Utilizing RT-DETR Model for Fruit Calorie Estimation

Shaomei Tang

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

2024

School of Engineering, Computer & Mathematical Sciences

Abstract

Estimating the calorie content of fruits is critical for weight management and maintaining overall health as well as aiding individuals in making informed dietary choices. Accurate knowledge of fruit calorie content assists in crafting personalized nutrition plans and preventing obesity and associated health issues. In this project, we investigate the application of deep learning models for estimating the calorie content in fruits, aiming to provide a more efficient and accurate method for nutritional analysis. We create a dataset comprising images of various fruits and employ random data augmentation techniques during training to enhance model robustness. We utilize the RT-DETR model integrated into ultralytics framework for implementation and conduct comparative experiments with YOLOv9 on the dataset. Our results show that the RT-DETR model achieved a precision rate of 99.01% in fruit detection, outperforming YOLOv9 in terms of F1 score, precision, and mAP. The results of our experiments provide a technical reference for more accurately monitoring individuals' dietary intake through estimating the calorie content of fruits.

Keywords: RT-DETR, YOLOv9, Deep learning, Calorie, mAP

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I affirm that the content submitted herein is entirely my own creation, and I attest, based on my understanding and conviction, that it does not include any content authored or published by someone else previously (unless explicitly credited), nor has it been substantially used for the fulfillment of any other academic degree or certification from any university or educational institution..

Signature: Shaomei Tang Date: 22 May 2024

Acknowledgment

Above all, I extend my heartfelt appreciation to my family for their unwavering support. The unwavering companionship and support in every aspect from them have enabled my successful completion of my Master's degree program at Auckland University of Technology (AUT), New Zealand..

Additionally, I am deeply grateful to Wei Qi Yan, my primary supervisor, for his invaluable guidance and expertise throughout this project. This is evident not only in his supervision of my project but also in his imparting of professional knowledge and the latest technologies to us students during our weekly meetings. His mentorship not only enriched my learning journey but also played a crucial role in helping me achieve my academic objectives. Furthermore, his meticulous and perfectionist scholarly attitude has greatly benefited me. I am also deeply grateful for his patience, especially when he tirelessly explained problems to me. Without his supervision, I could not have finished my studies at AUT. Meanwhile, I also appreciate Mr. Zhikang Chen for his valuable advice and support in establishing the development environment and the school administrators of AUT who provide supports throughout the Research Project.

Shaomei Tang

Auckland, New Zealand

May 2024

Chapter 1
Introduction

This chapter consists of five parts. The background and motivations are presented in the first section, followed by the research inquiry in the second part. Subsequently, the contributions, aims, and organization of this document are delineated.

1.1 Background and Motivation

Nowadays, we are more concerned about their health than ever before, and obesity has emerged as a significant global health issue due to its association with an increased risk of diseases such as heart disease, diabetes, and hypertension (Mansoor et al., 2022). An effective method to prevent obesity is through controlling the calorie intake in food (Rolls, 2007). In daily diets, fruits and vegetables play a crucial role as primary sources of nutrition. However, many individuals lack understanding regarding the calorie and nutritional content of various foods, necessitating a method to help them easily comprehend the calorie content of their food intake (Veni et al., 2021). With the advancement of technology, various artificial intelligence systems have been researched to facilitate people in understanding the daily calorie intake of fruits and vegetables, aiding them in better diet control, such as the research of Begum et al., (2022). This project proposes a deep learning model to calculate the calories in fruits.

According to Vaswani et al., (2023), transformer architecture was initially devised for tasks related to natural language processing (NLP) but has been so successful that deep learning models based on it have flourished and exhibited exceptional performance across various computer vision tasks, notably in object detection. The framework for real-time detection of objects utilizing the transformer architecture is Real-Time Detection Transformer (RT-DETR) (Lv et al., 2023), which has achieved impressive accuracy in real-time object detection. The motivation behind our project is to utilize the features of the RT-DETR model to create a system that can detect fruits in real-time using a camera feed and estimate their calorie content. By automating these processes, we can streamline workflows, improve efficiency, and provide users with valuable insights into their dietary habits.

Through this project, we explore the capabilities of RT-DETR in fruit detection and calorie estimation, evaluate its performance with existing methods like YOLOv8, and showcase its potential for practical use in dietary monitoring and nutrition analysis.

1.2 Research Questions

The report aims to investigate the application of deep learning models based on transformer architecture for estimating calorie content in fruits. Hence, the research queries addressed in this document are as follows,

(1) *What techniques can be utilized for fruits calorie estimation utilizing deep learning?*

(2) *How well does deep learning technology perform in estimating the calorie content of fruits.*

The project focuses on utilizing deep learning for fruits detection and calorie estimation. Therefore, appropriate techniques need to be selected for fruits identification, detection, and calorie estimation. In addition, the methods used in this research project require evaluation.

1.3 Contributions

The focus of this project is on utilizing the real-time detection capabilities of the model of object detection RT-DETR, employed the transformer architecture, for detecting fruits and estimating their calories. Moreover, we compare the performance of RT-DETR and other state-of-art object detection model YOLOv9. By evaluating these models using real-world fruit images captured from videos, we provide insights into their performance and weaknesses in practical scenarios. the project contributes to the advancement of computer vision research by showing how well transformer-based models such as RT-DETR perform in challenging detection tasks with various objects like fruits.

1.4 Objectives of This Report

The primary aim of the project is to investigate the application of deep learning techniques for estimating the calorie content of fruits. This involves utilizing a diverse dataset of fruit images captured from videos and implementing random augmentation techniques during training to enhance model robustness. The project aims to use the RT-DETR model integrated into the ultralytics framework for fruit detection and calorie estimation, aiming to achieve higher accuracy and efficiency compared to traditional methods. Additionally, comparative experiments between the RT-DETR model and YOLOv9 model are conducted to evaluate performance metrics such as F1 score, precision, recall, and mAP.

1.5 Structure of This Report

We outline the structure of the report here:

- In Chapter 2, We conduct an extensive literature review and delve into the research progress related to object detection, particularly its applications in food and fruit detection.
- In Chapter 3, We provide a detailed overview of our research methodology. This chapter encompasses the specifics of experimental design, data collection, and evaluation methods.
- In Chapter 4, We display the collected training and practical detection outcomes, illustrating them through visual aids such as charts and graphs. Additionally, we delve into the limitations of our study.
- In Chapter 5, We offer a comprehensive review and in-depth analysis of the experimental outcomes, aiming for a thorough understanding of our research findings.
- In Chapter 6, We summarize our research and its findings, and propose directions for future studies

Chapter 2
Literature Review

This report centers fruit detection using transformer-based methods. In the section, we will delve into the utilization of deep learning techniques in the evolution of object detection methods and explore related research.

2.1 Introduction

The detection of visual objects remains the cornerstone task in the realm of computer vision. Broadly, current object detection frameworks can be divided into two main groups: CNN-based and Transformer-based. Within CNN-based frameworks, a further division can be made between two-stage detection methods and one-stage detection methods. Two-stage detection methods include models like Faster R-CNN, while one-stage detection methods encompass SSD and the YOLO series. As for Transformer-based frameworks, the DETR series, such as Swin Transformer, represents the primary example (Arkin et al., 2023).

2.2 Convolutional Neural Network-Based Architecture

A specialized deep learning architecture tailored for image identification and classification tasks is the Convolutional Neural Network (CNN) (Bhatt et al., 2021). Its core architecture consists of hidden layers and a classification section. In the hidden layers, convolutional and pooling layers play crucial roles. Convolutional layers identify different characteristics in the image, including textures, edges, and shapes, by applying filters. Pooling layers decrease the dimensions and intricacy of the image, reducing computational load and helping the network better understand the overall structure of the image. In the classification section, fully connected layers transform the extracted features into prediction results, using activation functions to introduce non-linearity, allowing the network to better fit the data. Additionally, regularization layers are often used to prevent overfitting by limiting the complexity of the model, thereby improving generalization.

Training a CNN model typically involves two stages: Forward propagation and backward propagation. In the forward propagation stage, input images are passed through each network layer, generating prediction results. In the backward propagation stage, network parameters are updated using the gradient descent algorithm to minimize the loss

function based on the difference between the prediction results and the ground truth labels, continuously optimizing the model's performance. CNNs have wide applications in the fields of image processing and image classification. They have been successfully applied in various domains such as face recognition, object detection, and medical image analysis, achieving significant results.

2.2.1 Two-Stage Detectors

Visual object detectors based on CNN are classified into two main groupings: One-stage and two-stage. The primary difference between the two approaches lies in whether region proposals are generated. In general, two-stage detector methods typically consist of two stages: The first stage involves extracting deep features from the input images utilizing a backbone network like ResNet. Following that, the Region Proposal Network (RPN) is employed to generate potential regions, categorizing the image into background and target regions, and making preliminary predictions about position of the target. In the second stage, the Roi_pooling layer is utilized to precisely locate and refine the positions within the potential regions. These candidate targets are then mapped to corresponding feature regions on the feature map, followed by passing through a fully connected layer (FC) to acquire the respective feature vectors. Finally, the classification and regression branches are utilized to identify the class and position of candidate targets (Du et al., 2020; Lu et al. 2020).

The two-stage models include R-CNN (Region-based Convolutional Neural Network) (Bharati, & Pramanik, 2020), Fast R-CNN (Girshick, 2015), Faster R-CNN, R-FCN (Region-based Fully Convolutional Network) (Dai et al., 2016), FPN (Feature Pyramid Network) (Lin et al., 2017), and Mask R-CNN. Two-stage detectors typically achieve higher accuracy levels but operate at a slower speed compared to one-stage detectors. (Carranza-García et al., 2020).

Wasif et al. (2021) accommodated the Faster R-CNN method to detect ten different types of food and calculate their calories. They chose the Faster R-CNN algorithm

because it offers faster speed compared to other available algorithms. Faster R-CNN is a specialized method for object detection, comprising three main components: a backbone network (CNN) that is to extract features of objects. Another component is the Region Proposal Network (RPN), which generates target bounding boxes. Lastly, the detection network performs classification and regression of targets (Sarda et al., 2020). The detection algorithm obtained over 90% precision for all images.

2.2.2 One-Stage Detectors

One-stage object detectors, which utilize a single feedforward fully convolutional network to directly provide target bounding boxes and classifications. Early models like Single Shot MultiBox Detector (SSD) (Liu et al., 2016) and You Only Look Once (YOLO) (Jiang et al., 2022) pioneered this unified architecture, eliminating the need for per-proposal computation. However, these models often struggle with extreme foreground-background class imbalances, limiting their accuracy. This disparity poses a challenge in real-world scenarios where object and background proportions vary significantly. To address this, researchers propose enhancement techniques like sample weighting and adjusted loss functions to improve detector performance and accuracy (chen et al., 2021).

YOLO relies on CNN models and is predominantly employed for tasks like object identification, character segmentation, and precise target localization through annotations. This algorithm stands out as a well-established approach for extracting features in real-time scenarios (Jiang et al., 2022).

Xiao et al. (2024) conducted a study on identification of fruit ripeness utilizing the YOLOv8 model. The research involved extracting visual features from images of fruit and analysing peel characteristics to predict fruit categories. They utilized a custom dataset created by themselves and employed PyTorch as the experimental platform. It was observed that using all the dataset for model training led to redundant visual features, prompting manual removal of some data. Ultimately, they trained the model using two thousand samples. Experimental results showed a significant improvement in

classification accuracy, reaching 99.5%, with the application of the C2f module in the YOLOv8 model. During the training process, the authors also evaluated the model's performance and noted that insufficient training iterations could affect the convergence of the model.

To leverage the benefits of both one-stage and two-stage detectors while addressing their respective limitations, some studies have explored the effectiveness of hybrid architectures in object detection (Arkin et al., 2021). For instance, Agarwal et al., (2023) employed a hybrid architecture to predict food calories. In this study, image segmentation was initially conducted using Mask R-CNN, followed by feature extraction and food classification using the YOLOv5 framework. Subsequently, the dimensions of food items were determined by identifying them, and their quantities and calories were computed using the estimated dimensions. These methods achieved an accuracy of 97.12% on the training dataset, surpassing other classification models such as CNN, YOLO, and Mask RCNN across various evaluation criteria, leading to higher accuracy and fewer errors. Mask RCNN is an extension of Faster RCNN that adds a segmentation task to the original classification and regression tasks. It introduces a binary mask for each region of interest, enabling precise image segmentation alongside object classification and bounding box regression. This approach improves accuracy by accurately delineating object boundaries in images (He et al., 2017).

