to fit the medical equipment detection task in Cath Lab, while COCO dataset is also used for pre-training weights but won't be discussed here.

Table 4.1: Distribution of labels in each dataset.

Label	991-image	Video 1007	Video 1008
All	4402	43825	46592
Cardiologist	0	1400	1469
Lab Assistant	961	3921	4038
Patient	0	2713	2865
Instrument Table	434	2628	2745
Operating Table	800	3725	3965
Control Panel Display	0	4450	4625
Control Panel Buttons	0	3838	4003
X-Ray Detector	818	3063	3425
X-Ray Source	304	1888	2652
Lead Shield	484	5075	5302
Radiation Light	0	440	465
Display	991	2523	2459

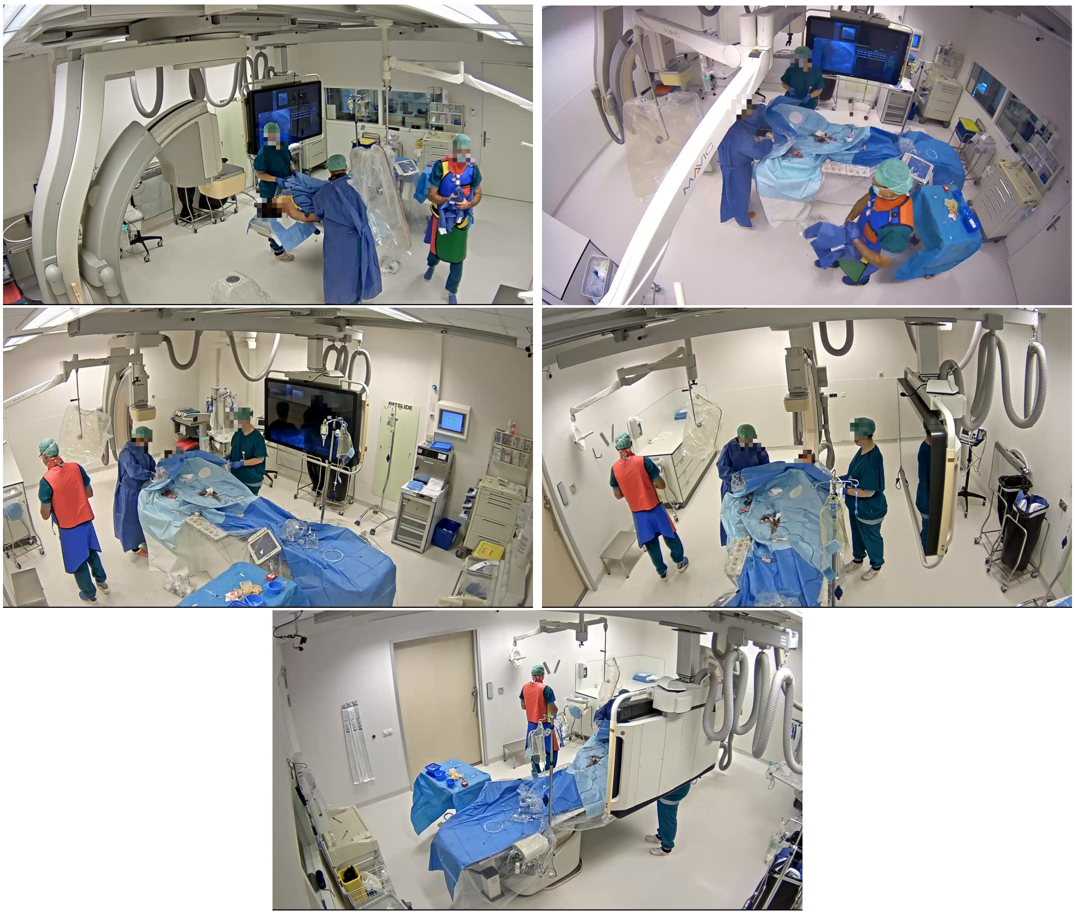


Figure 4.2: Example images in Cath Lab Dataset taken from different viewpoints.

4.1.2 Evaluation Metrics

4.1.2.1 Precision and Recall Scores

Precision and recall rates are the most popular performance indicators in machine learning. The performance of any machine learning system predicting a binary label can be assessed by comparing the prediction to the ground truth. Every prediction can be labeled with True/False and Positive/Negative. In our object detector, predicted boxes are refined by the IoU and confidence score thresholds before output. While the confidence score threshold determines the Positive/Negative label of predicted boxes, i.e. a "pass" to output, IoU threshold determines the correctness in fact. Thus, the Positive/Negative label is assigned to predicted bounding boxes with a confidence score value larger/smaller than the threshold, and the True/False label is assigned to predicted boxes with the same/different Positive/Negative label to the actual correctness. The combination of labels and complete analysis is shown in table 4.2.

