micromachines

MDPI

Article

Deep Learning for Clothing Style Recognition Using YOLOv5

Yeong-Hwa Chang [1,2,*] and Ya-Ying Zhang [1]

[1] Department of Electrical Engineering, Chang Gung University, Taoyuan City 333, Taiwan
[2] Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City 243, Taiwan
* Correspondence: yhchang@mail.cgu.edu.tw

Abstract: With the rapid development of artificial intelligence, much more attention has been paid to deep learning. However, as the complexity of learning algorithms increases, the needs of computation power of hardware facilities become more crucial. Instead of the focus being on computing devices like GPU computers, a lightweight learning algorithm could be the answer for this problem. Cross-domain applications of deep learning have attracted great interest amongst researchers in academia and industries. For beginners who do not have enough support with software and hardware, an open-source development environment is very helpful. In this paper, a relatively lightweight algorithm YOLOv5s is addressed, and the Google Colab is used for model training and testing. Based on the developed environment, many state-of-art learning algorithms can be studied for performance comparisons. To highlight the benefits of one-stage object detection algorithms, the recognition of clothing styles is investigated. The image samples are selected from datasets of fashion clothes and the web crawling of online stores. The image data are categorized into five groups: plaid; plain; block; horizontal; and vertical. Average precison, mean average precison, recall, F1-score, model size, and frame per second are the metrics used for performance validations. From the experimental outcomes, it shows that YOLOv5s is better than other learning algorithms in the recognition accuracy and detection speed.

Keywords: clothing style recognition; deep learning; one-stage detection; YOLO

Citation: Chang, Y.-H.; Zhang, Y.-Y. Deep Learning for Clothing Style Recognition Using YOLOv5. *Micromachines* **2022**, *13*, 1678. https://doi.org/10.3390/mi13101678

Academic Editor: Arman Roohi

Received: 27 July 2022
Accepted: 30 September 2022
Published: 5 October 2022

1. Introduction

Artificial intelligence and deep learning with the research of related technologies have been rapidly growing. Image recognition has not only appeared in industry but also in daily life, such as the automatic license plate identification system in a parking garage [1], and the maturity of fruits and vegetables [2], etc. In the fashion industry, image recognition can assist in clothing design, clothing accessories collocation and related data analysis and integration [3]. Due to the rise of e-commerce, online consumption has become a popular consumption behavior with the benefits of time and cost savings. One way the clothing industry can attract consumers is the focus on visual feelings, so the style of image presentation is very important. The official website of a clothing store can show the color and style of the clothes. Furthermore, they could display many try-on pictures and a selection of accessories. This should help the customers who make the clothing matching easier. There are also many applications in agriculture, such as automatic identification of fruits [4], the quality detection of cherries [5], and the identification of the maturity of strawberries [6]. In the research of [7], pests and diseases can be identified from the inspection of the leaves of sweet peppers. Other interesting applications can be also found, such as the identification of trees infested by beetles [8], tomato virus [9], the position of fruit picking [10], and apple picking by robots [11].

Recently, one-stage objection has attracted a lot of attention, such as YOLO and SSD [12]. The emergence of YOLO has greatly improved the speed of object detection. The application areas of YOLO are very wide. For example, in the medical field, related identification applications include cervical cancer [13], blood cells [14], colorectal cancer [15],

venipuncture [16], etc., Applications also include the auxiliaries for hearing and visually impaired people [17] and sign language identification [18]. Furthermore, due to the influence of epidemics, there were many mask identification studies based on the use of different algorithms [19–22]. There has been a lot of interest in self-driving cars such as automatic navigation [23], driver's assistance [24], vehicle detection [25], vehicle tracking [26], and blind-spot detection [27]. The use of deep learning for target identification improves the accuracy of identification and also accelerates the speed of identifying targets. In the study of [28], 42 different patterns of traditional clothes were considered, and the classification accuracy was higher than 90% by using convolutional neural networks. In [29], VGG19 was used as the feature extractor to identify the Indonesia dress pattern. Faster R-CNN and some other convolutional neural networks were considered for the recognition of traditional handicraft weaving patterns. The results of the study showed that the accuracy of Faster R-CNN reached 82.14% [30].