Two-stage detectors typically achieve higher accuracy but are slower in comparison to one-stage detectors, as noted by Carranza-García et al. (2020). One-stage detectors excel in quick processing, making them suitable for real-time applications. However, their lower precision poses challenges for tasks requiring high accuracy. The future trend is more centered on combining precision and speed in real-time applications to achieve high accuracy (Cao et al., 2021; Bharati, & Pramanik, 2020).

2.3 Transformer-Based Architecture

Since the successful application of the transformer in NLP tasks, there has been ongoing

effort within the industry to adapt the Transformer architecture for applications in computer vision (CV) (Jamil et al., 2023; Bi et al., 2021).

Xiao et al. (2023) utilized Swin Transformer model to identify apple ripeness from digital images. The Swin Transformer was developed by the research team at Microsoft Research Asia. It is a deep learning model that utilizes a transformer architecture and employs hierarchical grouping attention mechanisms. It has demonstrated impressive efficacy across numerous tasks of CV, like image categorization, target identification, and semantic partitioning (Liu et al., 2021). In the research the researchers also evaluated the detection outcomes of the Swin Transformer with the results of the YOLOv5 model and DETR. Based on the detection outcomes, the YOLO model exhibits superior detection performance and greater stability. The Mask RCNN and Swin Transformer both demonstrate rapid and consistent detection capabilities. Nevertheless, integrating the transformer mechanism into the YOLO model did not result in improved outcomes.

Based on recent publications, transformers show significant promise in tackling computer vision tasks. Additionally, the integration of CNNs and transformers has led to enhanced efficiency (Arkin, 2023). The key lies in combining the strengths of Object detection techniques employing CNNs and Transformers to achieve rapid and precise detection of targets in real-world scenarios.

Chapter 3
Methodology

This part primarily elaborates on the methodology of research adopted in this report, encompassing the detection of fruits using models based on the Transformer architecture, along with the evaluation methods employed in this project.

3.1 Transformer

Transformer architecture (Vaswani et al., 2017), is specifically crafted for processing data of sequence, for example, the words in sentence. It processes incoming sequences and converts them into other sequences. It utilizes self-attention exclusively to calculate its input and output representations, eliminating the need for sequence-aligned RNNs (Recurrent Neural Networks) (Manaswi, 2018) or convolution. The architecture comprises an encoder and a decoder, illustrated in Figure 3.1.

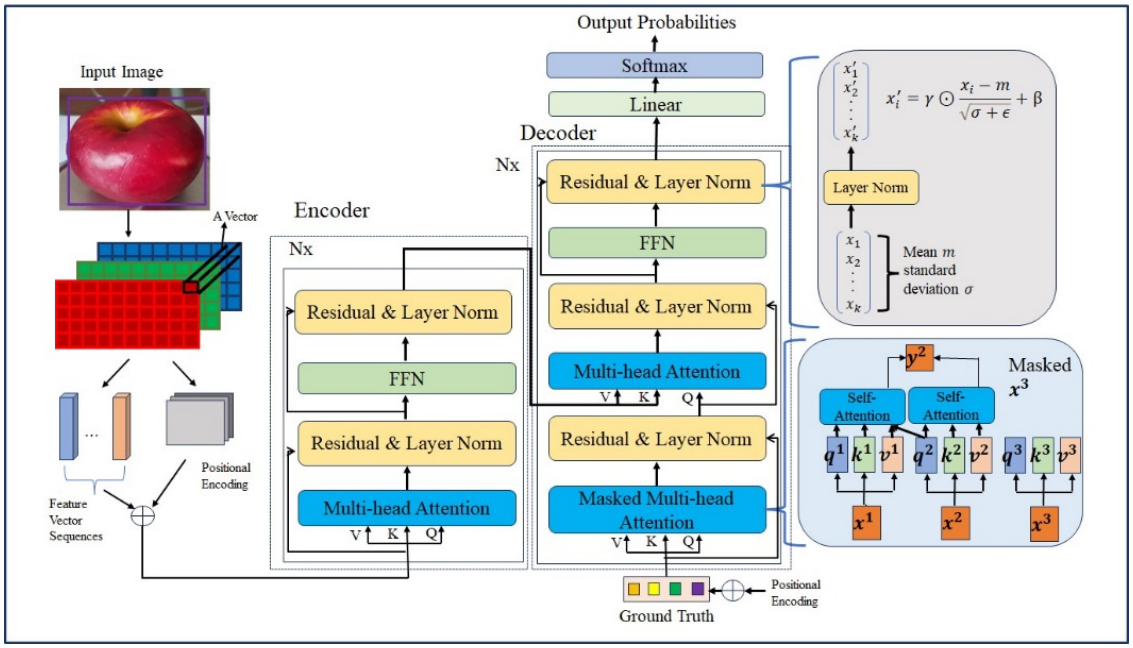


Figure 3.1: The transformer architecture

3.1.1 Encoder

The Encoder is composed of N (equal to6) identical layers, where each layer corresponds to a unit depicted on the "Encoder" block of the diagram, denoted as *Nx* where *x* ranges from 1 to 6. Each layer contains two sub-layers: A multi-head self-attention and a FFN (fully connected feed-forward network). Both sub-layers include residual connection (He et al., 2016) and layer normalization (Yu et al., 2023), allowing the output of the sub-layer to be represented as follows:

$$sub\_layer\_output = LayerNorm(x + Sublayer(x)) \quad (3.1)$$

where *Sublayer*(x) is defined as the operation performed by the sub-layer itself.

The LayerNorm isrepresented as follows, illustrating in the diagram:

$$x_i' = \gamma \odot \frac{x_i - m}{\sqrt{\sigma + \epsilon}} + \beta \tag{3.2}$$

where $x \in R^{N \times D}$ represents a target tensor consisting of N tokens $x_i \in R^{1 \times D}$, $x(i)$, m and σ indicate the mean and variance of x_i, respectively. The variable ϵ represents a small constant used for division stability, and the operation $\odot$ indicates element-wise multiplication. The trainable affine transformation coefficients are denoted by γ and β belonging to $R^{1 \times D}$.

(1) Encoder sub-layer 1: Multi-head self-attention mechanism

An attention function can be described as a mechanism that takes a query and a collection of pairs of key-value as input and produces an output. This process involves vectors representing the query, keys, values, and output. In practice, we calculate the attention function for a set of queries concurrently, which are arranged into a matrix Q. Correspondingly, matrices K and V organize the keys and values. The output can be represented as,

$$attention\_output = Attention(Q, K, V) \tag{3.3}$$

where the attention calculation takes use of a scaled dot-product mechanism:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

where d_k represents the dimensions of the queries and keys.

Multi-head attention involves projecting *Q*, *K*, and *V* through *h* distinct linear transformations, followed by concatenating the resulting attention results:

$$MultiHead(Q, K, V) = Concat(head_i, \dots, head_h)W^O \tag{3.5}$$

$$\text{where } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3.6}$$

where the projections consist of matrices that act as parameters $W_i^Q \in R^{d_{model} \times d_k}, W_i^K \in R^{d_{model} \times d_k}, W_i^V \in R^{d_{model} \times d_v}$ and $W^o \in R^{hd_v \times d_{model}}$,

However self-attention uses the same *Q*, *K*, and *V*.

(2) **Encoder sub-layer 2: position-wise fully connected feed-forward network**

This layer is a fully connected layer and mainly provides linear transformations.

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_2) \tag{3.7}$$

While the linear transformations remain consistent regardless of the position, they employ distinct parameters from one layer to another.

3.1.2 Decoder

The decoder, akin to the encoder, also comprises N layers stacked together, being divided into 3 sublayers (as shown in the right part of Figure 3.1). From the diagram we can see that:

(1) Decoder sub-layer 1 utilizes masked Multi-Headed Attention. This masking, along with the adjustment of shifting the output embeddings at each position, guarantees that the forecasts for position *i* can depend entirely on the established outputs at positions prior to i. For instance, as illustrated in the bottom right corner of the Figure 3.1, the inputs q^1, k^1, v^1 from x^1, and q^2, k^2, v^2 from x^2 are processed to compute the output y^2, while x^3 is masked and not included in the computation.

(2) Decoder sub-layer 2 is an encoder-decoder multi-head attention. The input to this layer comes from the encoder's key and value, as well as the query of the next layer in the decoder.

3.2 DETR

The transformer was primarily developed for natural language processing, but due to its powerful modeling capability and parallel computing ability, researchers have applied it to fields such as visual object detection (Samplawski & Marlin, 2021). The task of Visual object detection from digital images involves identifying objects in images and determining their positions and categories, typically involving image or video data (Carion et al., 2020).

In the aspect of feature acquisition, transformers have larger receptive fields, more adaptable weight settings, and better global modeling capabilities compared to CNNs, making Transformer-based backbone networks potentially deliver feature inputs of superior quality inputs for subsequent tasks. In 2020, Google introduced the Vision Transformer (ViT) developed by Dosovitskiy et al. (2021), successfully applying this model to image classification tasks. Subsequently, transformer innovations in the realm of CV emerged.

The DETR (Carion et al., 2020) model was investigated in the same year, employing an end-to-end transformer architecture to transform the object detection task into a sequence-to-sequence problem, simplifying the detection process and effectively eliminating the necessity for many manually crafted components like NMS or anchor generation, achieving good performance. However, DETR has a few drawbacks, such as slow training, large computational overhead, and poor performance on detection of small objects. (Carion et al., 2020). As a result, various variants of DETR have been proposed, each targeting specific challenges: PnP-DETR introduces a “poll and pool” sampling module to adaptively sample features of different granularity, balancing computational overhead and performance (Wang et al., 2022).

Deformable DETR reduces computational overhead by altering the attention mechanism calculation method, leveraging deformable convolutions to improve small object detection performance (Zhu et al., 2021). Sparse DETR further reduces computational costs by selectively updating only a portion of encoder tokens, maintaining detection performance (Roh et al., 2022). Conditional DETR decouples appearance and position features to speed up convergence by learning conditional space queries (Meng et al., 2023). Anchor DETR model shows a new object query design by using anchor points to guide optimization and accelerate convergence (Wang et al., 2022). DAB-DETR builds on Anchor DETR by introducing 4D reference points to further accelerate convergence (Liu et al., 2022). DN-DETR addresses slow convergence by stabilizing training with noisy ground truth and query inputs (Li et al., 2022). These methods aim to expedite DETR's convergence speed and have demonstrated significant effectiveness in

experiments conducted on the COCO dataset. Throughout innovative approaches targeting different aspects of the detection process, these variants contribute to advancing the performance and efficiency of object detection models.

3.3 RT-DETR

In our project, a DETR model named RT-DETR is employed for fruit detection and calorie estimation.

While the DETR series has made significant progress in recent years and has to some extent disrupted the dominance of CNNs in the realm of detection of objects, in terms of “practicality” DETR still cannot fully replace, or even match, the YOLO series. The high computational expenses and typically extended training durations associated with DETR have imposed limitations on its broad implementation within practical production operations. However, with the advent of RT-DETR (Li et al., 2023), this impasse was decisively overcome. RT-DETR not only effectively tackled the issue of "two sets of thresholds" but also made substantial strides in enhancing its practical utility, thereby simplifying deployment processes. These advancements have empowered RT-DETR to fulfill the demands of detection of real-time and have found extensive utilization in practical applications.

The architecture of RT-DETR is illustrated in Figure 3.2. In terms of structure, RT-DETR consists of three blocks: Backbone network, neck network, and decoder.