Then based on the attributes mentioned above, the precision and recall can be defined as

$$Precision = \frac{TP}{TP + FP}, \tag{4.1}$$

Table 4.2: Examples of TP, TN, FP, FN.

	IoU>threshold	IoU<threshold
Confidence score>threshold	TP	FP
Confidence score<threshold	FN	TN

$$Recall = \frac{TP}{TP + FN}, \tag{4.2}$$

where TP, FP, TN and FN are the number of bounding boxes that fall into their respective labels. Precision is then the fraction of outputted correct bounding boxes out of all predicted boxes passed by the algorithm. Recall is the fraction of outputted correct bounding boxes out of all actually correct bounding boxes. By setting different thresholds for confidence scores, different precision and recall values can be obtained and used to draw the precision-recall curve (PR curve). To analyze the relationship between precision and recall scores. A PR curve obtained from the results of Scaled-YOLOv4 is shown in fig. 4.3. From the figure, it is obvious that the precision-recall is downward sloping. To obtain the optimal precision and recall scores, we find the confidence score thresholds using the following criterion: equal error rate ($Precision = Recall$), unique precision score ($Precision \geq 0.995$) and unique recall score ($Recall \geq 0.995$).

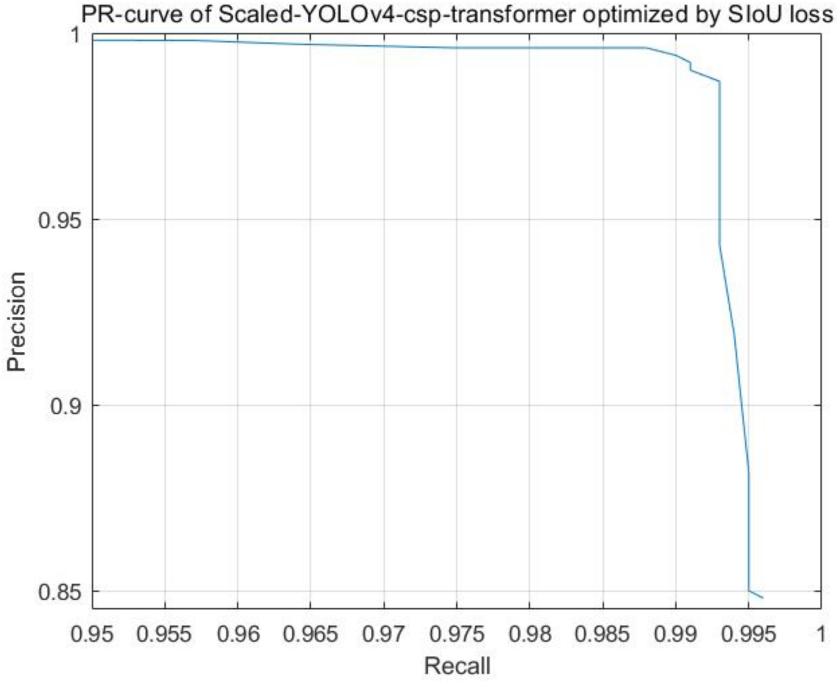


Figure 4.3: The PR curve obtained in this thesis.

4.1.2.2 Mean Average Precision

Mean Average Precision (mAP) is the most frequently used index to evaluate the performance of bounding box prediction in object detection. As the name suggests, mAP is the mean value of average precision (AP) across all classes, while AP is calculated by taking the expectation of precision over different thresholds in a single class. Since the precision values for different confidence score thresholds are shown in the PR curve, the value of AP can be obtained by the area under the PR curve. To achieve this, it needs to integral confidence score from 0 to 1 by definition. However, as a discrete system, the computer is unable to apply continuous integration to precision. Instead, we divide the range of confidence score into 100 steps (101 points) and calculate the expectation of precision at 100 steps, which is given as:

$$AP = \frac{1}{101}\sum_{i=1}^{101} P(c_i), \tag{4.3}$$

where c_i is the confidence score at i^{th} step and $P(\dot{})$ is the precision function at given IoU threshold. A confidence score of 0.001 is used in the beginning, because the confidence score from 0 to 0.001 (almost no effect on AP) can be ignored to speed up AP calculation. In addition to the confidence score, the IoU threshold, as another key factor in the evaluation of the object detection system, limits the achieved AP by a tighter threshold. In this thesis, since we only care about predicted bounding boxes with enough overlapping area to ground truth, a range of thresholds from 0.50 to 0.95 with a step size of 0.05 are used. In order to evaluate the object detector performance under normal IoU threshold and tighter thresholds, we use both AP at 0.5 IoU threshold and average AP over 10 IoU thresholds. Finally, we take the average over all classes (and all thresholds), which is given as:

$$mAP = \frac{1}{N_{IoU}N_c}\sum_{j=1}^{N_{IoU}}\sum_{k=1}^{N_c} AP_k^{IoU_j}, \tag{4.4}$$

where N_{IoU} and N_c are the numbers of IoU thresholds and classes respectively.