In [31], the discrimination of clothing types were addressed, where the multilayer perceptrons and convolutional neural networks were applied on the Fashion-MNIST dataset. Moreover, images from DeepFashion and FashionMNIST datasets were selected for the recognition of clothing styles, and YOLO and ResNet were used for the study of accuracy improvement [32]. In the application of a surveillance system of clothing recognition [33], categories like suits, shirts, and jeans, etc., were considered, where the average recall rate was 80%. The study in [34] showed that with the combination of batching normalization and residual skip connections, CNNs can make the overall improvement of accuracy up to 92.54%. In [35], the R-CNN network framework combined with Softmax was applied for extracting features of shirt images, and the results indicated that an accuracy of 73.59% and a recall rate of 83.84% can be attained. In the research of clothing recognition using deep learning techniques, DeepFashion and DeepFashion2 are two datasets that have attracted lots of attention [36–38]. For example, fashion style recognition can help e-commerce clothing retrieval and recommendation. In order to solve the problem of classification errors caused by the same style of clothing images in different visual appearances, a joint fashion style recognition model was proposed, which was verified using the DeepFashion dataset [37]. In practice, it is necessary to establish the target object before performing garment inspection. To reduce the lengthy process of labelling, a R-CNN network was used to resolve this problem, where the DeepFashion2 dataset was considered for verification [38].

In [39], a CNN model was proposed for clothing classification, where the algorithms YOLOv3 and Tiny YOLO were analyzed. Furthermore, a learning framework was proposed for automatically classifying clothing genres based on the visually differentiable style elements [40]. In [41], an imbalanced deep learning framework was presented for large scale visual data learning, where a class rectification loss function was characterized by batch-wise incremental minority class rectification with a scalable hard mining principle. In [42], data mining and symmetry-based learning techniques were addressed to create a classification model for predicting the garment category. Based on a fashion attributes recognition network, the multi-task learning framework to improve fashion recognition was proposed to leverage the noisy labels and generate corrected labels [43]. In [44], both deep learning and image processing techniques were applied to automatically recognize and classify logos, stripes, colors, and other features of clothing. An intelligent fashion technique based on deep learning for efficient fashion product searches and recommendations were proposed, including a sketch-product fashion retrieval model and a user preference fashion recommendation model [45]. In [46], CNN networks were used to train images of different fashion styles, in which the performance was validated using the Fashion-MNIST dataset. In addition, the problem of landmark point detection in clothes was considered, where a deep learning framework was proposed to predict clothing categories and attributes [47]. In [48], CNN networks combined with a self-attention mechanism were proposed to represent clothing attributes that were more fine-grained.

The target objects of this study are the tops of clothes. The sample images are selected from the datasets of fashion clothes, DeepFashion and DeepFashion2, and Web

crawling. The styles of the chosen objects are divided into five categories: plaid; plain; block; horizontal; and vertical. Google Colab virtual machine is adopted to complete the learning model training and testing. Both the two-stage and one-stage object algorithms, R-CNN and YOLO series, are discussed for performance comparisons. The differences of the R-CNN and related modified ones like Fast R-CNN and Faster R-CNN will be concisely explained. Furthermore, the key concepts and crucial differences in YOLOv1~YOLOv5 will be highlighted. The main contributions of this paper are listed as follows:

1. The process of building an integrated environment based on Google Colab is concisely explained so that those interested in deep learning can easily get involved in the study, especially for beginners who lack their own powerful GPU computer;
2. The crucial differences among R-CNN, Fast R-CNN, and Faster R-CNN are explained concisely such that readers can have easier access to the key concepts of typical two-stage algorithms;
3. The essential modifications in the development of the YOLO series are succinctly explained such that the readers will know better about the cores of each generation of one-stage YOLOs;
4. Experimental results about the recognition of clothing styles are provided along with each essential step. Furthermore, the integration of experimental outcomes are given for performance validations. The indexes of average precision (AP), mean average precision (mAP), recall, F1-score, model size, and frames per second (FPS) are investigated.