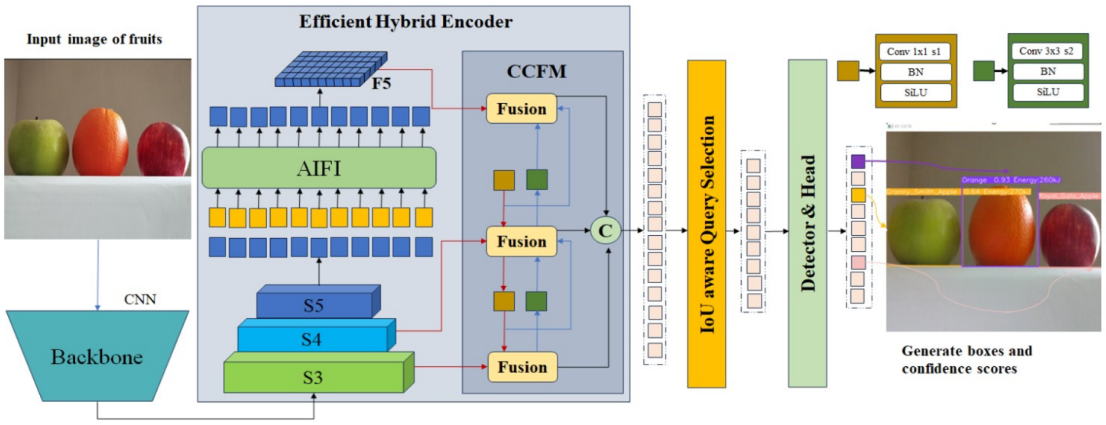


Figure 3.2: The architecture of RT-DETR

3.3.1 Backbone

The backbone network of RT-DETR takes use of ResNet50, ResNet101 (Dosovitskiy et al., 2021; He et al., 2019; He et al., 2016), and HGNet-v2 (Yao et al., 2023). These backbone networks all utilize CNN architecture. The table 3.1 lists backbones used in RT-DETR. The backbone can be scaled, and the publicly available HGNetv2 has two versions: L and X. Like previous detectors, RT-DETR also extracts outputs at three scales, S3, S4, and S5, from the backbone network. In this project, we trained the RT-DETR-L model on our dataset by using the HGNetv2 backbone. This choice was motivated by the limitations of our training environment, which consists of only one GPU. Given the lower parameter count of the RT-DETR-L model, it was deemed more suitable for our setup.

Table 3.1 Backbones used in RT-DETR

Model	Backbone	Parameters
RT-DETR-R50	R50	42
RT-DETR-R101	R101	76
RT-DETR-L	HGNetv2	32
RT-DETR-X	HGNetv2	67

3.3.2 Neck: Hybrid Encoder

For the neck network, RT-DETR employs a solitary layer of Transformer encoder, exclusively processing the S5 features outputted from the backbone network, as shown in Figure 3.2, called AIFI (i.e., the Attention-based Intra-scale Feature Interaction) module. The mathematical operations of AIFI can be represented as follows:

$$Q = K = V = Flatten(S_5) \tag{3.8}$$

$$F_5 = Reshape(Attn(Q, K, V)) \tag{3.9}$$

where *Attn* means the multi-head self-attention, and *Reshape* is utilized to revert the feature's shape back to that of S_5.

The two-dimensional S5 features undergo flattening into a vector before being passed to the AIFI module. The computational process entails multi-head self-attention and FFN (Feed-Forward Network). Subsequentially, the output is reshaped back into two dimensions, represented as S5, for further “cross-scale feature fusion.” According to the RT-DETR research team, the decision of RT-DETR to only process the final S5 feature through AIFI is based on two considerations:

(1) Previous DETR models, such as Deformable DETR, concatenated features from multiple scales into one long sequence vector. While this approach facilitates ample interaction between features at different scales, it also leads to significant computational overhead and time consumption. RT-DETR considers this as one of the primary reasons for the slow computation speed of existing DETR models.

(2) In RT-DETR, compared to shallower features like S3 and S4, the S5 features possess deeper, more advanced, and enhanced semantic information. These semantic features offer greater value and utility for Transformers to distinguish between different objects. In contrast, shallow features lack significant semantic information and are less effective.

The RT-DETR demonstrates that applying the Encoder only to the S5 features can significantly reduce computational complexity, improve computation speed, and maintain model performance.

IoU-aware Query Selection. IoU-aware query selection is introduced to guide the model during training. This approach enhances the classification by assigning higher scores to features with high IoU (Yan, 2023) scores and lower scores to those with low IoU scores. This improves the quality of initial object queries for the decoder, thereby enhancing detection performance.

Therefore, to address the latency issues caused by NMS (Non-Maximum Suppression) (Jiang et al., 2019) in current real-time detectors, RT-DETR introduces a real-time end-to-end detector which comprises two critical enhancements. Firstly, a hybrid encoder is designed to efficiently process multi-scale features. Secondly, IoU-

aware query selection enhances the initialization of object queries. The combination of these improved components enhances the performance of our detector in real-time scenarios.

3.3.3 Decoder

RT-DETR supports flexible tuning of inference speed by employing varying numbers of decoder layers, eliminating the necessity for retraining, enabling the model to adapt to various real-time scenarios.

3.4 Training Data

3.4.1 Data Selection

Due to seasonal variations in data collection, the artificial neural network in this project is trained utilizing the following fruit categories to cover the most fruits likely to be encountered by the calorie detection system: Royal Gala Apple, Rose Apple, Granny Smith Apple, Ambrosia Apple, JAZZ Apple, Orange and Kiwifruit.

3.4.2 Dataset

We created a dataset comprising 1,866 images of various fruits for fruit detection. Through using a camera, we captured videos of each fruit from multiple angles at equidistant distances and the images were obtained by extracting frames the videos. The dataset consists of seven classes of local fruits products. To identify fruit categories and estimate their calorie content, each fruit was classified into weight categories of equal intervals, resulting in a total of 22 categories.

For example, for Royal Gala Apples, weight categories were defined as follows: “Royal_Gala_Apple 1” for weights up to 140g, “Royal_Gala_Apple 2” for weights between 140g and 180g, and “Royal_Gala_Apple 3” for weights exceeding 180g. Due to the equidistant capture method, the images of fruits of different weights have varying

dimensions, allowing the deep learning model to estimate calorie content based on image dimensions. The calorie and nutrient composition data for the 7 classes of fruits were sourced from “The Concise Food Composition Tables” jointly own by The New Zealand Institute for PFR (Plant & Food Research Limited) and MoH (the Ministry of Health, New Zealand) (Lister, 2018). These data are utilized for energy estimation during the fruit detection process. Additionally, starting with these 1,866 fruit images as a foundation, we employed various data augmentation techniques to generate a dataset comprising 4,478 images. This dataset was subsequently partitioned into training, validation, and testing sets in the proportions of 87%, 8%, and 4% respectively. Specifically, the dataset includes 3918 images for the training, 374 images for the validation, and 186 images for the testing.

3.4.2 Data Pre-processing

To ensure the neural networks for tasks such as classification of images and object detection are trained effectively, it is essential to adjust the size of the images to a predetermined size that matches the initial input layer of the neural network. This is the reason why convolutional layers in neural networks analyze images pixel by pixel and the interactions with neighboring pixels to identify features. Given the use of the ultralytics framework in this project, we have standardized image dimensions to 640*640, compatible with ultralytics specifications.

In the field of deep learning, data augmentation is a method to increase the scale and diversity of training data by transforming inputs. These transformations involve operations such as rotation, flipping, scaling, cropping, and color transformations. By exposing the model to variations in angles, lighting, and scales, data augmentation aids in learning diverse features. Additionally, it reduces the model's reliance on specific samples, thereby improving the model's generalization ability and robustness. (Shorten & Khoshgoftaar, 2019) In this report, we employed various data augmentations, (1) Horizontal flipping with a 50% probability. (2) No rotation, clockwise rotation, and anti-clockwise rotation in 90-degree increments. (3) Random cropping of 0% to 20% of image size. (4) Random rotation between -15 and +15 degrees. (5) Horizontal shearing between

-10° and +10°, and vertical shearing between -10° and +10°. (6) Random brightness adjustment between -15% and +15%. (7) Random grayscale application to a subset of the training set with a 15% probability. If these data augmentation techniques are applied to the images, we can obtain results similar to the Figure 3.3.

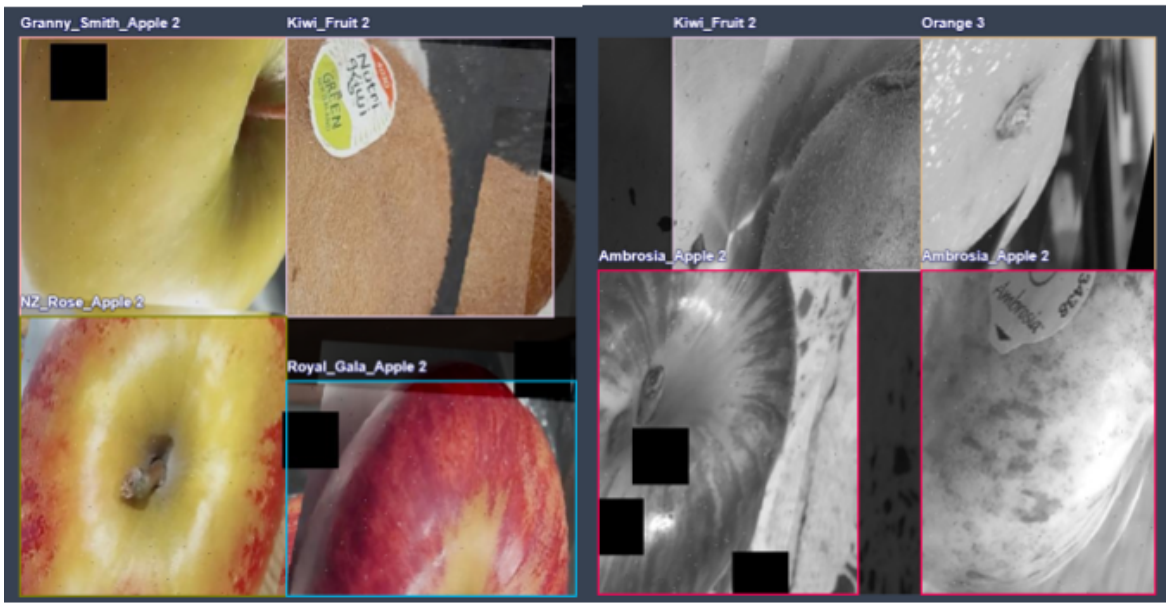


Figure 3.3: The images used in data augmentations

The augmentation mentioned should only be utilized on the training dataset and should not be applied to the validation or testing datasets. It is advised to maintain the testing and validation datasets as similar to the original dataset as feasible to evaluate the robustness of the training conducted using augmented data.

3.5 Program Implementation

The operating environment required in this thesis includes Microsoft Windows 10 or above, Python 3.10 or above programming language, PyTorch 2.1.0 or above deep learning framework, ultralytics object detection framework and CUDA 11.7 or above for accelerated computing.

Furthermore, in order to enhance the robustness of this proposed model, we utilized random data augmentation from the ultralytics framework during model training. ultralytics employs various approaches of data augmentation, and during training process,

we can enhance the generalization ability of the model and reduce overfitting by setting parameters in the configuration file to utilize these random data augmentation techniques. Furthermore, random data augmentation techniques not only generate training samples with different variations, thus enhancing the variety of the training dataset but also help the model generalize better to unobserved data and improve robustness. Moreover, they can generate more training samples by transforming existing data without the need to store additional raw data. Specifically, these data augmentation techniques include:

3.5.1 Random Affine Transformation

This technique involves translating, shearing, rotating, and scaling the image based on the specified parameter values. Then, these transformations are combined to form a comprehensive transformation matrix.