4.1.2.3 Object Detection Speed

The last significant assessment index is the detection speed, which can be easily evaluated through the detection time for one image. Actually, due to sophisticate spatial coverage between objects, some images may take more processing time. For purposes of ablation study and demonstrating approximate speed, we use a video of a complete procedure consisting of more than 80,000 frames for testing and use the average detection rate as the speed index. The hardware we use for object detection is another factor that may lead to differences between results got from different computers. Since the final objective of this thesis is to run the well-designed object detection on the computer in the RdGG hospital to detect medical equipment, all experiments are performed on the same computer in the hospital to avoid unexpected influence.

4.2 Results

In this section, the results of Scaled-YOLOv4 on detecting medical equipment are shown and analyzed. First, the hyperparameters used to train the Scaled-YOLOv4 are explained in section 4.2.1. In section 4.2.2, the results of fine-tuned models with different IoU losses are shown and analyzed, followed by the comparison of performance of different network architecture in section 4.2.3. Section 4.2.4 presents the results of KM filter.

4.2.1 Training Schedule

Training schedule, also called training strategy, is a set of hyperparameters that controls the learning process of the network. A good training schedule can accelerate the convergence of optimization and contribute to a well-trained model with better performance. In deep learning, hyperparameters are optimized to achieve the best performance of the network on the testing dataset. In this section, we discuss the optimal combination of hyperparameters used in the training stage and the reason for setting them.

The hyperparameters, defined before training, consist of the learning rate, momentum, weights decay, loss weights, and parameters used in some modules. The list of hyperparameters we use for fine-tuning and training from scratch is shown in Table table 4.3. In addition to the hyperparameters used in neural network (learning rate, momentum, weight decay, loss gains, IoU thresholds), we also define hyperparameters for data augmentation and anchor setup (HSV augmentation fractions, image translation fraction, image scale gain, image flip left-right probability and mixup probability.

When fine-tuning the object detector based on pre-trained weights, we use a slower learning rate to generate the optimal model. In contrast, we use a larger learning rate to learn the representation faster in the beginning when training from scratch. Since the fine-tuned model has been pre-trained on another object detection task before training, it is able to make accurate predictions about bounding boxes of objects. However, the pre-trained model needs to be fine-tuned for the Cath Lab dataset to learn the classification of new labels. Therefore, we increase the gains for objectness and classification losses when fine-tuning Scaled-YOLOv4 to make the object detector pay less attention to bounding box regression than to classification and objectness score. To train a better model from scratch, we also change some data augmentation parameters, which are obtained by experiments.

4.2.2 Fine-tuned Scaled-YOLOv4

The Scaled-YOLOv4 model is first fine-tuned on the Cath Lab dataset with pre-trained weights, which help the model converge faster (within 100 epochs). The precision, recall rates, and mAP results of the fine-tuned Scaled-YOLOv4-p5 network with SIoU are shown in table 4.5. From the results in table 4.5 we know that our fine-tuned Scaled-YOLOv4 model achieves very good overall results in detecting medical equipment in Cath Lab. An example of the image with detected bounding boxes is shown in fig. 4.4. As discussed in section 4.1.2.2, the precision and recall are dependent on the threshold

Table 4.3: Hyperparameters

Hyperparameters	Description	Fine-tune	Train from scratch
lr0	Initial learning rate	0.01	0.015
momentum	momentum	0.937	0.937
weight_decay	Optimizer weight decay	0.0005	0.0005
iou	IoU loss gain	0.05	0.055
cls	Class loss gain	0.5	0.3
obj	Objectness loss gain (scale with pixels)	1.0	0.55
iout	IoU training threshold	0.2	0.2
anchor$_t$	Anchor-multiple threshold	4.0	5.0
hsc$_h$	Image HSV-Hue augmentation fraction	0.015	0.015
hsc$_s$	Image HSV-Saturation augmentation fraction	0.7	0.6
hsv$_v$	Image HSV-Value augmentation fraction	0.4	0.4
translate	Image translation fraction	0.5	0.25
scale	Image scale gain	0.8	0.75
fliplr	Image flip left-right probability	0.5	0.6
mixup	Image MixUp probability probability	0.2	0.15

of the confidence score. To find the optimal confidence score threshold in the trade-off of precision and recall, we test the recall and precision using different thresholds. The results are shown in table 4.4. The PR curve drawn using the data in table 4.4 is shown in Figure 4.3. We set the threshold of confidence score to 0.585 to detect medical equipment because the corresponding recall and precision are equal and both larger than 0.99.

Confidence threshold	0.01	0.1	0.2	0.3	0.35	0.4	0.5	0.58
Recall	0.996	0.995	0.995	0.994	0.993	0.993	0.993	0.991
Precision	0.848	0.85	0.882	0.919	0.943	0.971	0.987	0.99
Confidence threshold	0.585	0.6	0.7	0.75	0.8	0.85	0.9	0.95
Recall	0.991	0.991	0.99	0.988	0.975	0.964	0.957	0.829
Precision	0.991	0.992	0.994	0.996	0.996	0.997	0.998	0.999

Table 4.4: Precision and recall rates of Scaled-YOLOv4 model with a range of confidence score thresholds.