2. Materials and Methods

2.1. Object Detection

In general, object detection algorithms can be classified into different groups in accordance with one- or two-stage, whether to use anchor frames, and the labeling methods. The YOLO series and SSD are typical one-stage algorithms, while the popularly addressed two-stage ones are R-CNN, Fast R-CNN and Faster R-CNN. Anchor boxes are a set of predefined boxes that are helpful to identify the detected objects with the information scale and aspect ratio. Anchor-based algorithms include YOLOv2 to YOLOv5, Faster R-CNN and SSD, as shown in Table 1 [49]. As the labelling techniques, regional proposals are commonly adopted in R-CNN series. On the other hand, the intersection of union (IoU) is the labelling method used in YOLOv2~YOLOv5.

Table 1. Object detection algorithms [49].

Stage	One-stage	YOLOv1~YOLOv5, SSD
	Two-stage	R-CNN, Fast R-CNN, Faster R-CNN
Anchor	Anchor-free	YOLOv1
	Anchor-based	YOLOv2~YOLOv5, SSD, Faster R-CNN
Labelling	Regional proposal	R-CNN, Fast R-CNN, Faster R-CNN
	Key point	YOLOv1
	IoU	YOLOv2~YOLOv5, SSD, Faster R-CNN

2.1.1. Two-Stage

R-CNN, Fast R-CNN, and Faster R-CNN are typical two-stage detection algorithms, where the feature extraction and classification are two unique steps for object detection. In Figure 1, the concept of Regional of Interest (RoI) was proposed in Fast R-CNN such that the feature extraction can be more efficient. To further reduce the detection time, a Regional of Proposal Network was presented for Faster R-CNN. Due to the increase in architecture complexity, Faster R-CNN requires high efficiency computation capability for real-time object detection.

