processes

MDPI

Article

Real-Time Classification of Chicken Parts in the Packaging Process Using Object Detection Models Based on Deep Learning

Dilruba Şahin [1], Orhan Torkul [1,*], Merve Şişci [2,3], Deniz Demircioğlu Diren [3], Recep Yılmaz [4] and Alpaslan Kibar [5]

[1] Industrial Engineering Department, Sakarya University, Sakarya 54050, Türkiye; dilruba.sahin@ogr.sakarya.edu.tr
[2] Industrial Engineering Department, Kütahya Dumlupınar University, Kütahya 43300, Türkiye; mervesisci@sakarya.edu.tr
[3] Department of Information Systems and Technologies, Sakarya University, Sakarya 54050, Türkiye; ddemircioglu@sakarya.edu.tr
[4] Business School, Sakarya University, Sakarya 54050, Türkiye; rcyilmaz@sakarya.edu.tr
[5] Department of Management Information Systems, Sakarya University, Sakarya 54050, Türkiye; kibar@sakarya.edu.tr
* Correspondence: torkul@sakarya.edu.tr

Abstract: Chicken meat plays an important role in the healthy diets of many people and has a large global trade volume. In the chicken meat sector, in some production processes, traditional methods are used. Traditional chicken part sorting methods are often manual and time-consuming, especially during the packaging process. This study aimed to identify and classify the chicken parts for their input during the packaging process with the highest possible accuracy and speed. For this purpose, deep-learning-based object detection models were used. An image dataset was developed for the classification models by collecting the image data of different chicken parts, such as legs, breasts, shanks, wings, and drumsticks. The models were trained by the You Only Look Once version 8 (YOLOv8) algorithm variants and the Real-Time Detection Transformer (RT-DETR) algorithm variants. Then, they were evaluated and compared based on precision, recall, F1-Score, mean average precision (mAP), and Mean Inference Time per frame (MITF) metrics. Based on the obtained results, the YOLOv8s model outperformed the other models developed with other YOLOv8 versions and the RT-DETR algorithm versions by obtaining values of 0.9969, 0.9950, and 0.9807 for the F1-score, mAP@0.5, and mAP@0.5:0.95, respectively. It has been proven suitable for real-time applications with an MITF value of 10.3 ms/image.

Keywords: chicken parts; deep learning; image processing; object detection; reducing waste and costs; RT-DETR; YOLOv8

check for updates

Academic Editors: Honggang Chen, Yuan Li and Junyu Guo

Received: 27 February 2025
Revised: 21 March 2025
Accepted: 23 March 2025
Published: 27 March 2025

Citation: Şahin, D.; Torkul, O.; Şişci, M.; Diren, D.D.; Yılmaz, R.; Kibar, A. Real-Time Classification of Chicken Parts in the Packaging Process Using Object Detection Models Based on Deep Learning. *Processes* **2025**, *13*, 1005. https://doi.org/10.3390/pr13041005

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Chicken meat is one of the most consumed types of meat worldwide because it is high in protein and can be purchased at an affordable price. Due to the increase in chicken meat consumption worldwide, the need for production has increased, which has led to the expansion of the chicken sector and its transformation into a competitive market environment. The White Meat Industrialists and Breeders Association reported that Türkiye ranks 7th in world trade in chicken meat production [1]. According to the data from the Türkiye Statistical Institute [2], Türkiye's chicken meat production in 2023 was 2,328,791 tons. Globally, it has steadily increased yearly, reaching approximately 103,549,000 tons by 2023. It is also predicted to exceed 104.2 million by the end of 2024 [3].

In the chicken industry, where such intensive production is carried out, and intense competition is experienced worldwide, eliminating human (operator) intervention, increasing the process speed, and reducing the costs are extremely important. The life cycle stages of the chicken production and consumption chain consist of chicken farming, poultry processing, meat slaughtering and packaging, distribution, retail sales, consumer, and end-of-life cycle. Sorting, classification, and packaging are important stages of the chicken production process [4]. The procedures applied throughout the production process, including the packaging, must ensure that the quality and safety of food are maintained [5].

In most chicken production facilities, the separation of chicken parts is performed manually by operators using visual inspection. The chicken parts are manually separated into different containers and then sent away to be packaged [6]. These manual aspects of the process cause a loss of time and thus reduce efficiency. They are also prone to errors and cannot guarantee product safety. As a result, they can cause many disadvantages, such as longer production times, product waste due to errors, and increased customer complaints. Therefore, the need for automatic systems in the process of sorting and classifying chicken parts has reached a significant level.

Developments in the fields of artificial intelligence, machine learning, and computer vision enable businesses to digitalize their production processes, minimizing human or manual intervention, which is one of the requirements of Industry 4.0, and enable high automation in the production processes. The success of Industry 4.0 lies largely in the integration and adaptation of these technologies into the existing manufacturing process [7]. The use of these technologies has become increasingly widespread in various production and business operations, such as quality control [8–14], human–robot collaboration [15–17], weld seam tracking [18–20], predictive maintenance [21–24], logistics and inventory [25–29], and worker safety [30–32]. Existing and developing technologies have begun to show their impact on production facilities belonging to the chicken sector, where intensive production is experienced due to high demand.

Object detection, one of the core tasks in computer vision, involves finding and classifying target objects in images. In the field of object detection, the integration of deep learning algorithms has led to significant progress [33]. Among these algorithms, two-stage algorithms, such as Faster R-CNN, offer a high accuracy but work slower [34]. Single-stage algorithms such as YOLO (You Only Look Once) have the feature of being speed-oriented [35,36]. They have also gained popularity in the field of computer vision because they provide a high accuracy despite being small-sized, and can be trained on a single GPU [37]. You Only Look Once version 8 (YOLOv8), one of the most up-to-date and advanced versions of the YOLO series, offers an ideal option for real-time performance owing to its fast operation and high accuracy [38]. However, it has been reported that the Detection Transformer (DETR) algorithm [39], which combined the transformer architecture with object detection for the first time, does not require complex intermediate steps in object detection and provides good results in object detection implementations [40]. The Real-Time Detection Transformer (RT-DETR) algorithm [41], which was proposed as a result of the improvements made to DETR, further increased the detection speed and accuracy of the DETR algorithm. It has been stated that the algorithm provides more successful results than other real-time methods thanks to its superior performance when using high-power platforms such as CUDA [42].

This study aims to develop an effective model for real-time detection and classification of chicken parts to eliminate the aforementioned problems by providing automation in the packaging process in the chicken industry. For this purpose, seven object detection models were developed using five different versions of the deep learning-based YOLOv8 algorithm, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, and two different

versions of the RT-DETR algorithm, namely RT-DETR-Large and RT-DETR Extra-Large, on image data of chicken parts, and the performances of these models were compared. The primary contributions of this study are as follows:

- The YOLOv8 and RT-DETR algorithms were applied for the first time in the field of chicken part classification.
- The effectiveness of the detection models was evaluated and compared on a 5-category chicken parts classification image dataset with different combinations, using original images of chicken parts taken in an experimental environment.
- As a result of the evaluation of seven deep learning-based object detection models with high prediction accuracy using versions of the YOLOv8 and RT-DETR algorithms, it has been proven that the YOLOv8s model is more successful than the RT-DETR models and other YOLOv8 versions on this dataset.

The remainder of this paper is organized as follows: in the second section, studies on automation using image-processing technology in the chicken industry are examined. In the third section, the methods used in this study are explained. The methodology and application proposed in this study are presented in the fourth section. The fifth section analyzes the experimental results obtained from the implementation. Finally, in the sixth section, the conclusions are presented.

2. Literature

Different studies in the literature are supported by artificial intelligence and image processing in the chicken industry. Asmara et al. [43] classified the freshness levels of chicken parts as fresh, medium freshness, and not fresh, using image processing and machine learning algorithms based on their color and texture characteristics. In this study, the Support Vector Machine (SVM) algorithm was used for classification purposes and compared with the Decision Tree and Naive Bayes algorithms. The model trained using the SVM algorithm yielded the best prediction results. This model achieved an accuracy rate of 58.33% for images obtained using a smartphone camera, 98% for images obtained using a webcam, and 79.1% for images obtained using a digital microscope. Putra and Prakasa [44] determined the freshness of chicken breast meat using computer image processing techniques and deep learning algorithms. Chicken breasts are divided into two categories: fresh and stale. A Convolutional Neural Network (CNN) algorithm was used for classification. The developed model achieved a 92.9% accuracy rate in determining the freshness of chicken meat. The research conducted by Garcia et al. [45] was aimed at determining the freshness level of only the breast part of chicken meat using the VGG16 architecture of the CNN. After performing various image preprocessing operations, such as thresholding and morphological transformation, the trained model achieved an overall accuracy of 94.11% in classifying chicken breast meat as fresh and stale. Nyalala et al. [46] provided a real-time estimation of the weights of chicken carcasses in the production line using image processing and Artificial Neural Network (ANN) methods. In the study, Linear SVM, Quadratic SVM, Cubic SVM, Radial Basis Function Support Vector Machine (RBF SVM), and Bayesian ANN models were developed for comparison purposes. The Bayesian ANN model performed better than other models in estimating carcass, head, and neck weights, and RBF SVM in estimating drumstick, breast, and wing weights. This study presents an example of the real-time use of neural network algorithms in the chicken industry. You et al. [47] proposed a quality assessment method by examining color images of chicken meat. Images of the chicken meat were collected by placing color cards next to them for color correction to extract the captured color information. Hierarchical clustering was used to examine the adjusted colors of the samples and produce three distinct quality levels. It has been reported that a new meat sample can be categorized into

one of three quality classes using the proposed clustering model, and its color can be used to reflect the quality assessment outcome. The reviewed studies have proven the usability of computer vision, machine learning, and deep learning methods in various areas of the chicken industry.