The Scaled-YOLOv4 achieves higher than 95% mAP for most labels when the IoU threshold is set to 0.5, indicating that the annotated objects can always be detected by our object detector with 50% overlap area. The results of mAP over higher IoU thresholds are a little worse since only bounding boxes with more overlap area are labeled as positive. While the results of mAP@.5:.95 are excellent (higher than 90%) for most labels, the mAP for radiation light is lower than expected. This is because the small errors in the prediction are magnified on the radiation lights that are so

Table 4.5: Precision, recall and mAP results for each label of fine-tuning pre-trained ScaledYOLOv4-p5 network with SIoU loss function and confidence threshold 0.4.

Label	Precision	Recall	mAP@.5	mAP@.5:.95
All	0.993	0.987	0.976	0.926
Cardiologist	0.982	0.982	0.985	0.963
Lab Assistant	0.983	0.989	0.985	0.928
Patient	1	0.992	0.995	0.953
Instrument Table	0.993	1	0.995	0.963
Operating Table	0.954	0.998	0.995	0.961
Control Panel Display	0.994	0.989	0.985	0.951
Control Panel Buttons	0.987	0.974	0.974	0.921
X-Ray Detector	0.995	0.987	0.985	0.959
X-Ray Source	0.99	1	0.995	0.98
Lead Shield	0.995	0.995	0.995	0.957
Radiation Light	1	0.83	0.83	0.635
Display	1	1	0.995	0.995

small that they only occupy hundreds of pixels in input images. Compared with large objects that contain more than 10,000 pixels, both intersection and union areas of small objects are more severely affected by prediction errors. This contributes to a significant decrease in the percentage of overlap area (IoU value) and thus reduces the mAP. However, the mAP for radiation lights is detected well with a lower IoU threshold (0.5), indicating that our object detector can correctly recognize the radiation lights but the prediction has some errors. Considering the small size of bounding boxes, the errors are acceptable in practice. Benefiting from the obvious plastic cover and unchanging shape, the transparent lead shield, which was thought difficult in object detection, achieves high mAP.

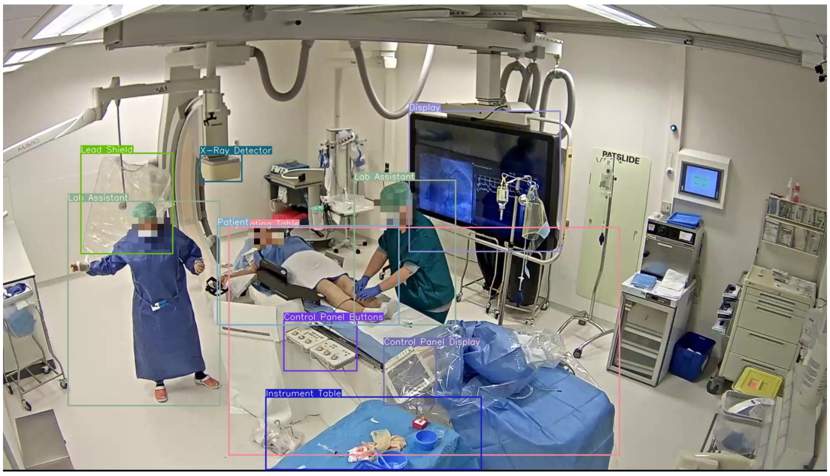


Figure 4.4: Medical equipment detected by fine-tuned Scaled-YOLOv4.

The excellent precision is also attributed to the overfitting of our model on Cath Lab datasets since all samples used for training are captured for the medical equipment in the Cath Lab from specific perspectives. Indeed, our trained object detector performs well when detecting the medical equipment in Cath Lab but may fail in another operating room due to the different appearance of the equipment.

Table 4.6: mAP@.5:.95 results of fine-tuned Scaled-YOLOv4 with different IoU losses for each label.

