Computational Analysis of Table Tennis Games from Real-Time Videos Using Deep Learning

Hong Zhou

A thesis submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2023

School of Engineering, Computer & Mathematical Sciences

Abstract

Utilizing a multiscale training dataset, YOLOv8 leverages deep learning to deliver rapid inference capabilities and exceptional accuracy in detecting visual objects, particularly smaller ones. The performance surpasses that of transformer-based deep learning models, positioning YOLOv8 as a leading algorithm in its field. While the effectiveness of visual object detection is generally assessed using pre-trained models on enhanced datasets, fine-tuning becomes crucial for specific situations like table tennis matches and coaching sessions. The unique challenges in these contexts include rapid ball movement, uniform color, fluctuating lighting conditions, and bright reflections due to intense illumination. In this thesis, we introduce a motion-centric algorithm to augment YOLOv8 model, aiming to improve the accuracy of predicting ball trajectories, impact points, and velocity in the realm of table tennis. This adaptive model not only elevates its utility in real-time sports coaching but also demonstrates its potential in other fast-paced settings. Experimental results indicate a significant improvement in detection rates and a reduction in false positives.

Keywords: YOLOv8, Moving balls, DETR, Image pre-process, Image post-process, Background subtraction, Deep learning

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:  Date: 15 September 2023

Acknowledgment

I am profoundly grateful for the support and guidance I've received throughout my time at the Auckland University of Technology (AUT) while pursuing my master's degree. This journey would have been impossible without the steadfast encouragement from my family and the invaluable mentorship from my supervisors.

First and foremost, my heartfelt appreciation goes out to my family. Their consistent love, encouragement, and faith in my abilities have been my bedrock of support. Their unwavering backing during both the challenges and triumphs of this academic journey has been indispensable in shaping my success. Their sacrifices and enduring belief in my potential have driven me to excel and persist.

Additionally, I am deeply indebted to Dr Wei Qi Yan and Dr Minh Nguyen for their extraordinary guidance and steadfast commitment. Their profound insights, patience, and mentorship have greatly enriched my academic experience and broadened my perspectives. Their skill in providing constructive criticism, thoughtful suggestions, and invaluable direction has been pivotal in determining the success of this research. Meanwhile, sincerely thanks to Mr. Rodney Bygrave for providing hands-on guidance and assistance in table tennis techniques, training methods, and judgment. I am grateful for their investment of time and effort to guide me through the complexities of my research project, as well as for fostering an environment where creativity and innovation could flourish.

Hong Zhou

Auckland, New Zealand

September 2023

Chapter 1
Introduction

This chapter is composed of five parts: The first part introduces the background and motivations, the second part includes the research question, followed by the contributions, objectives, and structure of this thesis.

1.1 Background and Motivation

Deep learning methods have gained traction in sports competitions, particularly in tasks such as determining the placement of balls in table tennis. This addresses inherent challenges in table tennis, such as the diminutive size of balls and subtle texture patterns. Compared to other sports balls, table tennis can be hard to distinguish from background textures, complicating the process of determining their landing points and velocities.

The benefits of utilizing deep learning in machine learning are quite evident, particularly in the domain of computer vision. Voulodimos et al (2018) conducted research work to assess the advantages and constraints of deep learning, and also discussed the future directions of computer vision design based on various practical applications.

Human and animal brain can process and understand diverse types of information, enabling the recognition of complex structures in large-scale data (Liang and Yan, 2024). Deep learning emulates this mechanism by establishing numerous data abstraction and computational layers. Unsupervised and supervised feature learning algorithms, hierarchical probability models, and neural networks are all examples of deep learning (Cao & Yan, 2022; Chen & Yan, 2024). When confronted with a large volume of complex data, deep learning has been shown to outperform previous technologies.

The development of neural networks was spurred by McCulloch and Pitts (1943) desire to create an artificial brain, with the MCP model serving as the earliest neuron model. LeNet (LeCun et al, 1989) and Long Short-Term Memory (LSTM) (Graves & Graves, 2012) have also made significant contributions to the field. However, the true era of deep learning began in 2006, after Hinton et al (2006) made a major breakthrough with a Deep Belief Network that employed multiple layers of Restricted Boltzmann Machines. This structure can facilitate layer-by-layer local training and learning without supervision, which is why deep learning frameworks and algorithms have gained popularity in recent decades (An & Yan, 2021).

Open high-quality large datasets and GPUs with high computational capabilities have greatly improved model training, thereby promoting the development of network and machine learning. Other factors that have contributed to this progress include addressing issues such as gradient disappearance that are caused by out-of-saturation activation functions, and the emergence of more powerful frameworks such as discarding, batch normalization, and data augmentation, as well as new regularization technologies like Mxnet, TensorFlow, and Theano (Bastien et al, 2012).

Deep learning has made significant strides in addressing visual problems such as semantic segmentation (Noh, 2015), (Long, 2015), human motion tracking (Doulamis & Voulodimos, 2016; Doulamis, 2018), human action recognition (Lin et al, 2016; Cao & Nevatia, 2016), visual object detection (Ouyang et al, 2016; Diba et al, 2017), and human pose estimation (Toshev & Szegedy, 2014; Chen & Yuille, 2014). The three most typical deep learning frameworks in this context are Stacked (Denoising) Autoencoders, Deep Belief Networks (DBNs), and Convolutional Neural Networks (CNNs).

In 2016, YOLO was created by Redmon et al (2016), and YOLOv8 means the eighth- version of YOLO models, not appear suddenly, but evolve from an earlier version of YOLOv5 by ultralytics, the initial group which created YOLO. The improvement from YOLOv5 to YOLOv8 includes the Backbone and head structure, anchor and training strategy of the epoch. In the backbone part, C2f structure with richer gradient flow in YOLOv8 is selected to replace the C3 structure in YOLOv5. In order to reduce the number of blocks of the largest stage in the backbone network, the models with different scaling factors N/S/M/L/X no longer share a set of model parameters. The M/L/X large model also reduces the number of output channels of the last stage, further reducing the number of parameters and calculations. Meanwhile, anchor-based was an alternative by Anchor-Free, TAL (Task Alignment Learning) dynamic matching adopted, and DFL (Distribution Focal Loss) and CIoU Loss are utilized as the loss function of the regression branch, which makes classification tasks correspond to regression tasks.

While addressing the complexities associated with detecting and identifying objects

in fast-paced environments such as table tennis, the selection of the most optimal model emerges as an indispensable step. Currently, deep learning predominantly features two mainstays: YOLOv8 algorithm and the transformer-based algorithms tailored for computer vision tasks. While tailoring solutions for the dynamic and real-time requirements of table tennis training and actual competitions, the speed of real-time detection becomes paramount. This elevates inference time to a critical determi-nant in the algorithm selection process.

Among the contenders, YOLOv8 distinctively stands out. Not only is it more streamlined, but it also boasts a markedly rapid inference time, making it a preferable choice over many transformer-based algorithms. To provide a holistic understanding of YOLOv8's efficacy, this thesis embarks on a comparative analysis with DETR (End-to-End Object Detection with Transformers), a flagship representation of trans-former-based algorithms. This comparison underscores the relative advantages of YOLOv8, particularly spotlighting its superior performance in inference speed and proficiency in detecting smaller objects. Beyond its speed, the YOLOv8 algorithm is underpinned by a cutting-edge architectural design coupled with avant-garde training methodologies. These components synergize, enabling the algorithm to achieve heightened precision in localizing and recognizing diminutive objects within images, even in the most challenging scenarios. This positions YOLOv8 as not just an alternative, but potentially the future gold standard in object detection for real-time applications.

The solid color of table tennis can cause them to be mistaken for light sources. To address this issue, we've integrated a module that focuses on the ball's consistent motion patterns. This module employs back-ground subtraction to differentiate between static background and moving foreground elements, based on parameters detailed in this algorithm. This approach enhances density estimation, clustering similar data points together. Once the back-ground is removed from the video sequence, the ball's form and path are evident, aiding the YOLOv8 model in extracting visual cues and predicting outcomes.

High-speed cameras can potentially mitigate motion blur challenges faced with capturing swift table tennis. Still, YOLOv8 inference time struggles to match this camera's speed. To ensure proper detection, the training dataset must accommo-date various ball shapes, including those distorted by motion blur (Pan & Yan, 2018; Pan & Yan, 2020; Pan et al., 2021). Adjusting the camera's capture speed can provide a more diverse training dataset for the model.

Measuring the actual velocity of a table tennis in the real-world entails determining its three-dimensional path, emphasizing the importance of object distance (Luo et al., 2021; Luo et al., 2022). While LiDar can detect smaller, reflective objects like the glossy table tennis, inaccuracies can arise due to the laser's interaction with such surfaces. Camera calibration presents a more reliable method for determining the ball's depth across successive frames.

Properly pinpointing where the ball lands on the table mandates precise detection, especially within the table boundaries. Traditional evaluation techniques, such as comparing predicted bounding boxes with ground truth boxes, may not suffice. Thus, we propose a novel evaluation technique focusing on the ball's impact point on the table.

Measuring ball speed provides objective data for evaluating a player's bat power. It allows players to track their progress in increasing their bat speed over time. The valuable statistical data of landing spots in terms of player's bat placement, strategies, and bat consistency can be gathered for statistical analysis, that can be used to analyze player performance and make informed decisions on training and competition strategies (Lu et al., 2018).

With the help of a real-time video table tennis analysis system, the players can receive immediate feedback on the ball speed during practice sessions (Lu et al., 2017; Lu et al., 2020; Lu et al., 2021). This feedback can help them adjust their technique to generate more powerful bats or to achieve greater accuracy at different speeds. In addition, coaches and players can use the recorded data to identify weaknesses and strengths in their game. This information can be employed to tailor training programs and improve specific

aspects of a player's performance, such as accuracy and bat placement.

Understanding where opponents tend to place their bats can identify weaknesses in offense and defense and provide insights into their tactical preferences and strategies. This information can be mined by coaches and players to develop effective game plans for matches. For example, knowing the speed of an opponent's bats can help a player adjust their positioning and timing during a match. Ball speed can reflect which strokes are aggressive. Real-time tracking of landing spots can provide instant feedback to players and coaches during practice sessions. This immediate feedback can help players adjust their technique and make corrections to improve their bat placement. This explosive record can motivate players to improve their shooting ability, thereby improving their skills and game level. However, consistently high-intensity exertion is detrimental to the player's body, leading to excessive fatigue and even injury. Monitoring the speed of shots can help players avoid overexertion, coaches can use data to ensure players maintain a balanced training regimen. While using a pitching machine to assist in training, the ball speed can be adjusted according to the player's technical level, providing a customized training experience.

Concatenating the video analysis tools with landing spot data can be used in conjunction to replay specific points or bats. This allows for in-depth analysis of key moments in matches and helps players and coaches identify areas for improvement.

In professional table tennis live broadcasts, displaying ball speed data on-screen can enhance the viewing experience and provide viewers with real-time insights into the game,sharing landing spot data with fans and viewers can enhance their understanding of the game and make live broadcasts more engaging. Visualizations of batting placements can add excitement to the viewing experience. Ball speed data can be valuable for researchers and equipment manufacturers. It can be used to study trends in shot speed, playing styles, and the impact of equipment choices on the game.

This thesis systematically delves into literature reviews, methodologies, outcomes, discussions, and conclusions. It comprehensively covers model structures, experi-mental

strategies, and algorithm deployment.

1.2 Research Questions

In this thesis, we focus on integrating deep learning techniques for computing and analysis, achieving real-time video capture and information recording, and enhancing both accuracy and efficiency. Accordingly, the research questions we aim to answer are as follows:

(1) *What is the predictive performance of the improved algorithmic model on future (unseen) data?*

(2) *How can we calculate the speed and landing point of a table tennis ball to analyze athletes' performance levels effectively?*

The core premise of this project centers on determining the speed and landing locations of table tennis balls using a custom training dataset captured through real-time video. To achieve optimal results in object detection and information recording, a range of suitable techniques must be implemented. Additionally, the methodologies we've employed in this research project warrant evaluation. Prior to implementing a motion-based algorithm for video processing, data collection and augmentation are required to achieve more accurate results.

1.3 Contributions

The application of deep learning techniques for understanding table tennis scenes aids in evaluating players' performance during both training and competition, particularly with the challenges posed by small and high-speed moving objects. This research addresses the existing gap in table tennis as a subject for deep learning training, paving the way for real-time calculations of spatial relationships, velocity magnitudes, and directional movements within video feeds. In this thesis, we carry out the following:

(1) Collect and augment a custom training dataset;

(2) Utilize a subtraction algorithm for the preprocessing of input frames;

(3) Extract the results of detected bounding boxes to calculate three-dimensional spatial relationships, taking into account camera calibration;

(4) Finally, we evaluate the outcomes of table tennis object detection and assess the skill levels of athletes.

1.4 Objectives of This Thesis

Firstly, we propose a method for creating a custom training dataset specifically geared towards table tennis detection. We also compare the inference time of YOLOv8 and transformer-based deep learning models through experiments to determine which model is better suited for real-time scenarios. Additionally, a motion-based algorithm is integrated with the YOLOv8 model to enhance detection accuracy. Secondly, we employ computer vision techniques to transform from a two-dimensional frame to real-world space. This enables us to calculate and evaluate the speed and landing spots of table tennis balls by determining their positional relationship to the table surface, using the optimized and motion-based YOLOv8 algorithm. These results can then be used to assess the performance of athletes in both training and competitive table tennis settings.

1.5 Structure of This Thesis

The structure of this thesis is described as follows:

- In Chapter 2, we conduct a literature review and discuss the relevant studies in terms of behavior and technical indicators that receive attention in table tennis competitions. Meanwhile, we combine the development history of deep learning models and compare the achievements of important technologies, models, and methods for object detection using deep learning in many fields.
- In Chapter 3, we introduce research methods and experimental design, including custom datasets, model structures, and algorithms in this chapter.
- In Chapter 4, we present the collected data and research results obtained through the proposed algorithm through images. Limitations of the method are specified in detail.

- In Chapter 5, we summarize and analyze the experimental results.
- We draw the conclusion and state future work in Chapter 6.

Chapter 2
Literature Review

The focus of this thesis is on unfulfilled goals and expected research directions in previous research, this chapter will introduce massive approaches and the relevant knowledge of visual object detection using deep learning.

2.1 Introduction

The velocity that a flying ball in table tennis games is specific feature. This speed can even be described as a flash, generated by the players' explosive power of swinging the bat and hitting a ball. By considering the speed factors, the landing spot where a player hits the ball on the table is also an evaluation of playing skills.

2.2 Understanding of Table Tennis

The coordination of head, eye, and arm movements during forehand hitting in table tennis can reflect the proficiency of participants (Rodrigues, 2010). It seems that participants are able to adapt to the limitations imposed by early warning conditions by using shorter duration of quiet eyes, earlier shift of quiet eyes, and reduced arm speed during contact. Under the conditions of later prompts, modifications to gaze, head, and arm movements are not sufficient to maintain accuracy. In time limited, goal-oriented actions, there appears to be functional coupling between perception and action. These may be factors that affect ball speed and landing spot.

Fuchs et al (2018) provided a historical overview of table tennis game analysis, surveyed a detailed review of various game analysis methods, including performance indicators, simulation methods, momentum analysis, footwork analysis, and expert knowledge analysis. The game analysis for the Chinese and Japanese national teams was offered as two examples of "best practices". The impact of different matching analysis is summarized for future developments.

A study has discovered the importance of a table tennis ball landing spots in return strategies and attempted to manipulate competitive robots to specify and control the landing spots of the table tennis ball. A learning-based landing spot control method is proposed to minimize the error between the actual landing spot and the designated landing spot based on the method of specifying the expected landing spot based on the level of

competitiveness. The study only set the required landing spot at 1.6 meters and calculated the error between the required landing spot and the actual landing spot. The accuracy of the landing spot was evaluated by calculating the mathematical mean and standard deviation of the actual landing spot of the ball, in order to determine the effectiveness of this method in reducing errors. However, it did not explain how to obtain the location information of the landing spot (Li, 2015).

The same situation also existed (Ding, 2022), which discusses the challenge of learning high-speed and precise table tennis ball on physical robots. After comparing reinforcement learning and imitation learning methods for goal-oriented control in the real world, researchers have presented evidence that iterative imitation learning can be extended to the goal-oriented behavior of real robots in dynamic environments. A direct and scalable method is discovered to achieve continuous robot learning without the need for complex reward design, value function learning, or simulation to real transfer, and can train on physical robots within a few hours. The resulting strategy performs equally or better in the real world than amateur humans in the task of sending the ball back to a specific target on the table. An appropriate amount of unstructured demonstration data accelerates the convergence of the target, reflecting the impact of the initial undirected guidance dataset on performance. But there is still no mention of the method of tracking the landing spot.

In previous studies (Zhang, 2023), around 56 college students were randomly selected from four complete classes. Two teachers used PP and Skills Centered Instruction (SI) to analyze the impact of Teaching Test Game Practice (PP) teaching on table tennis. The experiment adopts a non-equivalent control/control group experimental design with pre and post measurements. Three separated ANOVA and repeated measurements (time effects) were conducted to examine the impact of PP and SI on each of the three dependent variables: (a) Forehand stroke accuracy, (b) forehand attack, and (c) serve. The results indicate that both PP and SI conditions can effectively improve participants' forehand hitting, forehand attacking, and serving skills from pre-test to post test. However,

compared to SI, PP is more effective in improving participants' forehand attack and serve skills. From this observation, it is obviously that if one can collect and analyze the performance data and find clear practice goals, it will have significant benefits for technical improvement.

2.3 Related Work

Darknet-53 is described as a more powerful network than Darknet-19. It is more efficient than ResNet-101 or ResNet-152. In terms of ImageNet results, the Top-1 accuracy of Darknet-53 is 77.2%, and the Top-5 accuracy is 93.8%. Its performance is comparable to the state-of-the-art classifiers, but it has fewer floating-point operations and faster speed. Darknet-53 is better and 1.5 times faster than ResNet-101. It has similar performance to ResNet-152 but is twice as fast. In addition, Darknet-53 achieved the highest measurement of floating-point operations per second, indicating that it better utilizes the GPU, evaluates more efficiently, and is therefore faster (Redmon & Farhadi, 2018).

Intentionally perturbing input data to make a model to generate incorrect predictions or classifications. These attacks exploit vulnerabilities in machine learning models, particularly deep neural networks, by adding small, carefully crafted perturbations to input data. Fast Gradient Sign Method (FGSM) method perturbs the input image by taking a small step in the direction of the gradient of the loss function with respect to the input. It is a one-step attack method.

Projected Gradient Descent (PGD) method is an iterative version of FGSM. It took multiple steps in the direction of the gradient and projects the perturbed image back into a valid range at each step. Deep fool method iteratively finds the smallest perturbation that can cause a misclassification by linearizing the decision boundary of the model. To improve the robustness of detection models, multitask learning can be used. By training the model on multiple related tasks simultaneously, the model can learn more robust and generalizable features. This can assist the model to better handle adversarial attacks (Zhang & Wang, 2019).

The impact of different training methods, including CutMix, Mosaic, Class label smoothing, Mish activation, and Cross mini–Batch Normalization (CmBN) have been proposed for improving the existing models (Bochkovskiy et al., 2020).

YOLO algorithm and its evolutionary version have been described (Viswanatha et al., 2020), which demonstrated the effectiveness of object detection without loss of accuracy compared to other models. The CNN-based architecture model can eliminate highlights in any given image and identify objects. The advantage of YOLO is that it is a one-step algorithm that directly predicts bounding boxes and class probabilities from complete images, making it faster and more efficient than other object detection algorithms.

While object detection is being adopted in intelligent system, accuracy and computational efficiency are crucial for real-time scene applications. Fast moving targets, such as detecting badminton, are challenging, and modifying the loss function can help improve the speed of small object detection (Qi et al., 2022; Qi et al., 2023; Zhang et al., 2022). If more semantic information of small objects can be retained, the network can achieve high-precision detection at the fastest speed (Cao et al, 2021).

A 3D CAD model that appears to be created by creating objects and the elements, imports the model into the rendering software, which can generate any number of high-resolution images in virtual frames. By utilizing different textures, geometric shapes, lighting effects, and camera views, realistic and diverse images can be synthesized. Additional data augmentation techniques can also be applied to make images more realistic and diverse. This method (Kapusi ea al., 2022) allows for the generation of datasets of any size without the need to create real images, which can solve the time-consuming and error prone problem of creating datasets.

Deep learning has greatly improved the accuracy of visual object detection while posing challenges in terms of high computational time (Yu & Yan, 2020; Yang & Yan, 2024). Improving the YOLO network by using small convolution operations to reduce

the number of parameters appears to shorten object detection time. It seems necessary to eliminate the influence of image background through image preprocessing before training with the YOLO model (Lu et al, 2019).