When the literature is examined, it can be seen that great progress has been made by conducting research in areas such as evaluating chicken meat quality, freshness, and estimating the weight of chicken carcasses. However, the number of studies focusing on the classification of chicken parts during the packaging process and using image processing and computer vision implementations is limited. In a study conducted by Teimouri et al. [6], a system was developed that classified chicken parts as breasts, filets, wings, legs, and drumsticks using image processing and ANN. This is an example of a system that can separate chicken parts in real time using vision-based intelligent modeling. Geometrical aspects, color, and textural features were extracted from the image data of chicken parts obtained using a charge-coupled device (CCD) camera. Classification models were developed using Partial Least-Squares Regression, Linear Discriminant Analysis, and ANN methods on the obtained features. The ANN model with 22.2 ms processing time for an image showed the best performance with a 93% overall accuracy rate. Salma et al. [48] classified poultry as chicken, turkey, chicken farmer, and Fayoumi. Images of the drumstick, wing, neck, and breast parts of the animals of the desired species, purchased from the animal market, were collected. The classification was performed using the pre-trained MobileNetv2 algorithm. The model attained an accuracy of up to 98%. In a study by Chen et al. [49], real-time detection and classification of chicken parts in the production line were achieved with image processing and the YOLOv4 algorithm. Then, YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16 detection methods were compared, and it was concluded that the algorithm with the best real-time recognition performance was YOLOV4. The YOLOV4-CSPDarknet53 model, with a processing time of 15 ms, achieved an accuracy rate of 98.86%. Peng et al. [50] developed a detection model with the Swin-Transformer algorithm for automatic classification of chicken parts in the production line, and its performance was compared with that of the YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16 models. The developed Swin-Transformer model produced more successful results than other models with a 97.21% mean average precision (mAP) value and 19.02 ms average detection time. These studies conducted by Teimouri et al. [6], Chen et al. [49], and Peng et al. [50] can be seen as pioneering studies on the implementation of real-time classification of image processing, artificial neural networks, and deep learning methods in the sorting and packaging stages of chicken parts, which we consider to be lacking in the literature, and which we will contribute to the industry.

Although the models developed in the reviewed studies showed a high performance, there is a need for models with a more accurate and faster predictive power. Therefore, more up-to-date and advanced versions of these algorithms, such as YOLOv8 and RT-DETR, have been developed. The success of these algorithms, which have applications in different fields, compared to other algorithms, has been experimentally proven. Tamang et al. [51] studied an image processing model that detects and classifies face mask-wearing situations, which emerged as a need after the COVID-19 pandemic which shook the world in recent years. They classified mask-wearing situations as mask-worn, mask-not-worn, and improperly worn. They used YOLOv5 and YOLOv8 algorithms for object detection and classification and compared their performances. In almost all performance measures, the YOLOv8 algorithm performed better than the YOLOv5 algorithm. In a study conducted by Bawankule et al. [52], an automatic waste sorting system was developed to address the issue of solid waste and to make the process more efficient and reliable. The YOLOv8 model was used to detect and classify solid wastes, such as tin cans, batteries, food waste,

soft plastics, bottles, and cardboard. In this study, the performances of the Faster R-CNN, SSD, YOLOv4, YOLOv5, and YOLOv7 models were compared, and the YOLOv8 model left the other models behind with its high accuracy rate of 97.7%. The study by Jun et al. [53] aimed to provide aerial detection of objects in urban areas. To achieve this goal, they used Urban Zone Aerial Object Detection datasets with four classes: people, small vehicles, medium vehicles, and large vehicles. They developed object detection models using three different versions of the YOLOv8 algorithm and two different versions of the RT-DETR algorithm and evaluated their performance. As a result, it was stated that a better model was obtained compared to the other models in the study by obtaining a mAP@0.5:0.95 of 0.598 with the developed RT-DETR-r50 model. The YOLOv8n model was the fastest, with an inference speed of 30.4 frames per second (FPS). Guemas et al. [54] used the RT-DETR algorithm to detect Plasmodium species. A dataset of 24,720 images collected from 475 thin blood smears was used to develop the model. The performance of the RT-DETR model was compared with models developed using the YOLOv8x and YOLOv5x algorithms, and its superiority was proven. Because of the high performance of the models developed with YOLOv8 and RT-DETR algorithms in the aforementioned studies, in this study, unlike the reviewed studies on the classification of chicken parts with image processing, YOLOv8 and RT-DETR algorithms, which show high performance in real-time object detection applications from learning algorithms, were used.

The literature shows that the number of studies conducted with image processing and deep learning algorithms in the packaging process in the chicken industry is limited. Additionally, to the best of our knowledge, no implementation of the YOLOv8 or RT-DETR algorithms has been found in this area. By leveraging these advanced algorithms, which have demonstrated superior classification performance in other fields, this research fills a critical gap in the literature and provides a novel approach to automating chicken part classification in real time.

3. Methods

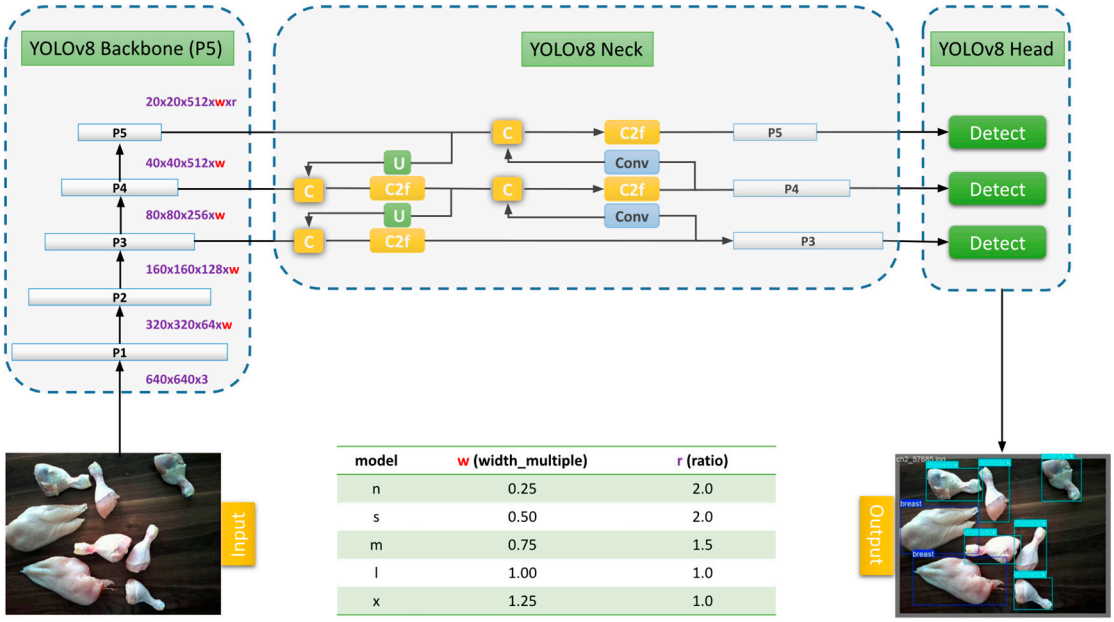
This study, which aimed at the automatic classification of chicken parts, required a fast and effective object detection model in real-time object recognition. Therefore, this study utilized two different versions of deep-learning-based models, YOLOv8 and RT-DETR.

3.1. You Only Look Once Version 8 (YOLOv8)

The YOLOv8 (You Only Look Once version 8) algorithm [55], which was developed by Ultralytics to achieve a better performance and includes the high-performance features of the previous YOLO versions, also includes a new backbone network, an anchor-free detection head, and a new loss function in addition to other YOLO versions [56]. This algorithm, which uses a single CNN and is configured to detect and segment objects in images, estimates bounding boxes and class probabilities [57]. In the YOLOv8 model, unlike the YOLOv5 (You Only Look Once version 5) model, instead of using anchors to measure the distance of an object, the center of the object is estimated with anchor-free detection [58]. The network structure of the algorithm, consisting of the input layer, backbone, neck, and head sections, is shown in Figure 1.

The image fed to the input layer is scaled to a fixed size, and preprocessing operations are performed. It is then forwarded to the CSPDarknet53 backbone network, which is also used in Yolov7, to perform the feature extraction task. Here, the features are extracted from the images using convolutional layers [59]. The backbone section contains the CSPBottleneck with two convolution (C2F) blocks and a spatial pyramid pooling fast (SPPF) block. The C2F module used for feature extraction improves the detection results by obtaining richer gradient flow information. It also shrinks the network to make the model

lighter. The SPPF module, located at the end of the backbone, reduces the number of layers in the network using the maximum pooling layer to eliminate unnecessary operations. The neck part, which performs the multi-scale aggregation task, contains a path aggregation network (PAN) and feature pyramid network (FPN). This part combines the low- and high-level features transmitted to it by the neck network and then sends them to the head part [41,60].



model	w (width_multiple)	r (ratio)
n	0.25	2.0
s	0.50	2.0
m	0.75	1.5
l	1.00	1.0
x	1.25	1.0

Figure 1. The structure of YOLOv8.