Label	CIoU	EIoU	ECIoU	α-EIoU	SIoU
All	0.936	0.941	0.942	0.44	0.946
Cardiologist	0.959	0.959	0.962	0.966	0.968
Lab Assistant	0.926	0.929	0.927	0.93	0.933
Patient	0.954	0.952	0.949	0.957	0.961
Instrument Table	0.971	0.971	0.972	0.973	0.982
Operating Table	0.972	0.971	0.969	0.976	0.977
Control Panel Display	0.955	0.963	0.961	0.974	0.971
Control Panel Buttons	0.925	0.921	0.928	0.933	0.948
X-Ray Detector	0.98	0.953	0.971	0.978	0.977
X-Ray Source	0.979	0.982	0.985	0.982	0.982
Lead Shield	0.968	0.967	0.964	0.973	0.978
Radiation Light	0.645	0.733	0.725	0.695	0.679
Display	0.995	0.995	0.995	0.994	0.995

Since fine-tuning a pre-trained model is very fast, it is used to test the techniques without changing the network architecture, i.e. different loss functions. Table 4.6 shows the results of mAP@.5:.95 of fine-tuned Scaled-YOLOv4 with different IoU losses. As the GIoU and DIoU losses are concluded in other research [35] that they are outperformed by CIoU, we only test the state-of-the-art IoU losses: CIoU, EIoU [63], ECIoU, α-EIoU, and SIoU. It can be seen from the results of table 4.6, CIoU used by the initial YOLOv4 model is outperformed by other IoU losses by more than 1% mAP. By considering more geometric information, EIoU and ECIoU achieve better performance (around 92% mAP) than CIoU. The best result is achieved by α-IoU and SIoU, at approximately 92.5% mAP. However, the α-IoU and SIoU also spend more time in training since they spend more time on the computation of additional geometric information. To achieve better performance and convergence speed, we choose the SIoU loss and use it in other experiments.

4.2.3 Network Architectures

Scaled-YOLOv4 provides networks with different depths and widths for different needs. We compare the performance and speed of different network architectures and the results are shown in table 4.7. Since the pre-trained weights of the Scaled-YOLOv4-p5, p6, and p7 models are provided, they are tested by fine-tuning and compared with the Scled-YOLOv4-csp model that is trained from scratch. It can be seen from table 4.7 that there is a trade-off between speed and performance. Scaled-YOLOv4-p7 with deeper network architecture and more parameters achieve higher mAP, while p5 and csp detect objects faster. Benefiting from the well-designed network and large dataset, all models achieve excellent mAP (higher than 94%) so we pay more attention to the speed in this trade-off. The Scaled-YOLOv4-csp network, which has the same depth but fewer parameters than the p5 model, runs much faster than other models and we use it as the base of the transformer layer and KM-filter in this thesis.

Table 4.7: mAP@.5:.95 results of fine-tuned Scaled-YOLOv4 with different network architectures.

Label	YOLOv4-csp	YOLOv4-p5	YOLOv4-p6	YOLOv4-p7
All	0.942	0.946	0.946	0.948
Cardiologist	0.946	0.968	0.96	0.964
Lab Assistant	0.866	0.933	0.928	0.935
Patient	0.957	0.961	0.958	0.941
Instrument Table	0.966	0.982	0.977	0.975
Operating Table	0.966	0.977	0.974	0.972
Control Panel Display	0.964	0.971	0.973	0.97
Control Panel Buttons	0.934	0.948	0.933	0.93
X-Ray Detector	0.975	0.977	0.979	0.975
X-Ray Source	0.983	0.982	0.981	0.983
Lead Shield	0.964	0.978	0.973	0.967
Radiation Light	0.763	0.679	0.721	0.767
Display	0.995	0.995	0.995	0.995
Time per Image	17ms	22ms	28ms	34ms

Then, we implement a self-attention layer, as mentioned in section 3.1.2, on Scaled-YOLOv4-csp to improve the ability to focus on regions with objects. To test the performance of the transformer layer, we add a randomly initialized transformer layer on the end of the backbone and train it from scratch. The results of Scaled-YOLOv4-csp models with transformer layer are shown in table 4.8. It shows that the object detector with transformer layer significantly improves the overall mAP by about 1%. It is worth mentioning that, the detection of lab assistant has noticeable poor performance when models are trained from scratch. That's because the weights used in fine-tuning models are pre-trained on large dataset and can enhance the ability of network to extract distinguishable features, while training scratch weakened the ability of networks. In the same time, the samples of lab assistant contains a variety of poses. Sometimes the lab assistants stretch their arms so that the hands and arms cannot be detected since they are too far away from the body, which can be easily detected. This issue does not occur in patient and cardiologist due to their simpler poses and behavior.

4.2.4 KM-filter

The KM-filter is designed to adjust predicted bounding boxes with large errors by considering the correct boxes from other viewpoints. We compare the mAP@.5:.95 of the object detector tested in the previous section with the results adjusted by the KM-filter,

Table 4.8: mAP@.5:.95 results of Scaled-YOLOv4 models trained from scratch with/without transformer layers for each label.