By combining the target detection results of RGB cameras and LiDAR using a weighted average scheme, it appears that mAP (mean Average Precision) can be improved to achieve the goal of improving detection performance. This method from Kim and Cho (2020) is robust to external environmental changes.

In the field of computer vision, drawing accurate bounding boxes around detected visual objects is a major task. YOLOv3 for visual object detection as the basic model, combined with edge detection and pixel values in the region, appears to improve the accuracy of the bounding box. The edges required for bounding box construction will be found on the side of the image and is not suitable for images with complex backgrounds or cluttered scenes (Blue and Brindha, 2019).

Transformers dominate natural language processing due to the ability to pre-training using a large amount of data for specific tasks. Directly applying transformer model to images appears to achieve competitive results in benchmark classification tasks. However, for more complex tasks such as detection or segmentation, it relies on high input resolution and large capacity training sets (Beal et al, 2020).

DETR (Sun et al., 2021) is a transformer-based visual object detection method that achieves the state-of-the-art performance but has a slow convergence speed. The difficulties of optimization methods are caused by the issue with the Hungarian loss and Transformer cross-attention mechanism.

Transformer pre-training model scaling strategy has limitations in vision. In contrast, YOLO models pre-trained on ImageNet-1k achieved competitive performance on the COCO object detection benchmark (Fang et al, 2021).

Due to advances in deep neural networks, computing power, and big data access, deep learning models have made significant progress in computer vision tasks (Zhang et al., 2020). Data augmentation is a method that improves the size and quality of training datasets to improve the performance of deep learning models, especially in areas with limited data. The effectiveness of data augmentation depends on specific tasks and datasets, it is important to consider factors such as label preservation, computational costs, and enhancing the interpretability of data. The image enhancement algorithms include geometric transformations, color space enhancement, kernel filters, mixed images, random erasure, feature space enhancement, adversarial training, generative adversarial networks (GANs), neural style transfer, and meta learning (Shorten and Khoshgoftaar, 2019).

Based on CIFAR-10 and other datasets, the results (Rebuffi et al., 2021) indicate that using data augmentation to improve the robustness of adversarial training can significantly update robust accuracy combined with model weight averaging.

An improved feature pyramid model AF-FPN is proposed, which utilizes adaptive attention and feature enhancement modules to enhance the representation ability of the feature pyramid. The original feature pyramid network in YOLOv5 has been replaced by AF-FPN, which improves the detection performance of multi-scale targets while ensuring real-time detection. A new automatic learning data augmentation method is also proposed to enrich the dataset and improve the robustness of the model (Wang et al, 2023).

Deep learning methods based on large amounts of data require a large number of annotated samples, which is impractical in real-world scenarios. Inspired by the ability of humans to learn quickly from a small number of samples, small sample learning has become an important research direction in deep learning. The methods of small sample learning include generative models based on probability reasoning and discriminative models based on meta learning. Multiscale meta relational networks take use of model independent meta learning algorithms to search for optimal parameters and combine the idea of meta learning with metric learning and initial representation optimization seems

to improve the generalization ability of learning measurements (Zheng et al., 2021).

Few-Shot Learning (FSL) is a machine learning method used to generalize from given examples in a limited sample dataset. The core method (Wang et al., 2020) is to minimize the unreliable empirical risk, which includes three aspects: data, models, and algorithms. A limited sample size cannot increase the training volume. The model limits the complexity of hypothesis and makes learning feasible. From an algorithmic perspective, prior knowledge changes the search strategy for the best hypothesis. Less lens learning is a low-cost solution that can reduce the need for large amounts of data (Parnami and Lee, 2020).

Visual object detection, as a challenging issue in computer vision, usually involves two types of detectors, namely, two-stage detectors and single-stage detectors. YOLO is a popular single-stage object detection algorithm that has faster inference time, but lower detection accuracy compared to two-stage detectors (Diwan et al, 2023).

YOLO is a virally and widely used algorithm for its object detection characteristics. YOLOv2, YOLOv3, YOLOv4, and YOLOv5 are subsequent versions of the YOLO algorithm, each with its own improvements and characteristics, and continuous improvements. The versions have differences in conceptual design and implementation, but similarities in structure and object detection methods (Jiang et al, 2022).

Computer vision is inspired by the complexity of human vision, exhibiting visual characteristics such as spatial perception, temporal perception, and brightness perception (Zhang, 2023).

The time the vehicle moves in the video can be directly measured by the image frame rate, but the actual unique information cannot be directly obtained and needs to be calibrated through the camera. This depends on the set model of camera imaging, which can reflect the mapping relationship between image coordinates and the actual three-dimensional road coordinate system, in order to obtain the actual displacement. This

understanding of the way that visual objects move follows Newton's second law. However, as movement of a table tennis ball, there is no corresponding guideline that can be used as a reference (Cheng et al, 2020).

A real-time detection and tracking solution for golf balls using convolutional neural networks (CNNs) and discrete Kalman filters (Zhang et al., 2020). Three classic CNN-based ball detection models (Faster R-CNN, YOLOv3, and YOLOv3 tiny) and a discrete Kalman filter were implemented to predict the position of the ball based on previous observations. Image patches are recommended instead of the entire image for detection to improve accuracy and speed. The difference from ball detection in table tennis is that the ball will rebound, the method of judging trajectory through the continuity of momentum is not feasible.

A few of computational methods for accurate speed measurement are proposed in a transportation system such as license plate surveys, CCTV video surveillance, Google traffic data, speed radar guns, and deep learning of vehicle speed. Data collection was conducted on seven vehicle categories at five locations in the Colombo district. This study had statistical analysis, including RMSE, MAE, and correlation, to compare the speed of each detection technique. The results show that deep learning technology ranks the first in accuracy and the second in Google Data. Radar gun speed method is somewhere in between. However, deep learning models require further being trained for a variety of vehicle categories. Research outcomes show that deep learning is a sustainable speed detection solution (Herath et al, 2021).

A new method was proposed for camera calibration in machine vision using ready-made television cameras and lenses (Tsai, 1987). This method includes a two-stage process to calculate the external position and direction of the camera, as well as focal length, lens distortion, and image scanning parameters. This method has advantages in accuracy, speed, and versatility. This theoretical framework is believed to be capable of real-time calibration with minor modifications.

The basic concepts of computer vision has been addressed in a book (Forsyth & Ponce, 2003), including computer vision professionals, computational geometry experts, computer graphics practitioners, and students interested in vision, all can understand the basic geometry and physics of imaging. It covers various topics, such as camera models, light and shadows, colors, linear filters, local image features, textures, stereo vision, motion structures, segmentation, tracking, object recognition, image-based modeling, etc.

The purpose of stereo camera calibration is to estimate the parameters of each camera both internally and externally. By using these parameters, the scene can be recognized and matched in two stereo images through using triangulation method (Weng et al, 1992).

Motion blur is usually considered as unwanted artifacts in an image which is usually removed before further processing. However, motion blur can be used as a means of estimating vehicle speed, rather than using radar or LiDAR equipment (Lin, 2005). It seems that motion blur parameters can be estimated based on a single motion blur image, and this length is employed for image restoration and can establish a link between the motion blur information of 2D images and the velocity information of moving objects. The restored image can be used to obtain other parameters for vehicle speed estimation.

Compared to video-based speed estimation methods, the error is smaller (Zhu et al., 2022). Traditional speed measurement methods take use of radar or laser-based devices, which are much expensive compared to passive camera systems. This method utilizes a single image captured by using vehicle motion to measure the speed. The motion blur in the image provides a visual basis for measuring the speed of moving objects. The entire process involves segmenting the target area, estimating blur parameters, deblurring the image, and exporting other parameters to calculate vehicle speed. The results indicate that there is an error of less than 5% between the calculated speed and the actual speed.

Programmable Graphics Processing Unit (GPU) is utilized for real-time image processing in computer vision. NVIDIA's CUDA architecture, as a computational resource for accelerating image processing algorithms, demonstrated the effectiveness of

their method by parallelizing and optimizing Canny's edge detection algorithm, and applying it to a video motion tracking algorithm called Vector Coherent Mapping (VCM). The results show that using GPUs in dense computer vision can significantly improve performance. This fully reflects the advantages of GPU's high computing power and low cost (Park et al, 2008)

The design and implementation of high-speed background subtraction algorithms (Hanchinamani et al., 2016) make it possible to detect moving objects. The algorithm includes converting the video into a stream, applying convolutional filters to remove noise, and using adaptive thresholds to detect moving objects. The results indicate that the proposed method has a number of advantages in image quality and computational speed, and provides a more efficient and accurate method for moving object detection (Gowdra et al., 2021).

The event camera is equipped with a different type of biologically inspired sensor from traditional cameras, known as a neuromorphic camera or event-based camera (Herrera et al., 2008). Unlike traditional cameras that capture images at fixed intervals, event cameras work based on the concept of asynchronous event driven sensing. By simulating certain aspects of human vision, it can exert unique advantages in special scenes, especially under fast motion and constantly changing lighting conditions. Using event cameras for high-speed moving object detection has advantages over traditional frame-based cameras in terms of high temporal resolution, high dynamic range, and minimal motion blur. Combining event cameras with traditional detection algorithms appears to be able to detect high-speed moving objects and reduce motion blur and data redundancy (Zhang et al, 2022).

The research work published in sports journals since 1980 has offered a strong foundation for the analysis of table tennis matches. The three stage evaluation method on different themes have played a positive role in the understanding of coaches and athletes in table tennis matches. It seems that computer-aided game analysis can improve the ability and speed of data processing, promote the tactical features through video feedback

and multimedia demonstrations. However, matching analysis based on different theories or models is still in the preliminary stage (Zhang et al, 2018).

An interactive visualization system was designed (Wu et al., 2018) to analyze and explore table tennis data. The system provides overall visualization of the entire game from three main perspectives: time orientation, statistics, and tactical analysis. The competition view includes a customized step chart and a score result bar to display score evolution and rebound information. The stroke view displays detailed stroke attributes on the table tennis table. The statistical view displays the correlation between stroke attributes within a stroke and between adjacent strokes. The tactical view allows for the detection of frequent tactical patterns (Zhu & Yan, 2022). The design of this system was guided by domain experts, starting from the characterization of table tennis analysis domain problems, and evaluated through case studies.

A neural network called TTNet, which can process high-resolution videos and provide temporal (event discovery) and spatial (ball detection and semantic segmentation) data, was once mentioned by Voeikov et al (2020). The network was trained on a multitasking dataset called OpenTTGames, which includes table tennis game videos labeled with events, semantic segmentation masks, and ball coordinates. It has achieved high accuracy in event discovery and ball detection, but the inference time for each input tensor on a single GPU is only around 6ms. More valuable information may lie in its automated data collection, support for referee decision-making, and provision of additional information about the competition process. Unfortunately, this study is based on OpenTTGames and has not shown any differences in data between this game and real-world table tennis matches.

The importance of accurate detection algorithms in the sports industry, especially in the table tennis industry, has been emphasized (Qiao, 2021). Its value lies in providing precise tactical analysis, improving training and performance evaluation. At the same time, it also affirms the challenges of rapid movement and the necessity of accurately capturing rotational information in table tennis. It is necessary to acknowledge the

limitations of traditional recognition and detection algorithms in accurately detecting fast moving and rotating balls in complex environments and clarify the necessity of real-time monitoring.

At that time, DCNN was employed for visual object tracking (Li et al., 2016), LSTM algorithm was used to predict the trajectory of the ball, and deep reinforcement network became the main method for extracting real-time motion features. However, the ball of table tennis is tiny in size, the pixel information available in the images might not be sufficient to distinguish the ball from the background of similar color. These factors may impact the accuracy of ball detection and tracking, especially in challenging lighting conditions. The proposed DCNN-LSTM model shows promising results, but it is worth noting that its performance was tested on a self-built video dataset without providing detailed information on the dataset, evaluation metrics, and potential biases. The robustness and universality of the model under different scenarios and conditions also seem uncertain.

The longest baseline scheme is a time of arrival (TDOA) localization method which can be applied in table tennis ball localization estimation (Jian and Hong, 2017). However, the disadvantage of the time difference localization method is that it requires at least three sensors to accurately estimate the position of two-dimensional targets. The limitation is that the accuracy of the TDOA method may be affected by using factors such as sensor position error and time difference estimation error, which may introduce errors in position estimation. Compared to this method, computer vision only requires one to two digital cameras to complete the project and make it more cost-effective.

Two detection methods YOLO and Mask R-CNN were proposed to implement ball detection tasks in handball scenes (Buric et al., 2018). A dataset containing custom handball lenses and Internet images has been created to evaluate the speed and accuracy of the model. In addition, the impact of additional training on model performance was also investigated. The results showed that YOLO and Mask R-CNN performed poorly on custom datasets, but significantly improved after additional training. YOLO models

outperform Mask R-CNN in speed, while Mask R-CNN provides more accurate object detection. Cross frame tracking of detected balls and the use of motion information for further improvement may also be a direction for future research.

Data preprocessing is an important stage of data analysis activities, which involves constructing the final dataset to provide it to deep learning algorithms. The preprocessing (Ashraf et al., 2022) in the model is to adjust the image to two variants, and the resolution for YOLOv5 input is 416 × 416 because it only accepts 32 and 240 CNN models × 240.

The second preprocessing based on images is to blur or remove the background in the image using different algorithms. In this model, it is a Gaussian blur operation, and this preprocessing are selected instead of any other preprocessing because compared to other techniques such as median filters, its speed requires sorting and reduces the computational speed. Gaussian filter is a low-pass filter that removes high-frequency components, and the pixels closest to the kernel center are given greater weight than those far from the kernel center. The intention of proposing this framework is to minimize false negatives and false positives in weapon detection while maintaining real-time detection speed. Performance indicators such as accuracy, recall rate, and F1 score are employed for evaluation. However, YOLOv8 algorithm can directly predict the center point coordinates of the bounding box of the target through anchor-free. Thereby, there should be no need to add filters for data preprocessing.

A framework (Peng et al., 2016) for detecting pedestrians near substations takes use of a combination of GMM (Gaussian Mixture Model) and YOLO (You Only Look Once) methods. GMM is employed to model the background and perform preliminary pedestrian detection, while the method YOLO based on convolutional neural networks (CNN) is also utilized for pedestrian detection. Combining these two results with different weights will obtain better detection results. It seems that the detection rate of this method has increased by 20% compared to a single method. However, the accuracy of YOLO and GMM is relatively low, only 70.9% and 68.3%, respectively.

Li et al (2003) proposed a new method for detecting foreground objects from videos containing complex backgrounds. This method takes use of Bayesian decision rules to classify the background and foreground based on the selected feature vectors, furthermore, it classifies different types of background objects by selecting appropriate feature vectors. Static background objects are described by color features, while moving background objects are represented by color co-occurrence features. Extracting foreground objects by fusing classification results from stationary and moving pixels. This method seems to be able to adapt to learning strategies of background gradient and mutation, and can extract foreground objects from complex backgrounds and achieve good results. The premise of using Bayesian decision rules to classify pixels as foreground or background is to establish statistical models for the pixel values of the foreground (table tennis player and ball) and background (playing field). From this perspective, it is essential to use probability density functions (PDF) to model the distribution of these pixel values.

Through using a neural network or model for making predictions or inferences on new, unseen data (inference images), the quality or accuracy of those predictions tends to be higher when the model has been trained on a dataset that is very similar or identical to the dataset from which the inference images are drawn. It learns patterns, relationships, and features from a training dataset when a neural network or machine learning model is trained. This training involves adjusting internal parameters of the model to minimize prediction errors or optimize a specific objective. After the model is trained, it can be harnessed to make predictions on new data, referred to as inference or test data. The predictions are much accurate with higher precision, or exhibit superior performance metrics when the model was trained on a dataset that closely resembles the dataset from which the inference images are taken (Takano & Alaghband, 2019).

Zhu et al (2016) and Liang et al (2022) investigated whether existing object recognition systems will continue to improve with the growth of training data, or whether their performance will be saturated. The focus is on defining differentiated training templates on directional gradient features. As the number of mixed components and the

amount of training data increase, the performance of template mixtures is investigated. The results show that classic mixture models saturate quickly, but compositional mixtures that share template parameters via parts can synthesize new templates which is not encountered during training, and yield significantly better performance. The maximum benefits of detection performance probably come from improved representation and learning algorithms that can effectively utilize large datasets. There are astonishing subtleties in expanding the training dataset. For fixed models, it is expected that performance typically increases with increasing data volume and ultimately saturates. There is a weird result from empirical perspective, that ready-made implementations show a decrease in performance with additional data increasing. Different object subcategories and viewpoints are expected to be captured by adding hybrid components in order to utilize additional training data. Even for non-parametric models that grow with the increase of training data, there is a problem of diminishing returns in performance with only a small amount of training data.

The table tennis is a sport (Akramjonovich et al., 2022) where a small racket is employed to return a ball to hit the table. Easy and fast movements, agile attack, and focused defense are the characteristics of this sport. Flexibility, speed, and endurance are all tests of athlete muscle strength. Speed is crucial for attacking and improving the pace of the game, as it depends on factors such as joint flexibility, muscle strength, and the flexibility of the nerve center. Endurance plays an important role in table tennis, as athletes with good skills but insufficient endurance may lose accuracy, attention, and normal breathing during matches. Agile is crucial for effectively executing various actions and techniques, requiring economical and agile actions.

The tactics of table tennis evolve over time, and tactical planning should consider the opponent's skills and weaknesses, aiming to leverage their own strengths and weaken the enemy. Special imitation exercises, such as using bicycle wheels or swinging pears, and skills such as “topspin” strokes, can help develop. Observing the opponent's game and analyzing their playing style can help develop tactical game plans. These conditions and

techniques seem to be inseparable from the judgment of the state of table tennis ball as the research object.

The fundamental transformation of software engineering is referred to as Software 2.0 (Whang and Lee, 2020). With the availability of big data and computing infrastructure, the use of machine learning has become a new norm for software. Therefore, a slew of software engineering practices require rethinking from scratch, and data, like code, becomes a first-class citizen. 80-90% of the time spent on machine learning development is spent on data preparation.

In addition, even the best machine learning algorithms cannot perform well without good data or at least handling biased and dirty data during model training. Data collection and quality are common challenges in deep learning applications. There are three main methods for data collection. (1) Data collection is the problem of finding suitable datasets for training models. (2) Supervised learning cannot do without data labeling, therefore, various scalable technologies such as semi supervised learning, crowdsourcing, and weak supervision are also ways to solve the problem of high data labeling costs. (3) Starting from scratch seems to be the lowest level strategy, and using transfer learning to reuse existing models or improve the quality of existing data is a better solution.

Compared to traditional machine learning, there is less demand for feature engineering, but it requires more data. Data validation and cleaning techniques that improve data quality are considered the most advanced data collection techniques in machine learning. Even if there are still issues with the data, hope is not lost, and fair and robust training techniques can be used to handle data biases and errors.

Trajectory similarity calculation is a research hotspot in spatial databases. The existing trajectory complementarity methods have limitations due to only considering spatial and temporal features. Activity trajectory data provides additional semantic information, which can improve trajectory complementarity. Li et al (2018) proposed the At2vec framework to combine spatiotemporal features with activity information to

generate robust trajectory representations. Representation learning, also known as feature learning or representation discovery, is a crucial concept in machine learning and deep learning. It refers to the process of learning meaningful and informative representations of data from raw input.

These learned representations capture relevant features or characteristics of the data, making it easier for machine learning models to perform tasks like classification, clustering, and prediction. Representation learning automates the process of feature extraction by allowing models to learn useful features directly from the data. Representation learning often involves creating a hierarchy of representations, where each level abstracts and refines the features from the previous level.

Deep learning architectures like neural networks are particularly well-suited for learning hierarchical representations. Representation learning can be supervised, unsupervised, or self-supervised. In unsupervised learning, the model learns representations without explicit labels. Self-supervised learning is a type of unsupervised learning where the model generates labels from the data itself. For example, predicting a rotated version of an image can be a self-supervised task. Once meaningful representations are learned from one task or dataset, which can be transferred and fine-tuned for other related tasks. This is known as transfer learning and is a powerful technique for leveraging pre-trained models and limited labeled data.

Network representation learning is a learning paradigm aimed at embedding network vertices into low dimensional vector space while preserving network topology, vertex content, and other auxiliary information. The large-scale information network makes network analysis tasks computationally expensive or difficult to handle. Embedding large-scale networks into new vector spaces can facilitate the analysis of large-scale networks.

A survey initiated by Zhang et al (2018) classified and summarized the most advanced network representation learning technologies based on learning mechanisms,

retained network information, and algorithm design, such as specific task NRL algorithms for link prediction, community detection based imbalanced learning, active learning, and information retrieval. Using network representation learning as an intermediate layer to solve the target task is to store as much information as possible in the new representation, which can further benefit subsequent tasks. Therefore, the expected task specific NRL algorithm must retain information crucial to the specific task in order to optimize its performance.