3.5.2 Random Mosaic Augmentation

In the ultralytics framework, mosaic augmentation for both 4-image and 9-image mosaics is defined. We chose to implement random mosaic augmentation using 4 images. To augment training data, four images are randomly cropped and then stitched together into a single image for training purposes. The effect is shown in Figure 3.4:

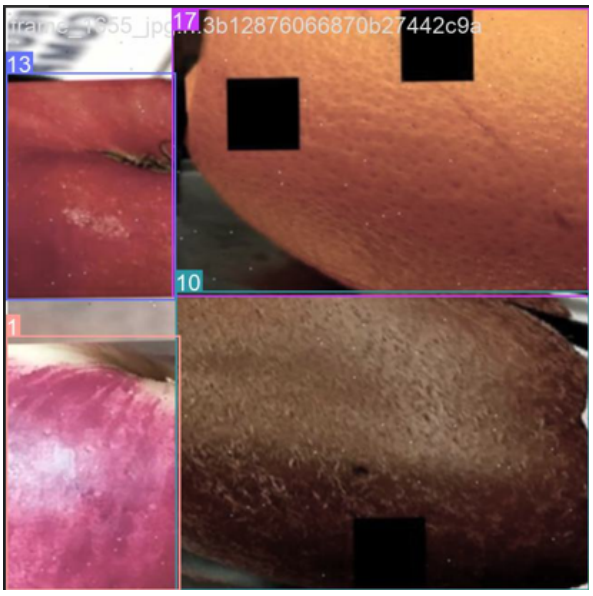


Figure 3.4: Random mosaic augmentation

3.5.2 HSV Augmentation

HSV stands for hue, saturation, and value, collectively representing a color space used to describe colors. HSV enhancement adjusts the values of these three parameters to modify the color and brightness of an image, aiming to improve image quality and enhance the robustness of models without altering the image geometry or structure (Liu et al., 2023). Figure 3.5 demonstrates the combined enhancement effect when these three parameters are applied together as perturbations.

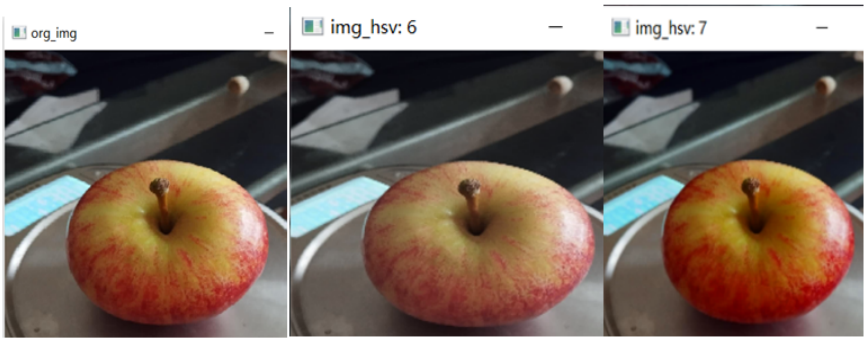


Figure 3.5: Combining HSV augmentation randomly

3.5.4 Albumentations Libraries

Albumentations is a Python library that provides powerful and concise interfaces tailored for various computer vision tasks such as object classification, segmentation, and detection. It surpasses other popular image enhancement tools in terms of speed and also offers a wide range of optimized transformation functions. This adaptability enables users to select augmentation techniques as needed and integrate them into more complex preprocessing pipelines, simplifying the process of establishing data augmentation workflows for various computer vision tasks. Albumentations rapidly enhances model generalization and its performance improvements enhance the efficiency of image enhancement, saving computational resources and time. (Buslaev et al., 2020). We harnessed the Albumentations toolkit for image augmentation, incorporating techniques such as Blur, MedianBlur, ToGray, and CLAHE. During training, we adjusted these parameters to enhance the images. It is essential to install this toolkit before using it.

Figure 3.6 showcases the effects of employing these augmentation techniques:

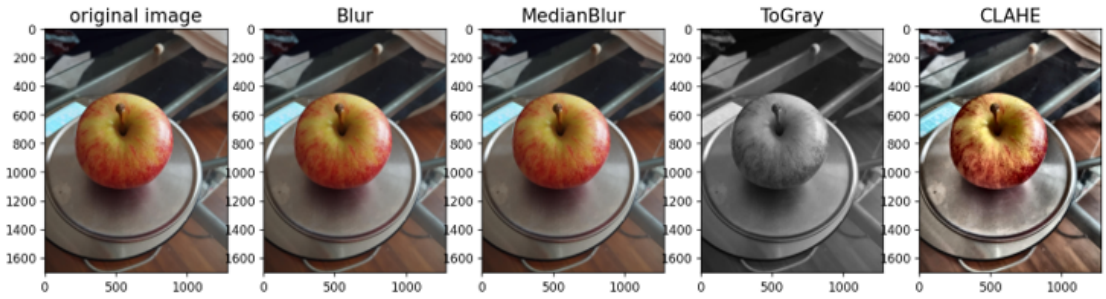


Figure 3.6: Effects of four augmentation techniques from the Albumentations Library

3.6 Evaluation Methods

In this report, we will evaluate RT-DETR by comparing its detection performance with that of YOLOv9 on our dataset since YOLOv9 is the cutting-edge object detector constructed upon CNN architecture. The training performance of YOLOv8 (Yan, 2023) is also used for comparison, as it is a high-performance and relatively stable version of the YOLO model.

In the field of object detection, You Only Look Once (YOLO) combines the conventional two-stage process of predicting the location and classification into a single-stage process, making it very fast in terms of detection speed. The YOLO series (Redmon et al., 2016) include YOLOv1 through YOLOv9. YOLOv9 (Aziz et al., 2024), based on YOLOv7 (Wang et al., 2023), provides two novel technologies, PGI (Programmable Gradient Information) and GELAN (Generalized ELAN), which not only address the issue of information bottleneck but also further push the boundaries of improving the accuracy and efficiency of object detection.

Deep neural networks may encounter problems such as information loss and unreliable gradients when dealing with complex tasks, especially as the network layers increase, leading to potential loss of original data, incomplete information usage during training process, and the generation of unreliable gradients and poor convergence. PGI, as a new auxiliary supervision framework, addresses these issues by introducing an auxiliary reversible branch and multi-level auxiliary information. The auxiliary reversible branch

aims to generate reliable gradients and update network parameters to avoid information loss problems. Multilevel auxiliary information aggregates gradient information containing all target objects and passes it to the main branch to address error accumulation issues. These methods enable the network to better retain information and generate reliable gradients, thereby improving the training effectiveness of deep neural networks. PGI does not require additional connections during inference, thus fully preserving the advantages of speed, parameter quantity, and accuracy.

GELAN integrates the design concepts of CSPNet (Wang et al., 2020) and ELAN (Wang et al., 2023), while considering computational complexity, lightweight, inference speed, and accuracy. This design allows users to choose suitable computation blocks according to various inference devices. Compared to the state-of-the-art depth-wise separable convolution design, GELAN makes use of only traditional convolutions but achieves a higher parameter utilization rate, while also possessing the advantages of being lightweight, fast, and accurate.

Chapter 4
Results

. This chapter primarily validates our research model by analyzing training and experimental results. Additionally, it explores the constraints of the project.

In this report, we evaluated the effectiveness of RT-DETR by conducting a comparative analysis with YOLOv9 on our dataset. In this thesis, we utilize several performance metrics, like confusion matrix, F1 curve, P-R curves, precision, loss-curves, etc. Additionally, all training results refer to the model being trained on our dataset for 100 epochs, with a batch size of 4.

4.1 Confusion Matrix

A confusion matrix serves as a concise overview of the classification outcomes in a given problem. It summarizes the counts of accurate and inaccurate predictions, segmented by each class, which is the pivotal aspect of the confusion matrix. The columns in the confusion matrix denote the predicted classes, while the rows correspond to the real classes. The entries along the diagonal of the matrix indicate the ratio of accurate predictions, while those off the diagonal represent the inaccurate predictions. Optimal performance is reflected by higher values along the diagonal, indicating a multitude of correct predictions (Fahmy, 2022). A typical binary classification confusion matrix is represented as follows:

True Positive (TP): The count of positive instances accurately predicted as positive, i.e., true positive when the actual is 0 and predicted as 0.

False Negative (FN): The count of positive instances inaccurately predicted as negative, i.e., false negative when the actual is 0 and predicted as 1.

False Positive (FP): The count of negative instances mistakenly classified as positive, i.e., false positive when the actual is 1 and predicted as 0.

True Negative (TN): The count of negative instances accurately predicted as negative, i.e., true negative when the actual is 1 and predicted as 1.

For a multi-class confusion matrix, the background is also listed as a separate class, so it also has its own TP, FP, TN, FN, etc. The last column and the last row both represent the FN and FP of the background class predictions. Hence, the bottom-right corner has no significance and no values.

Figures 4.1 and Figure 4.2 respectively display the confusion matrices for the YOLOv9 and RT-DETR models. It is apparent that the RT-DETR model demonstrates a reduction in misclassifications and fewer calorie estimation errors compared to the YOLOv9 model. Additionally, there are fewer missed and false detections for the background class in the RT-DETR model compared to YOLOv9, indicating that RT-DETR is less likely to misclassify the background as fruits. These performances suggest that the RT-DETR model achieves higher detection accuracy.