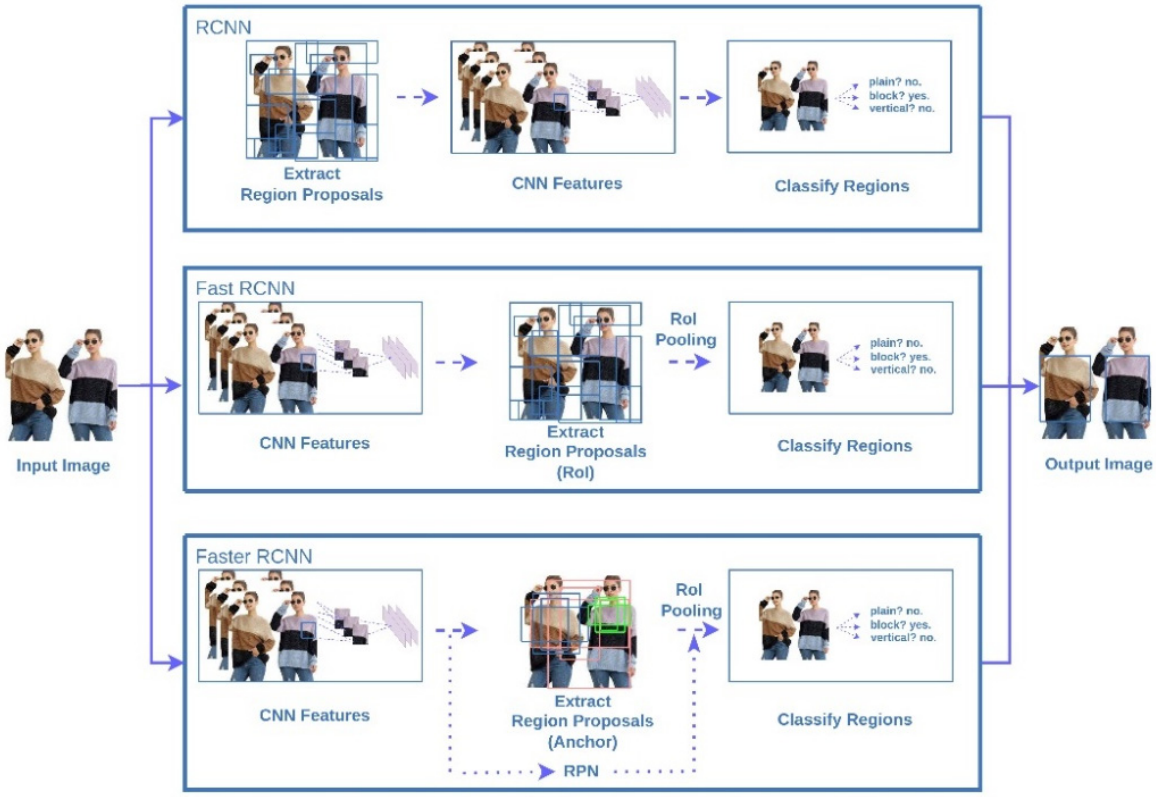

Figure 1. Two-stage object detection algorithms.

2.1.2. One-Stage

The You Only Look Once (YOLO) series, first presented in 2016, are one-stage objection algorithms. Compared to the two-stage Faster R-CNN, the regression-based classification is used to replace the RoI pooling layer such that detection time can be reduced, as shown in Figure 2. YOLOv5 is a newly developed algorithm in the YOLO series. Basically, YOLOv5 has a relatively small size that will make the implementation on mobile devices more feasible.

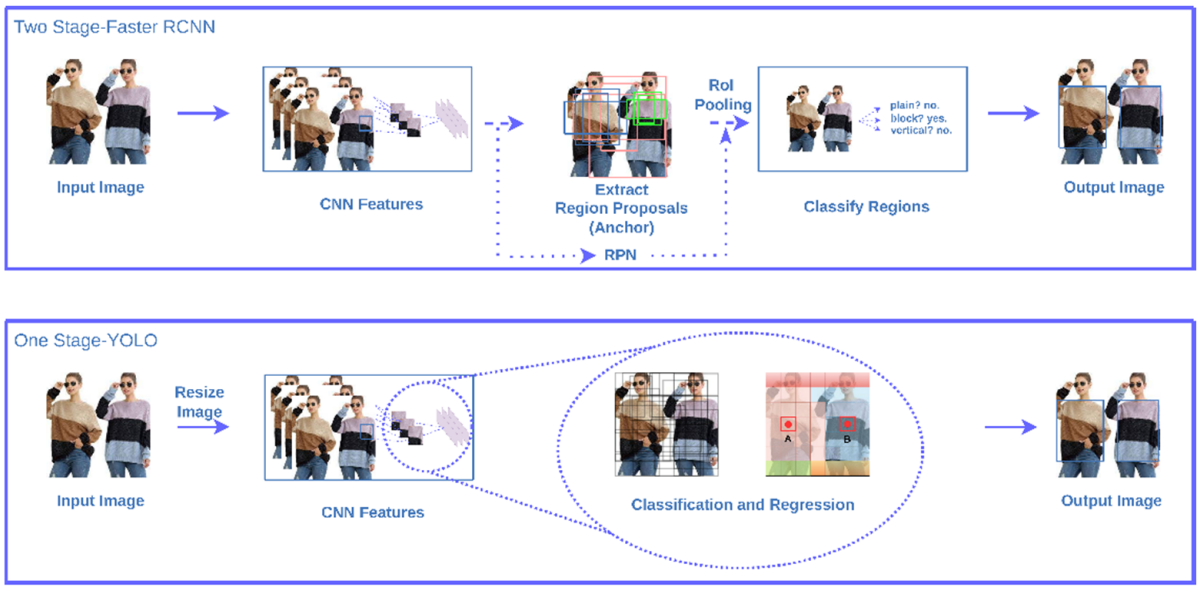

Figure 2. Comparison between one-stage and two-stage object detection algorithms.