An effective data augmentation scheme (Rajagopalan, 2023) was proposed for segmenting motion blurred regions without deblurring. Utilizing segmentation annotations can generate synthesized spatial variation blurs based on the CCMBA (Class Centered Motion Blur Enhancement) strategy. This method allows the network to simultaneously learn the semantic segmentation of clean images, self-motion blurred images, and dynamic scene blurred images. It is suitable for the universality of CNN and Vision Transformer semantic segmentation networks and is considered to perform better than baseline methods on datasets such as PASCAL VOC, Cityscapes, GoPro, and REDS.

Compared with aerial images, underwater images have lower accuracy when using deep convolutional neural networks (CNNs) for classification due to their uniqueness. Therefore, data augmentation is needed to improve classification capabilities. The optical conversion of raw data, such as proportion and aspect ratio enhancement and color enhancement, and the use of generative adversarial networks (GANs) to generate additional data are two data augmentation methods mentioned by Xu et al. (2017). These methods (Howard, 2013) include adding more image transformations to training data, adding more transformations during testing, and using complementary models applied to higher resolution images, all inspired by the entries in the 2013 Imagenet Large Scale Visual Recognition Challenge. This challenge achieved a classification error rate of 13.55% in the top five without external data, a relative improvement of 20% compared to the previous year's winners.

In addition, data conversion includes extending image cropping to additional pixels

and adding additional color operations, attach predictions from multiple scales and views to the data conversion used for testing, use a simple greedy algorithm to reduce the number of predictions to a manageable size. A higher resolution model is a supplement to the basic model and can be used to search for objects in enlarged images. In previous challenges, there were several types of image transformations to enhance the training set. Randomly extracting 224 ×224 pixels from 256×256 pixel images and capturing some translation variance were the first methods; Horizontal flipping of images is also a feasible method for capturing reflection invariance and adding randomly generated lighting in an attempt to capture the invariance of lighting changes and small color changes with additional transformations to extend translation invariance and color invariance.

Dropout is a technique used to prevent overfitting in neural networks. When the network performs well on training data but poorly on test data, overfitting occurs. It prevents complex collaborative adaptation of feature detectors by randomly omitting half of them in each training case. This forces each neuron to learn a feature that typically helps generate the correct answer in the various internal environments in which it must operate. Dropout has been proven to improve the performance of benchmark tasks such as speech and object recognition. The implementation of discarding includes using constraints on the L_2 norm of the input weight vector for each hidden unit, as well as using an average network during testing. Its effectiveness has been demonstrated in various benchmark tasks, including MNIST, TIMIT, CIFAR-10, and ImageNet. Adjusting dropout probabilities based on input is a recognized potential optimization and improvement method (Hinton et al, 2012).

Raschka (2018) summarized the crucial methods for model evaluation, model selection, and algorithm selection in machine learning research and applications. The combination of F-test and nested cross validation is more suitable for small datasets than the Holdout method. Bootstrap technology can be used to estimate performance uncertainty. Cross validation, such as omission and *k*-folding is more practical evaluation methods. While selecting models, the relative performance of these models is crucial, as

long as these models are compared under the same optimistic or pessimistic conditions, even if the overall value is underestimated or overestimated, it is not difficult to discover the optimal model from them.

Centroid tracking methods refer to a simple representation of a tracked object using its centroid, which is the mean of the segmented object's points in a 3D point cloud. These methods involve predicting the future location of the object based on a motion model and associating new observations with existing tracks. The centroid of the new observation is used to update the tracked position or adjust the Kalman Filter. Centroid tracking methods are often used as a benchmark for more complex tracking algorithms and can provide good velocity estimates. Morton et al (2011) designed an experiment, compared to technologies based on 3D appearance models, centroid tracking has been found to have more advantages. Pedestrian tracking experiments have shown that centroid tracking achieves up to 95% correct data association, while the method based on appearance models achieves 60%.

Compared with image-based analysis, LiDAR sensors provide precise depth information, allowing for accurate 3D measurements of objects and environments. This depth perception is not affected by lighting conditions or color variations. It is not affected by changes in lighting conditions, making them suitable for both indoor and outdoor applications. Based on the 3D point cloud data, it makes objects detection and tracking useful for applications such as autonomous driving, surveillance systems, and robotics. LiDAR sensors capture only distance values in a 3D space, ensuring that individuals' details cannot be easily identified, thus preserving privacy.

However, LiDAR sensors are generally more expensive compared to traditional cameras, which can limit their widespread adoption in some applications. Typically, the limited field of view may lead to infeasibility to capture the entire environment or all objects within the scene simultaneously. Processing and analyzing LiDAR data can be computationally intensive, requiring specialized algorithms and hardware to handle large point cloud datasets. It is different from image-based analysis, LiDAR sensors do not

capture color or texture information, which can be valuable for certain applications that rely on visual appearance (Hasan et al., 2022).

Liu and Ma (2021) proposed to improve the YOLOv4 small object detection model by integrating a dual head architecture A2-YOLO, effective channel attention ECA-Net, ALL-CNN, and Mish activation, while reducing the number of parameters and FLOPs. The results showed that on the MS COCO 2017 dataset, A2-YOLO outperformed the original YOLOv4 tiny, with an increase in AP50 of 3.3% and a decrease in model parameters of 7.26%. The experiment of improving YOLO Tiny has demonstrated the effectiveness of integrated technology in improving detection results.

Computer-aided detection (CAD) systems have limitations in real-time detection, as well as limited sensitivity and specificity. YOLOv5 object detection algorithm and artificial bee colony (ABC) optimization algorithm are employed to improve the performance of polyp detection models (Karaman et al., 2023). The ABC algorithm is a group based global optimization algorithm that shows higher accuracy than the original YOLOv5 algorithm after optimizing the activation function and hyperparameters of the YOLOv5 algorithm. The pre-trained COCO weights are employed to train the YOLOv5 algorithm by using default activation functions and hyperparameters, which are saved as the optimal model. On this basis, the ABC algorithm is applied to fine tune the model for several periods, and the process is repeated until meet the termination criteria.

A prior study (Tian et al., 2020) highlighted the formidable challenge of ball detection within the realm of computer vision, attributed to diminutive size and swift motion of the balls, even YOLOv8 model struggles with a variation of aspect ratio and accurately detects balls due to fast motion. While an anchor-free approach was proposed to counter this issue during the evolution of YOLO models. This challenge in detecting such balls is still difficult as the state-of-the-art algorithm. Nonetheless, the limitations may not only stem from false positives by using anchor-free algorithms, but the small size of sports balls could also cause to a significant influence.

A valuable reference is dedicated to track moving or airborne objects. In 2022, a method employing LSTM in deep learning and simple physical motion models corrected deviations, through establishing a binocular vision-based trajectory extraction system for table tennis that relies on digital cameras (Cai, 2022). The visual feature extraction was completed by using MobileNet and SSD models, a compromise between resource-constrained environments and accuracy. Nevertheless, it falls compared to the pyramid feature network in YOLOv8 architecture, particularly for challenging datasets and small visual objects.

After reviewed the video footages of 2017 Summer Universiade Men's Singles Final, it is found that persisting in achieving precise recognition and positioning of high-speed, mini balls is considered a challenge (Huang et al, 2019). The TrackNet model, built upon deep learning, can identify balls from single frames with blurred images and lingering trails, even unable to be seen from a visual perspective. However, the performance of Track-Net model heavily hinges on the training data it encounters, potentially faltering if exposed to visual objects or environments deviating significantly from the training data.

In a previous study (Blank et al., 2017), eight participants with different types of hitting balls were involved. Inertial sensors installed on the racket were used to estimate the speed and rotation of a table tennis ball, and high-speed cameras were used to record the impact of the racket for evaluation. The method includes assuming and simplifying the properties of the initial ball and the motion of the player and racket. The speed of the racket blades after impact was calculated, and a rebound model between the ball and rubber was used to predict the speed and rotation of the ball.

The results show that the accuracy of ball speed estimation for forward and backward spin shots is only 79.4% and 87.4%, respectively, while the accuracy of rotation estimation is 73.5% and 75.0%. This accuracy may not seem sufficient, but from this experiment, it illustrates that when hitting table with the ball, it is influenced by various variables such as the speed of the ball as it flies, the speed of the initial pass, the rotation of the ball itself, and the elastic model of the racket rubber. These are important reasons

that affect athletes' batting performance.

VAR (i.e., Video Assistant Referee) was available in the 2018 FIFA World Cup, volleyball matches, and fencing competitions. Conversely, it has not been applicable in table tennis competitions due to the exceptional speeds of balls up to 112.5 kilometers per hour (Moshayedi et al, 2019). As a widely participated sport with 800 million table tennis players globally, it laid the foundation for popularity ranking at the Olympics. Tracking and detecting the table tennis balls are anything but routine. Employing VAR introduces the risk of misjudgment constrained by the ball's incredible speed.

Apart from overcoming the challenges associated with tracking and detecting the table tennis balls, the involved issues are related to the relationship between training datasets and accuracy enhancement. A group of models struggle to achieve the officially announced accuracy, with actual detection results falling short of expectations. Fast or erratic object motion causes motion blur, making it difficult to comprehensively cover training datasets and assess detection outcomes.

An end-to-end BFAN (i.e., Blur-aid Feature Aggregation Network) for visual object detection has been proposed (Wu et al., 2020). However, the application of this approach seems unsuitable for table tennis due to its requirement for multiscale feature training datasets. Deblurring (Shi et al, 2014) may restore clarity to the balls in consecutive frames, yet distinguishing blurred foreground from background poses a significant challenge.

Optimizing the predicted bounding box scale might offer a solution. This entails learning scale features from minimal samples, as demonstrated by using MSNN (i.e., MultiScale Meta-relational Network) (Zheng et al, 2012). MSNN enhances the generalization capability of the proposed model for measurements and improving classification accuracy without necessitating model-independent meta-learning algorithms. While the omniglot dataset yielded positive results, further research work was required to fine-tune metalearning methods for improving the performance on other datasets.

In order to calculate the speed of a table tennis ball using computer vision, it is necessary to find the depth of the scene of table tennis in the image. The movement of a table tennis ball may be perpendicular to camera lens, which requires at least two fixed cameras to synchronously record from different angles to avoid the ball in table tennis being considered as not moving during two consecutive frames. A few of camera APIs provide timestamp information for captured video frames satisfying stereo vision. The frames from different cameras can be aligned using these timestamps. This approach might require prudently handling of the timestamps and proper synchronization logic.

A stereo camera installed on a robot has been studied (Zhan et al., 2007) for tracking the table tennis balls after being synchronized. It explores a method that captures and processes stereo images of the ball motion and analyses the disparities between corresponding points in the stereo images, that can determine the ball's 3D position in space. This method focuses on image synthesis and processing after asynchronous cameras capture images, even if only one video frame rate is known, it can be processed. However, this method increases the processing interval for each frame, which seems to significantly increase the detection time of motion-based YOLOv8 algorithm.

In this experiment, replacing the image information captured by the main camera with the image information captured by the auxiliary camera only occurs when the a table tennis ball captured in two consecutive frames in the main camera are in the same position. In other words, this calculation method only changes the data source from the main camera to an auxiliary camera, without increasing the computational workload under limited computing resources.

In summary, diversifying training dataset scales, deblurring the table tennis balls affected by motion blur, and employing the multiscale meta-relational network appear as viable avenues for investigation. The focus of this thesis is on dataset scale diversity while deblurring methods will be explored in subsequent research endeavors.

Chapter 3
Methodology

The main content of this chapter is to clearly articulate research methods, which satisfy the objectives of this thesis. These methods fully integrate the characteristics and technical requirements of table tennis itself, and process and operate real-time videos through model of deep learning in possible environments.

3.1 Customed Training Dataset

The training of YOLOv8 model is a supervised learning process, the primary objective is to learn a mapping or relationship between input data (features) and corresponding target labels or outputs. The algorithm aims to make predictions or classifications based on this learned relationship. It was employed for tasks like classification and regression. The scene of table tennis is specific with a moving small ball that is different from the pre-trained datasets. Thus, a customed training dataset needs to be tailored resulting in potentially better model performance. However, it requires effort in terms of data collection, annotation, and quality control. Fortunately, a huge number of parameters can be utilized from pre-trained models through transfer learning. A great number of factors will affect establishing a customed training dataset that needs to be addressed. The approach how to collect enough data is the first challenge.

According to actual scene of table tennis competitions and training, the table occupying the entire width of the video frame seems to be the optimal position with the appropriate angle, which can maximize the size of table as the target detected in the frame without missing any landing spots on the table. This is also conducive for prediction using YOLOv8 models. The real-time video footages captured under these conditions can serve as the main source of images in the training dataset. In addition, the scale of data needs to be enriched through methods based on computer vision to improve data diversity, such as random resizing. The shape of balls in table tennis is simulated at different depths using the frames by setting random scale factors.

On the other hand, the fast-moving object leads to motion blur after captured by a camera, even if the camera has 120 Hz plus the fastest inference time. It is easier to obtain the ball shape under this deformation by using a low-speed camera to capture images. Meanwhile, this type of images with motion blur also requires a randomly resizing. Finally, the balls in table tennis games with various textures and colors need to be used for sampling and recording.

3.2 Comparison between YOLOv8 and DETR Model

YOLO seems to be an algorithm born for real-time detection, with its related publications on Google Scholar reaching 1200 in 2015. In Fig 3.1, 10 shows the release dates of each version of the YOLO algorithm. The initial mAP of YOLOv1 algorithm was not as good as Fast R-CNN and later SSDs. However, with continuous optimization and improvement, the number of research work related to the new version of YOLO algorithm of that year has maintained a high level.

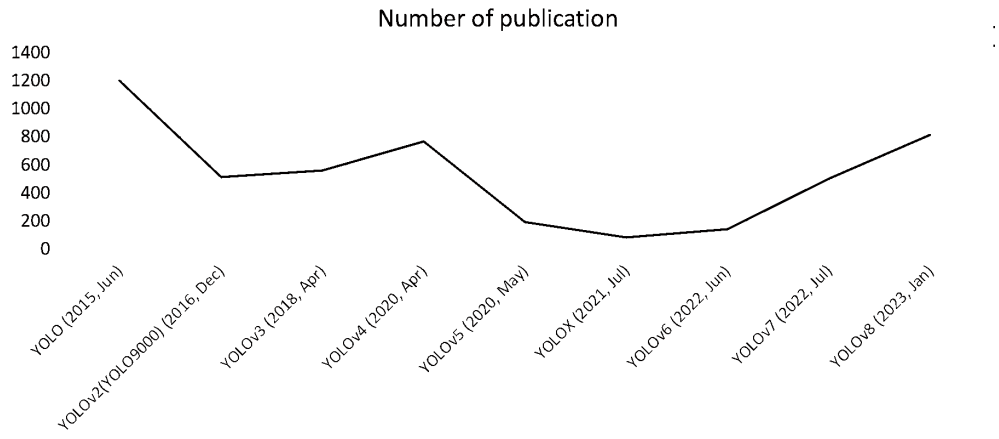


Figure 3.1. Number of publications related to YOLO models

Convolutional neural networks (CNN) are the core of the YOLO algorithm, which can understand the spatial layout of inputs and process them in relative terms. CNN sequentially traverses the learning of surrounding pixels starting from the central pixel rather than learning every pixel globally. Then, it looks at different parts of the image while sliding, searching for the same pattern in each region that is relatively centered. The difference between it and fully connected networks lies in weight sharing and locality. The fixed center position calculation method is applied to each position, and each calculation can only look at things that are quite close to the center position. For example, in the YOLOv8 algorithm, the image is divided into a x a grids, and a 3x3 pixel window is used to scan and extract features from the image by stride 2 with operation of 3x3 convolution. Each object is defined by its smaller features, and it does not need to analyze

whether the current target is a table tennis ball by observing things other than a ball.

As a result of this characteristic of CNN that text processing cannot use lead to the appearance of Transformer model based on recurrent neural network (RNN), which was first proposed by the Google team in 2017. The intricate connections between words in a sentence, coupled with the limited storage capacity of Scratchpad, may lead to RNN distraction even ambiguity. The attention mechanism compensates for this deficiency by pairing and comparing in the complex network of relationships formed between sentences and words, searching for matching words and calculating them separately, constructing connections for them and combining information from this position with information from other position, while also ignoring most irrelevant information globally. Different from CNN, this information does not need to be around the corner means not local. This method used for Natural Language Processing (NLP) also performs well in object detection; Thus, it is needed to determine which method will be used for the experiments of the table tennis ball detection.

Swin transformer (Liu et al., 2021) is a hybrid architecture which is good at large-scale image classification tasks that are efficient by using hierarchical windows and local self-attention mechanisms. Contrast with Swing transformers, DETR proposed by Carion (2020) is much versatile and efficient, the primary focus of DETR is on visual object detection.

The reason why DETR is selected instead of Deformable DETR with slightly higher accuracy due to inference time that is faster and more conducive to real-time object detection. Data on the comparison between AP and investment time between DETR and Deformable DETR can be found in the research publication of Zhou et al (2023). A bipartite matching loss is deployed for a set of prediction tasks for visual object detection. As a result of eliminating the need for anchor-based methods, object classes and locations are directly predicted in a single forward pass.

Thus, DETR is selected to determine which one is much suitable for this experiment

that will be conducted in a Google Collab virtual environment equipped with a A100 GPU (graphics processing unit) to satisfy the basic requirements of real-time object detection.

After a real-time video was processed and ball position is predicted by using YOLOv8 model, it is obvious to see that two textured regions in the background are recognized as balls in Figure 3.2. In this scenario, it is almost impossible to calculate the speed and landing spots of a ball. The detection results with these errors cannot be filtered by using shapes and colors, even texture. The correct detection of a real ball has become the key to visual object detection.

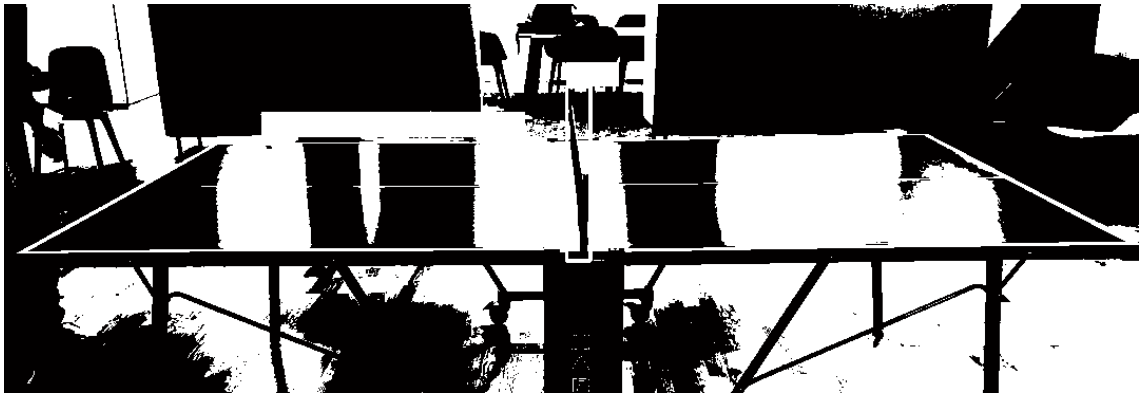


Figure 3.2. Original video is adopted as input of YOLOv8s model

Real-time recordings obtained from table tennis training and gaming contain abundant light spots and reflective patches in the background. MoG (Mixture of Gaussians) is employed for background subtraction to remove light spots and reflective patches in the pre-processing stage before the video as input to be predicted by using YOLOv8 model.

3.3 Weight Calculation and Backpropagation

In the architecture of CNN, weight calculation primarily occurs within convolutional layers. Each neuron as a filter in a convolutional layer has a set of learnable weights and a bias term. These weights are shared across all the input patches and subregions of the previous layer. The convolution operation involves sliding the neuron over the feature

maps and computing the weighted sum of the values within the receptive field. The weighted sum, along with the bias term, is passed through an activation function, such as sigmoid, to produce the output of the neuron. During training, the network learns the optimal values for these weights and biases using an optimization algorithm like stochastic gradient descent (SGD) or one of its variants. The goal is to minimize a loss function that measures the difference between the predicted output and the actual target values. Taking texture feature extraction as an example:

Extracting texture features from images can be various statistical or descriptor representations of textures, such as mean, variance, entropy, etc. Assuming two features including the mean of each image μ and variance σ^2 are extracted. In Figure 3.3, a simple neural network defined for texture classification consists of two input layers of neurons for mean and variance feature extraction, a hidden layer containing several neurons, and an output layer containing as many neurons as the category (e.g., three neurons for wood, metal, and fabric).