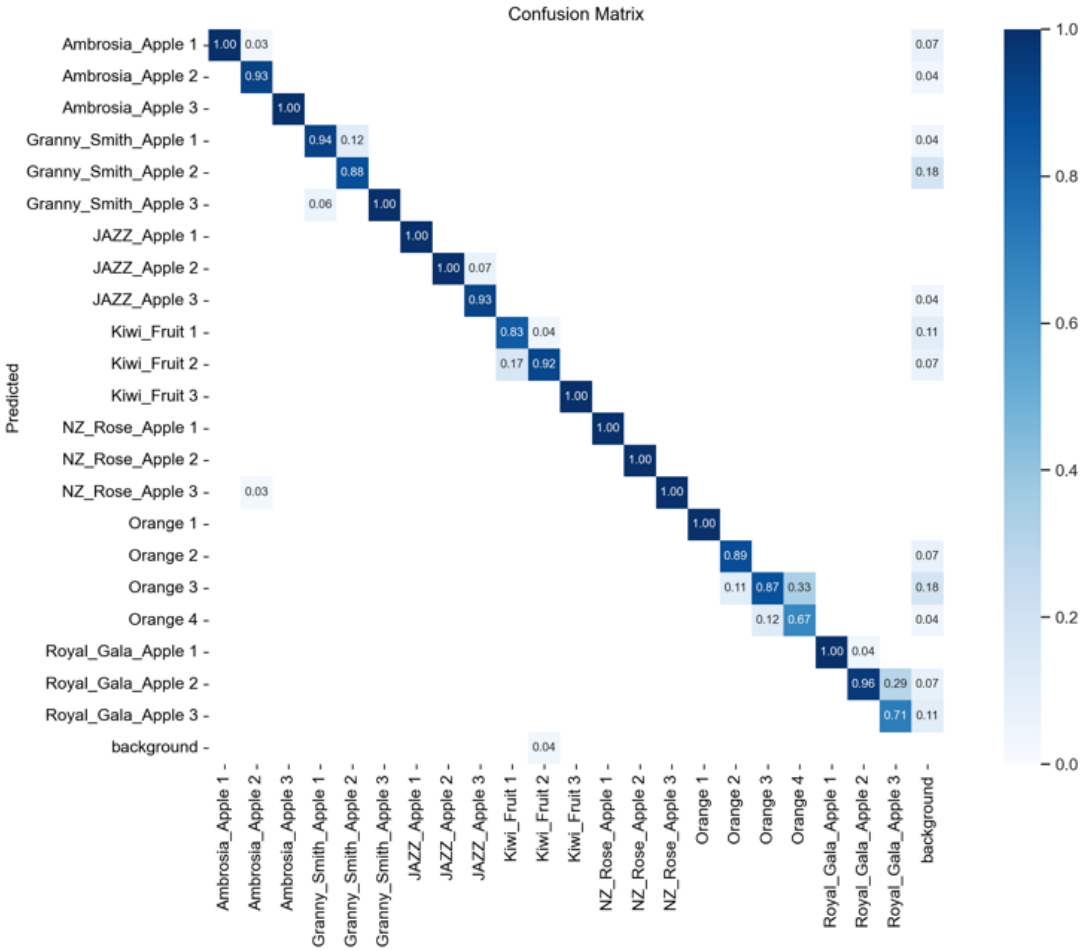


Figure 4.1: Confusion matrix for YOLOv9

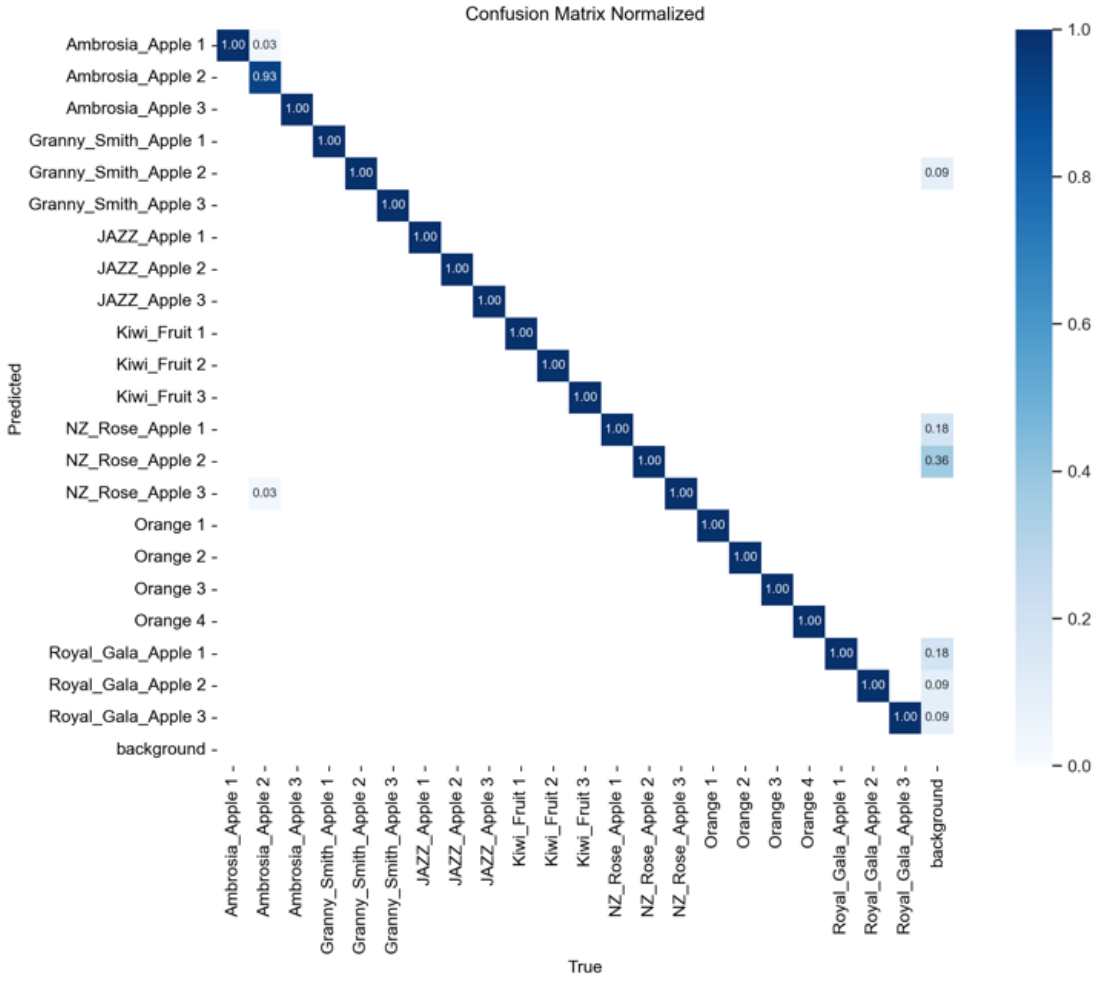


Figure 4.2: Confusion matrix for RT-DETR

4.2 F1-Confidence Curves

The F1 curve represents the harmonic mean of precision and recall (Zhao, & Li, 2020). It ranges from 0 to 1, where a value of 1 indicates optimal performance and 0 indicates poor performance. The F1 score curve shows how the F1 score changes at different thresholds. It is mathematically represented by eq.(4.1) and eq.(4.2) (Padilla et al., 2021):

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.1}$$

$$F_1 = 2TP/(2TP + FN + FP) \tag{4.2}$$

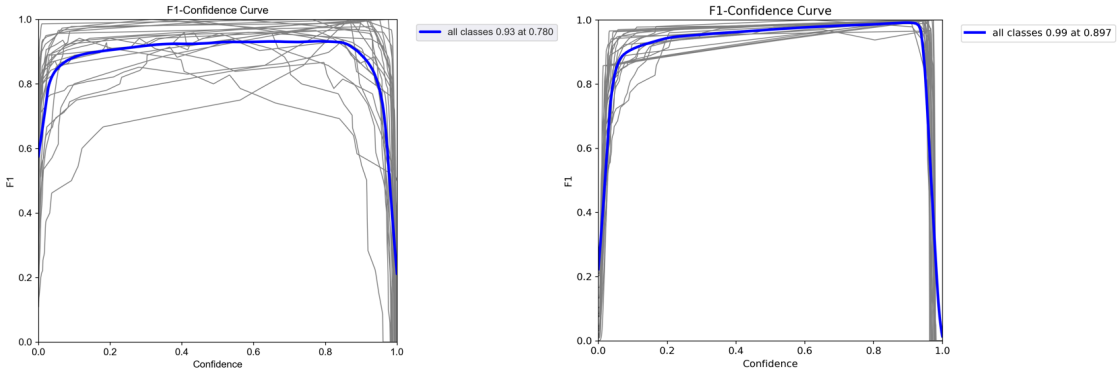
where precision measures the accuracy of detections by indicating the proportion of predicted bounding boxes that correspond to ground truth objects. It reflects how many of the predicted objects are correct. Recall evaluates the capacity of the model to detect

ground truth objects by indicating the proportion of actual objects that are accurately identified.

Figure 4.3 (a), (b) depict the F1-Confidence curves for the YOLOv9 and RT-DETR, respectively.

(1) The peak F1 score is higher, signifying the model's optimal performance. The maximum F1 score for the YOLOv9 is 0.93, while RT-DETR achieves 0.99, indicating an improvement of 0.06.

(2) The region beneath the F1-Confidence curve provides a summary of performance across all thresholds. A greater region indicates superior model performance. The results show that RT-DETR outperformed YOLOv9.



(a) (b)

Figure 4.3: F1 curves for YOLOv9 (a) and RT-DETR (b)

4.3 P-R curves

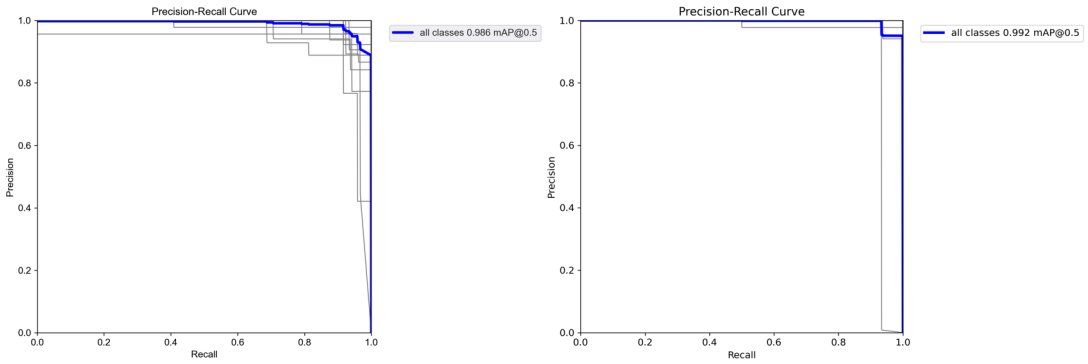
The PR curve showcases the trade-off between precision and recall as well as mAP stands for Mean Average Precision, indicates the average precision across all classes as proposed by Padilla et al. in 2020. It is observed that as precision increases, recall tends to decrease. Therefore, the ideal scenario is to achieve high precision while detecting as many instances of all classes as possible. Consequently, we aim for the curve to approach the point (1,1), indicating maximum precision and recall, and thus maximizing the area under the mAP curve as close to 1 as possible. Figure 4.4 (a), (b) illustrate the P-R curves for

the YOLOv9 and RT-DETR, respectively.

(1) We see that the curve of the RT-DETR model is higher than that of YOLOv9, indicating that the RT-DETR detector exhibits greater precision at different recall levels.

(2) The curve of the RT-DETR model is closer to the upper right corner compared to YOLOv9, suggesting that the overall precision and recall of the RT-DETR detector are better.

(3) The RT-DETR model has a higher AUC than YOLOv9, indicating that the RT-DETR model exhibits better performance.



(a) (b)

Figure 4.4: The P-R curves for YOLOv9 (a) and RT-DETR (b)

4.4 Loss curves

We are use of three loss functions to measure the extent to which the model's predictions deviate from the ground truth, aiming to extensively evaluate performance of the proposed model. They are:

(1) GIoU loss (Localization Loss): This loss function calculates the difference between predicted bounding boxes and ground truth bounding boxes. YOLOv9 model represents it as box_loss. The model employs Intersection over Union (IoU) as a metric to measure the overlap between two bounding boxes. GIoU loss measures the positional precision of the predicted boxes by computing the IoU between predicted and ground truth boxes and

converts it into a loss value. By minimizing the GIoU loss, the model can learn more accurate bounding box positions.

(2) Classification loss (Cls_loss): The model uses classification loss to measure the accuracy of classification. Cls_loss calculates the loss value of classification by comparing the difference between predicted class distribution and actual class labels. By minimizing the classification loss, the model can learn more accurate class classification.

(3) L1 loss (Feature Point Loss): The model utilizes feature points to predict object orientation and angle information, represented as dfl_loss in the YOLOv9 model. L1 loss is employed to compute the disparity between predicted feature points and ground truth feature points. By minimizing the L1 loss, the model can learn more accurate object orientation and angle information.

Figure 4.5 and 4.6 illustrate the loss curves for YOLOv9 and RT-DETR, respectively. It can be observed that:

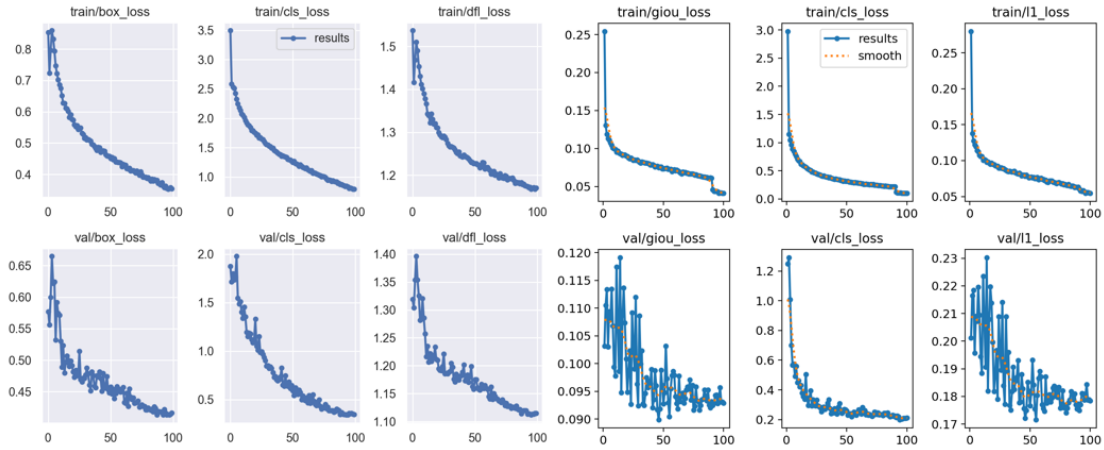
(1) RT-DETR's giou_loss is smaller than YOLOv9 in both training and validation phases, indicating more precise localization.

(2) RT-DETR's dfl_loss is smaller than YOLOv9 in both training and validation phases, indicating stronger capability in object detection.

(3) RT-DETR's cls_loss is smaller than YOLOv9's in both training and validation phases, indicating more accurate classification.

(4) The giou_loss and dfl_loss curves of RT-DETR exhibit more fluctuations during the validation process compared to YOLOv9, indicating that its localization and object detection during validation are more unstable.

The loss values in Table 4.1 further validate the above results.