Label	YOLOv4-csp	YOLOv4-csp-transformer	YOLOv4-csp-transformer-KMfilter
All	0.942	0.951	0.915
Cardiologist	0.946	0.94	0.963
Lab Assistant	0.866	0.886	0.914
Patient	0.957	0.968	0.959
Instrument Table	0.966	0.971	0.948
Operating Table	0.966	0.975	0.967
Control Panel Display	0.964	0.971	0.868
Control Panel Buttons	0.934	0.931	0.846
X-Ray Detector	0.975	0.978	0.939
X-Ray Source	0.983	0.988	0.923
Lead Shield	0.964	0.971	0.937
Radiation Light	0.763	0.836	0.717
Display	0.995	0.995	0.994

as shown in table 4.8. The results in table 4.8 show that the KM-filter actually reduces the overall precision. This is caused by the error generated by the keypoints clustering in KM-filter. As mentioned in section 3.2.3, the DBSCAN clustering technique is used to divide the matched keypoints from different objects. However, in some cases, some objects are very close to each other in either viewpoint or even in contact with each other. Some keypoints on the edges of two objects may have very small Euclidean distance between each other, leading to a larger keypoint cluster. Consequently, the bounding boxes may be over-expanded and have more union area (smaller IoU) with ground truth boxes.

Although there is not always this error, it performs worse than the original object detector if there is not enough room for improvement. Since the Scaled-YOLOv4-csp-transformer object detector we trained in the previous section performs excellently in medical equipment detection, it predicts a low probability of error. The KM-filter can only make small adjustments to the predicted boxes, which also has little effect on increasing IoU when the object is very large.

However, the KM-filter significantly improves the detection of lab assistant made by models trained from scratch. As mentioned before, the lab assistant is poorly detected because of their poses and stretched arms that is hard to identify. However, if the arms and posses are detected by one of the viewpoint then the KM-filter can help spread the information to other viewpoints. As a results, the probability of missing arms or other parts in bounding boxes of lab assistant is less and the detection results is improved. the KM-filter plays a role in ensuring the whole object is included in bounding boxes. In follow-up works, the predicted bounding boxes in 2D space will be used to find medical equipment in 3D space. Boxes containing complete objects generate the correct 3D objects while missing parts of objects can lead to incomplete 3D objects. In addition, both the location and scales of objects are used for workflow analysis, missing part of the object may produce errors when analyzing the workflow.

Conclusion 5

In this chapter, our work presented in this thesis is summarized and the recommendation for the follow-up study is given. The conclusion to this thesis is presented first in Section 5.1, followed by the discussion of future work to improve our methods or expand the application in Section 5.2.

5.1 Summary

As a significant examination tool for the diagnosis of cardiovascular diseases, Cath Lab integrates extensive medical equipment [8]. However, the procedures made in Cath Lab also poses threat to the safety of patient and cardiologist through potential medical errors [12] and radioactive imaging equipment [11], e.g. X-ray source. For purposes of protecting the personnel inside the Cath Lab and improving procedure efficiency, it is necessary to apply workflow analysis by observing the activities of personnel and medical equipment inside the Cath Lab [2]. With movable medical equipment, one of the steps in the automation of the workflow analysis in Cath Lab is to automatically locate the medical equipment. To achieve this, five finely calibrated cameras are mounted on different walls and corners of the Cath Lab to observe the location of medical equipment within the Cath Lab from different viewpoints.

In this thesis, we propose a pipeline for multiple camera object detector on Cath Lab dataset. Our proposed object detector is based on Scaled-YOLOv4 [13] with added transformer layers [57] and improved IoU loss function. The transformer layer, put at the end of the backbone network, uses a multi-head self-attention block to focus on the regions containing objects in extracted feature maps. To find the IoU loss function that can achieve optimal precision, we investigate multiple state-of-the-art IoU losses, including CIoU [62], EIoU [63], EC-IoU [63], α-EIoU [66] and SIoU [64]. A keypoint-matching-based filter (KM-Filter) is also developed to combine information from different perspectives to adjust predicted bounding boxes. The filter first detects keypoints of the object inside a bounding box by SuperPoint [14] then match them with keypoints from the same object but neighboring perspectives by SuperGlue [15]. A DBSCAN clustering technique [67] is performed to refine the matched keypoints that are used to determine the new boundaries of bounding boxes. Our KM-Filter performs keypoint detection for each image and keypoint matching with images from neighboring cameras. It is worth noticing that the keypoints from the current image are matched with the keypoints from neighboring cameras that have already been filtered and believed to be reliable.

In this thesis, experiments are made to test the combinations of different IoU losses and network architectures. The mean average precision (mAP) at different thresholds is used to evaluate the performance achieved by different solutions. We use mAP@.5:.95

to represent the expectation of mAP over all classes at a range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05. By fine-tuning a pre-trained network, we test the performance achieved by the Scaled-YOLOv4 network using different IoU loss functions. It is found the models with SIoU and α-IoU loss functions achieve higher mAP@.5:.95, at about 92.5% than other loss functions. However, due to fewer computations for calculating loss during training, SIoU diverges faster. Another experiment, training the scaled-YOLOv4 network from scratch, is to test the performance achieved by different network architectures. It shows that the network with transformer layer performs better, at 95.1% mAP@.5:.95, than the standard Scaled-YOLOv4 network (94.3% mAP@.5:.95). Consequently, Scaled-YOLOv4 with SIoU loss and additional transformer layer is used as the object detector to predict bounding boxes in this thesis. Finally, KM-Filter is added after Scaled-YOLOv4 and the 94.3% mAP@.5:.95 is improved for 0.2%.