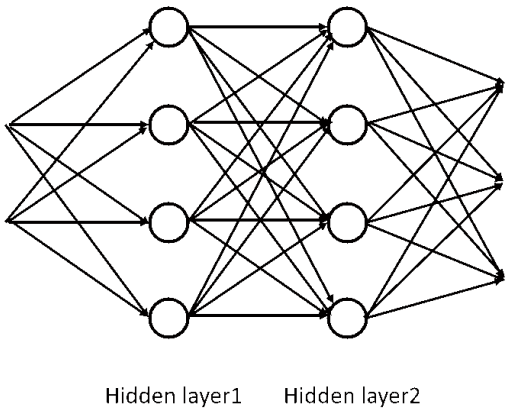


Figure 3.3. Propagation of weights

The mean and variance of the images as input to the input layer of the neural network are weighted and activated. Each neuron in the hidden layer calculates the weighted sum of its inputs and passes it to the activation function, and the same process occurs in the output layer. Initially, the weights of connections between neurons were randomly

initialized. Perform forward propagation on it to predict the given input features. Compare these predictions with real category labels to calculate losses. The gradients from loss with respect to the weights using the chain rule of calculus and backpropagation, indicate how much each weight needs to be adjusted to reduce the loss. Optimization algorithms such as random gradient descent is deployed to calculate gradients and update weights, with the aim of updating weights in the opposite direction of gradients to minimize losses. This process will be repeated for multiple epochs, where each epoch involves passing the entire dataset through the network, calculating gradients, and updating weights.

3.4 Transfer Learning with COCO Dataset

Transfer learning is a dee learning method, a model that has already been trained on a large and general COCO is further trained on a smaller, more specific dataset that is the table tennis dataset. In this situation, it can leverage the knowledge and features that the model has already learned from the COCO dataset and adapt it to the table tennis balls detection task without training the entire model from scratch, that leads to faster convergence and improved performance on specific task.

In the case of YOLOv8s, weights and parameters of the model can be fine-tuned on customed table tennis dataset by modifying the final layers of the model to match the number of classes in training dataset and train it using annotated data. The pre-trained YOLOv8s model has already learned to extract valuable features from images in COCO dataset, which are useful for object detection. Transfer learning allows to leverage these learned features, saving the effort and computational resources required to train a feature extractor from scratch.

Moreover, the model can learn to detect and classify objects specific to research while retaining the knowledge it gained from the COCO dataset including object shapes, textures, and contexts. Starting with a pre-trained model can lead to faster convergence during training on customed table tennis dataset due to the learned useful representations of the model which reduces the number of epochs needed to adapt it to specific task. Pre-

trained models often act as regularizers that can help prevent overfitting by providing a good initialization point, which can lead to better generalizatio. Transfer learning typically results in models that perform better on the target task compared to training from scratch, especially customed task with limited data.

3.5 Convolutional Neural Network with Feature Extraction

Figure 3.4 demonstrates the process using convolutional operation for feature extraction. In the left side, the image is provided as input, and a 3 × 3 filter focus on scanning and obtaining local information for each grid through translation. The edge of the image needs to be processed into size 640 × 640 by adding 0 due to the arbitrary image size as input. The valuable information on the edge of images is not easily omitted. For instance, the 640 × 640 input image pass the *Conv* [64, 3, 2] and the output is 320 × 320 feature map with 64 channels. The parameter for the entire model will be written in weights file which can be employed for prediction task in real-time videos.

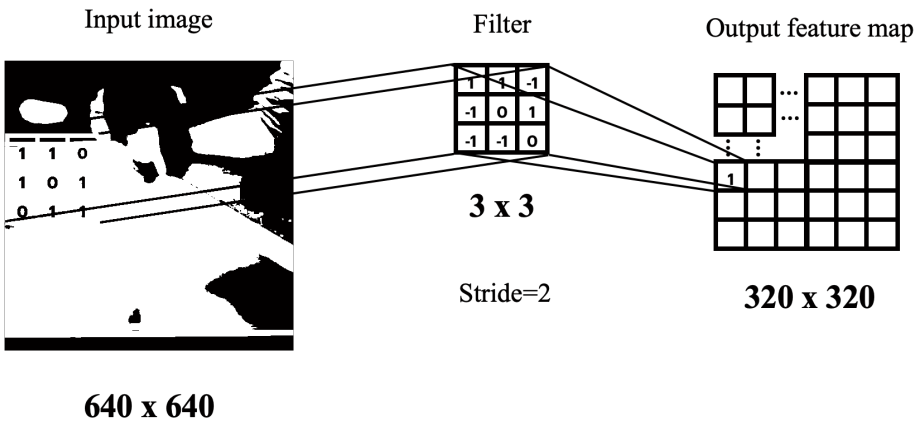


Figure 3.4. Filtering is applied to scan an image to extract features

YOLOv8 has two parts of the architecture consisted of the backbone and the head. According to the backbone, the input images will pass the layer with the different filters. By applying multiple filters with the different weights, the convolutional layer is able to extract multiple different features from the input, which are then used to build higher-level features in subsequent layers of the neural network. In Figure 3.5 related to the

backbone part, Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) are combined to aggregates features from multiple layers of a neural network using top-down and bottom-up pathways into a feature pyramid similar to FPN. The resolution of the images will be doubled through up-sampling as input. Concatenated P5, P7, P9 and Conv2d module build the connected structure from backbone to head leads to unshared parameters between classification and regression, passing these layers, bounding box and confidence scores of the detected object will be generated.

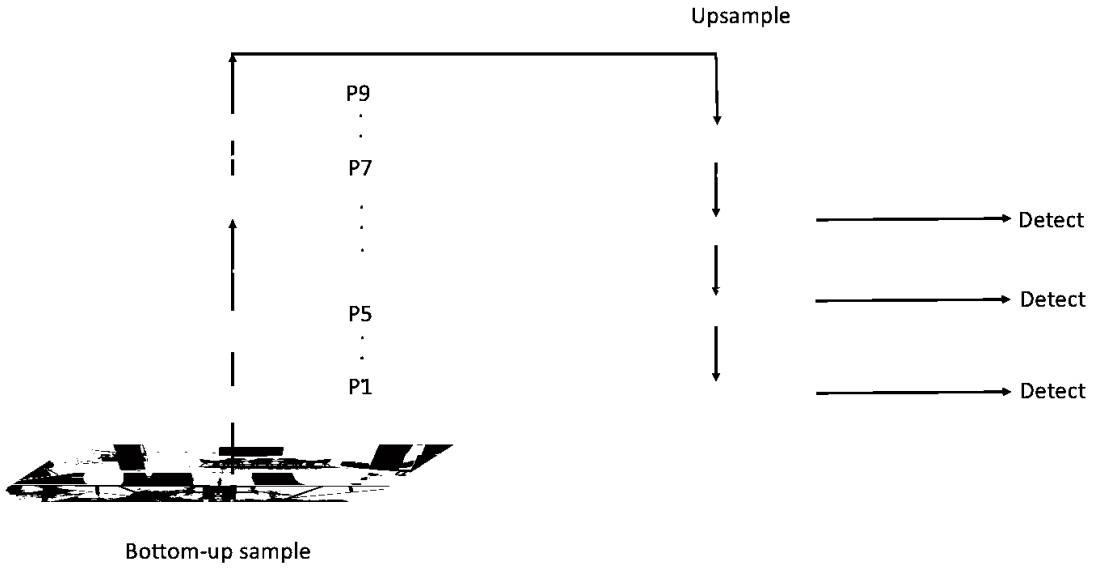


Figure 3.5. Backbone network architecture inspired by PANFPN

3.6 Motion-Based Method

The light spots and reflective patches are immovable in a video while using a static camera for ball detection in table tennis games, which provides feasibility for addressing these influencing factors. MoG (Mixture of Gaussians) can subtract background in video sequences based on the pixel intensities in the background. Generally, the initialized approach requires a trade-off between subtle differences in background and computational efficiency. If the pixels represented as Gaussian distribution mixtures are considered as background by one or more components, it is most possible to be evaluated as background by using MoG, as shown in eq.(3.1) and eq.(3.2).

$$N(\mu, cov) = \frac{1}{\sqrt{2\pi\, det(cov)}} e^{-0.5(\chi-\mu)'(\chi-\mu)inv(cov)} \tag{3.1}$$

$$P = \sum[weight_i \cdot N(\mu_i,\ cov_i)] \tag{3.2}$$

The probabilities assigned to each Gaussian component represent a potential class of a Gaussian distribution, such as background. The higher the likelihood, the greater the probability that it belongs to the background. By setting and adjusting the threshold, the accuracy of this classification method can be controlled. Figure 3.6 displays the frame of a real-time video that is added a mask to cover the background, the static objects including the light spots and reflective patches are removed after pre-processing. In this experiment, the background subtraction approach does not seem to reduce the accuracy of the table tennis balls detection though the color and texture of the ball in each frame is replaced by a white mask.

The results of YOLOv8 prediction involve a 2D tensor of bounding box coordinates. The center point of a table tennis ball that occurs on the screen is signed as the current box time and coordinates which need to be transferred to real-world coordinates using the perspective transformation with an initialized *z*-coordinate added as a 2D homogeneous point (*x, y,* 1). The camera projection matrix combines intrinsic and extrinsic parameters, the inverse matrix is used to transform points from image coordinates to normalized camera coordinates f_x and f_y are the focal lengths along x-axis and y-axes; c_x and c_y are the optical center coordinates. (*X, Y, Z*) is the center point coordinates of the ball in the real world.

$$(X, Y, Z) = \begin{bmatrix} 1/f_x & 0 & -c_x/f_x & 0 \\ 0 & 1/f_y & -c_y/f_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot (x, y, 1) \tag{3.3}$$

Figure 3.6. Separating the moving object from background in an image sequence

The displacement of a ball between the consecutive frames can be calculated based on the coordinate transformation, the instantaneous velocity will be acquired depending on this time interval, which is the frame rate.

3.7 Frame Difference Method

Frame difference method involves calculating the pixel-wise difference between consecutive frames of a video to identify regions of change or movement. This method is particularly useful for detecting motion in scenarios where the background is relatively static and only a few objects are moving. Differ from the Canny edge detector (Zhan, 2007), the center coordinates of the bounding box are used to calculate the difference between the center points of two images. Based on precise location of balls in table tennis games detected by YOLOv8 algorithm from video frames, the velocity of balls can be calculated through the variation of location between two consecutive frames corresponding to the frame rate. A camera with 120 Hz can effectively prevent the disappearance of the table tennis balls in each frame and ensure the surface of table

completely exists in the screen, maintaining an angle that allows for landing spot on the table. An auxiliary camera is fixed at a 90-degree angle to the main camera.

Once a table tennis ball moves perpendicular to the main camera lens in two consecutive frames, which will be replaced by the image information captured by the auxiliary camera in two consecutive frames. The velocity of a ball in each two-dimensional space needs to be mapped to real-world 3D coordinates by using camera calibration which depends on the perspective transformation of a black and white chessboard as a reference that was put in the scene on the table. The intrinsic and extrinsic parameters including focal length principal point, position and orientation obtained lead to depth information added into the coordinates of bounding boxes. The instantaneous velocity with spatial direction can be calculated through the mapped spatial displacement and frame rate.

3.8 Camera Calibration

The light from the real-world passes through the camera lens to form a 2D image, which defines the relationship between 3D world coordinates and 2D image coordinates. The camera matrix *K* contains information about the intrinsic parameters of the camera, such as focal lengths f_x and f_y and the principal point c_x and c_y. The equation reflects how real-world coordinates are projected onto the image plane to form two-dimensional coordinates. *u* and *P* denote 2D image coordinate and the 3D world coordinate, respectively.

$$u = K \cdot P \tag{3.4}$$

Translation Vectors *t* specify the distance from the camera to the origin of the world coordinate system representing the position of the camera in real-world space. Rotation Vectors *r* is parameter reflecting rotation of camera relative to the world coordinate system and it represents the orientation or pose of the camera in 3D space.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \quad (3.5)$$

where radial distortion ($k_1,\ k_2\ ,\ k_3$) affects points based on their distance from the principal point, causing them to bend inward or outward. Tangential distortion ($p_1,\ p_2$) corrects for small deviations that are not accounted for by radial distortion. These distortions can cause 2D points in the image to deviate from their ideal positions. In the calibration process,distortion coefficients are used to correct for radial and tangential distortions introduced by the camera lens. Using a reverse distortion model which is displayed in the equations to eliminate the distortion of 2D image points can make the image coordinates more accurate for three-dimensional reconstruction and obtain corrected 3D coordinates ($x_{corrected}, y_{corrected}$).

$$\begin{bmatrix} x_{corrected} \\ y_{corrected} \end{bmatrix} = \begin{bmatrix} x_{distorted} \\ y_{distorted} \end{bmatrix} + [k_1\ \ k_2\ \ p_1\ \ p_2\ \ k_3] \cdot \begin{bmatrix} r^2 \\ r^4 \\ 2x\,y \\ (r^2 + 2x^2) \\ (r^2 + 2y^2) \end{bmatrix} \quad (3.6)$$

$$r^2 = x_{distorted}^2 + y_{distorted}^2 \quad (3.7)$$

A chessboard is a flat object of known square size. A corner on the chessboard plane can be regarded as the origin of world coordinates (0, 0, 0), and the plane where the chessboard is located is the grid pattern of the X-Y plane, with the Z-axis perpendicular to the chessboard plane. In the grayscale image, corners are at where there are rapid changes in intensity in both the horizontal and vertical directions.

Harris corner detection algorithm discovers local intensity variations in the image that are characteristic of corners. Sobel operators is used to compute the gradients of the image in both the horizontal I_x and vertical I_y directions. The sensitivity constant *k* is set to 0.06, making the algorithm less sensitive to corners and only detecting the most prominent and well-defined corners in the image. Weak or less obvious corner like features will be filtered out.

$$R = I_x I_x \cdot I_y I_y - I_x I_y \cdot I_x I_y - k \cdot (I_x I_x + I_y I_y)^2 \quad (3.8)$$

Figure 3.7 displays the findings of the internal corner on the chessboard placed on the table for camera calibration. The surface of the chessboard is considered perpendicular to *z*-axis in world coordinates and coincident with the plane enclosed by using *x*-axis and *y*-axis. The intrinsic, distortion coefficients, rotation vectors and translation vectors can be computed (Liu, 2018) that the mapping points of the internal corners in the real-world due to the known number of rows and columns in the chessboard and the size of each square in the real world.

A series of images containing calibrated checkerboards need to be taken. These images contain multiple views of the checkerboard at different positions and angles, such as different lighting environments and focal lengths. For each image, using one corner of the checkerboard as the origin, we establish a correspondence between the detected corner pixel coordinates and the physical coordinates of the checkerboard and perform camera calibration using the established corner pixel coordinates and corresponding physical coordinates. We fit the camera matrix based on the least squares method to minimize the error between the actual pixel coordinates and the predicted pixel coordinates. This fitting process will determine various parameters in the camera matrix through optimization.

$$\text{Matrix} = \begin{bmatrix} Sum({G_x}^2) & Sum(G_x \cdot G_y) \\ Sum(G_x \cdot G_y) & Sum({G_y}^2) \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (3.9)$$

$$\lambda = \frac{(A+B) \pm \sqrt{(A+B)^2 - 4(AB - C^2)}}{2} \quad (3.10)$$

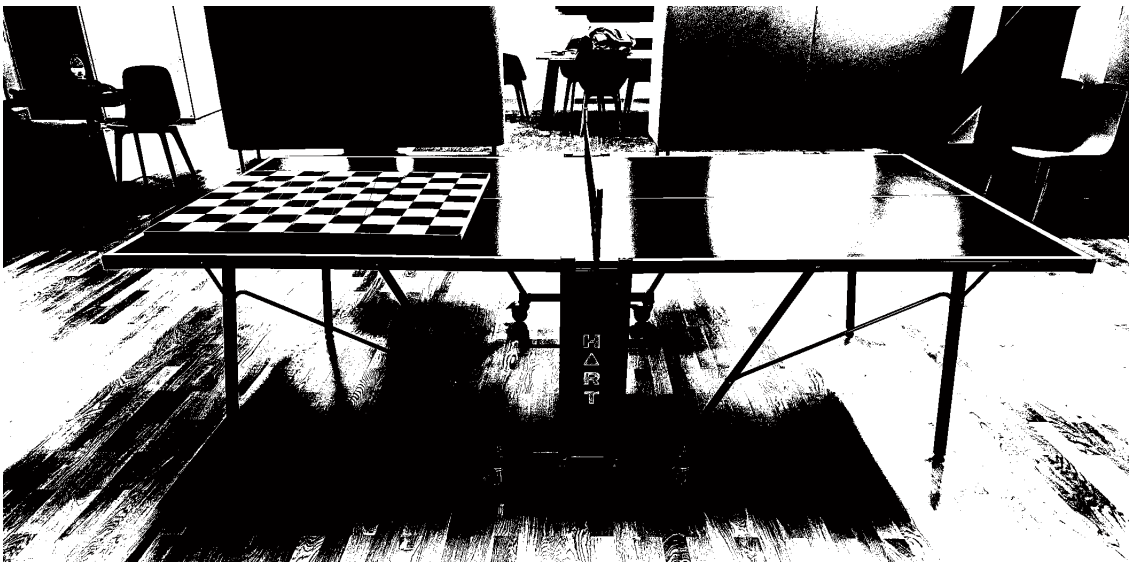


Figure 3.7. Finding the corners of a chessboard in an image for camera calibration

3.9 Speed Calculation

The speed and spin of table tennis balls are two important influencing factors in table tennis competitions. There have been many studies on the table tennis balls spin in the past. However, there has been little research on speed. In table tennis matches, the spin of the table tennis balls cannot be observed with human vision. Conversely, the speed changes of table tennis balls are easily transmitted to the brain through the visual nerve and then respond accordingly.

In fact, the speed change of table tennis balls can also be studied as a manifestation of the spin change of table tennis balls. Therefore, calculating and analyzing the speed of table tennis balls is valuable including the following reasons: (1) Players need to determine whether to bat the ball by moving footwork close to or away from the table based on the ball speed. In Figure 3.8, if the player stands at position *c* and bat the ball on position *a*, the ball will fly out of the boundaries of the table along the green arc. If the player moves to position *d* away from the table that can reduce the ball speed at position *b* and make the flight trajectory of the ball along the purple line to prevent the return stroke from going out of bounds.

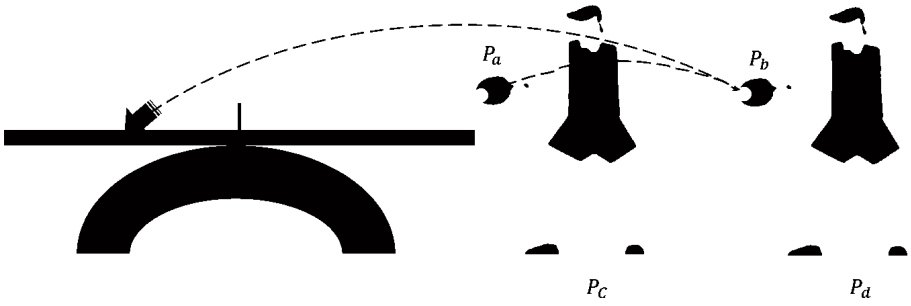


Figure 3.8. The changed position of batting with footwork based on the speed of the ball

(2) The direction of hitting the ball is also a way to avoid out. By raising the hitting angle from green arrow to purple one in Figure 3.9, which means swinging the racket in an angle-open manner, higher curve is similar to the purple lines that will prevent the table tennis ball from going out of the boundary.