(a) (b)

Figure 4.5: The loss curves for YOLOv9 (a) and RT-DETR (b)

Table 4.1: Loss values for RT_DETR, YOLOv9 and YOLOv8

	train			val		
	giou_loss/ box_loss	cls_loss	l1_loss/ dfl_loss	giou_loss/ box_loss	cls_loss	l1_loss/dfl_loss
YOLOv8	0.20	0.29	0.89	0.36	0.26	0.92
YOLOv9	0.36	0.85	1.17	0.42	0.34	1.11
RT-DETR	0.04	0.10	0.05	0.09	0.21	0.18

4.5 Precision

We utilize four performance metrics to describe the precision of the model: Precision, Recall, mAP50, and mAP50-95.

(1) Precision refers to the capability of this proposed model to correctly identify and classify only the objects that are pertinent to the given task. Precision evaluates the proportion of correctly predicted positive samples (true positives/all predicted positives). In object detection, a prediction is deemed accurate if the predicted bounding box intersects with the ground truth bounding box (Padilla et al., 2021).

(2) Recall assesses the fraction of all true positive samples that the model can identify. In object detection, a sample is considered correctly recalled if the ground truth bounding box overlaps with the predicted bounding box (Padilla et al., 2021).

(3) mAP50: mAP shorts for mean Average Precision, indicating the mean precision across different classes. mAP50 denotes the value of mAP at a 50% threshold of IoU. In

formal terms, the average precision (AP) for a specific class is derived from the region under the precision-recall curve. AP is obtained by integrating values of precision across all recall levels utilizing numerical techniques (Padilla et al., 2021).

$$AP = \int p(r)dr \tag{4.3}$$

The term *mAP* is utilized to calculate the mean values of AP across all classes.

$$mAP = \frac{1}{nc}\sum AP \tag{4.4}$$

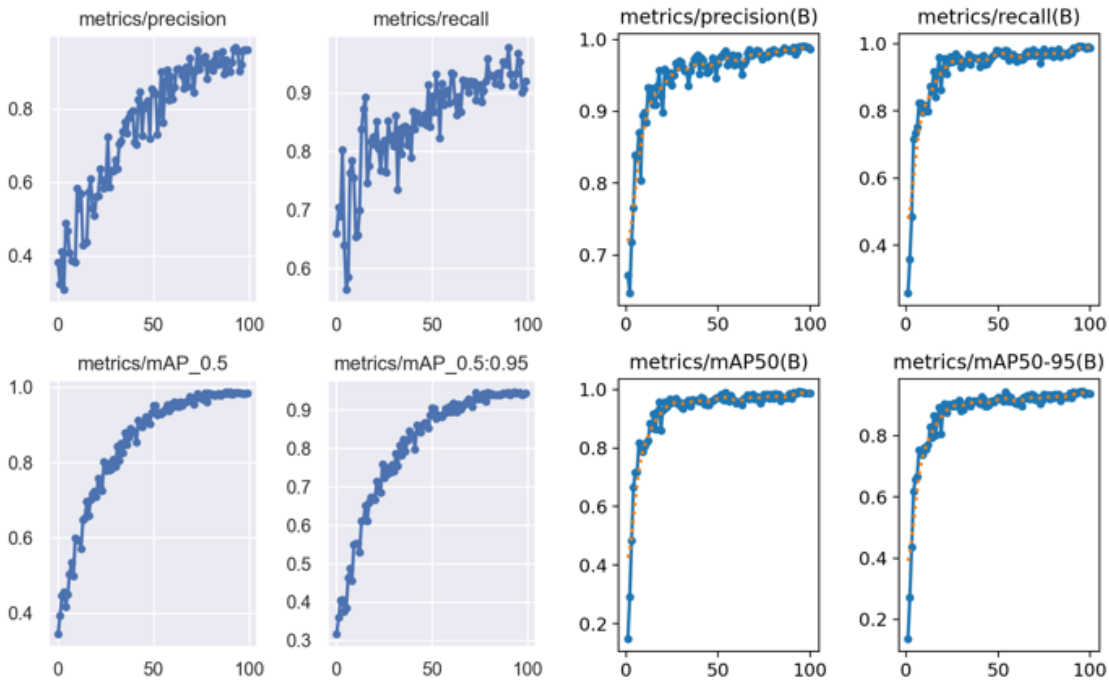
where *nc* is the total count of classes.

(4) The term mAP50-95 is a stricter evaluation metric as it computes the value of mAP across the range of 50-95% IoU thesholds (from 0.5 to 0.95, with increments of 0.05, i.e., 0.5, 0.55, 0.6, ..., 0.95), and then takes the average. This offers a more accurate evaluation of the effectiveness of the model at different IoU thresholds (Padilla et al., 2021). The four plots in Figure 4.7(a) and four in (b) respectively depict the performance of these four metrics for the YOLOv9 and RT-DETR models. From the graphs, we observe:

(1) The precision and recall achieved by the RT-DETR model surpass those of the YOLOv9 model.

(2) The PR curves of the YOLOv9 model exhibit more fluctuations compared to those of RT-DETR, which show relatively fewer fluctuations. Both curves steadily rise.

(3) The mAP50 and Map50-95 curves for both the RT-DETR and YOLOv9 models steadily increase.



(a) (b)

Figure 4.6: The precision, recall and mAP values curves for the YOLOv9 (a) and the RT-DETR model (b). The blue line represents the real metrics values, illustrating how the actual metrics changes with each epoch. The yellow dots depict smoothed results derived from the blue line, capturing the overall trend of the metrics value.

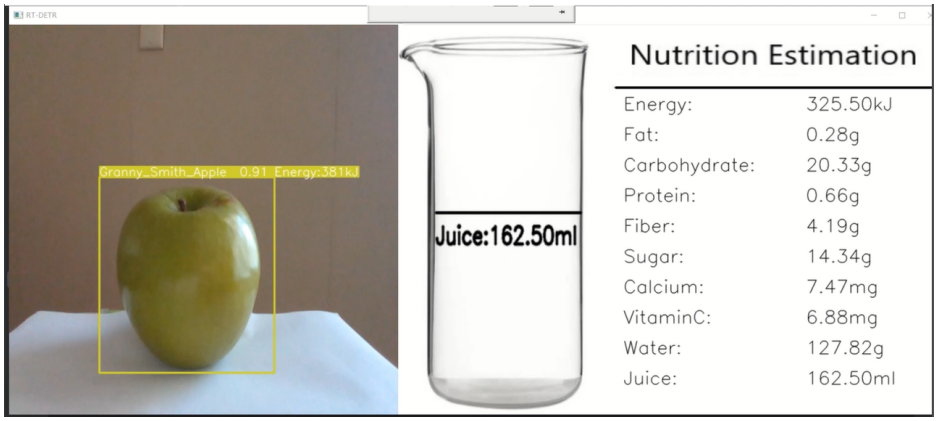
Table 4.2 displays the best performance values of these three models across these four metrics for 100 epochs. Overall, the metrics of the RT-DETR model surpass those of YOLOv8 and YOLOv9, indicating better performance of the RT-DETR model with higher precision in target detection and classification. However, the mAP50-95 of the RT-DETR model is slightly lower than that of YOLOv9, with values of 94.45% and 94.64% respectively. Specifically, the precision rates for RT-DETR, YOLOv9 and YOLOv8 are 99.01%, 96.63% and 94.57%, respectively. In addition, the training time for RT-DETR is shorter than that of YOLOv9 but is five times longer than YOLOv8.

Table 4.2: Performance values of YOLOv8, YOLOv9 and RT-DETR

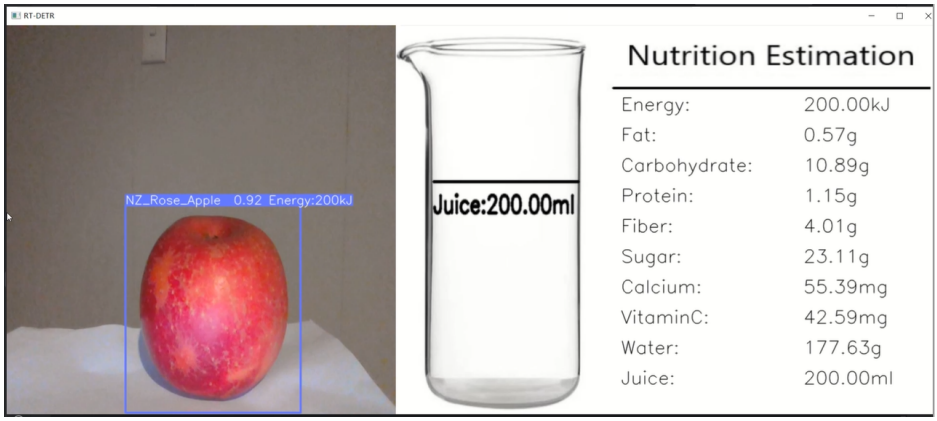
	Precision(B)	**Recall(B)**	**mAP50(B)**	**mAP50-95(B)**	**Training Time**
YOLOv8	94.57%	95.17%	97.87%	93.01%	54 ms
YOLOv9	96.63%	91.32%	98.56%	94.64%	9hrs23ms
RT-DETR	99.01%	99.20%	99.17%	94.45%	6hrs35ms

4.6 Real-Time Detection Results

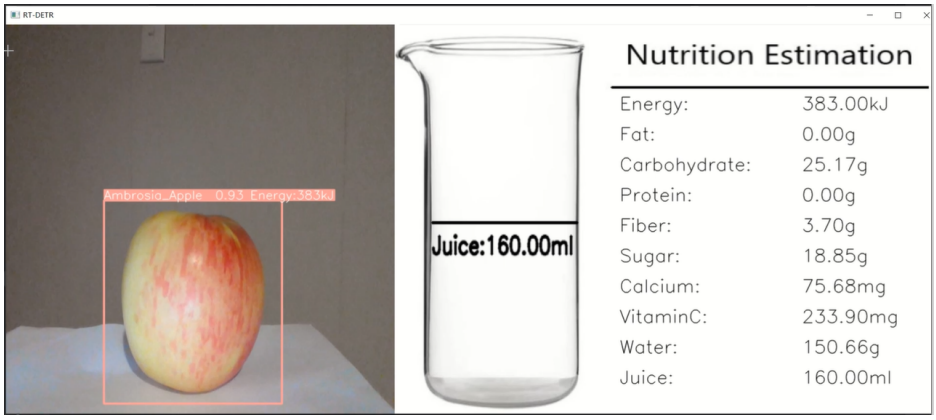
We took use several types of fruits with varying weights for real-time prediction. Additionally, the juice content of fruits of equal weight to the detected fruits were also reflected in the experimental results. Figures 4.7 (a) to (c) and Figure 4.8 (d) to (f) show the detection outcomes generated by using RT-DETR and YOLOv9 models, respectively. Overall, the RT-DETR model exhibits higher detection accuracy and better performance. Additionally, while it might be due to insufficient sample diversity, the two models have calorie estimation errors. For instance, Figures 4.9 (g) and (h) display that a NZ Rose apple originally containing 264kJ of energy is detected by both RT-DETR and YOLOv9 models as 200kJ. The two models occasionally exhibit detection errors. For instance, Figure 4.10 (i) shows the RT-DETR model misidentifying an Ambrosia apple as a Gala apple, while Figure 4.10 (j) shows the YOLOv9 model misidentifying the same apple as a JAZZ apple.