The head part of the algorithm consists of three basic layers: classification, bounding box regression, and confidence score. The classification layer processes the features from the FPN layer to determine the class of objects, the bounding box regression layer to estimate the coordinates of the bounding boxes for objects in the image, and the confidence score layer to estimate whether objects will appear in the image [61]. Two types of activation functions are used in the YOLOv8 algorithm: softmax and sigmoid. The softmax function classifies the object in the image, whereas the sigmoid function determines whether the object is within the bounding box. In addition, the use of distribution-focused loss (DFL) functions, and Complete Intersection over Union (CIoU) in the algorithm increases the detection accuracy for smaller objects [62]. The Ultralytics infrastructure allows five types of YOLOv8 algorithm implementations that differ in various features, such as the number of layers and parameters. YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x represent the nano, small, medium, large, and extra-large models of YOLOv8, respectively. Here, the YOLOv8n model had the least number of parameters.

3.2. Real-Time Detection Transformer (RT-DETR)

Developed for real-time object detection applications, the Real-Time Detection Transformer (RT-DETR) algorithm [63] was released by Baidu's Paddle team in January 2023 as an end-to-end object detection technology that integrates transformer architectures and CNN. The algorithm, which can significantly increase the processing speed while maintaining a high accuracy, was developed based on the DETR model [64,65]. Originally developed for natural language processing tasks, the DETR model, which uses transformers to encode and decode to directly predict object bounding boxes and classes, was developed by Facebook AI Research in 2020. This simplifies the object detection process by eliminating the post-

processing steps and traditional region proposals. It perfectly performs the task of multiple object detection, even in images with complex backgrounds [66]. In the DETR approach, which does not require predefined anchors or candidate boxes, the object-detection problem is transformed into an object-query problem. During the operation of the algorithm, the positions of the pixels are converted into a query vector in the form of a series of feature vectors using the transformer encoder. These query vectors are then used by the decoder to predict the class of objects through interactions with their bounding boxes [64]. However, to overcome the high computational cost of DETR, which is a significant challenge for real-time applications, the RT-DETR model was developed [67].

The RT-DETR model eliminates the nonmaximal suppression (NMS) component to solve the latency problem in existing real-time object detection models [68]. Key enhancements that provide improved performance in terms of speed and accuracy for real-time applications include a hybrid encoder that combines CNN and Transformers to efficiently handle multi-scale features by separating intra-scale interactions and inter-scale fusion, and minimum query selection that improves the initialization of encoder/object decoder queries [65,69,70]. The components of the RT-DETR algorithm are shown in Figure 2. These are a transformer decoder (with an auxiliary prediction header), a hybrid encoder, and a backbone network.

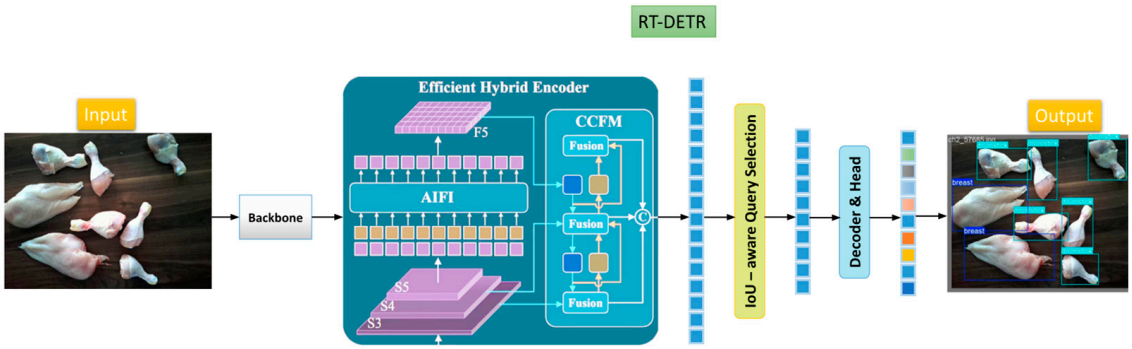


Figure 2. The architecture of RT-DETR.

With the image as input, the backbone network, which is the first processing layer, uses CNN structures such as ResNet series or HGNet to extract basic features at different levels, such as high-, medium-, and low-level features down-sampled by 32, 16, and 8 times, respectively [64,68]. The outputs are extracted from the backbone network part of the algorithm at three scales: S3, S4, and S5 [70].

The hybrid encoder, which converts multi-scale features from the backbone network into an image feature sequence, has two basic mechanisms: the CNN-based Cross-scale Feature Fusion (CCFM) module and the Attention-based Intra-scale Feature Interaction (AIFI) module [71]. The AIFI module is applied to extract more advanced high-level features. The CCFM module, which uses several convolutional layers, is applied to combine features at different levels [68].

The transformer decoder, which receives the processed multiscale features from the hybrid encoder, is responsible for further refining these features to produce the final object detection results. These detection results consist of bounding boxes and the classification scores of the objects in the image. This process is performed using a series of transformer decoder layers that iteratively refine the object queries. The auxiliary prediction header it contains produces intermediate detection results while also ensuring that the predictions to be produced by the model are improved throughout the stages [67].

4. Methodology and Implementation

The methodology proposed in this study for developing chicken part classification models is illustrated in Figure 3. After the chicken piece images were obtained in the experimental environment, two data preprocessing processes, namely data labeling and data splitting, were applied to the dataset to ensure that the dataset was in a format ready for models. Then, seven chicken part detection models based on YOLOv8 and RT-DETR algorithms were developed, and their performances were evaluated.

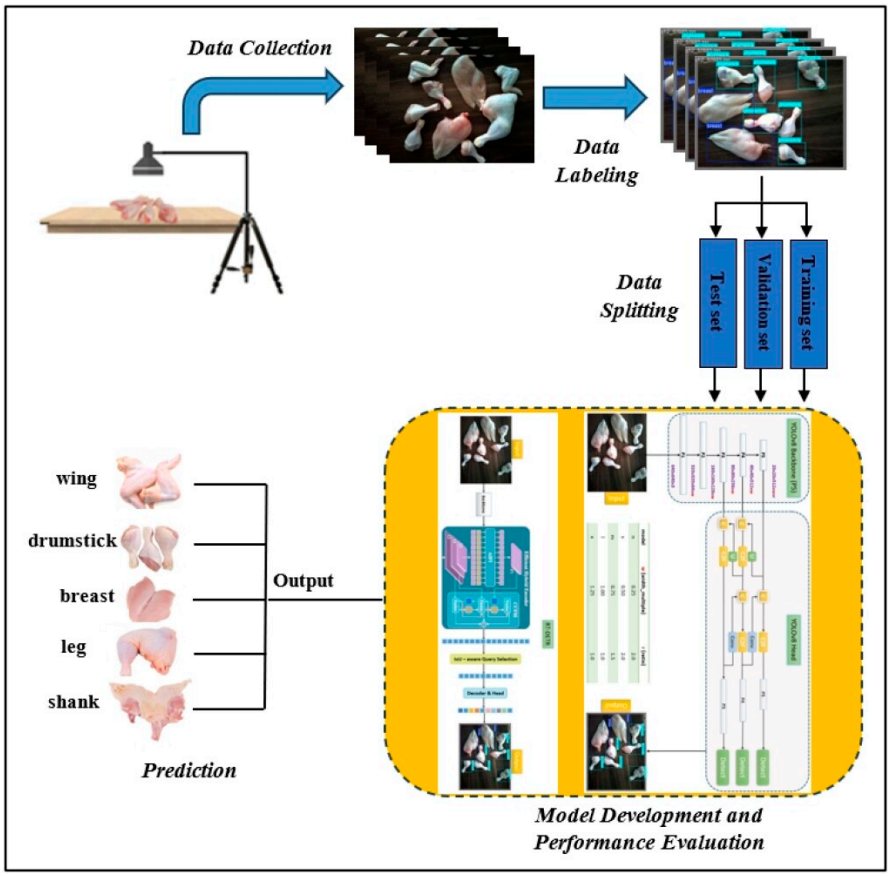


Figure 3. The steps of the study methodology.

4.1. Chicken Parts Identification Dataset Collection

For a successful deep learning model, the model must be trained with a sufficient dataset that contains a sufficient number of examples per class and can represent the entire data space [72]. Therefore, in this study, images were collected for five types of chicken parts: leg, breast, shank, wing, and drumstick, and a chicken part classification image dataset was generated. For this purpose, packaged products for each type of chicken were purchased from the butcher section of a supermarket, in a sufficient number for each type, taking into account the production dates and freshness of the products. The resulting chicken pieces were placed in different combinations of varieties at various positions and angles in the experimental sorting and packaging area. A diagram of the mechanism used to obtain the images of the chicken parts is shown in Figure 4. The Arducam 8MP 1080P USB Camera module used for image collection was adjusted to a suitable position to keep the chicken pieces in the focal plane and in the center of view, and was connected to a laptop computer with a 2.80 GHz Intel(R) Core(TM) i7-7700HQ CPU processor and 16 GB RAM. With the help of the camera, laptop, and codes developed using the Python programming language, images of chicken parts were collected at 640 × 480 resolution at 30 fps and saved

on a laptop. The OpenCV library, which is widely used in image-processing applications, was used to develop the codes.