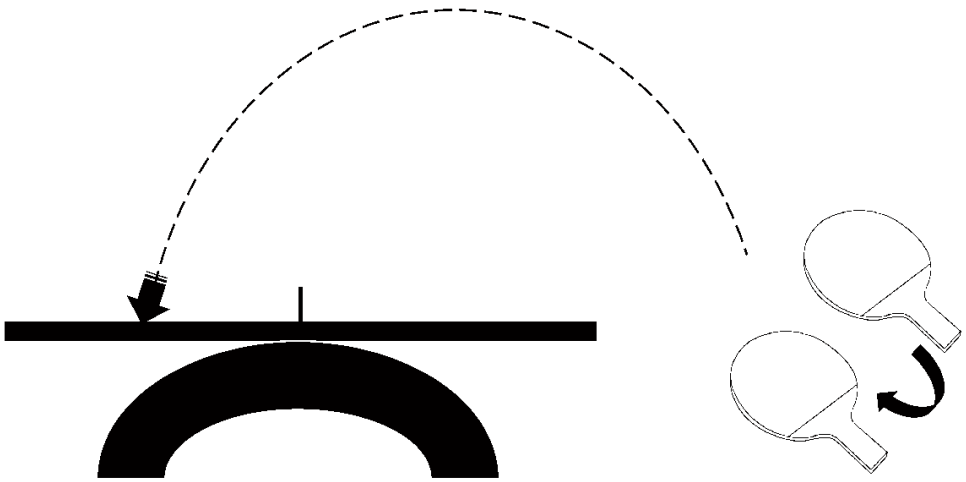
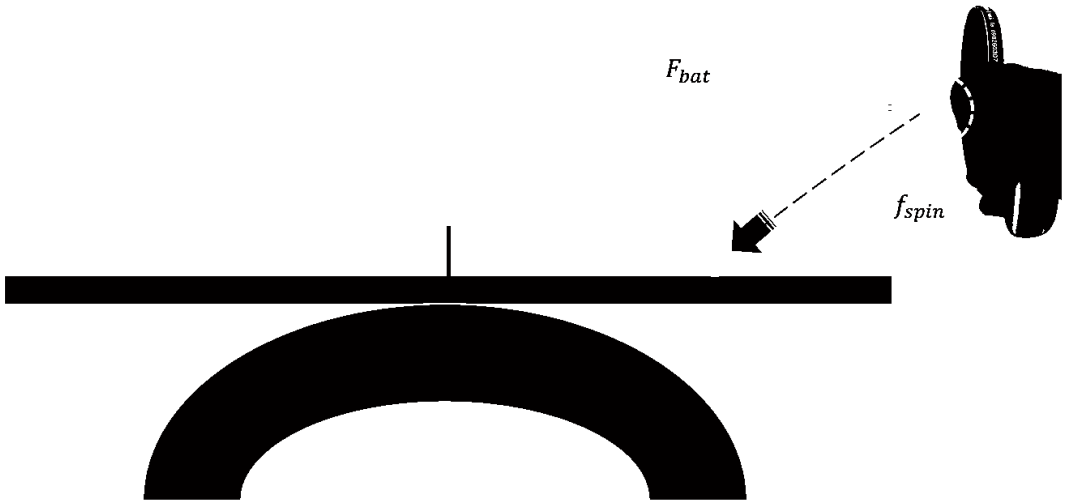


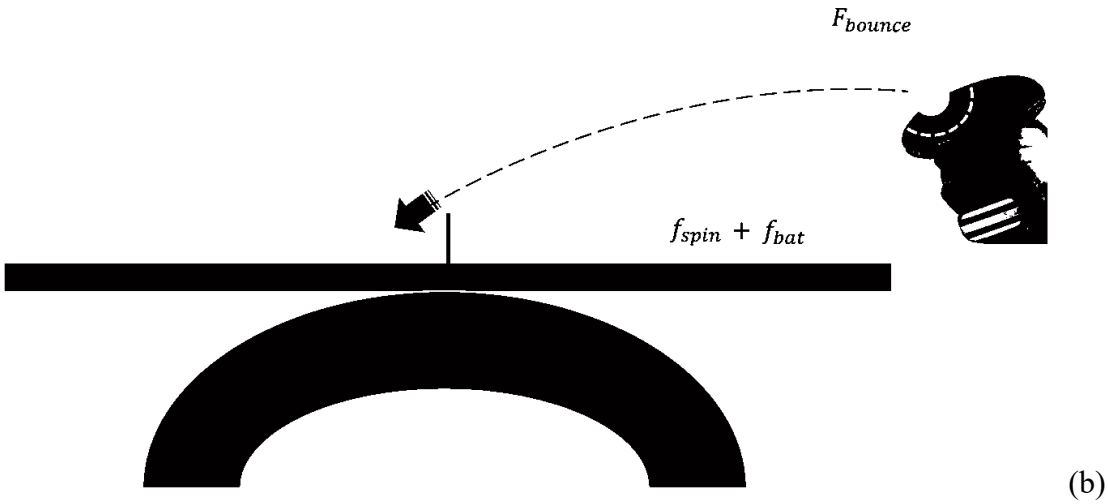
Figure 3.9. Changing the batting angle based on the ball speed

(3) The table tennis balls returned by the opponent slows down after hitting the ball, indicating that it is a backspin ball. When returning this type of ball, in addition to adjusting the hitting angle mentioned above, it is also necessary to consider how to counteract the backward spin force by changing the direction of friction when the racket contacts the ball. If the player swings the racket along the direction of F_{bat} to block the ball as seen in the Figure 3.10 (a), under the combined force F_{bat} and friction f_{spin}, the

movement of the ball roughly follows a purple trajectory until it lands on the table. If the player changes the direction of swinging racket along the f_{spin} in Figure 3.10 (b), the combined force F_{bounce} and friction $f_{spin}+f_{bat}$ *will push the ball over the net.*



(a)



(b)

Figure 3.10. Batting a ball after rebound deceleration due to the backspin

(4) Vice versa, the phenomenon of a table tennis ball bouncing faster from the table is caused by the forward spin of the ball. Generally, it is necessary to swing the racket with angle-closed to rub the surface of the ball to complete the return stroke like Figure 3.11 (b). If a player would like to bat the ball like Figure 3.11 (a) result in ball out of bounds.

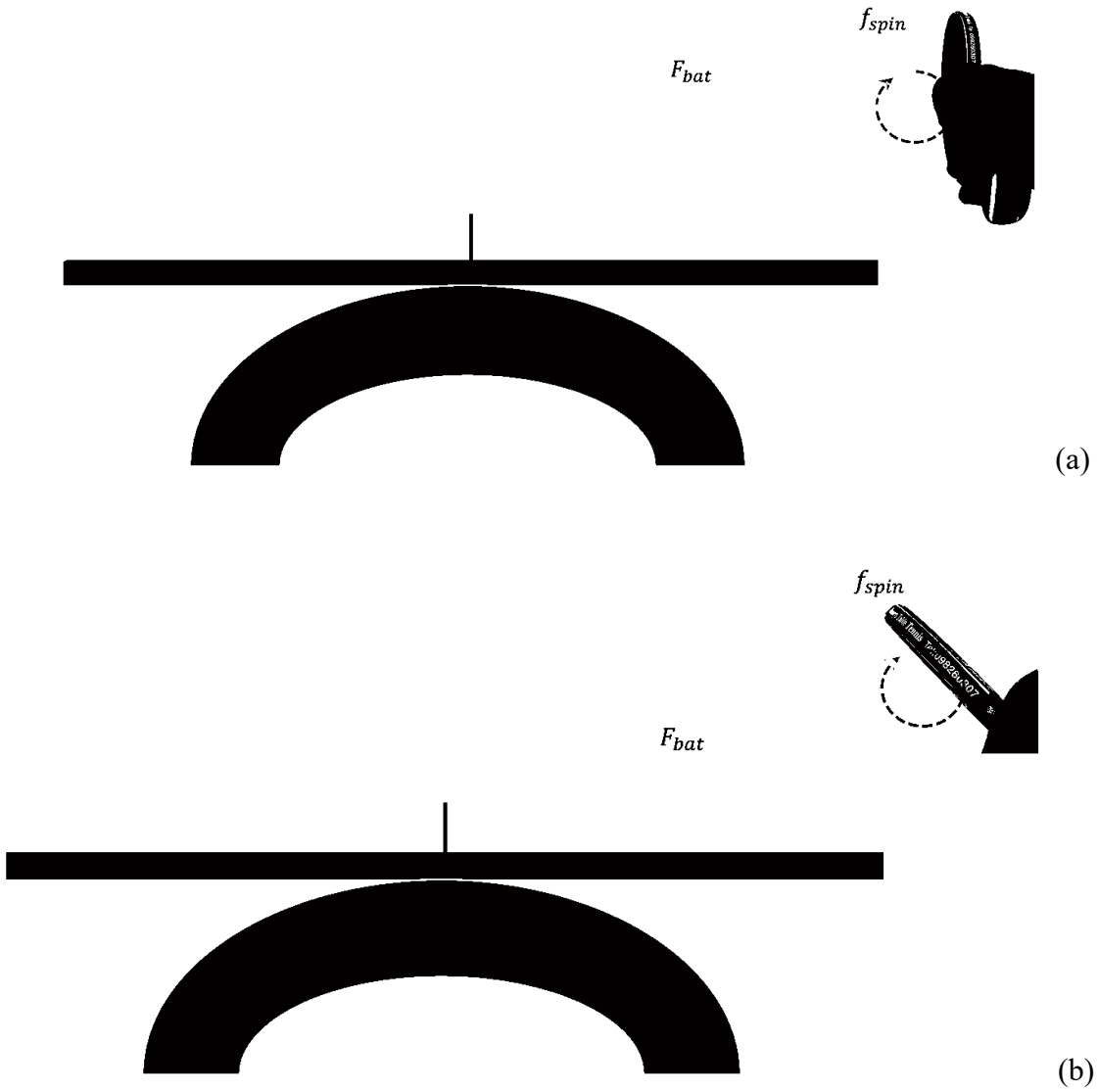


Figure 3.11. Batting a ball after rebound acceleration due to topspin

(5) Using ball speed as a skill indicator is more intuitive than spin avoiding misunderstandings. A number of table tennis beginners have misconceptions about the return stroke of a spinning ball, mistakenly believing that rubbing a ball with only a specific direction and angle of friction can achieve effective return stroke. For instance, a backspin ball needs to be rubbed under the ball to counteract the spin when hitting back. In this case, if the angle closed friction ball is accompanied by a faster speed, a return stroke can also be completed. The return stroke for the spinning ball is not based on the friction method, but on the speed and swing angle. When the ball flies off the edge of the table, the athlete can have enough space to swing arms, both the chopping and loop-drive

can handle arbitrary spinning ball.

This experiment only focused on one of the factors and simplified the model to the simplest collision model, enabling athletes to make more reasonable responses when facing different ball speed, thereby improving sports performance. The speed that affects the ball flying can be abstracted into three situations: with forward rotation, with backward rotation, and without rotation. When a ball with forward spinning is placed on the table, the rotated ball will act on the table in a frictional manner, and under the influence of the reaction force, the ball will be accelerated. Conversely, a ball with a backward rotation will slow down when it comes into contact with the tabletop. The speed of a ball without rotation itself is basically not affected by hitting the table. Inexperienced athletes often have an improper understanding of the movement and speed changes of the ball, resulting in inappropriate return force. However, computer vision-based calculation and analysis system can accurately reflect the speed changes of the ball.

While using the frame difference method to subtract the center position of the target in two consecutive frames, it is necessary to find the center coordinate of the detected target, which evinces the advantages of the YOLOv8 model in the table tennis balls detection and speed calculation. The reason is that the anchor free algorithm in the head of model architecture can predict bounding boxes systematically including coordinates (e.g., center x, center y, width, height) and confidence scores through pre-trained data. In the head, deploying two parallel branch heads for object classification and localization, this optimization was inspired (Feng et al., 2021) and achieved through two subtasks in one stage object detectors.

Task aligned Head was proposed by TOOD to enhance the interaction between classification and localization tasks and provide greater flexibility to learn alignment via a task aligned predictor. The core of Anchor free is to align ground truth bounding boxes and predicted bounding boxes through this task aligned assigner. The gray line in the Figure 3.12. divides the image into multiple grids, and feature extraction is performed on each grid separately. The center point of the anchor may be located on any grid, as

evidenced by green annotations of the ground truth bounding box. The center point of which predicted bounding boxes located in the range of the ground truth bounding box are considered as positive samples. If an anchor corresponds to multiple grounding truth bounding boxes, the one with the highest IoU will be successfully matched.

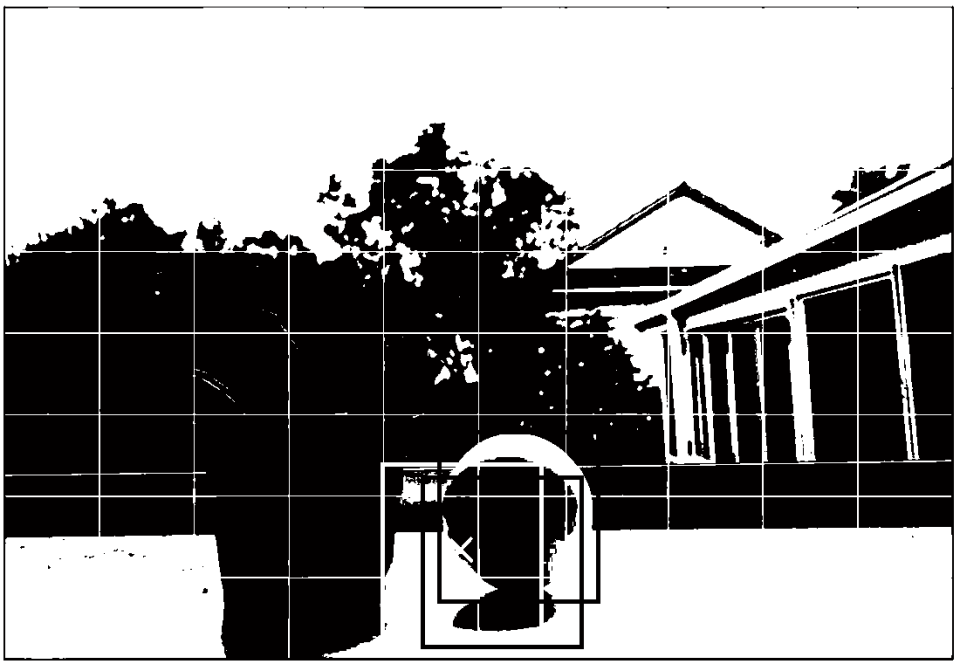


Figure 3.12. Anchor-free for bounding boxes prediction

The speed change of table tennis balls is influenced by various factors, the most important of which include the force of hitting, the force and direction of the ball's rotation. These factors not only affect the speed of the ball flying in the air, but also directly affect the direction and speed of the ball bouncing after hitting the table. Therefore, simulating and calculating speed is an extremely challenging task.

The biggest advantage of considering using computational and visual methods to calculate the instantaneous speed of a ball is that the speed of the ball can be determined directly from the position changes of a table tennis ball in consecutive frames without too much consideration of the reasons for the formation of speed. Scaccia (2006) and oshayedi (2019) both mentioned that the speed of a ball in table tennis can reach nearly 112 kilometers per hour (approximately 0.031 meters per millisecond) in professional

athlete competitions. According to the experimental results, the inference time of the YOLOv8 model is 7.2 milliseconds. A 120 Hz camera captures a frame every 8.3 milliseconds, and during this process, the table tennis ball moves up to a distance of about 0.2573 meters in the real world. Therefore, using computer vision to record and analyze the speed of ball in table tennis is feasible, and it will not exceed the perception range of the camera, resulting in two consecutive frames where the target cannot be captured.

3.10 Landing Spots Computing

There are 116 Spanish national level players were tested and found no significant difference in reaction time based on horizontal advantage. However, male players have lower movement time than female players, while female players have significantly lower reaction time values. The study also found that lower reaction time and exercise time can be considered key variables in table tennis performance. Thus, these factors should be considered while evaluating the performance of table tennis players (Castellar et al, 2019). The main reason why athletes need to move horizontally in table tennis is due to the changes in the landing point of the opponent's counterattack. Therefore, it can be inferred that mastering the information of the ball landing point is the key to mobilizing the opponent and gaining an advantage in defense and attack.

In order to detect the landing spots, the surface of table is segmented from images as shown in Figure 3.13. Each side of the table is split into nine regions respectively. One table has left side or right side from the camera viewpoint. The landing spots of a table tennis ball on both sides of the table will be continuously recorded and displayed to the players in percentage form based on the number of times the region of table has been hit. The probability of each region hit by the ball can be analyzed for understanding the players' skills in the aftermath of a match.

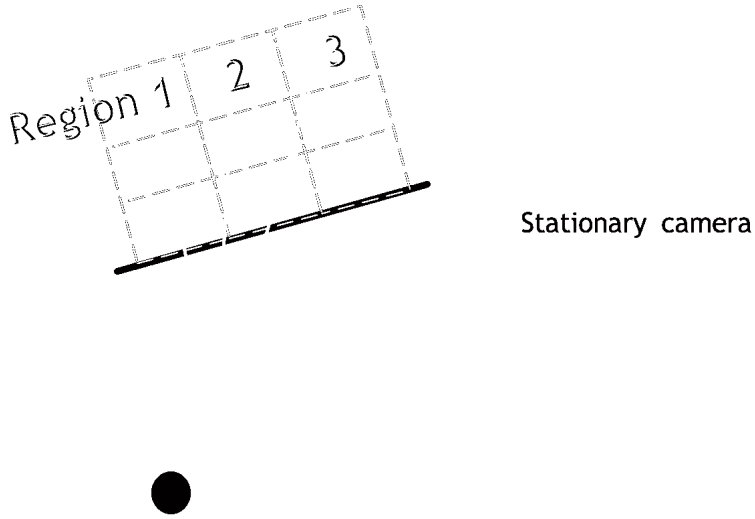


Figure 3.13. The sketch of a table division in table tennis

During each real-time video shoot of a table tennis match, there will be minor changes in the relative position and angle of the table and camera; Therefore, it is necessary to calibrate the tabletop area before shooting which is a complex process. Based on the segmentation approach, the tabletop can be separated from a frame as binary grayscale image that is shown in Figure 3.14.

According to approximate contour method, the contour is simplified and displayed in Figure 3.15. through compressing horizontal, vertical, and diagonal segments, while retaining only its endpoints. This provides the conditions for dividing the tabletop region. Contour approximation is the Douglas-Peucker algorithm which simplifies the contour by replacing closely spaced point sequences with fewer points that approximate the same shape. This algorithm recursively divides the contour into simpler parts and only preserves points that are crucial for approximating the contour shape. The vertical distance *d* from a given point *(x, y)* to a line segment defined by two points $(x_1,\ y_1)$ and $(x_2,\ y_2)$. If the maximum value *d* of this distance is less than or equal to the acceptable deviation ε, all points in the contour can be approximated by a line connecting the first and last points. In this case, only the first and last points are returned as simplified contours. Vice versa, if d_{max} is greater than ε, contour should be divided into two segments at the point with the maximum distance *d*. The final simplified contour will be

obtained through recursive call after finding and connect the simplified contours of two-line segments.

$$d = \frac{|(x_2 - x_1)\cdot(y_1 - y) - (x_1 - x)\cdot(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{3.11}$$

Figure 3.14. The binary grayscale image of a table tennis table

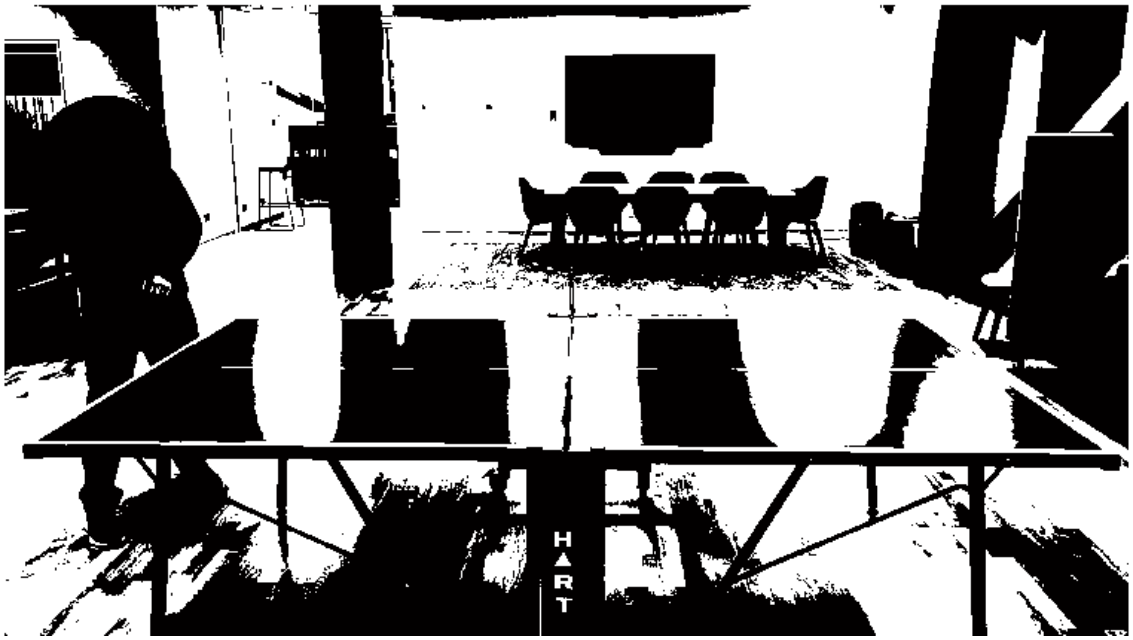


Figure 3.15. Calibrate tabletop area

If a table tennis ball moves to contact with the table, it undergoes elastic deformation.

From the perspective of camera placement in Figure 3.16, the contour below the table tennis ball in the image can almost be considered as the position of contact with the desktop. If it can be determined that the table tennis ball hit the table at this frame.