(a)

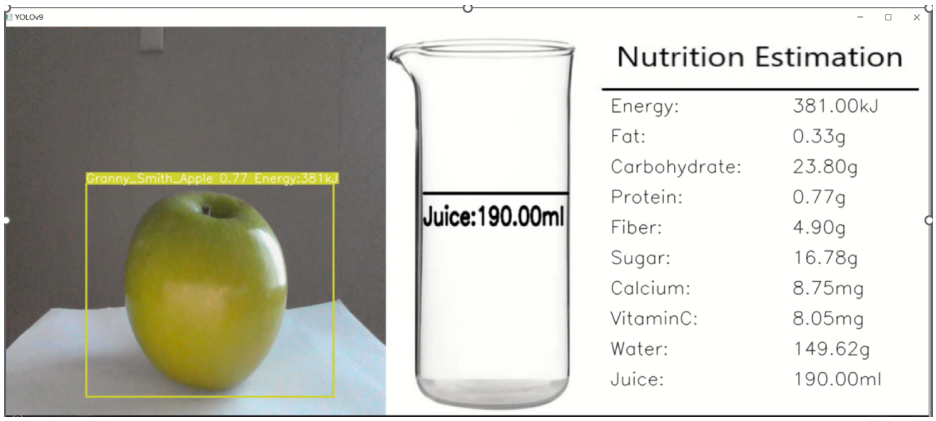


(b)



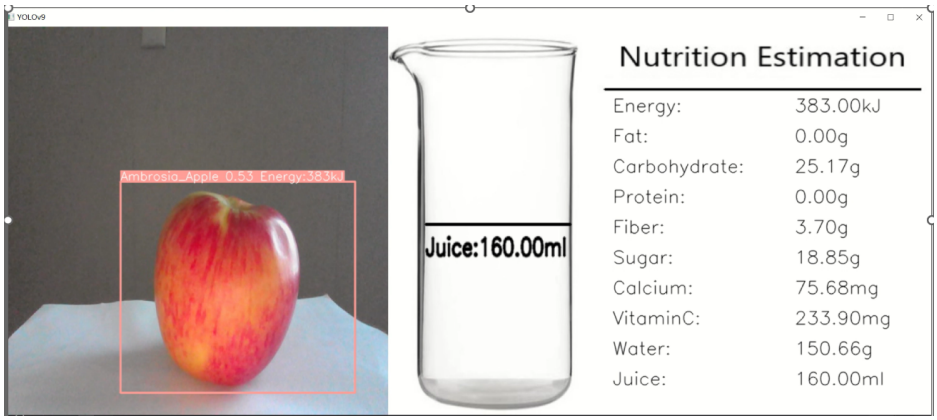
(c)

Figure 4.7 (a) to (c) Prediction of RT-DETR model



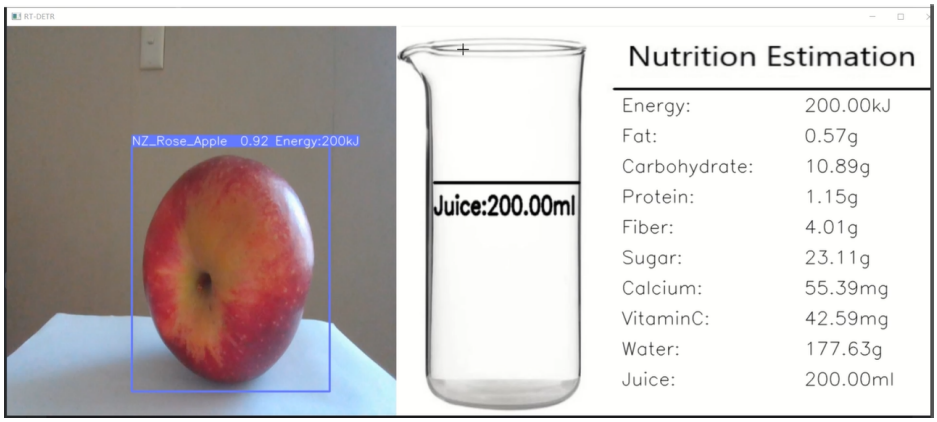
(d)

(e)



(f)

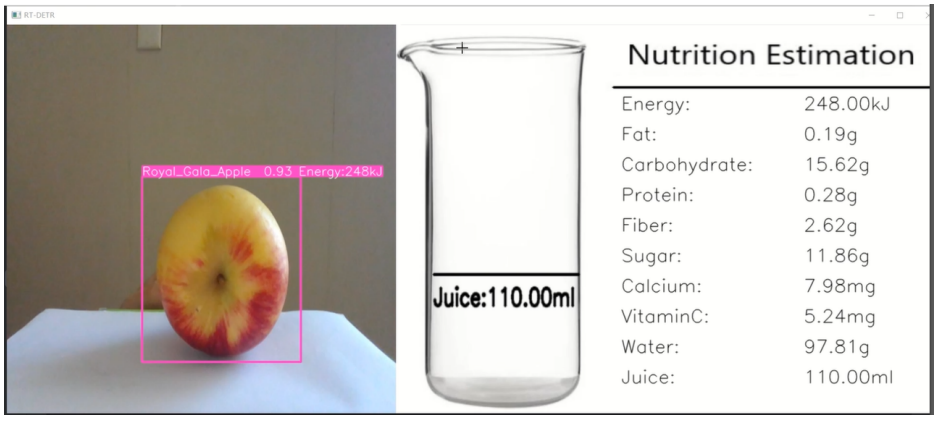
Figure 4.8: (d) to (f) Prediction of YOLOv9 model



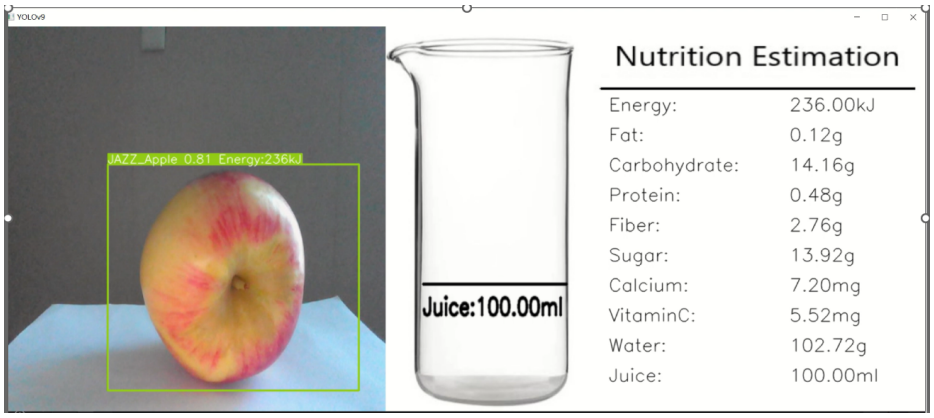
(g)

(h)

Figure 4.9 (g) Calorie estimation error for RT-DETR model and (h) for YOLOv9 model



(i)



(j)

Figure 4.10: (i) and (j) RT-DETR and YOLOv9 models incorrectly detected an Ambrosia apple as a Gala apple and a JAZZ apple respectively

4.7 Limitations of the Research

(1) Our dataset originates from videos capturing fruits in supermarkets, where fruits are typically selected and displayed for sale based on similar sizes. This uniformity in fruit sizes results in a lack of diversity in our samples, which may be insufficient for comprehensive training.

(2) Currently, our project only considers estimating the calorie content of individual fruits. To further extend our capabilities to estimate the total calorie content of multiple types of fruits, we need to expand our approach.

(3) While the research team claims that the model surpasses the YOLOv8 model in real-time detection, the RT-DETR-L model we selected requires longer training time on our dataset compared to YOLOv8. Additionally, during real-time detection, it does not perform as smoothly as the latter, indicating the need for further improvement in this aspect.

(4) The model exhibits false detections during real-time detection, highlighting the need for further improvement to enhance precision.

Chapter 5
Analysis and Discussions

This part primarily analyzes and discusses the outcomes of experiment along with the potential underlying reasons.

5.1 Analysis

Firstly, from the training results, it is apparent that the RT-DETR model achieved a precision of 99.01% on our dataset, indicating excellent model performance. The utilization of random data augmentation during training played a crucial role in achieving this high accuracy. Secondly, regarding real-time detection results, the model performed as expected, accurately detecting fruits and estimating their calories. In addition, the model may experience false detections due to insufficient samples, a situation that can be remedied by either increasing the sample size or refining the model to enhance detection performance.

5.2 Discussions

While examining the curves representing model performance, such as F1-confidence curves, P-R curves, loss curves, precision, recall, and mAP curves derived from training results, the RT-DETR model outperforms the YOLOv9 and YOLOv8 models in terms of performance. Additionally, the real-time detection outcomes show that the RT-DETR model has higher accuracy compared to the YOLOv9 model. These results demonstrate that RT-DETR has been proven to be effective in our project. These achievements can be attributed to several improved components of RT-DETR. The design of the hybrid encoder in RT-DETR enables the effective learning of more comprehensive multi-scale fruit features., while IoU-aware query selection facilitates the model in locating target fruits within images more effectively. This combination not only enhances the real-time detection capabilities of the RT-DETR detector but also improves detection accuracy, thereby enhancing overall detection performance.

Chapter 6
Conclusion and Future Work

In this part, our main focus is to encapsulate the methodologies and findings of our project. Furthermore, we outline potential avenues for future research based on the results obtained and the prevailing technological landscape.

6.1 Conclusion

This project aims to investigate the application of deep learning models for estimating the calorie content in fruits. To achieve this goal, we generated a dataset from videos captured by ourselves. We employed random augmentation during training to increase the robustness of the model in target detection. For the implementation, we adopted the RT-DETR model and integrated it into the framework of ultralytics. We also compared the training results of the RT-DETR model, YOLOv9, and YOLOv8 on our dataset and conducted real-time detection experiments to evaluate the performance of the RT-DETR model.

The findings of the study are promising, with a precision rate of 99.01% and 94.45% mAP50-95 achieved in fruit detection from digital images. In comparison with the YOLOv9 and YOLOv8 models, our chosen RT-DETR model demonstrates higher F1 score, precision, and mAP on our dataset. From various performance curves and real-time detection outcomes, it is apparent that the RT-DETR model surpasses the YOLOv9 and YOLOv8 models.

6.2 Future Work

Future research efforts can focus on collecting more diverse samples to improve detection precision. Additionally, incorporating fruit weight as a parameter in model training to estimate calorie content based on weight could be explored. Furthermore, detecting and estimating total calories of multiple fruits together could be considered. Lastly, there is still room for improvement in the model's detection precision and real-time performance. In the end, current efforts are focused on integrating CNN and Transformer architectures to achieve optimal results. In the future, it is plausible that Transformers may entirely replace CNNs in the realm of CV, as further fine-tuning progresses.

References

Agarwal, R., Choudhury, T., Ahuja, N. J., & Sarkar, T. (2023). Hybrid Deep Learning Algorithm-Based Food Recognition and Calorie Estimation. *Journal of Food Processing & Preservation*, pp. 1-15.

Amjoud, A. B., & Amrouch, M. (2023). Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access*, vol. 11, pp. 35479-35516.

Arkin, E., Yadikar, N., Muhtar, Y., & Ubul, K. (2021). A Survey of Object Detection Based on CNN and Transformer. *IEEE International Conference on Pattern Recognition and Machine Learning (PRML)* pp. 99-108.

Arkin, E., Yadikar, N., Xu, X., Aysa, A., & Ubul, K. (2023). A Survey: Object Detection Methods from CNN to transformer. *Multimedia Tools and Applications*, 82(2023), pp.21353–21383.

Al-Sarayreh, M., Reis, M., Yan, W., Klette, R. (2019) A sequential CNN approach for foreign object detection in hyperspectral images. International Conference on Information, Communications and Signal.

Al-Sarayreha, M. (2020) Hyperspectral Imaging and Deep Learning for Food Safety. PhD Thesis. Auckland University of Technology, New Zealand.

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

An, N. (2020) Anomalies Detection and Tracking Using Siamese Neural Networks. Master's Thesis. Auckland University of Technology, New Zealand.

Aziz, F., Saputri, D. U. E., Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). Efficient Skin Lesion Detection using YOLOv9 Network. Medical Informatics Technology, 2(1).

Begum, N., Goyal, A., & Sharma, S. (2022). Artificial Intelligence-Based Food Calories

Estimation Methods in Diet Assessment Research. *Artificial Intelligence Applications in Agriculture and Food Quality Improvement,* pp. 15.

Bharati, P., & Pramanik, A. (2020). Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey. *Computational Intelligence in Pattern Recognition*.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics,* 10(20), pp.2470.

Bi, J., Zhu, Z., & Meng, Q. (2021). Transformer in Computer Vision. *IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI),* pp. 1-5.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information,* 11(2), pp.125.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *Computer Vision – ECCV 2020. Lecture Notes in Computer Science (Vol. 12346).*

Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. *Remote Sensing,* 13(1), pp.89.

Cao, Y., Jin, K., & Wang, Y. (2021). A Survey of Deep Learning Based Object Detection. *3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)* pp. 602-607.

Chen, K., Lin, W., Li, J., See, J., Wang, J., & Zou, J. (2021). AP-Loss for Accurate One-Stage Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11), pp. 3782-3798.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-Based Fully Convolutional Networks. *30th Conference on Neural Information Processing Systems (NIPS 2016).*

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*.