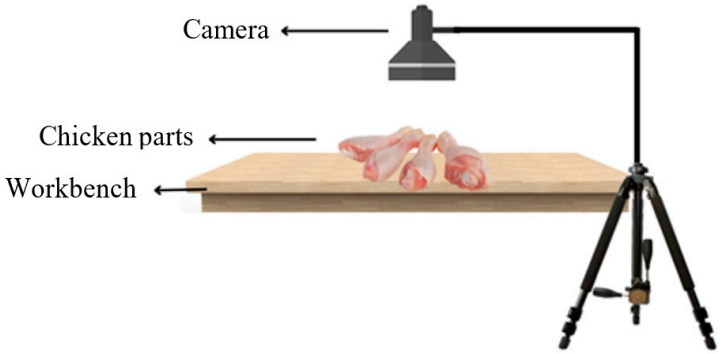


Figure 4. A representative image of the image acquisition mechanism.

Chicken images were obtained in daylight with the top open to include the flexibility of the environmental conditions, such as movement and light. 907 images were obtained from all varieties by placing the pieces at different positions on the table, rotating them in various ways, and collecting their images. Images obtained from the chicken pieces are shown in Figure 5. The entire dataset included 547 breasts, 1028 drumsticks, 464 wings, 209 legs, and 154 shanks images.

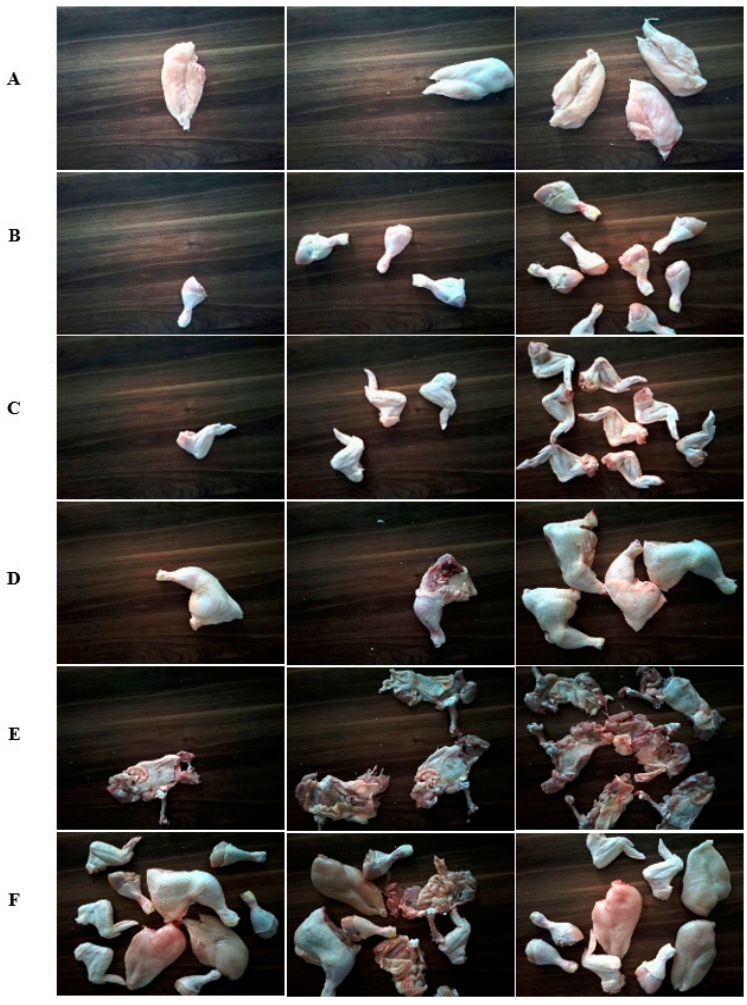


Figure 5. Examples of the images obtained from the following chicken part varieties: (**A**) breast, (**B**) drumstick, (**C**) wing, (**D**) leg, (**E**) shank and (**F**) whole chicken part varieties.

4.2. Data Preparation and Preprocessing

Data preparation and preprocessing were performed in two steps. First, the images were labeled using LabelImg 1.8.6 software, an open-source Python-based data-labeling tool. Subsequently, 70% of the labeled images were randomly allocated to be used for training the models, 15% for validation, and 15% for testing the trained models. The number of images and chicken pieces in the datasets are presented in Table 1.

Table 1. The number of images and chicken parts in the datasets as a result of resulted from data splitting.

	Training Dataset		Validation Dataset		Test Dataset	
Class	**Images**	**Instances**	**Images**	**Instances**	**Images**	**Instances**
All	634	1639	136	400	137	363
breast	261	384	51	73	58	90
drumstick	303	696	74	198	61	134
wing	106	315	21	63	24	86
leg	112	146	30	40	17	23
shank	38	98	9	26	9	30

4.3. Performance Evaluation Metrics

Seven different chicken part detection models, based on YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x) and RT-DETR (RT-DETR Large and RT-DETR Extra-Large) which were trained on chicken part images, were tested on the image data allocated for testing during the data preparation phase. To evaluate the test results, performance evaluation metrics of various object detection models including precision, recall, F1-Score, mAP@0.5, mAP@0.5:0.95, and Mean Inference Time per Frame (MITF) were used.

The precision performance metric, which measures how accurately the model detects, measures the percentage of correctly detected chicken pieces from the total number of detected pieces. The recall performance metric, which measures the number of chicken pieces accidentally missed by the model, is calculated as the ratio of correctly detected chicken pieces to the total number of chicken pieces that are both correctly detected and missed. This is important in applications where there should be no misses, such as detecting chicken pieces [73]. F1-Score, which is calculated as the harmonic mean of the Precision and Recall values, is used to measure the balance between the two performance metrics [74].

Precision, recall, and F1-Score performance metrics can be calculated as in Equations (1)–(3) [75], where CN_{TP} is the number of chicken pieces correctly detected by the model, CN_{TN} is the number of chicken pieces correctly missed, and CN_{FP} is the number of chicken pieces wrongly detected.

$$Precision = \frac{CN_{TP}}{CN_{TP} + CN_{FP}} \quad (1)$$

$$Recall = \frac{CN_{TP}}{CN_{TP} + CN_{FN}} \quad (2)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

Another important performance metric in object detection models is the mean average precision (mAP). This metric expresses the average precision of the model for different

classifications. The mean precision (AP) value for a single class is obtained by calculating the area under the precision-recall (PR) curve, as expressed in Equation (4) [70].

$$AP = \int_0^1 Precision(Recall)d(Recall) \quad (4)$$

The mAP performance criterion is determined by calculating the average AP values for all classes, as shown in Equation (5). Here, k represents the relevant chicken piece class, and N represents the total number of classes.

$$mAP = \frac{1}{N}\sum_{k=1}^{N} = AP_k \quad (5)$$

The intersection-union (IOU), a ratio calculation, is used in the mAP calculations and is obtained by dividing the intersection of the actual bounding box area by the total area [76]. When this value is accepted as the threshold value, the performance criterion values of mAP@0.5 and mAP@0.5:0.95 can be calculated. The rate of objects detected with a 50% accuracy (mAP@0.5) is obtained when the IOU value is set to 0.5. In contrast, the rate of objects detected with 50–95% accuracy is obtained by giving multiple values for the IOU value between 0.5 and 0.95, increasing by 0.05 intervals.

The MITF criterion used in this study expresses the amount of inference time the model requires for detecting in a single image [77].

5. Experiments and Results

5.1. Experimental Design

Python programming language was used in the training and testing phases of all models based on the YOLOv8 and RT-DETR algorithms. The experiments were performed in a development environment with a combination of the TensorFlow deep learning infrastructure, Python 3.12.4, Pytorch 2.3.1, Ultralytics 8.3.96, CUDA12.5, and CUDNN9.2 on a Windows 10 operating system. The hardware used for the experiments is a workstation with a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz processor, 16 GB RAM, and an NVIDIA GeForce RTX 3060 GPU graphics card.

Table 2 presents the hyperparameter values used to train the models. All images in the dataset were resized to 640 × 640 pixels, and the network was fed these images during the training and testing processes. Different parameter values set during the training process of deep learning-based models such as YOLOv8 and RT-DETR lead to different results. One of these parameters, batch size, is a parameter that affects training time and performance [56]. As the batch size increases, the training time decreases. In this study, different batch sizes of 2, 4, 8, 16, and 32 were tested. The batch size that gives the best values in terms of performance criteria was found to be 2. The epoch size was set to 300. A warm-up strategy was applied in training. This strategy improves the performance of the model and increases the convergence speed [78]. In the applied warm-up strategy, the learning rate was set to gradually increase over three epochs. The number of epochs for the patience was set to 100. In other words, if no improvement were observed in the last 100 epochs during the models' training, the model's training would be stopped. The optimizer was set automatically. The optimizer algorithm automatically determined by the library for training was the Adam optimization algorithm [79]. The initial learning rate and momentum determined by the optimizer were 0.001111 and 0.9, respectively. The mosaic augmentation technique was applied to the training dataset. In this technique, four images are randomly taken from the training dataset. Then, random cropping is performed on these images and combined into a single mosaic image [80].

Table 2. Training parameters and values for object detection models.