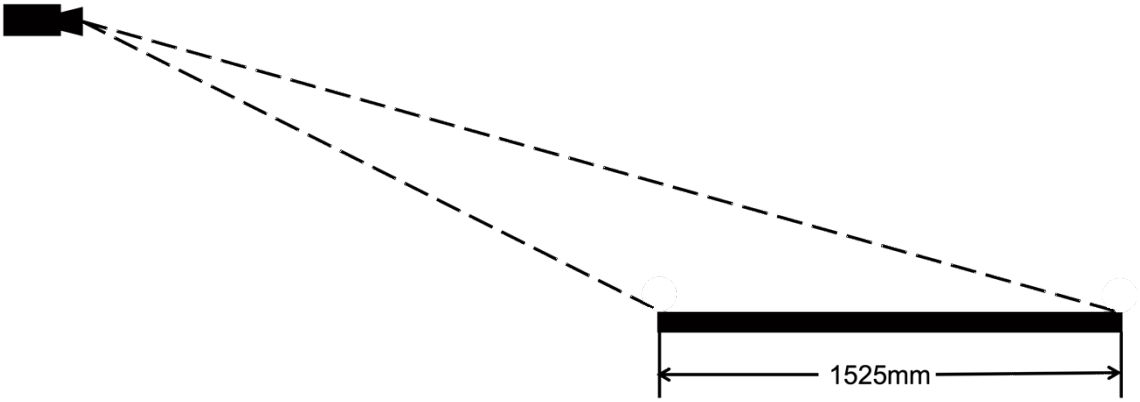


Figure 3.16. The angle of a camera to capture the table tennis ball hitting the table

On the tabletop, the most significant manifestation of a table tennis ball bouncing after hitting the surface of the table is the change in velocity direction in y-axis when the camera and the table are pointing in the same plane. That means, *y*-coordinates for the center point of the table tennis ball change in the vertical axis in consecutive three frames. The function *sign* ($y_m - y_f$) represents the sign of position difference in y-axis between the previous two frames consecutively, *sign* ($y_c - y_m$) shows the sign of change of ball positions in y-axis between the current frame and the previous one. *LS = -1* means the ball hits on the table and then bounces back. Or else, the table tennis ball is considered flying without hitting the table. The bottom of the bounding box for the table tennis ball in the previous frame is compared with the regions to determine where it lands.

$$LS = \begin{cases} not\ hit, & if\ sign\ (y_m - y_f) \cdot sign(y_c - y_m) = 1 \\ hit, & if\ sign(y_m - y_f) \cdot sign(y_c - y_m) = -1 \end{cases} \quad (3.12)$$

Chapter 4

Results

The main content of this chapter is to collect video data and display the experimental results. In the end, this chapter will also discuss the limitations of the project.

4.1 Experimental Environment

Using a 120Hz camera for real-time video capturing means that the inference time required by the YOLOv8s algorithm needs to be less than 8.3ms. According to data released by the Ultratics team, with the support of A100 TensorRT, the inference time is about 1.2ms. Therefore, Google Colab with A100 GPU displayed in Figure 4.1. is employed as the experimental environment to avoid affecting the experimental results due to insufficient GPU performance. In this experimental environment, there were no instances of training interruption caused by memory overflow during training.

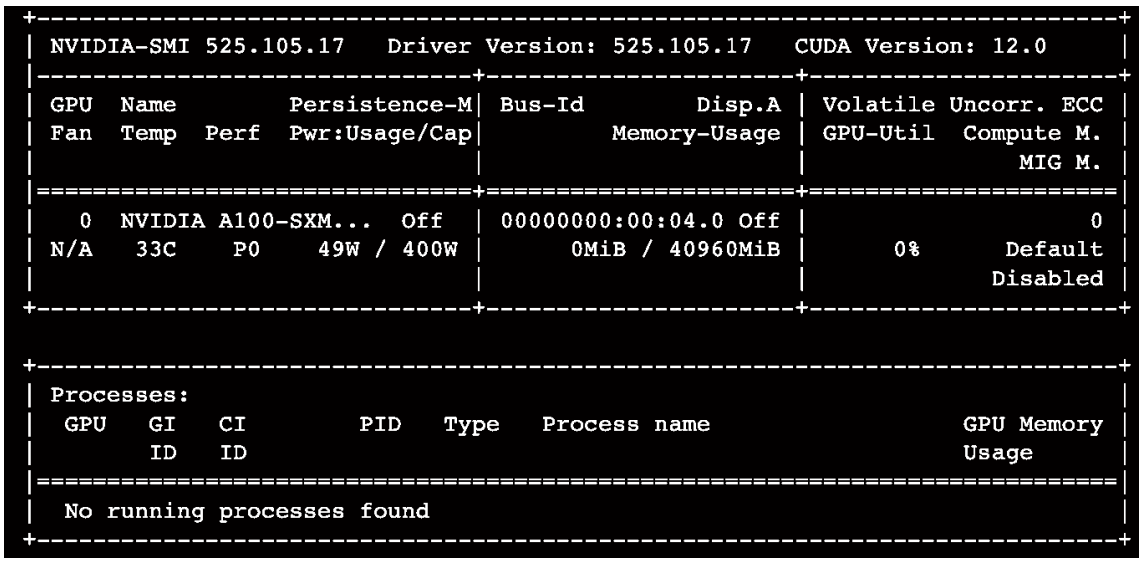


Figure 4.1. NVIDIA-SMI in the Google Colab environment

4.2 Data Collection

The collection of custom datasets is conducted in multiple steps. Firstly, a 120Hz high-definition camera is used to capture a movement video of a table tennis ball thrown into a living environment. These videos are segmented into separate frames using OpenCV, and frames in which the position of the ball in table tennis does not change significantly in consecutive frames are removed to prevent overfitting and obtain 300 images that can be used for training. Figure 4.2 demonstrates that the contour of the table tennis ball in the retained images is clearly visible, and a certain scale can be obtained while the table

tennis ball is thrown away from the camera, and there are also instances where the ball is partially obstructed.

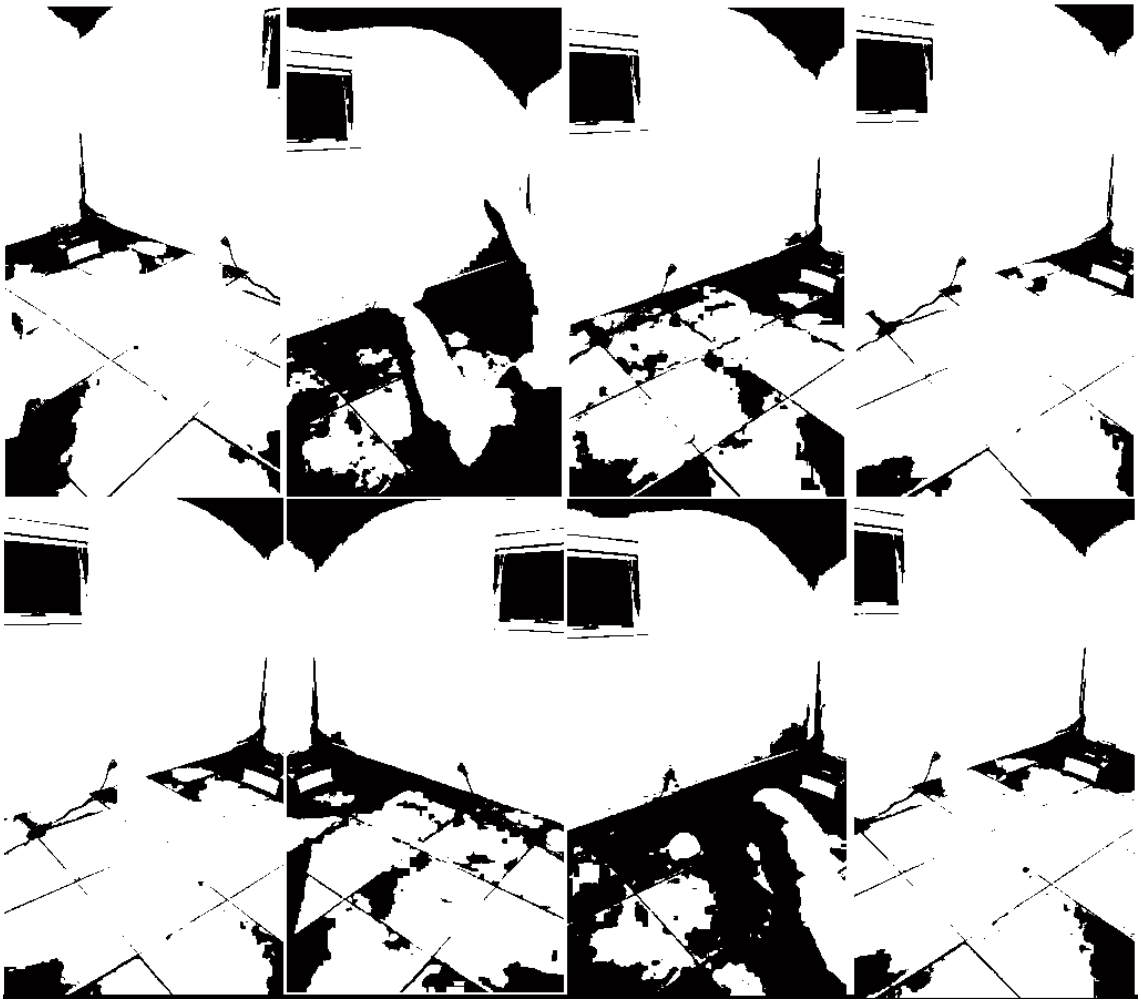


Figure 4.2. Display of samples collected for the first time

In this case, while using a 120 Hz stationary camera to capture a real-time video, the static table tennis balls can already be detected by the YOLOv8 algorithm, although the balls for table tennis games are made up of different colors and textures. Five types of colors table tennis balls detected by using YOLOv8s algorithm are display in Figure 4.3.

Figure 4.3. Five types of colors for the balls in table tennis games are detected by using YOLOv8s model

However, the result obtained from using real table tennis training videos as the validation dataset illustrates that the table tennis balls in almost fast motion cannot be detected with only 15.6% mAP value in Figure 4.4.

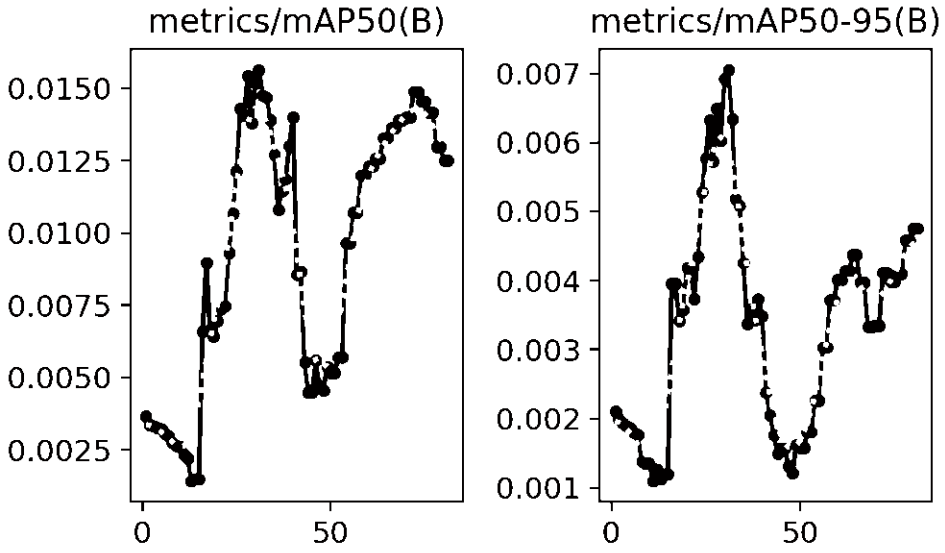


Figure 4.4. Unsatisfied accuracy after the first data collection and training

After analyzed the images of table tennis in 120 Hz real-time videos, it was found that most of the table tennis balls in training videos exhibit motion blur due to their fast speed, resulting in significant changes in the shape of the balls. Compared to the original

table tennis balls, it seems necessary to add table tennis balls with this characteristic as the target to the training set if there are two types of objects. A camera with a lower image acquisition frequency is more likely to capture motion blur caused by rapid movement; Thus, the 30 Hz and 60 Hz camera are utilized to capture the additional images from training of table tennis that is shown in Figure 4.5. The total number of images used for training also reached 1,774.

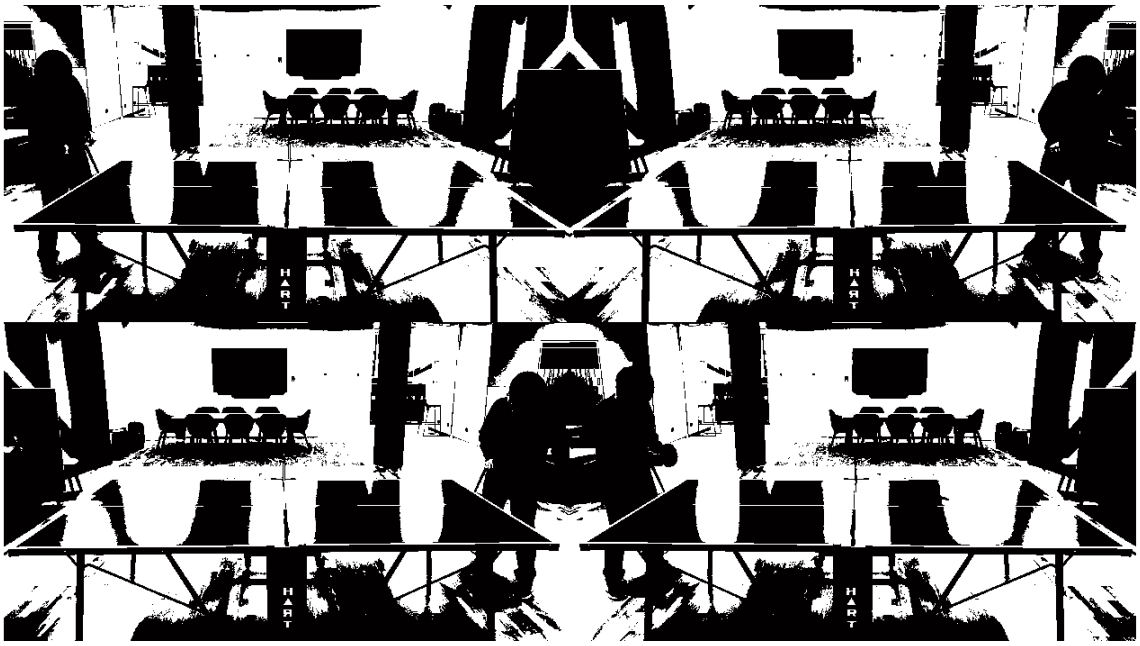


Figure 4.5. Training images of table tennis with significant motion blur

In Figure 4.6, the accuracy of the table tennis balls detection is 54%, which has been significantly improved compared to the results of the first training. Through observation of real-time videos, it appears that the table tennis balls can be accurately detected, but the accuracy is insufficient. The observation of the validation video reveals that the existence of false positives may be caused by indoor lighting environment which are illustrated in Figure 4.7, and the light spots and reflective patches in the video background are sometimes considered the table tennis balls.

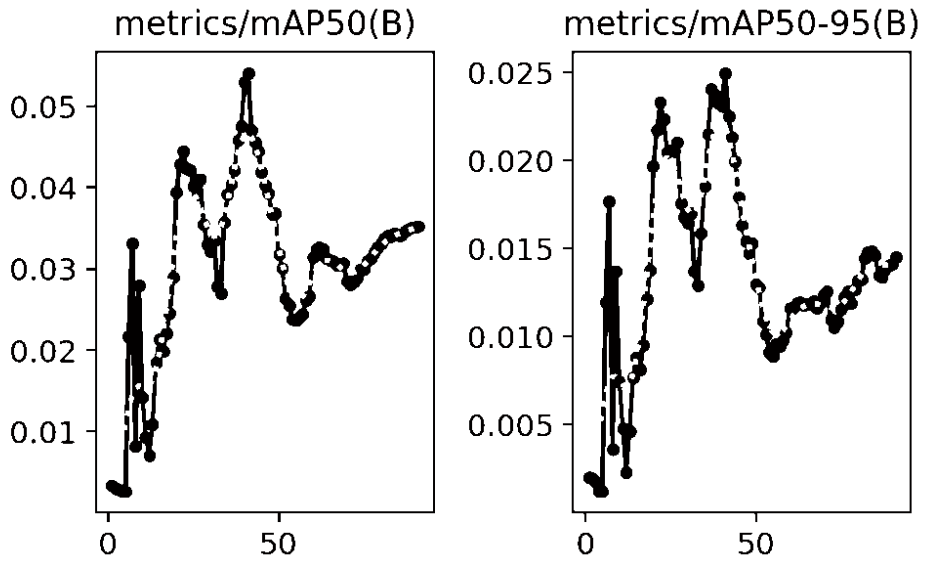


Figure 4.6. The mAP 50 prepresents the accuracy after the second time training

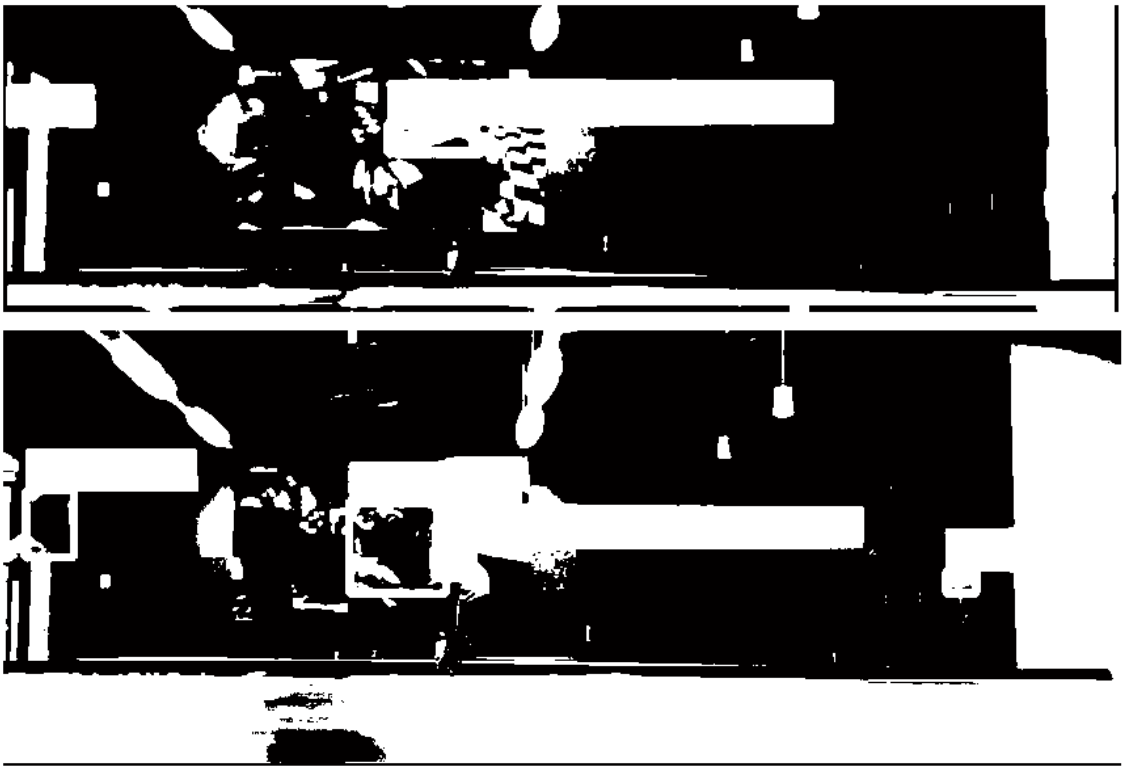


Figure 4.7. The light spots are mistakenly recognized as the table tennis balls

The improved YOLOv8s algorithm has been enabled to accurately detect the table tennis balls during training and competition. In Figure 4.8, based on the characteristics of a table tennis ball in continuous moving, the pixel intensity of the light spot and reflective

patches is considered as the background by the MoG component and masked, and will no longer be considered as a ping pong ball during inference.