In addition to the precision, we also pay attention to the speed of running object detection pipeline to Cath Lab dataset. The Scaled-YOLOv4-csp network is the fastest network architecture, predicting bounding boxes with a speed of 58 FPS on RTX 3090.

5.2 Future Work

There are lots of ideas to further improve the object detector pipeline in this thesis. However, due to time constrains and thesis topic, some of them are not studied. To further improve the performance of our pipeline or explore new potential application, these unrealized ideas deserve to be studied in depth. The potential follow-up study direction is listed below:

- **Larger dataset** The dataset we use in this thesis only contains videos of two procedures in addition to a 991-image dataset. In fact, there are numerous videos of procedures that can be used to enlarge our dataset. Our pipeline can generate accurate predictions of medical equipment for these videos, which can be used as annotation after checking and adjusting. The larger dataset can make our pipeline fit better when training and perform more accurately.

- **Better object detector as baseline.** Because this thesis started since last year, the Scaled-YOLOv4, as the latest version of the YOLO object detector at the time, was chosen as the baseline from the very beginning. Scaled-YOLOv4 achieves 55.5% AP for the MS COCO dataset. However, another latest version of YOLO, YOLOv7, was published last month. YOLOv7 achieves 56.8% AP and a faster speed for MS COCO dataset by using more bag-of-freebies methods, "extend" and "compound scaling" methods. Therefore, it is necessary to transplant our methods to YOLOv7 for better precision and faster speed.

- **Apply KM-Filter based on the confidence score of predicted boxes.** As mentioned in section 4.2.4, the KM-Filter performs excellently on adjusting bounding boxes with little IoU with the ground truth and has little effect on well-predicted bounding boxes. Since the confidence score is positively related to the precision of predicted boxes in most cases. The KM-Filter can be applied to those predicted boxes with low confidence scores and skip those with high scores. As a consequence, the incorrectly predicted boxes can be improved and the time for will-correct boxes can be saved.

- **Use our pipeline to detect objects in 3D space or 3D point clouds.** The application of this thesis is to provide locations in 3D space for workflow analysis. To achieve this, we can match the bounding boxes in 2D (this has been down by KM-Filter) and use finely calibrated cameras to find the bounding boxes in 3D space. Moreover, we can determine their 3D location by using 3D point clouds. With camera calibration information, the coordinates of matched keypoints in 3D space can be obtained and form a point cloud for each object.

Bibliography

[1] Y. Jiang, R. Dai, J. Zeng, R. Butler, T. Vijfvinkel, Y. Wang, J. van den Dobbelsteen, M. van der Elst, J. Dauwels, Object detection and person tracking in cathlab with automatically calibrated cameras, in: 42nd WIC Symposium on Information Theory and Signal Processing in the Benelux (SITB 2022), 2022.

[2] M. Vankipuram, K. Kahol, T. Cohen, V. L. Patel, Toward automated workflow analysis and visualization in clinical environments, Journal of Biomedical Informatics 44 (3) (2011) 432–440.

[3] P. Bellido-Montesinos, F. Lozano-Galant, F. J. Castilla, J. A. Lozano-Galant, Experiences learned from an international bim contest: Software use and information workflow analysis to be published in: Journal of building engineering, Journal of Building Engineering 21 (2019) 149–157.

[4] J. Redmon, S. Divvala, R. Girshick, 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.

[5] M. Maity, S. Banerjee, S. S. Chaudhuri, Faster r-cnn and yolo based vehicle detection: A survey, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 2021, pp. 1442–1447.

[6] J. V. Kanna, S. E. Raj, M. Meena, S. Meghana, S. M. Roomi, Deep learning based video analytics for person tracking, in: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), IEEE, 2020, pp. 1–6.

[7] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, P. Dario, Visual-based defect detection and classification approaches for industrial applications—a survey, Sensors 20 (5) (2020) 1459.

[8] T. M. Maddox, P. M. Ho, M. Roe, D. Dai, T. T. Tsai, J. S. Rumsfeld, Utilization of secondary prevention therapies in patients with nonobstructive coronary artery disease identified during cardiac catheterization: insights from the national cardiovascular data registry cath-pci registry, Circulation: Cardiovascular Quality and Outcomes 3 (6) (2010) 632–641.

[9] M. J. Kern, H. Samady, Current concepts of integrated coronary physiology in the catheterization laboratory, Journal of the American College of Cardiology 55 (3) (2010) 173–185.

[10] A. C. Glatz, X. Zhu, M. J. Gillespie, B. D. Hanna, J. J. Rome, Use of angiographic ct imaging in the cardiac catheterization laboratory for congenital heart disease, JACC: Cardiovascular Imaging 3 (11) (2010) 1149–1157.