Parameter	Value
Epoch	300
Batch size	2
Momentum	0.9
Input image size	640 × 640
Warmup epochs	3
Warmup momentum	0.8
Warmup bias learning rate	0.1
Patience	100
Workers	8
Optimizer	Adam
Weight decay	0.0005
Initial learning rate	0.001111
Final learning rate	0.01

The design features of the deep learning-based object detection models developed in this study, such as the number of parameters, number of layers, giga floating-point operations per second (GFLOPs), number of completed epochs, training time, and comparative information about the training process, are given in Table 3. The maximum epoch number of 300 epochs was completed for all YOLOv8 models. Because there was no improvement during the last 100 epochs, the training of both RT-DETR models was stopped early. The training of the RT-DETR Large model was completed in 244 epochs, and the training of the RT-DETR Extra-Large model was completed in epoch 158. As shown in the table, the training time per epoch increased as the number of parameters and layers increased in the YOLOv8 and RT-DETR models. The training time per epoch in both RT-DETR models is higher than that of all YOLOv8 models. The lowest training time per epoch was observed in the YOLOv8n model, whereas the highest training time was observed in the RT-DETR Extra-Large model.

Table 3. Models training details.

Model	Parameters	Layers	Number of Completed Epochs	Training Time (hours)	Training Time per Epoch (hours)	GFLOPs
YOLOv8n	3,006,623	168	300	2.244	0.00748	8.1
YOLOv8s	11,127,519	168	300	4.380	0.0146	28.4
YOLOv8m	25,842,655	218	300	5.469	0.01823	78.7
YOLOv8l	43,610,463	268	300	6.494	0.021647	164.8
YOLOv8x	68,128,283	268	300	8.693	0.028977	257.4
RT-DETR Large	31,994,015	494	244	8.400	0.034426	103.5
RT-DETR Extra-Large	65,477,711	638	158	11.176	0.070734	222.5

5.2. Experimental Results

To perform a comparative evaluation of the models trained with the YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, RT-DETR Large, and RT-DETR Extra-Large algorithms in terms of detection accuracy and efficiency, the models were tested on the test dataset. The precision, recall, mAP@0.5, mAP@0.5:0.95, and MITF values obtained from the experiments are listed in Table 4. MITF values consist of the inference time the models spend while processing an image in the test phase. When the table is examined, it can be seen that all models provide values above 0.95 for precision, recall, F1-Score, mAP@0.5, and mAP@0.5:0.95. The highest precision value obtained was 0.9981 by the

RT-DETR Extra-Large model, followed by the YOLOv8s model with a value of 0.9961. The highest recall value (1.0) was observed in the YOLOv8n and RT-DETR Large models. The highest F1-Score, mAP@0.5, and mAP@0.5:0.95, were achieved by the YOLOv8s model, with values of 0.9969, 0.9950, and 0.9807, respectively. The lowest precision value of 0.9801 was obtained for YOLOv8n. The lowest F1-Score, recall, mAP@0.5, and mAP@0.5:0.95 values were achieved by the RT-DETR Extra-Large model with values of 0.9835, 0.9694, 0.9888, and 0.9588, respectively. It was observed that as the number of parameters of the models increased, the MITF values generally increased. It was concluded that the fastest model was the YOLOv8n model with a value of 6.1 ms/image, and the second fastest model was the YOLOv8s model with a value of 10.3 ms/image. The model with the slowest inference speed feature is the RT-DETR Extra-Large model, with a speed value of 45.8 ms/image. Considering both speed and other performance criteria, it can be said that the most suitable performance can be obtained when the YOLOv8s model is used.

Table 4. Comparison of performance metric values of YOLOv8 and RT-DETR models.

Model	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95	MITF (ms/Image)
YOLOv8n	0.9801	1.0	0.9900	0.9948	0.9720	6.1
YOLOv8s	0.9961	0.9977	0.9969	0.9950	0.9807	10.3
YOLOv8m	0.9864	0.9911	0.9888	0.9926	0.9802	15.2
YOLOv8l	0.9949	0.9925	0.9937	0.9945	0.9738	22.7
YOLOv8x	0.9884	0.9910	0.9897	0.9937	0.9742	32.8
RT-DETR Large	0.9908	1.0	0.9954	0.9938	0.9670	30.5
RT-DETR Extra-Large	0.9981	0.9694	0.9835	0.9888	0.9588	45.8

The precision–recall curve, F1–confidence curve, and recall–confidence curve graphs of the YOLOv8s model, which provided the most successful results, and the RT-DETR Extra-Large model, which showed the least success, are shown in Figure 6. The precision–recall (PR) curve is a curve where the recall values are on the *x*-axis, the precision values are on the *y*-axis, and the area under it corresponds to the mAP value. The large size of this area—the high mAP value—indicates that the proposed object detection model is successful [81]. As can be seen in the figure, the area under the precision–recall (PR) curve of the YOLOv8s model (Figure 6d) is larger than the area under the precision–recall (PR) curve of the RT-DETR Extra-Large model (Figure 6b). For all classes, the area under the receiver operating characteristic curve (AUC) value was 0.989 for the RT-DETR Extra-Large model and 0.995 for the YOLOv8s model. The F1–confidence curve can provide a visual confidence score that optimizes recall and precision. The confidence score should be as high as possible [38]. According to the test results, the optimum value of the F-measure confidence was 0.831 for the YOLOv8s model and 0.586 for the RT-DETR Extra-Large model. While the maximum F1-Score value of the YOLOv8s model was 1.00, the maximum F1-Score value of the RT-DETR Extra-Large model was 0.98. In this case, the YOLOv8s model exhibited an improvement of 0.02.

Examples of the test data accuracy results for the YOLOv8s model developed in this study are shown in Figure 7. It can be seen that it showed a very high performance in both the class estimation and the bounding box estimation, even in the detection of overlapping parts.

To further evaluate the superiority of the YOLOv8s model with the highest detection power in detecting and classifying chicken parts, the performance of the model was also evaluated by comparing it with other existing studies aimed at the classification of chicken parts. A comparative summary is presented in Table 5. When the table is examined, it

can be seen that the YOLOv8s model developed in this study has the best performance among the other studies in terms of both accuracy and speed in classifying and detecting chicken parts. This comparative analysis confirmed the superiority of the YOLOv8s model in effectively detecting chicken parts compared to the models developed in other studies.

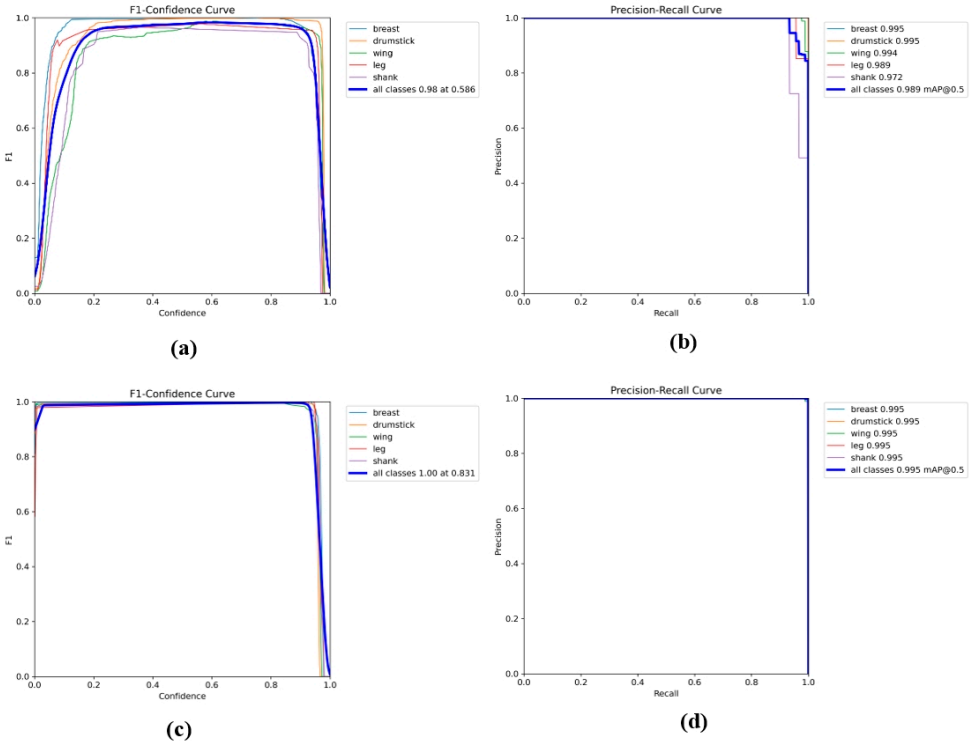


Figure 6. Precision–recall and F1–confidence curves (**a**) F1–confidence Curve of the RT-DETRX model, (**b**) precision–recall curve of the RT-DETRX model, (**c**) F1–confidence Curve of the YOLOv8s model, (**d**) precision–recall curve of the YOLOv8s model.

Figure 7. Examples of the accuracy results on the test data for the trained YOLOv8 model.

Table 5. Comparison between the proposed methodology and the existing studies in the literature.