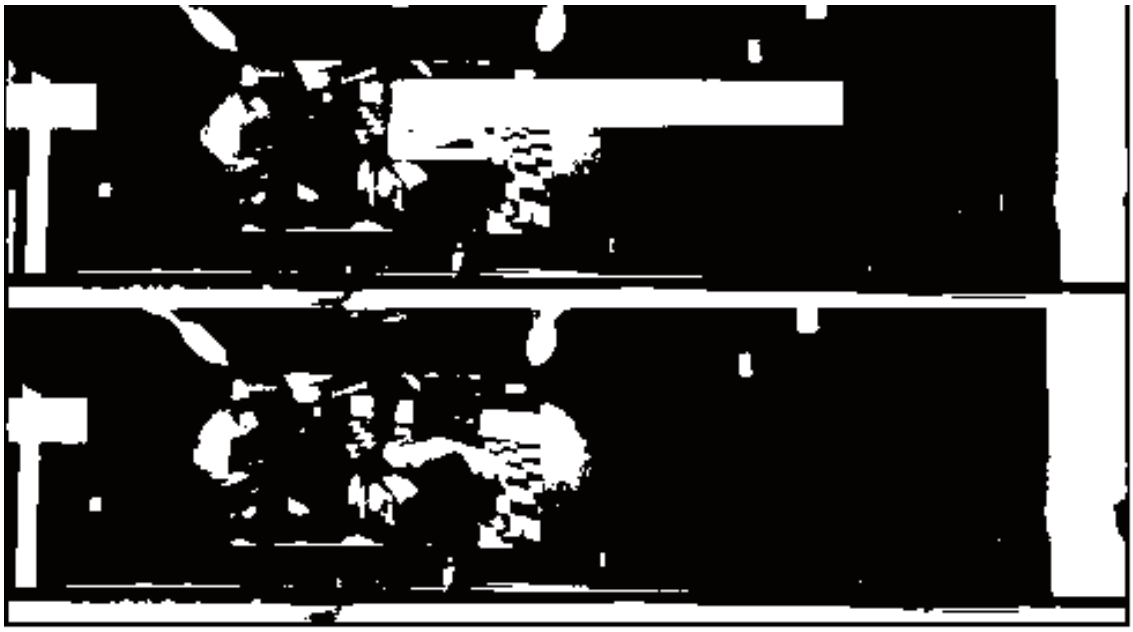


Figure 4.8. The results of a table tennis ball detection using motion-based YOLOv8s algorithm

Figure 4.9 is an example after an original image is resized, different resized images will be obtained with 10 random scale factors set as variables for the balls in table tennis games. Figure 4.10 demonstrates the resized images with motion blur as an example. This deformation exhibits a variety of forms due to a diversity of directions and velocities of motions, such as rectangles, arches, and shapes approximate to the letter ‘v’.

Figure 4.9. The example after an original image is resized

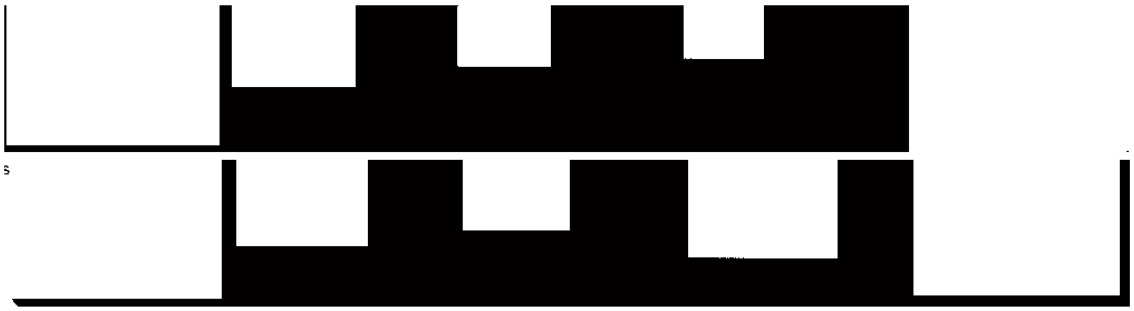
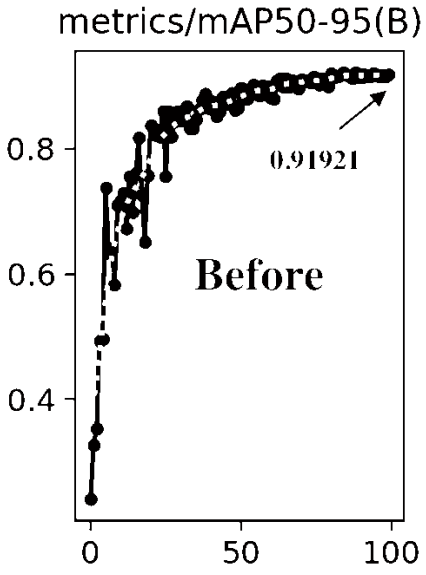


Figure 4.10. The example after an original image with motion blur is resized

The weights and parameters of the pretrained COCO dataset through the YOLOv8 model are employed to start with the original 1,774 images captured by 120 Hz, 30 Hz and 60 Hz cameras in table tennis training and competition. Figure 4.11 (a) shows that with the support of background subtraction in the pre-process of the video, the accuracy of the table tennis balls detection has been fundamentally improved to 91.9% of the figure; However, this has not yet met expectations. 90% of 1774 images still need to be resized following the approach mentioned above to expand the dataset. In Figure 4.11 (b), after the data in the training dataset was augmented and processed, the data capacity increased by 10 times to 15, 960 images and the accuracy of training result is approximate to 93.5% and the mAP 50 can even exceed 98%.



(a)

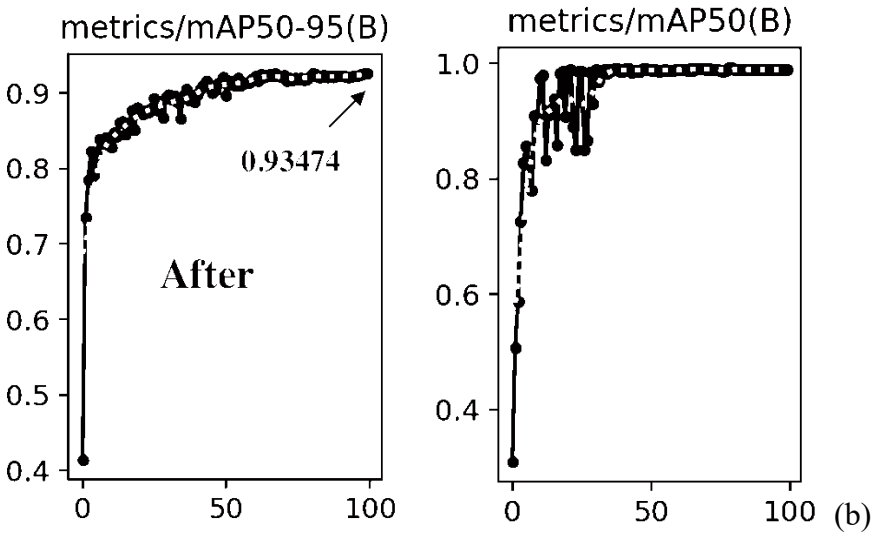


Figure 4.11. Comparison of mAP50-95 in 100-th epoch before and after resizing by random scale factor

4.3 Model selection

There are 15, 960 images which were employed as training datasets, and the remaining images are left for the validation dataset and testing dataset. The inference time and the detection accuracy of the table tennis balls are significant evaluations. Table 4.3 illustrates that the inference time of the YOLOv8 algorithm with a shorter inference time compared with the DETR algorithm in the experiment under Google Collab virtual environment equipped with A100 GPU.

Table 4.3. Comparisons between DETR and YOLOv8 algorithm

Name	Size(pixel)	Backbone	Inf_time (ms) A100 GPU
DETR	640	ResNet-50	75
YOLOv8s	640	CSPDarknet53	7.2

Figure 4.12 reflects that YOLOv8s, as a lightweight model, contains 11,139,470 parameters for training process. The extracted feature of images as input start from layer 0, and then the output of the previous convolutional network will become the input of the

next layer. The richer information of gradient flow can be obtained by the c2f module, and the feature information from different levels are concatenated in steps 11, 14, 17, and 20 respectively to achieve the detection results based on the structure of PAN-FPN (Path Aggregation Network and Feature Pyramid Network).

```
Ultralytics YOLOv8.0.173 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (NVIDIA A100-SXM4-40GB, 40514MiB)
engine/trainer: task=detect, mode=train, model=yolov8s.pt, data=data/fall.yaml, epochs=100, patience=50, batch=32, imgsz=
Overriding model.yaml nc=80 with nc=10

                   from  n    params  module                                       arguments
  0                  -1  1       928  ultralytics.nn.modules.conv.Conv             [3, 32, 3, 2]
  1                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  2                  -1  1     29056  ultralytics.nn.modules.block.C2f             [64, 64, 1, True]
  3                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  4                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  5                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  6                  -1  2    788480  ultralytics.nn.modules.block.C2f             [256, 256, 2, True]
  7                  -1  1   1180672  ultralytics.nn.modules.conv.Conv             [256, 512, 3, 2]
  8                  -1  1   1838080  ultralytics.nn.modules.block.C2f             [512, 512, 1, True]
  9                  -1  1    656896  ultralytics.nn.modules.block.SPPF            [512, 512, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    591360  ultralytics.nn.modules.block.C2f             [768, 256, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 16                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    493056  ultralytics.nn.modules.block.C2f             [384, 256, 1]
 19                  -1  1    590336  ultralytics.nn.modules.conv.Conv             [256, 256, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  1   1969152  ultralytics.nn.modules.block.C2f             [768, 512, 1]
 22        [15, 18, 21]  1   2119918  ultralytics.nn.modules.head.Detect           [10, [128, 256, 512]]
Model summary: 225 layers, 11139470 parameters, 11139454 gradients
```

Figure 4.12. Display of model architecture for training

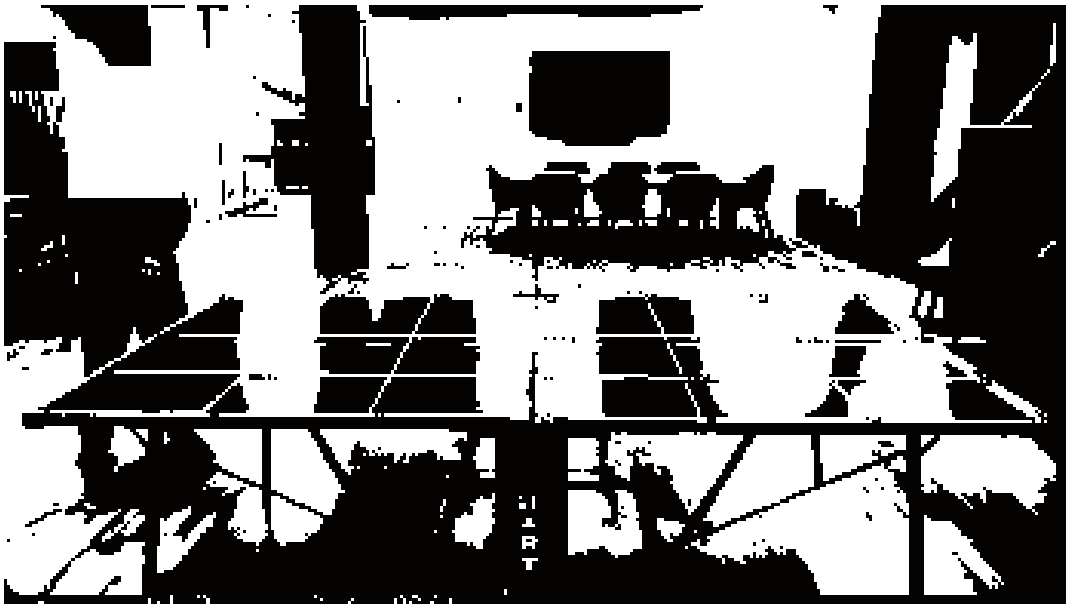
Figure 4.13 demonstrates that closing mosaic data augmentation in the last 10 epochs of the YOLOv8s model improved the model's generalization ability. In the early stages of training, data augmentation can help the model better learn the features of the data and prevent overfitting. However, in the later stages of training, the model has already learned enough features. If data augmentation is continued, it may lead to the model overfitting the training data result in reducing the model's generalization ability. Therefore, turning off data augmentation can help the model better generalize to the test dataset and improve the accuracy of the model.

```
Closing dataloader mosaic
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     91/100      8.52G      0.942     0.6169     0.8556          8        640: 100% 3/3 [00:02<00:00,  1.14it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  3.03it/s]
                   all         38        138     0.0965     0.0211     0.0353     0.0145
```

Figure 4.13. Mosaic data augmentation are closed in last 10 epoch

Figure 4.14 displays the table which is automatically divided into nine regions on each side using visual object detection and region segmentation. The bounding boxes of balls touch these regions on the table surface are the landing spots.


(a)


(b)

Figure 4.14. In real-time scene, the table is automatically segmented into nine regions on each side. (a) color images with 9 regions on each side (b) Binary images of the table

Figure 4.15 demonstrates the real-time analysis system of table tennis matches. On the left side, it is the video footage of ongoing competition captured by a 120 Hz stationary camera. The statistics and analysis listed on the right side consist of the

instantaneous flying speed of the ball and the percentages of the regions that are hit by the ball on the table. Through this system, both the player and coach can accurately grasp the player's actions that can set training plan for further improvement.

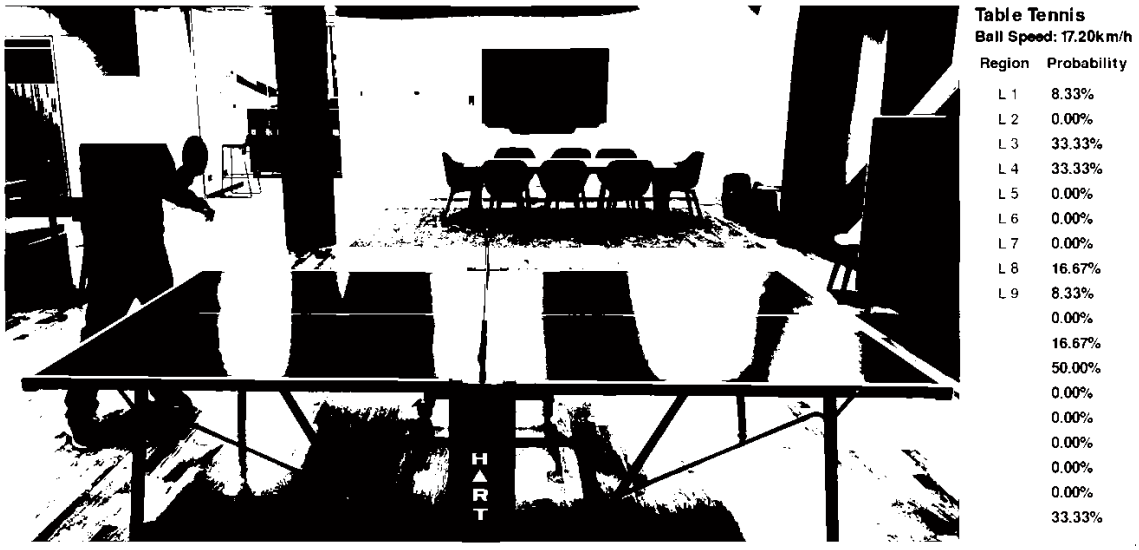


Figure 4.15. The interface of real-time analysis of table tennis matches

4.4 Limitations of the Research

(1) The accuracy of the ball speed estimation has not been verified by other feasible tools and methods.

(2) The landing spots of the edge ball in a matchh of table tennis cannot be calculated and analyzed using this system.

Chapter 5
Analysis and Discussions

In this chapter, experimental results are analyzed and compared. Comparisons of the results under various conditions will be mentioned.

5.1 Analysis

In summary, while initially using only a 120Hz camera to capture and label the table tennis balls as a custom training dataset, the training results were able to capture the balls in table tennis matches with less than 15.6% accuracy. Additional using a mixture of 30Hz and 60Hz cameras to capture videos, during the manual data labeling process, it is evident that there are many deformations in the aspect ratio of the table tennis balls due to motion blur, which is correspond with the essence of rapid movement of a ball in table tennis. This movement will cause significant changes in the color, texture, and shape of a table tennis ball in the image. The integration of these multi-scale data into training significantly improved the accuracy to 54%. This result falls far short, it can be clearly seen from the inference video that light spots and reflective patches are considered as the table tennis balls .

From the perspective of motion characteristics, adopting motion-based algorithm, the accuracy of the table tennis balls detection significantly improved to 91.9% from 54% compare with using original YOLOv8s algorithm. Light spots and reflective patches as stationary backgrounds are no longer considered as the table tennis balls. The accuracy of the YOLOv8s model trained with data augmentation exceeded 93.5%, slightly improved compared to the previous 91.9%. This may be closely related to the camera position used in analyzing the video and ultimately capturing real-time table tennis scenes. After all, in order to clearly and completely record the movement of a table tennis ball above the table, the camera will not be placed everywhere; Thus, there will be no significant changes in aspect ratio.

Accurately detecting the table tennis balls in real-time videos only lays the foundation for calculating speed and landing spots. Table tennis matches are real movements that exist in three-dimensional space; Therefore, it is necessary to translate coordinate changes in two-dimensional images into displacements in the real world. As for the landing point of a table tennis ball, the motion model can be transformed into a mathematical model to consider. After hitting a horizontally placed tabletop, a table tennis will rebound, which means that there is a minimum value of the position of the table

tennis ball in the direction perpendicular to the tabletop, that is the minimum value of y when in contact with the tabletop. These variables required for calculation can be obtained from the sensor of the YOLOv8 model, making it possible to calculate and analyze table tennis matches.

If the player is too slow to adapt to the ball speed, coaches should focus on improving the physical fitness of players, focusing on agility, speed, and reflexivity training to respond quickly to quick bats and rebounds. Relaxation and composure in facing a fast ball are also crucial, such as relaxed grip and calm demeanor, which requires efficient footwork practice to avoid being caught off guard. Tension can hinder the players' reflexes and control. Reading the opponent's ball speed, studying their playing style, and predicting the landing spots of the opponent's stroke can all give athletes an advantage in the game. Encountering faster ball speeds requires athletes to quickly return to their prepared positions after each hit in order to effectively respond to the next attack. A Table tennis ball emphasizes the variation of rhythm. If an athlete can control the variation of ball speed and landing spots, it will be difficult for opponents to guess the rhythm of the attack. Table tennis is not only a physical exercise, but also a mental exercise that requires training players' attention and psychological resilience to maintain focus and mental agility throughout the game, avoiding tactical interference from opponents. Combining the server with this computational analysis system of table tennis match can simulate training with different pace.

The materials for making table tennis rackets and surfaces may be wood, carbon, or a combination of multiple composite materials. These components will affect the hardness of the racket and thus the ball speed. A harder blade often transfers more power from the player to the ball, resulting in higher speed. In Table 5.1, speed is the rating of the power generated by the blade. It can be broken down into loop speed and volley speed (speed on flat contact). Using a faster blade produces faster shots which at high speeds. Assuming that an athlete using a Nittaku Rutis racket now has a maximum ball speed of 40 kilometers per hour calculated and analyzed by the system. In this condition, the player can directly replace the racket with Butterfly Schlager Carbon, Yasaka Extra Offer 7 Power, or Nittaku Rutis Power to directly improve the ball speed and make the attack

more lethal.

Table 5.1. Comparison of table tennis racket and ball speed batted

	Name of table tennis racket	Structure	Thickness	Speed (grade)
1	Butterfly Timo Boll Spirit	5 wood plies (Koto, Limba, Kiri), 2 Arylate-carbon (ALC)	5.8 mm	8.5
2	Butterfly Schlager Carbon	Three wood plies+2 Carbon	7.4mm	9.5
3	Yasaka Extra Offensive 7 Power	5 wood +2 carbon		9.2
4	Nittaku Rutis	3 plywood, G-carbon 2	5.5mm	8.9
5	Nittaku Rutis Power	Walnut outer plies, AD Carbon	5.8mm	9.6
6	Butterfly Timo Boll ALC	2 plies of Arylate Carbon, 2 plies of Koto, 2 plies of Limba, center ply is Kiri	5.7mm	8.9
7	Andro Super Core Carbon Light ALL	Ayous, Lima	6.4-6.8mm	8.6
8	Stiga Offensive Classic	Ayous, spruce, Lima	5.4-5.5mm	7.6
9	Yasaka Extra	Ayous, Lima	6.0mm	7.4
#	Stiga Carbo 7.6 WRB	7 plies of wood; 6 plies of Carbon	6.1-6.3mm	8.6
#	Nittaku Redshank	2 Ayous, 2Limba	6.2mm	8.9

The rubbers on a racket, such as stickiness, inward and outward, and its thickness, can affect how the ball grasps the surface, causing the player to rotate and speed. Compared to thinner rubber, thicker rubber often provides higher speed. In addition, the tension of the rubber and the hardness of the sponge underneath can affect the effect of the trampoline and the speed of the ball. Tight rubber and harder sponge can hit the ball faster, especially when the ball comes into contact with the racket with a lot of force. Even with the same racket, the player's technique and swing speed can significantly affect the ball's speed. Regardless of the characteristics of the racket, a player who strikes the ball forcefully and at the right time can produce higher ball speed. On the other hand, the contact point between the racket and the ball, as well as the angle of the racket surface,

can affect the trajectory and speed of the ball. Clicking on the ball with the best stroke of the racket can maximize speed. From the above perspectives, the applicability of the racket and the correctness of its movements can be analyzed through ball speed.