Du, H., Zhou, S., Yan, W., Wang, S. (2023) Study on DNA storage encoding based IAOA under innovation constraints. Current Issues in Molecular Biology, 45 (4), 3573-3590.

Du, L., Zhang, R., & Wang, X. (2020). Overview of Two-Stage Object Detection Algorithms. *Journal of Physics*, 1544(1), 012033.

Fahmy, M. M. (2022). Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial. *Journal of Engineering Research*, 6(5).

Fu, Y., Nguyen, M., Yan, W. (2022) Grading methods for fruit freshness based on deep learning. Springer Nature Computer Science.

Fu, Y. (2020) Fruit Freshness Grading Using Deep Learning. Master's Thesis. Auckland University of Technology, New Zealand.

Gao, X., Nguyen, M., Yan, W. (2023) A high-accuracy deformable model for human face mask detection. PSIVT.

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.

Gowdra, N. (2021) Entropy-Based Optimization Strategies for Convolutional Neural Networks. PhD Thesis, Auckland University of Technology, New Zealand.

Girshick, R. (2015). Fast R-CNN. *IEEE International Conference on Computer Vision*

(ICCV), pp. 1440-1448.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 27-30.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2019). Bag of Tricks for Image Classification with Convolutional Neural Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558-567.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of YOLO Algorithm Developments. *Procedia Computer Science*, 199, pp.1066-1073.

Jiang, S., Xu, T., Li, J., Huang, B., Guo, J., & Bian, Z. (2019). IdentifyNet for Non-Maximum Suppression. *IEEE Access*, 7, pp.148245-148253.

Jiao, Y., Weir, J., Yan, W. (2011) Flame detection in surveillance. Journal of Multimedia 6 (1).

Jamil, S., Piran, M. J., & Kwon, O. J. (2023). A Comprehensive Survey of Transformers for Computer Vision. *Drones,* 7(5), pp.287.

Latif, G., Alsalem, B., Mubarky, W., Mohammad, N., & Alghazo, J. (2020). Automatic Fruits Calories Estimation through Convolutional Neural Networks. *International Conference on Computer and Technology Applications,* pp. 17-21.

Le, R., Nguyen, M., Nguyen, Q., Nguyen, H., Yan, W. (2020) Automatic data generation for deep learning model training of image classification used for augmented reality on pre-school books. International Conference on Multimedia Analysis and Pattern Recognition.

Le, R., Nguyen, M., Yan, W. (2020) Machine learning with synthetic data – A new way

to learn and classify the pictorial augmented reality markers in real-time. International Conference on Image and Vision Computing New Zealand.

Le, R. (2022) Synthetic Data Annotation for Enhancing the Experiences of Augmented Reality Application Based on Machine Learning (PhD Thesis). Auckland University of Technology, New Zealand.

Li, E., Wang, Q., Zhang, J., Zhang, W., Mo, H., & Wu, Y. (2023). Fish Detection under Occlusion Using Modified You Only Look Once v8 Integrating Real-Time Detection Transformer Features. *Applied Sciences*, 13, 12645.

Li, F., Zhang, H., Liu, S., Guo, J., Ni, L. M., & Zhang, L. (2022). DN-DETR: Accelerate DETR Training by Introducing Query DeNoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4), pp.2239-2251.

Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., & Yang, J. (2020). Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *34th Conference on Neural Information Processing Systems (NeurIPS).*

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lister, C. (2018). Easily Accessible Food Composition Data: Accessing Strategic Food Information for Product Development and Marketing. Food New Zealand, 18(5), pp.17-19.

Liu, B., Bai, H., & Liu, Y. (2023). HSVAugRegNet: Random Augmentation to Rotate Image Recognition. *IEEE International Conference on Control, Electronics and Computer Technology (ICCECT),* pp. 126-130.

Liu, J., Pan, C., Yan, W. (2022) Litter detection from digital images using deep learning. Springer Nature Computer Science.

Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., & Zhang, L. (2022). DAB-DETR: Dynamic Anchor Boxes Are Better Queries for DETR. *International Conference on Learning Representations*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *ECCV*, pp. 21–37.

Lu, X., Li, Q., Li, B., & Yan, J. (2020). MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection. *Computer Vision – ECCV,* pp. 12359.

Liu, Z., Yan, W., Yang, B. (2018) Image denoising based on a CNN model. International Conference on Control, Automation and Robotics.

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. *IEEE/CVF International Conference on Computer Vision (ICCV),* pp. 9992-10002.

Manaswi, N. K. (2018). RNN and LSTM. *Deep Learning with Applications Using Python,* pp. 115–126.

Mansoor, S., Jain, P., Hassan, N., Farooq, U., Mirza, M. A., Pandith, A. A., & Iqbal, Z. (2022). Role of Genetic and Dietary Implications in the Pathogenesis of Global Obesity. *Food Reviews International*, 38(S1), pp.434–455.

Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., & Wang, J. (2021). Conditional DETR for Fast Training Convergence. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3631-3640.

Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A survey on Performance Metrics for Object-Detection Algorithms. *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237-242.

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & da Silva, E. A. B. (2021). A Comparative Analysis of Object Detection Metrics with a Companion Open-Source

Toolkit. *Electronics*, 10(279), pp.279.

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.

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. (2022) Waste classification from digital images using ConvNeXt. Pacific-Rim Symposium on Image and Video Technology (PSIVT).

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

Qi, J., Nguyen, M., Yan, W. (2024) NUNI-Waste: Novel semi-supervised semantic segmentation for waste classification with non-uniform data augmentation. Multimedia Tools and Applications.

Redmon, J., Girshick, R., Divvala, S., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.

Roh, B., Shin, J., Shin, W., & Kim, S. (2022). SPARSE DETR: Efficient End-to-End Object Detection with Learnable Sparsity. *International Conference on Learning Representations*.

Rolls, E. T. (2007). Understanding the Mechanisms of Food Intake and Obesity. *Obesity*

Reviews, 8(Suppl. 1), pp.67–72.

Samplawski, C., & Marlin, B. M. (2021). Towards Transformer-Based Real-Time Object Detection at the Edge: A Benchmarking Study. *IEEE Military Communications Conference (MILCOM),* pp. 898-903.

Sarda, E., Deshmukh, P., Bhole, S., & Jadhav, S. (2020). Estimating Food Nutrients Using Region-Based Convolutional Neural Network. *International Conference on Computational Intelligence and Data Engineering,* pp. 435–444.

Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. International Conference on Control, Automation and Robotics.

Shen, H., Kankanhalli, M., Srinivasan, S., Yan, W. (2004) Mosaic-based view enlargement for moving objects in motion pictures. IEEE ICME'04.

Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), pp.60.

Sohan, M., Sai Ram, T., & Rami Reddy, C.V. (2024). A Review on YOLOv8 and Its Advancements. *Algorithms for Intelligent Systems*. pp 529-545.

Tang, S., Yan, W. (2024) Utilizing RT-DETR model for fruit calorie estimation from digital images. Information.

Tao, X., Li, Y., Yan, W., Wang, M., et al. (2021) Heritable variation in tree growth and needle vegetation indices of slash pine (Pinus elliottii) using unmanned aerial vehicles (UAVs). Industrial Crops and Products.

Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning & Knowledge Extraction*, 5(4), pp.1680-1716.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł.,

& Polosukhin, I. (2017). Attention Is All You Need. *31st Conference on Neural Information Processing Systems,* pp. 6000-6010.

Veni S., Krishna Sameera A., Samuktha V., & Anand R. (2021). A Robust Approach Using Fuzzy Logic for the Calories Evaluation of Fruits. *IEEE International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 1-6.

Wang, C.-Y., Bochkovskiy, A., Liao, & H.-Y. M. (2023). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464-7475.

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A New Backbone that Can Enhance Learning Capability of CNN. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Wang, C.-Y., Liao, H.-Y. M., & Yeh, I.-H. (2023). Designing Network Design Strategies through Gradient Path Analysis. *Journal of Information Science and Engineering (JISE)*, 39(4), pp. 975–995.

Wang, L., Yan, W. (2021) Tree leaves detection based on deep learning. International Symposium on Geometry and Vision.

Wang, S., Zhou, S., Yan, W. (2022) An enhanced whale optimisation algorithm for DNA storage encoding. Mathematical Biosciences and Engineering, 19 (12), 14142-14172.

Wang, T., Yuan, L., Chen, Y., Feng, J., & Yan, S. (2021). PnP-DETR: Towards Efficient Visual Analysis with Transformers. *IEEE/CVF International Conference on Computer Vision (ICCV),* pp. 4641-4650.

Wang, Y., Zhang, X., Yang, T., & Sun, J. (2022). Anchor DETR: Query Design for Transformer-Based Object Detection. *The AAAI Conference on Artificial Intelligence (AAAI-22)*.

Wasif, S. M., Thakery, S., Nagauri, A., Pereira, S. I. (2021). Food Calorie Estimation

Using Machine Learning and Image Processing. *International Journal of Advance Research, Ideas and Innovations in Technology,* 5(2), ISSN: 2454-132X.

Xia, Y., Nguyen, M., Yan, W. (2022) A real-time Kiwifruit detection based on improved YOLOv7. International Conference on Image and Vision Computing New Zealand (IVCNZ)

Xia, Y., Nguyen, M., Yan, W. (2023) Kiwifruit counting using KiwiDetector and KiwiTracker. IntelliSys conference.

Xia, Y., Nguyen, M., Yan, W. (2023) Multiscale Kiwifruit detection from digital images. PSIVT.

Xiang, Y., Yan, W. (2021) Fast-moving coin recognition using deep learning. Springer Multimedia Tools and Applications.

Xiao, B., Nguyen, M., Yan, W. (2021) Apple ripeness identification using deep learning. International Symposium on Geometry and Vision.

Xiao, B., Nguyen, M., Yan, W. (2023) Apple ripeness identification from digital images using transformers. Multimedia Tools and Applications, Springer.

Xiao, B., Nguyen, M., Yan, W. (2023) Fruit ripeness identification using transformers. Applied Intelligence, Springer.

Xiao, B., Nguyen, M., Yan, W. (2023) A mixture model for fruit ripeness identification in deep learning. Handbook of Research on AI and ML for Intelligent Machines and Systems, pp.1-16, Chapter 16, IGI Global.

Xiao, B., Nguyen, M., Yan, W. (2023) Fruit ripeness identification using YOLOv8 model. Springer Multimedia Tools and Applications.

Xiao, B. (2024) Fruit Ripeness Identification from Digital Images Using Deep Learning. PhD Thesis, Auckland University of Technology, New Zealand.

Xin, C. (2020) Detection and Recognition for Multiple Flames Using Deep Learning. Master's Auckland University of Technology, New Zealand.

Xin, C., Nguyen, M., Yan, W. (2020) Multiple flames recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 296-307.

Xue, Y. Yan, W. (2023) YOLO models for fresh fruit classification from digital videos. Handbook of Research on AI and ML for Intelligent Machines and Systems, pp. 421-435, Chapter 17, IGI Global.

Xue, Y. (2024) YOLO Models for Fresh Fruit Classification from Digital Videos. Master's Thesis. Auckland University of Technology, New Zealand.

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

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

Yao, T., Li, Y., Pan, Y., & Mei, T. (2023). HGNet: Learning Hierarchical Geometry from Points, Edges, and Surfaces. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 21846-21855.

Yu, R., Wang, Z., Wang, Y., Li, K., Liu, C., Duan, H., Ji, X., & Chen, J. (2023). LaPE: Layer-adaptive Position Embedding for Vision Transformers with Independent Layer Normalization. *IEEE/CVF International Conference on Computer Vision (ICCV).*

Zhao, K. (2021) Fruit Detection Using CenterNet. Master's Thesis, Auckland University of Technology, New Zealand.

Zhao, K., Yan, W. (2021) Fruit detection from digital images using CenterNet. International Symposium on Geometry and Vision.

Zhao, L., & Li, S. (2020). Object Detection Algorithm Based on Improved YOLOv3. *Electronics*, 9(3), pp.537.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *AAAI Conference on Artificial Intelligence*, *34*(07), pp.12993-13000.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable Transformers for End-to-End Object Detection. *International Conference on Learning Representations*.