Study	Classes (Types of Chicken Parts)	Dataset Details	Methods	The Most Successful Method	The Most Successful Results
Teimouri et al. [6]	breast, leg, filet, wing, and drumstick	100 samples	Partial least squares regression, linear discriminant analysis, and artificial neural network	Artificial neural network	Overall accuracy: %93, MITF: 15 ms/image
Chen et al. [49]	wing, leg, and breast	600 images	YOLOV4-CSPDarknet53, YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16	YOLOV4-CSPDarknet53	mAP%: 98.86%, MITF 22.2 ms/image
Peng et al. [50]	wing, leg, and breast	2000 images	Swin-Transformer, YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16	Swin-Transformer	mAP% = 97.21%, MITF: 19.02 ms/image
This Study	breast, drumstick, wing, leg and shank	907 images	YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x) RT-DETR (RT-DETR Large and RT-DETR Extra-Large)	YOLOv8s	F1-Score: 0.9969, mAP@0.5: 0.9950, mAP@0.5:0.95: 0.9807, MITF: 10.3 ms/image

6. Conclusions

Manual separation of the chicken parts by the operator during the packaging process in chicken production facilities can lead to human-induced errors, unnecessary and repetitive operations, food waste, and even customer dissatisfaction from faulty products. Therefore, there is a need to develop an automation-based system to quickly and accurately separate the chicken parts during this process. This study aims to develop an automatic detection model for use in the packaging process of a basic chicken production facility. The system that will detect chicken parts will also classify them. To achieve this goal, deep learning-based computer vision methods were used in this study. Before starting the process of developing the models, the chicken packaging process of a company operating in the chicken sector in Türkiye was analyzed in detail. First, the images of the breast, drumstick, wing, leg, and shank parts included in the packaging process at the company were collected in the prepared experimental setup. The dataset consisted of 907 images; to simulate real-world scenarios encountered in the company's packaging line variations in lighting conditions, angles, and slight changes in the positioning of chicken pieces were provided. To increase robustness, images were collected from different chickens, allowing for natural variability in size, shape and texture. Additionally, preprocessing operations, including resizing and mosaic augmentation techniques, were performed to improve the generalization ability of the dataset. The variability of the dataset is crucial to train a model that can effectively adapt to different production environments and maintain a high accuracy in various conditions. Then, deep learning-based object detection models were developed using different versions of the YOLOv8 and RT-DETR algorithms (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, RT-DETR Large, and RT-DETR Extra-Large), and their performances were investigated. It is necessary to develop models with a high accuracy and real-time speed for the classification of chicken parts in the packaging process.

In this study, YOLOv8 and RT-DETR models, which are state-of-the-art object detection models based on deep learning, were selected due to their success in studies conducted in different fields [51–54] and to fill the gap in the field of classification of chicken parts in the packaging process. The applicability of the developed models in an industrial environment was evaluated by considering their real-time working ability and high accuracy rate. Although it has been emphasized in the studies carried out in different fields and belonging to Jin and Zhang [82], Yu and Chen [71], and Wu et al. [83] that the RT-DETR object detection model performed better than the YOLOv8 model in terms of accuracy and speed, when the YOLOv8s model developed in this study was compared in terms of accuracy, it produced more accurate results than the RT-DETR models developed in this study with values of 0.9969 F1-Score, 0.9950 mAP@0.5, 0.9807 mAP@0.5:0.95. This finding contrasts with some prior studies but aligns with Parekh [84], who reported that YOLOv8 models outperformed RT-DETR in traffic object detection within advanced driver assistance systems. These high values obtained by the model show that the classification success is at a usable level in production lines. The model also outperformed other YOLO versions. When evaluated in terms of speed, although YOLOv8n is the model that can make the fastest inference with an MITF value of 6.1 ms/image, YOLOv8s stands out as the ideal model in terms of speed with an MITF value of 10.3 ms/image. Both models developed within the scope of the study operated with a low mean inference time (MITF) and exhibited a performance that can be easily integrated into industrial automation systems. These results support the selection of YOLOv8s as the most effective model for real-time classification of chicken parts in packaging processes, given its superior performance in both accuracy and inference speed.

It has been proven that the model developed in this study can be applied efficiently and effectively in deep learning-supported real-time object detection and classification processes in the food industry. It is foreseen that the following benefits will be achieved using this model in chicken production facilities:

- Reducing the need for manual intervention by automating the process of separating chicken parts,
- Minimizing errors caused by manual classification by the human eye,
- Reducing and eliminating labor waste and faulty product packaging waste,
- Increasing the degree of automation of chicken plant production processes,
- Reducing customer complaints due to faulty product packaging and increasing customer satisfaction,
- Directing the workforce to areas where it can be used more efficiently,
- Increasing the speed and efficiency of the production process and reducing production costs.

The model developed in this study has demonstrated usable performance in industrial chicken packaging lines with high accuracy and low inference time. However, the model has some limitations. First of all, the dataset was collected in an experimental environment created similar to a production line, not in a real production line. Changes in the equipment, camera angle, imaging systems and especially lighting conditions used in the real production environment may affect the performance of the model. For example, factors such as low light, shadowing or reflection may negatively affect the detection accuracy. In addition, difficulties in the real-world deployment of the model include factors such as detecting fast-moving objects on the production line and ensuring appropriate hardware integration. Future work will progress towards training the model on data collected from real production environments, increasing its robustness to variable lighting conditions and optimizing its integration into real production processes. Another limitation of the study is that the background in the production line examined is regular, clean, and uniform. Therefore, the model developed in this study was trained and evaluated by creating

an experimental environment that would simulate this real production environment. A new research direction can be created in future studies to examine the effect of different background conditions on model performance. In addition, this model, which has proven successful in detecting chicken parts before the packaging, can be trained on datasets collected from different packaging processes, such as duck or turkey, in future studies, and its performance can be investigated for automatic classification purposes.

Author Contributions: Conceptualization, D.Ş., O.T. and M.Ş.; methodology, D.Ş., O.T. and M.Ş.; software, D.Ş., M.Ş., D.D.D. and A.K.; validation, D.Ş., O.T., M.Ş., D.D.D., R.Y. and A.K.; formal analysis, D.Ş., O.T., M.Ş. and R.Y.; investigation, D.Ş., O.T., M.Ş. and R.Y.; data curation, D.Ş., M.Ş. and D.D.D.; writing—original draft preparation, D.Ş., O.T., M.Ş., D.D.D., R.Y. and A.K.; writing—review and editing, D.Ş., O.T., M.Ş., D.D.D., R.Y. and A.K.; visualization, M.Ş., D.D.D. and A.K.; supervision, O.T. and R.Y; project administration, O.T.; funding acquisition, D.Ş. and O.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2209-A TUBITAK (Scientific and Technological Research Council of Türkiye) Student Project, grant number 1919B012335040.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AIFI	Attention-based Intra-scale Feature Interaction
ANN	Artificial Neural Networks
AP	Mean Precision
AUC	Area Under The Receiver-Operating Characteristic Curve
C2F	CSPBottleneck with 2 convolutions
CCD	Charge-coupled Device
CCFM	CNN-based Cross-scale Feature Fusion
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network
CN_{TP}	Number of chicken pieces correctly detected by the model
CN_{TN}	Number of chicken pieces correctly missed
CN_{FP}	Number of chicken pieces incorrectly detected
DETR	Detection Transformer
DFL	Distribution Focused Loss
FPN	Feature Pyramid Network
FPS	Frames per Second
GFLOPs	Giga Floating-Point Operations Per Second
GPU	Graphics Processing Unit
IOU	Intersection Over Union
mAP	Mean Average Precision
MITF	Mean Inference Time per Frame
PAN	Path Aggregation Network
PR	Precision–recall
R-CNN	Region Based Convolutional Neural Networks
RT-DETR	Real-Time Detection Transformer
SPPF	Spatial Pyramid Pooling Fast
YOLOv4	You Only Look Once version 4
YOLOv5	You Only Look Once version 5
YOLOv5x	You Only Look Once version 5-extra large

YOLOv7	You Only Look Once version 7
YOLOv8	You Only Look Once version 8
YOLOv8l	You Only Look Once version 8-large
YOLOv8m	You Only Look Once version 8-medium
YOLOv8n	You Only Look Once version 8-nano
YOLOv8s	You Only Look Once version 8-small
YOLOv8x	You Only Look Once version 8-extra large