[11] C. E. Chambers, K. A. Fetterly, R. Holzer, P.-J. P. Lin, J. C. Blankenship, S. Balter, W. K. Laskey, Radiation safety program for the cardiac catheterization laboratory, Catheterization and Cardiovascular Interventions 77 (4) (2011) 546–556.

[12] L. L. Leape, D. M. Berwick, Five years after to err is human: what have we learned?, Jama 293 (19) (2005) 2384–2390.

[13] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Scaled-yolov4: Scaling cross stage partial network, in: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, 2021, pp. 13029–13038.

[14] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 224–236.

[15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.

[16] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., Deep high-resolution representation learning for visual recognition, IEEE transactions on pattern analysis and machine intelligence 43 (10) (2020) 3349–3364.

[17] M. R. Hribar, S. Read-Brown, I. H. Goldstein, L. G. Reznick, L. Lombardi, M. Parikh, W. Chamberlain, M. F. Chiang, Secondary use of electronic health record data for clinical workflow analysis, Journal of the American Medical Informatics Association 25 (1) (2018) 40–46.

[18] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, arXiv preprint arXiv:1905.05055 (2019).

[19] P. Piccinini, A. Prati, R. Cucchiara, Real-time object detection and localization with sift-based clustering, Image and Vision Computing 30 (8) (2012) 573–587.

[20] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, International journal of computer vision 60 (1) (2004) 63–86.

[21] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Transactions on information theory 21 (1) (1975) 32–40.

[22] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE transactions on neural networks and learning systems 30 (11) (2019) 3212–3232.

[23] R. Girshick, J. Donahue, T. Darrell, 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.

[24] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE transactions on pattern analysis and machine intelligence 37 (9) (2015) 1904–1916.

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

[26] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

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

[28] W. S. Noble, What is a support vector machine?, Nature biotechnology 24 (12) (2006) 1565–1567.

[29] O. Chapelle, P. Haffner, V. N. Vapnik, Support vector machines for histogram-based image classification, IEEE transactions on Neural Networks 10 (5) (1999) 1055–1064.

[30] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International journal of computer vision 111 (1) (2015) 98–136.

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

[32] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

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

[34] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[35] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[36] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint arXiv:2107.08430 (2021).

[37] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, arXiv preprint arXiv:2207.02696 (2022).

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[39] 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 115 (3) (2015) 211–252.

[40] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, PMLR, 2015, pp. 448–456.

[41] X. Jiang, J. Ma, G. Xiao, Z. Shao, X. Guo, A review of multimodal image matching: Methods and applications, Information Fusion 73 (2021) 22–71.

[42] C. Chen, B. Wang, C. X. Lu, N. Trigoni, A. Markham, A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence, arXiv preprint arXiv:2006.12567 (2020).

[43] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, Ieee, 2011, pp. 2564–2571.

[44] J. Revaud, C. De Souza, M. Humenberger, P. Weinzaepfel, R2d2: Reliable and repeatable detector and descriptor, Advances in neural information processing systems 32 (2019).

[45] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, T. Sattler, D2-net: A trainable cnn for joint description and detection of local features, in: Proceedings of the ieee/cvf conference on computer vision and pattern recognition, 2019, pp. 8092–8101.

[46] A. Jakubović, J. Velagić, Image feature matching and object detection using brute-force matchers, in: 2018 International Symposium ELMAR, IEEE, 2018, pp. 83–86.

[47] P. Chamoso, A. Rivas, J. J. Martín-Limorti, S. Rodríguez, A hash based image matching algorithm for social networks, in: International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer, 2017, pp. 183–190.

[48] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[49] P.-E. Sarlin, C. Cadena, R. Siegwart, M. Dymczyk, From coarse to fine: Robust hierarchical localization at large scale, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12716–12725.

[50] L. Cavalli, V. Larsson, M. R. Oswald, T. Sattler, M. Pollefeys, Adalam: Revisiting handcrafted outlier detection, arXiv preprint arXiv:2006.04250 (2020).

[51] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[52] T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).

[53] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023–6032.

[54] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, Advances in neural information processing systems 27 (2014).

[55] J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, arXiv preprint arXiv:1601.06733 (2016).

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

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

[59] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

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

[61] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658–666.

[62] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 12993–13000.

[63] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, T. Tan, Focal and efficient iou loss for accurate bounding box regression, Neurocomputing 506 (2022) 146–157.

[64] Z. Gevorgyan, Siou loss: More powerful learning for bounding box regression, arXiv preprint arXiv:2205.12740 (2022).

[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[66] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, X.-S. Hua, α-iou: A family of power intersection over union losses for bounding box regression, Advances in Neural Information Processing Systems 34 (2021) 20230–20242.

[67] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited, revisited: why and how you should (still) use dbscan, ACM Transactions on Database Systems (TODS) 42 (3) (2017) 1–21.