5.2 Discussions

In experiments, various methods have been used to collect data and enhance data to improve the accuracy of the table tennis balls detection, and even motion-based algorithms have been used to preprocess videos. The results achieved by using these methods are also shown and compared, reflecting the progressiveness of the improved and optimized YOLOv8 algorithm in fast moving small target detection. Meanwhile, compared to transformer-based algorithms, YOLOv8, which is itself a lightweight model, has more advantages in inference time.

Previous research work on table tennis has required the use of various sensors. Installing sensors to measure the rotation of a table tennis ball can cause changes in the weight of the ball and interfere with normal competition. Detecting and tracking objects are a computationally intensive process, complexity of which may lead to delays in obtaining real-time speed measurements. Compared with computer vision, the limitation of lidar makes use of only the first observation of an object as the appearance model, the tracked object may rotate relative to the sensor over time, exposing previously unseen faces. This can result in a decrease in similarity between the appearance model and the observed points, making it difficult for the ICP process to find a good match. Additionally, directly replacing the appearance model with each new observation can lead to tracking failure if an incorrectly associated observation is used for subsequent registration steps (Morton et al, 2011).

However, motion-based YOLOv8s algorithm in this experiment can only complete the entire data collection and real-time monitoring through a traditional camera and a running environment with GPU, It will not be affected by the rotation of a table tennis ball and the brief computability time. While achieving calculation and analysis of ball speed and landing point, it has high cost-effectiveness. This calculation and analysis

system can also be integrated with the serving machine to dynamically adjust the serving speed and landing point according to the analysis results, improving the quality of training. At the same time, it can also be cross verified with the referee system to avoid misjudgment.

Chapter 6
Conclusion and Future Work

In this chapter, the subject and method of this project will be summarized, and new future direction of the research will be presented according to the result and insufficiency of the experiment.

6.1 Conclusion

In this comprehensive study, we delve deeply into the intricate integration of motion-based features with the formidable capabilities of the YOLOv8 model for precise ball detection in table tennis matches. The research objectives go beyond mere detection and aim for accurate estimation of landing spots and ball velocity. To obtain a nuanced understanding, we harness the capabilities of high-resolution cameras operating at both 30 Hz and 60 Hz. These video feeds are further enriched through the use of multiscale variation techniques specifically designed for data augmentation. This methodological enhancement not only boosts the Average Precision (AP) value but also significantly reduces the instances of false positives, often triggered by intrusive light flares and reflective interferences.

A novel aspect of this research methodology was the use of stereoscopic cameras, which are often employed to capture depth and dimension. This approach offered a unique advantage by facilitating the extraction of multiple perspectives and depth information, all while avoiding the usual computational overheads associated with depth extraction. This strategic application of technology sets the stage for more accurate computations of both landing spots and ball velocity, employing deep learning to decode and interpret real-world video data from table tennis tournaments.

6.2 Future Work

Guiding this research is a central ambition: to transform table tennis competitions and training programs through the seamless integration of motion-centric algorithms. However, this journey is not without its hurdles. A significant obstacle has been the false recognition of a ball's shadow as a tangible object, a problem that even ground subtraction techniques have not been able to eliminate fully. While it might be possible to differentiate between the actual object and its shadow based on their respective positions during landing spot calculations, a more comprehensive solution is worth considering at an early

stage. By enhancing pre-processing techniques to systematically remove shadows from video frames, the accuracy and reliability of the detection mechanism could see significant improvement.

References

Akramjonovich, Y. I., Abdumalikovich, U. A., Urinboyevna, U. Z., Abduxamidovich, M. Y. I., Azamovich, A. M., & Umidovich, A. B. (2022). Main characteristics of table tennis in international sport and technologies of playing it. *Journal of Positive School Psychology*, *6*(10), 2183-2189.

An, N., Yan, W. (2021) Multitarget tracking using Siamese neural networks. *ACM Transactions on Multimedia Computing, Communications and Applications.*

Ashraf, A. H., Imran, M., Qahtani, A. M., Alsufyani, A., Almutiry, O., Mahmood, A., Attique, M., & Habib, M. (2022). Weapons detection for security and video surveillance using cnn and YOLO-v5s. *CMC-Comput. Mater. Contin*, *70*, 2761-2775.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., & Bengio, Y. (2012). Theano: New features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Beal, J., Kim, E., Tzeng, E., Park, D. H., Zhai, A., & Kislyuk, D. (2020). Toward transformer-based object detection. *arXiv preprint,* arXiv:2012.09958.

Blank, P., Groh, B. H., & Eskofier, B. M. (2017). Ball speed and spin estimation in table tennis using a racket-mounted inertial sensor. In *ACM International Symposium on Wearable Computers,* (pp. 2-9).

Blue, S. T., & Brindha, M. (2019). Edge detection based boundary box construction algorithm for improving the precision of object detection in YOLOv3. *In International Conference on Computing, Communication and Networking Technologies (ICCCNT),* (pp. 1-5). IEEE.

Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint,* arXiv: 2004.10934.

Buric, M., Pobar, M., & Ivasic-Kos, M. (2018). Ball detection using YOLO and Mask R-CNN. In *International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 319-323). IEEE.

Cai, G. L. (2022). A method for prediction the trajectory of table tennis in multirotation state based on binocular vision. *Computational Intelligence and Neuroscience*.

Cao, S., & Nevatia, R. (2016). Exploring deep learning based solutions in fine grained activity recognition in the wild. In *International Conference on Pattern Recognition (ICPR)* (pp. 384-389).

Cao, X. and Yan, W. (2022) Pose estimation for swimmers in video surveillance. Multimedia Tools and Applications, Springer.

Cao, Z., Liao, T., Song, W., Chen, Z., & Li, C. (2021). Detecting the shuttlecock for a badminton robot: A YOLO based approach. *Expert Systems with Applications* (Vol. 164, 113833). IEEE.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision* (pp. 213-229).

Castellar, C., Pradas, F., Carrasco, L., La Torre, A. D., & González-Jurado, J. A. (2019). Analysis of reaction time and lateral displacements in national level table tennis players: Are they predictive of sport performance? *International Journal of Performance Analysis in Sport*, *19*(4), 467-477.

Chen, Z., Yan, W. (2024) Real-time pose recognition for billiard player using deep learning. Deep Learning, Reinforcement Learning and the Rise of Intelligent Systems, IGI Global.

Cheng, G., Guo, Y., Cheng, X., Wang, D., & Zhao, J. (2020). Real-time detection of vehicle speed based on video image. *International Conference on Measuring Technology*

and Mechatronics Automation (ICMTMA), (pp. 313-317). IEEE.

Chen, X., & Yuille, A. L. (2014). Articulated pose estimation by a graphical model with image dependent pairwise relations. *Advances in Neural Information Processing Systems*, *27*.

Diba, A., Sharma, V., Pazandeh, A., Pirsiavash, H., & Van Gool, L. (2017). Weakly supervised cascaded convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 914-922).

Diwan, T., Anirudh, G. & Tembhurne, J. V. (2023). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed Tools Appl 82,* 9243–9275.

Ding, T., Graesser, L., Abeyruwan, S., D'Ambrosio, D. B., Shankar, A., Sermanet, P., Sanketi, P. R., & Lynch, C. (2022). Learning high speed precision table tennis on a physical robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* (pp. 10780-10787).

Doulamis, N., & Voulodimos, A. (2016). FAST-MDL: Fast adaptive supervised training of multi-layered deep learning models for consistent object tracking and classification. In *IEEE International Conference on Imaging Systems and Techniques (IST)* (pp. 318-323).

Doulamis, N. (2018). Adaptable deep learning structures for object labeling/tracking under dynamic visual environments. *Multimedia Tools and Applications*, *77*, 9651-9689.

Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., & Liu, W. (2021). You Only Look at one sequence: Rethinking transformer in vision through object detection. *In Neural Information Processing Systems 34 (NeurIPS).*

Feng, C., Zhong, Y., Gao, Y., Scott, M. R., & Huang, W. (2021). Tood: Task-aligned one-

stage object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 3490-3499).

Forsyth, D., & Ponce, J. (2003). Computer Vision: A Modern Approach. *Prentice Hall.*

Gowdra, N., Sinha, R., MacDonell, S., Yan, W. (2021) Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting. Pattern Recognition.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior analysis and prediction in image sequences using rough sets. International Machine Vision and Image Processing Conference (pp.71-76)

Weng, J.Y., Cohen, P., & Hernion, M. (1992). Camera calibration with distortion models and accuracy evaluation. *IEEE PAMI, 14*(10), 965-980.

Fuchs, M., Liu, R., Lanzoni, I. M., Munivrana, G., Straub, G., Tamaki, S., Yoshida, K., Zhang, H., & Lames, M. (2018). Table tennis match analysis: A review. *Journal of Sports Sciences, 36*(23), 2653-2662.

Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised Sequence Labelling with Recurrent Neural Networks*, 37-45.

Hanchinamani, S. R., Sarkar, S., & Bhairannawar, S. S. (2016). Design and implementation of high-speed background subtraction algorithm for moving object detection. *Procedia Computer Science,* (Vol. 93, pp. 367-374). IEEE.

Hasan, M., Hanawa, J., Goto, R., Suzuki, R., Fukuda, H., Kuno, Y., & Kobayashi, Y. (2022). LiDAR-based detection, tracking, and property estimation: A contemporary review. *Neurocomputing*.

Herath, H., Perera, R., Ayesha, B., Sivakumar, T., Kumarage, A., & Perera, A. (2021). A comparison of speed data by different speed detection techniques. *Researchgate.*

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Howard, A. G. (2013). Some improvements on deep convolutional neural network-based image classification. *arXiv preprint arXiv:1312.5402*.

Huang, Y., Liao, I., Chen, C., İk, T., & Peng, W. (2019). TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications. *In IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* (pp. 1-8). IEEE.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of yolo algorithm developments. *Procedia Computer Science,* (Vol. 199, pp. 1066-1073).

Jian, Z., & Hong, H. (2017). Time difference of arrival passive positioning technology and its application research in table tennis ball landing spot estimation. *International Journal of Information and Electronics Engineering*, *7*(4).

Kapusi, T. P., Erdei, T. I., Husi, G., & Hajdu, A. (2022). Application of deep learning in the deployment of an industrial SCARA machine for real-time object detection. *Robotics, 11*(4), 69.

Karaman, A., Pacal, I., Basturk, A., Akay, B., Nalbantoglu, U., Coskun, S., ... & Karaboga, D. (2023). Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC). *Expert Systems with Applications*, *221*, 119741.

Kim, J., & Cho, J. (2020). Exploring a multimodal mixture-of-YOLOs framework for advanced real-time object detection. *Applied Sciences, 10*(2), 612.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, *2*.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Li, L., Huang, W., Gu, I. Y., & Tian, Q. (2003). Foreground object detection from videos containing complex background. In *ACM International Conference on Multimedia* (pp. 2-10).

Li, J., Chen, X., Huang, Q., Chen, X., Yu, Z., & Duo, Y. (2015). Designation and control of landing spots for competitive robotic table tennis. *International Journal of Advanced Robotic Systems, 12*(7).

Li, X., Zhao, K., Cong, G., Jensen, C. S., & Wei, W. (2018). Deep representation learning for trajectory similarity computation. In *IEEE International Conference on Data Engineering (ICDE)* (pp. 617-628).

Liang, C., Lu, J., Yan, W. (2022) Human action recognition from digital videos based on deep learning. ACM ICCCV.

Liang, C., Yan, W. (2024) Human action recognition based on YOLOv7. Deep Learning, Reinforcement Learning, and the Rise of Intelligent Systems. IGI Global.

Lin, H. Y. (2005). Vehicle speed detection and identification from a single motion blurred image. In *IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05),* (Vol. 1, pp. 461-467). IEEE.

Lin, L., Wang, K., Zuo, W., Wang, M., Luo, J., & Zhang, L. (2016). A deep structured model with radius–margin bound for 3D human activity recognition. *International*

Journal of Computer Vision, *118*, 256-273.

Liu, Y., & Liu, L. (2018). Accurate real-time ball trajectory estimation with onboard stereo camera system for humanoid ping-pong robot. *Robotics and Autonomous Systems,* (Vol. 101, pp. 34-44).

Liu, Y., & Ma, C. W. (2021, November). Improving tiny YOLO with fewer model parameters. In *International Conference on Multimedia Big Data (BigMM)* (pp. 61-64).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision,* (pp. 10012-10022).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440).

Lu, J., Shen, J., Yan, W., Boris, B. (2017) An empirical study for human behaviors analysis. International Journal of Digital Crime and Forensics 9 (3), 11-17.

Lu, J., Nguyen, M., Yan, W. (2018) Pedestrian detection using deep learning. IEEE AVSS.

Lu, J., Nguyen, M., Yan, W. (2020) Human behavior recognition using deep learning. International Conference on Image and Vision Computing New Zealand.

Lu, J., Nguyen, M., Yan, W. (2020) Comparative evaluations of human behavior recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 176-189.

Lu, J., Nguyen, M., Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. International Symposium on Geometry and Vision.

Lu, S., Wang, B., Wang, H., Chen, L., Ma, L., & Zhang, X. (2019). A real-time object

detection algorithm for video. *Computers & Electrical Engineering* (Vol. 77, pp. 398-408). IEEE.

Luo, Z., Nguyen, M., Yan, W. (2022) Kayak and sailboat detection based on the improved YOLO with Transformer. ACM ICCCV.

Luo, Z., Nguyen, M., Yan, W. (2021) Sailboat detection based on automated search attention mechanism and deep learning models. International Conference on Image and Vision Computing New Zealand.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115-133.

Morton, P., Douillard, B., & Underwood, J. (2011). An evaluation of dynamic object tracking with 3D LIDAR. In *Australasian Conference on Robotics & Automation (ACRA)* (p. 38).

Moshayedi, A. J., Chen, Z., Liao, L., & Li, S. (2019). Kinect based virtual referee for table tennis game: TTV (Table Tennis Var System). *In International Conference on Information Science and Control Engineering (ICISCE),* (pp. 354-359). IEEE.

Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision* (pp. 1520-1528).

Ouyang, W., Zeng, X., Wang, X., Qiu, S., Luo, P., Tian, Y., Li, H., Yang, S., Wang, Z., Li, H., Wang, K., Yan, J., Loy, C., & Tang, X. (2016). DeepID-Net: Object detection with deformable part based convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(7), 1320-1334.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception.

Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.

Parnami, A., & Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint,* arXiv: 2203.04291.

Park, S. I., Ponce, S. P., Huang, J., Cao, Y., & Quek, F. (2008). Low-cost, high-speed computer vision using NVIDIA's CUDA architecture. *In IEEE Applied Imagery Pattern Recognition Workshop,* (pp. 1-7). IEEE.

Peng, Q., Luo, W., Hong, G., Feng, M., Xia, Y., Yu, L., Hao, X., Wang, X., & Li, M. (2016). Pedestrian detection for transformer substation based on gaussian mixture model and YOLO. In *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 562-565).

Qi, J., Nguyen, M., Yan, W. (2022) Small visual object detection in smart waste classification using Transformers with deep learning. International Conference on Image and Vision Computing New Zealand (IVCNZ).

Qi, J., Nguyen, M., Yan, W. (2023) CISO: Co-iteration semi-supervised learning for visual object detection. Multimedia Tools and Applications

Qiao, F. (2021). Application of deep learning in automatic detection of technical and tactical indicators of table tennis. *PLOS ONE, 16*(3), e0245259.

Rajagopalan, A. N. (2023). Improving robustness of semantic segmentation to motion-blur using class-centric augmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10470-10479).

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in

machine learning. *arXiv preprint arXiv:1811.12808*.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint* arXiv: 1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788).

Rebuffi, S., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., & Mann, T. A. (2021). Data augmentation can improve robustness. *In Neural Information Processing Systems 34 (NeurIPS).*

Rodrigues, S. T., Vickers, J. N., & Mark, W. A. (2010). Head, eye and arm coordination in table tennis. *Journal of Sports Sciences, 20*(3), 187-200.

Scaccia, J. (2006). Tabling Tennis. *The New York Times*, A15-L.

Shi, J., Xu, L., & Jia, J. (2014). Discriminative blur detection features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (pp. 2965-2972).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*.

Sun, Z., Cao, S., Yang, Y., & Kitani, K. M. (2021). Rethinking transformer-based set prediction for object detection. *In IEEE/CVF International Conference on Computer Vision (ICCV),* (pp. 3611-3620). IEEE.

Takano, N., & Alaghband, G. (2019). SRGAN: Training dataset matters. *arXiv preprint arXiv:1903.09922*.

Tian, B., Zhang, D., & Zhang, C. (2020). High-speed tiny tennis ball detection based on deep convolutional neural networks. *In International Conference on Anti-counterfeiting, Security, and Identification (ASID),* (pp. 30-33).

Toshev, A., & Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1653-1660).

Tsai, R.Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the shelf TV camera and lenses. *Journal of Robotics and Automation, 3*(4), 323-344.

Viswanatha, V., Chandana, R. K., & Ramachandra, A. C. (2022). Real time object detection system with YOLO and CNN models: A review. *arXiv preprint,* arXiv: 2208.00773.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, *2018*.

Wang, J., Chen, Y., & Dong, Z. (2023). Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Comput & Applic, 35,* 7853–7865.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys, 53*(63), 1-34.

Whang, S. E., & Lee, J. G. (2020). Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment*, *13*(12), 3429-3432.

Wu, Y., Lan, J., Shu, X., Ji, C., Zhao, K., Wang, J., & Zhang, H. (2018). iTTVis: Interactive visualization of table tennis data. In *IEEE Transactions on Visualization and Computer Graphics,* (Vol. 24, pp. 709-718).

Wu, Y., Zhang, H., Li, Y., Yang, Y., & Yuan, D. (2020). *Video object detection guided by object blur evaluation,* IEEE Access (Vol. 8, pp. 208554-208565).

Yan, W. (2023). Computational Methods for Deep Learning: Theory, Algorithms, and Implementations. *Springer*.

Yan, W. (2019). Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. *Springer*.

Yang. B, W. Yan. (2024) Real-time billiard shot stability detection based on YOLOv8. Deep Learning, Reinforcement Learning, and the Rise of Intelligent Systems, IGI Global.

Yu, X. (2023). Evaluation of training efficiency of table tennis players based on computer video processing technology. *Optik*, *273*, 170404.

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. International Conference on Image and Vision Computing New Zealand.

Zhan, C., Duan, X., Xu, S., Song, Z., & Luo, M. (2007). An improved moving object detection algorithm based on frame difference and edge detection. *IEEE International Conference on Image and Graphics (ICIG 2007),* (pp. 519-523).

Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. *IEEE Transactions on Big Data*, *6*(1), 3-28.

Zhang, H., & Wang, J. (2019). Towards adversarially robust object detection. *In IEEE/CVF International Conference on Computer Vision (ICCV),* (pp. 421-430). IEEE.

Zhang, H., Zhou, Z., & Yang, Q. (2018). Match analyses of table tennis in China: A systematic review. *Journal of Sports Sciences, 36*(23), 2663-2674.

Zhang, P., Ward, P., Li, W., Sutherland, S., & Goodway, J. (2023). Effects of play practice on teaching table tennis skills. *Journal of Teaching in Physical Education, 31*(1), pp. 71-85.

Zhang, S., Wang, W., Li, H., & Zhang, S. (2022). EventMD: High-speed moving object detection based on event-based video frames. *Available at SSRN 4006876*.

Zhang, X., Zhang, T., Yang, Y., Wang, Z., & Wang, G. (2020). Real-time golf ball detection and tracking based on convolutional neural networks. *IEEE International*

Conference on Systems, Man, and Cybernetics (SMC), (pp. 2808-2813). IEEE.

Zhang, Y. J. (2023). Camera calibration. *In 3-D Computer Vision. Springer, Singapore*.

Zheng, K., Yan, Q., Nand, P. (2017) Video dynamics detection using deep neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence.

Zheng, W., Liu, X., &Yin, L. (2021). Research on image classification method based on improved multi-scale relational network. *PeerJ Computer Science.*

Zheng, W., Liu, X., & Yin, L. (2012). Research on image classification method based on im-proved multi-scale relational network. *J Computer Science,* (Vol. 7, p. 613). IEEE.

Zhou, H., Nguyen, M., Yan, W. (2023) Computational analysis of table tennis matches from real-time videos using deep learning. PSIVT 2023.

Zhou, Q., Li, X., He, L., Yang, Y., Cheng, G., Tong, Y., Ma, L., & Tao, D. (2022). TransVOD: End-to-end video object detection with spatial-temporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhu, X., Vondrick, C., Fowlkes, C. C., & Ramanan, D. (2016). Do we need more training data? *International Journal of Computer Vision*, *119*(1), 76-92.

Zhu, Y., Peng, B., Yan, W. (2022) Ski fall detection from digital images using deep learning. ACM ICCCV.

Zhu, Y., Yan, W. (2022) Image-based storytelling using deep learning. ACM ICCCV.