References

1. Omrak, H. Türkiye ranks 10th in the world in poultry meat production. *Agric. For. J.* **2021**. Available online: http://www.turktarim.gov.tr/Haber/702/turkiye-kanatli-eti-uretiminde-dunyada-10--sirada#:~:text=Sekt%C3%B6r,%2015%20bin%20adet%20kay%C4%B1tl%C4%B1,ise%20d%C3%BCnyada%207%E2%80%99nci%20s%C4%B1radad%C4%B1r (accessed on 22 February 2025). (In Turkish)
2. Türkiye Statistical Institute. Available online: https://data.tuik.gov.tr/Bulten/Index?p=Kumes-Hayvanciligi-Uretimi-Ocak-2024-53562 (accessed on 21 November 2024). (In Turkish)
3. Statista. Global Chicken Meat Production 2012–2024. Available online: https://www.statista.com/statistics/237637/production-of-poultry-meat-worldwide-since-1990/#:~:text=This%20statistic%20depicts%20chicken%20meat,million%20metric%20tons%20by%202024 (accessed on 21 November 2024).
4. Cooreman-Algoed, M.; Boone, L.; Taelman, S.E.; Van Hemelryck, S.; Brunson, A.; Dewulf, J. Impact of consumer behaviour on the environmental sustainability profile of food production and consumption chains–a case study on chicken meat. *Resour. Conserv. Recycl.* **2022**, *178*, 106089. [CrossRef]
5. Marcinkowska-Lesiak, M.; Zdanowska-Sąsiadek, Ż.; Stelmasiak, A.; Damaziak, K.; Michalczuk, M.; Poławska, E.; Wyrwisz, J.; Wierzbicka, A. Effect of packaging method and cold-storage time on chicken meat quality. *CyTA-J. Food* **2016**, *14*, 41–46. [CrossRef]
6. Teimouri, N.; Omid, M.; Mollazade, K.; Mousazadeh, H.; Alimardani, R.; Karstoft, H. On-line separation and sorting of chicken portions using a robust vision-based intelligent modelling approach. *Biosyst. Eng.* **2018**, *167*, 8–20. [CrossRef]
7. Nayyar, A.; Kumar, A. (Eds.) *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*; Springer Nature: Berlin/Heidelberg, Germany, 2019; ISBN 978-3-030-14543-9.
8. Chen, Y.W.; Shiu, J.M. An implementation of YOLO-family algorithms in classifying the product quality for the acrylonitrile butadiene styrene metallization. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 8257–8269. [CrossRef] [PubMed]
9. Kulkarni, U.; Patil, A.; Devaranavadagi, R.; Devagiri, S.B.; Pamali, S.K.; Ujawane, R. Vision-Based Quality Control Check of Tube Shaft using DNN Architecture. In Proceedings of the ITM Web of Conferences 2023, Gujarat, India, 28-29 April 2023; EDP Sciences. Volume 53, p. 02009. [CrossRef]
10. Chetoui, M.; Akhloufi, M.A. Object detection model-based quality inspection using a deep CNN. In Proceedings of the Sixteenth International Conference on Quality Control by Artificial Vision, Albi, France, 6–8 June 2023; Volume 12749, pp. 65–72. [CrossRef]
11. Ardic, O.; Cetinel, G. Deep Learning Based Real-Time Engine Part Inspection with Collaborative Robot Application. *IEEE Access* **2024**. [CrossRef]
12. Akgül, İ. A novel deep learning method for detecting defects in mobile phone screen surface based on machine vision. *Sak. Univ. J. Sci.* **2023**, *27*, 442–451. [CrossRef]
13. Alpdemir, M.N. Pseudo-Supervised Defect Detection Using Robust Deep Convolutional Autoencoders. *Sak. Univ. J. Comput. Inf. Sci.* **2022**, *5*, 385–403. [CrossRef]
14. Güngör, M.A. A New Gradient Based Surface Defect Detection Method for the Ceramic Tile. *Sak. Univ. J. Sci.* **2022**, *26*, 1159–1169. [CrossRef]
15. Fan, J.; Zheng, P.; Li, S. Vision-based holistic scene understanding towards proactive human–robot collaboration. *Robot. Comput.-Integr. Manuf.* **2022**, *75*, 102304. [CrossRef]
16. Zhang, R.; Lv, J.; Li, J.; Bao, J.; Zheng, P.; Peng, T. A graph-based reinforcement learning-enabled approach for adaptive human-robot collaborative assembly operations. *J. Manuf. Syst.* **2022**, *63*, 491–503. [CrossRef]
17. Li, C.; Zheng, P.; Yin, Y.; Pang, Y.M.; Huo, S. An AR-assisted Deep Reinforcement Learning-based approach towards mutual-cognitive safe human-robot interaction. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102471. [CrossRef]
18. Banafian, N.; Fesharakifard, R.; Menhaj, M.B. Precise seam tracking in robotic welding by an improved image processing approach. *Int. J. Adv. Manuf. Technol.* **2021**, *114*, 251–270. [CrossRef]
19. Rout, A.; Deepak, B.B.V.L.; Biswal, B.B.; Mahanta, G.B. Weld seam detection, finding, and setting of process parameters for varying weld gap by the utilization of laser and vision sensor in robotic arc welding. *IEEE Trans. Ind. Electron.* **2021**, *69*, 622–632. [CrossRef]

20. Chen, C.; Chen, T.; Cai, Z.; Zeng, C.; Jin, X. A hierarchical visual model for robot automatic arc welding guidance. *Ind. Robot. Int. J. Robot. Res. Appl.* **2023**, *50*, 299–313. [CrossRef]
21. Susto, G.A.; Schirru, A.; Pampuri, S.; McLoone, S.; Beghi, A. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Trans. Ind. Inform.* **2014**, *11*, 812–820. [CrossRef]
22. Paolanti, M.; Romeo, L.; Felicetti, A.; Mancini, A.; Frontoni, E.; Loncarski, J. Machine learning approach for predictive maintenance in industry 4.0. In Proceedings of the 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Oulu, Finland, 2–4 July 2018. [CrossRef]
23. Liu, C.; Zhu, H.; Tang, D.; Nie, Q.; Zhou, T.; Wang, L.; Song, Y. Probing an intelligent predictive maintenance approach with deep learning and augmented reality for machine tools in IoT-enabled manufacturing. *Robot. Comput.-Integr. Manuf.* **2022**, *77*, 102357. [CrossRef]
24. Zhuang, L.; Xu, A.; Wang, X.L. A prognostic driven predictive maintenance framework based on Bayesian deep learning. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109181. [CrossRef]
25. Knoll, D.; Prüglmeier, M.; Reinhart, G. Predicting future inbound logistics processes using machine learning. *Procedia CIRP* **2016**, *52*, 145–150. [CrossRef]
26. Yu, X.; Liao, X.; Li, W.; Liu, X.; Tao, Z. Logistics automation control based on machine learning algorithm. *Clust. Comput.* **2019**, *22*, 14003–14011. [CrossRef]
27. Abosuliman, S.S.; Almagrabi, A.O. Computer vision assisted human computer interaction for logistics management using deep learning. *Comput. Electr. Eng.* **2021**, *96*, 107555. [CrossRef]
28. Gregory, S.; Singh, U.; Gray, J.; Hobbs, J. A computer vision pipeline for automatic large-scale inventory tracking. In Proceedings of the 2021 ACM Southeast Conference, New York, NY, USA, 15–17 April 2021; pp. 100–107. [CrossRef]
29. Denizhan, B.; Yıldırım, E.; Akkan, Ö. An Order-Picking Problem in a Medical Facility Using Genetic Algorithm. *Processes* **2024**, *13*, 22. [CrossRef]
30. Zhafran, F.; Ningrum, E.S.; Tamara, M.N.; Kusumawati, E. Computer vision system based for personal protective equipment detection, by using convolutional neural network. In Proceedings of the 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 27–28 September 2019. [CrossRef]
31. Khandelwal, P.; Khandelwal, A.; Agarwal, S.; Thomas, D.; Xavier, N.; Raghuraman, A. Using computer vision to enhance safety of workforce in manufacturing in a post COVID world. *arXiv* **2020**, arXiv:2005.05287. [CrossRef]
32. Cheng, J.P.; Wong, P.K.Y.; Luo, H.; Wang, M.; Leung, P.H. Vision-based monitoring of site safety compliance based on worker re-identification and personal protective equipment classification. *Autom. Constr.* **2022**, *139*, 104312. [CrossRef]
33. Zhang, H.; Ma, Z.; Li, X. Rs-detr: An improved remote sensing object detection model based on rt-detr. *Appl. Sci.* **2024**, *14*, 10331. [CrossRef]
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149.
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
36. Güney, E.; Bayilmiş, C.; Çakan, B. An implementation of real-time traffic signs and road objects detection based on mobile GPU platforms. *IEEE Access* **2022**, *10*, 86191–86203. [CrossRef]
37. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [CrossRef]
38. Inui, A.; Mifune, Y.; Nishimoto, H.; Mukohara, S.; Fukuda, S.; Kato, T.; Furukawa, T.; Tanaka, S.; Kusunose, M.; Takigami, S.; et al. Detection of elbow OCD in the ultrasound image by artificial intelligence using YOLOv8. *Appl. Sci.* **2023**, *13*, 7623. [CrossRef]
39. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229. [CrossRef]
40. Nguyen, D.; Hoang, V.D.; Le, V.T.L. V-DETR: Pure Transformer for End-to-End Object Detection. In *Asian Conference on Intelligent Information and Database Systems*; Springer Nature: Singapore, 2024; pp. 120–131. [CrossRef]
41. Zhao, X.; Song, Y. Improved ship detection with YOLOv8 enhanced with MobileViT and GSConv. *Electronics* **2023**, *12*, 4666. [CrossRef]
42. Wang, A.; Xu, Y.; Wang, H.; Wu, Z.; Wei, Z. CDE-DETR: A Real-Time End-To-End High-Resolution Remote Sensing Object Detection Method Based on RT-DETR. In Proceedings of the IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024.
43. Asmara, R.A.; Hasanah, Q.; Rahutomo, F.; Rohadi, E.; Siradjuddin, I.; Ronilaya, F.; Handayani, A.N. Chicken meat freshness identification using colors and textures feature. In Proceedings of the 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 25–29 June 2018. [CrossRef]

44. Putra, G.B.; Prakasa, E. Classification of chicken meat freshness using convolutional neural network algorithms. In Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Sakheer, Bahrain, 20–21 December 2020; pp. 1–6. [CrossRef]
45. Garcia, M.B.P.; Labuac, E.A.; Hortinela, C.C., IV. Chicken meat freshness classification based on vgg16 architecture. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 13–15 September 2022; pp. 1–6. [CrossRef]
46. Nyalala, I.; Okinda, C.; Makange, N.; Korohou, T.; Chao, Q.; Nyalala, L.; Zhang, J.; Zuo, Y.; Yousaf, K.; Liu, C.; et al. On-line weight estimation of broiler carcass and cuts by a computer vision system. *Poult. Sci.* **2021**, *100*, 101474. [CrossRef]
47. You, M.; Liu, J.; Zhang, J.; Xv, M.; He, D. A novel chicken meat quality evaluation method based on color card localization and color correction. *IEEE Access* **2020**, *8*, 170093–170100. [CrossRef]
48. Salma, S.; Habib, M.; Tannouche, A.; Ounejjar, Y. Poultry Meat Classification Using MobileNetV2 Pretrained Model. *Rev. D'intelligence Artif.* **2023**, *37*, 275–280. [CrossRef]
49. Chen, Y.; Peng, X.; Cai, L.; Jiao, M.; Fu, D.; Xu, C.C.; Zhang, P. Research on automatic classification and detection of chicken parts based on deep learning algorithm. *J. Food Sci.* **2023**, *88*, 4180–4193. [CrossRef]
50. Peng, X.; Xu, C.; Zhang, P.; Fu, D.; Chen, Y.; Hu, Z. Computer vision classification detection of chicken parts based on optimized Swin-Transformer. *CyTA-J. Food* **2024**, *22*, 2347480. [CrossRef]
51. Tamang, S.; Sen, B.; Pradhan, A.; Sharma, K.; Singh, V.K. Enhancing COVID-19 safety: Exploring yolov8 object detection for accurate face mask classification. *Int. J. Intell. Syst. Appl. Eng.* **2023**, *11*, 892–897. Available online: https://ijisae.org/index.php/IJISAE/article/view/2966 (accessed on 20 February 2025).
52. Bawankule, R.; Gaikwad, V.; Kulkarni, I.; Kulkarni, S.; Jadhav, A.; Ranjan, N. Visual detection of waste using YOLOv8. In Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 14–16 June 2023; pp. 869–873. [CrossRef]
53. Jun, E.L.T.; Tham, M.L.; Kwan, B.H. A Comparative Analysis of RT-DETR and YOLOv8 for Urban Zone Aerial Object Detection. In Proceedings of the 2024 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 29 June 2024; pp. 340–345. [CrossRef]
54. Guemas, E.; Routier, B.; Ghelfenstein-Ferreira, T.; Cordier, C.; Hartuis, S.; Marion, B.; Bertout, S.; Varlet-Marie, E.; Costa, D.; Pasquier, G. Automatic patient-level recognition of four Plasmodium species on thin blood smear by a real-time detection transformer (RT-DETR) object detection algorithm: A proof-of-concept and evaluation. *Microbiol. Spectr.* **2024**, *12*, e01440-23. [CrossRef] [PubMed]
55. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: https://docs.ultralytics.com/models/yolov8 (accessed on 20 February 2025).
56. Yu, Z.; Wan, L.; Yousaf, K.; Lin, H.; Zhang, J.; Jiao, H.; Yan, G.; Song, Z.; Tian, F. An enhancement algorithm for head characteristics of caged chickens detection based on cyclic consistent migration neural network. *Poult. Sci.* **2024**, *103*, 103663. [CrossRef] [PubMed]
57. Dumitriu, A.; Tatui, F.; Miron, F.; Ionescu, R.T.; Timofte, R. Rip current segmentation: A novel benchmark and yolov8 baseline results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1261–1271.
58. Aishwarya, N.; Kumar, V. Banana ripeness classification with deep CNN on NVIDIA jetson Xavier AGX. In Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 11–13 October 2023; pp. 663–668. [CrossRef]
59. Soylu, E.; Soylu, T. A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition. *Multimed. Tools Appl.* **2024**, *83*, 25005–25035. [CrossRef]
60. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An optimized YOLOv8 network for tiny UAV object detection. *Electronics* **2023**, *12*, 3664. [CrossRef]
61. Kumari, S.; Gautam, A.; Basak, S.; Saxena, N. Yolov8 based deep learning method for potholes detection. In Proceedings of the 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Gwalior, India, 10–11 December 2023; pp. 1–6. [CrossRef]
62. Pereira, G.A. Fall detection for industrial setups using yolov8 variants. *arXiv* **2024**, arXiv:2408.04605. [CrossRef]
63. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
64. Lu, L.L.; Li, X.L.; Wu, Y.W.; Chen, B.C. Enhanced RT-DETR for Traffic Sign Detection: Small Object Precision and Lightweight Design. Research Square [Preprint]. 12 November 2024. Available online: https://www.researchsquare.com/article/rs-5351138/v1 (accessed on 20 February 2025).
65. Liu, M.; Wang, H.; Du, L.; Ji, F.; Zhang, M. Bearing-detr: A lightweight deep learning model for bearing defect detection based on rt-detr. *Sensors* **2024**, *24*, 4262. [CrossRef] [PubMed]

66. Dai, L.; Wang, D.; Song, F.; Yang, H. Concrete Bridge Crack Detection Method Based on an Improved RT-DETR Model. In Proceedings of the 2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC), Mianyang, China, 5–7 July 2024; pp. 172–175.
67. Liu, Y.; Cao, Y.; Sun, Y. Research on Rail Defect Recognition Method Based on Improved RT-DETR Model. In Proceedings of the 2024 5th International Conference on Computer Engineering and Application (ICCEA), Hangzhou, China, 12–14 April 2024; pp. 1464–1468. [CrossRef]
68. Li, X.; Cai, M.; Tan, X.; Yin, C.; Chen, W.; Liu, Z.; Wen, J.; Han, Y. An efficient transformer network for detecting multi-scale chicken in complex free-range farming environments via improved RT-DETR. *Comput. Electron. Agric.* **2024**, *224*, 109160. [CrossRef]
69. Cao, X.; Wang, H.; Wang, X.; Hu, B. DFS-DETR: Detailed-feature-sensitive detector for small object detection in aerial images using transformer. *Electronics* **2024**, *13*, 3404. [CrossRef]
70. Tang, S.; Yan, W. Utilizing RT-DETR model for fruit calorie estimation from digital images. *Information* **2024**, *15*, 469. [CrossRef]
71. Yu, C.; Chen, X. Railway rutting defects detection based on improved RT-DETR. *J. Real-Time Image Process.* **2024**, *21*, 146. [CrossRef]
72. Bayraktar, E.; Basarkan, M.E.; Celebi, N. A low-cost UAV framework towards ornamental plant detection and counting in the wild. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 1–11. [CrossRef]
73. Mamdouh, N.; Khattab, A. YOLO-based deep learning framework for olive fruit fly detection and counting. *IEEE Access* **2021**, *9*, 84252–84262. [CrossRef]
74. Ghorbanzadeh, O.; Crivellari, A.; Ghamisi, P.; Shahabi, H.; Blaschke, T. A comprehensive transferability evaluation of U-Net and ResU-Net for landslide detection from Sentinel-2 data (case study areas from Taiwan, China, and Japan). *Sci. Rep.* **2021**, *11*, 14629. [CrossRef]
75. Torkul, O.; Selvi, İ.H.; Şişci, M.; Diren, D.D. A New Model for Assembly Task Recognition: A Case Study of Seru Production System. *IEEE Access* **2024**. [CrossRef]
76. Jang, W.S.; Kim, S.; Yun, P.S.; Jang, H.S.; Seong, Y.W.; Yang, H.S.; Chang, J.S. Accurate detection for dental implant and peri-implant tissue by transfer learning of faster R-CNN: A diagnostic accuracy study. *BMC Oral Health* **2022**, *22*, 591. [CrossRef]
77. Qadir, H.A.; Shin, Y.; Solhusvik, J.; Bergsland, J.; Aabakken, L.; Balasingham, I. Toward real-time polyp detection using fully CNNs for 2D Gaussian shapes prediction. *Med. Image Anal.* **2021**, *68*, 101897. [CrossRef] [PubMed]
78. Su, Y.; Cheng, B.; Cai, Y. Detection and Recognition of Traditional Chinese Medicine Slice Based on YOLOv8. In Proceedings of the 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT), Qingdao, China, 21–24 July 2023; pp. 214–217. [CrossRef]
79. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
80. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-time flying object detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972. [CrossRef]
81. Ding, J.; Zhang, J.; Zhan, Z.; Tang, X.; Wang, X. A precision efficient method for collapsed building detection in post-earthquake UAV images based on the improved NMS algorithm and faster R-CNN. *Remote Sens.* **2022**, *14*, 663. [CrossRef]
82. Jin, M.; Zhang, J. Research on Microscale Vehicle Logo Detection Based on Real-Time DEtection TRansformer (RT-DETR). *Sensors* **2024**, *24*, 6987. [CrossRef]
83. Wu, M.; Qiu, Y.; Wang, W.; Su, X.; Cao, Y.; Bai, Y. Improved RT-DETR and its application to fruit ripeness detection. *Front. Plant Sci.* **2025**, *16*, 1423682. [CrossRef]
84. Parekh, A. Comparative Analysis of YOLOv8 and RT-DETR for Real-Time Object Detection in Advanced Driver Assistance Systems. Master's Thesis, The University of Western Ontario, London, ON, Canada, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.