YOLO is a family of object detection architectures and models pretrained on the COCO dataset. In contrast to the two-stage detectors based on the region proposal method, the representative one-stage detector, YOLO, uses the idea of regression to predict all the categories along with the corresponding confidence and bounding box information, which can speed up the detection greatly, although this comes at the expense of slightly reduced precision. YOLOv5 is a state-of-art deep learning framework, and the whole network is composed of four parts: input; backbone; neck; and head. In the series of YOLOv5, YOLOv5s has the benefit of less model size that would provide the potential for future interesting applications such as edge AI and machine learning on a micro-controller unit (MCU). In this paper, the details about the key technologies and how to build the YOLOv5s machine learning framework will be discussed. Furthermore, the feasibility will be validated for the clothing style recognition. The performance including the computation cost and recognition accuracy will be compared with the YOLOv3, YOLOv4 and traditional two-stage R-CNN frameworks.

2.2. Implementation of Deep Learning Framework

System Built with Google Colab

Deep learning usually relies on a GPU computer for high-efficiency computation. For beginners who are interested in the deep learning topics, system building may not be affordable. In this paper, an open-resource Google Colab is adopted to build the environment for learning model training and testing. In Figure 3, three steps are performed for the YOLOv1~YOLOv4 installation and corresponding model training and testing. In the creation of the Google Colab project, a dataset is required to be well-prepared and uploaded into the created YOLO folder. Next, the user must confirm all the settings of the GPU and CUDA before the installation of YOLO algorithms. Finally, provided with the pre-defined weights, the model training and testing of the selected YOLO algorithm can be sequentially performed. It is noticed that Darknet is the neural network framework here.

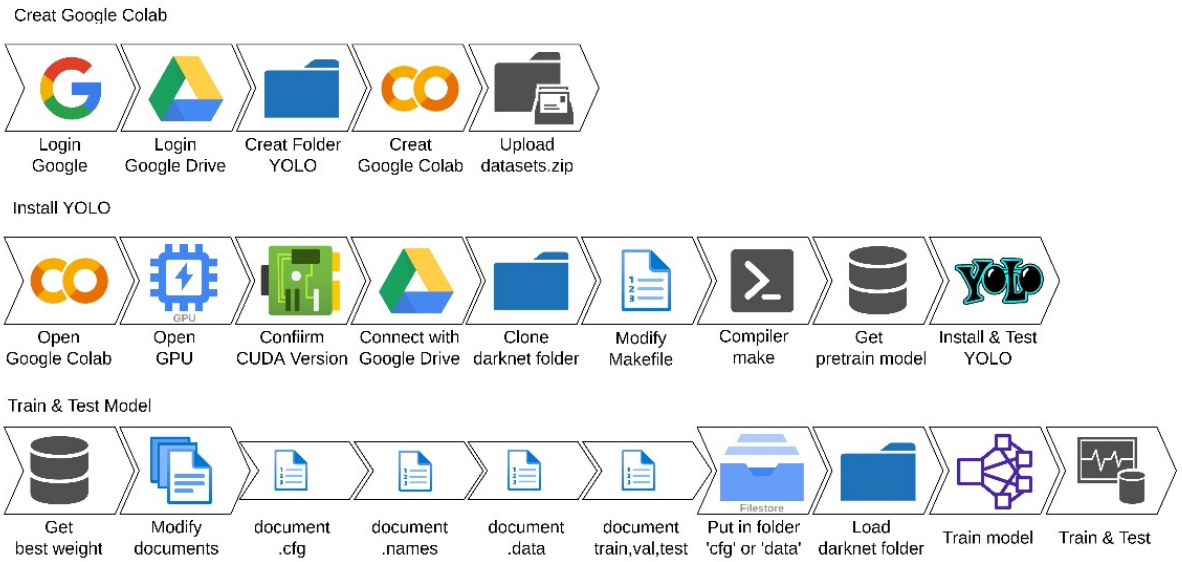

Figure 3. Implementation of YOLOv1~YOLOv4 for model training and testing with Google Colab.

The building process of YOLOv5 is like the process in Figure 3. While installing the environment of YOLOv5, the neural network framework is PyTorch, as shown as Figure 4. A typical YOLOv5 series includes YOLOv5x, YOLOv5l, YOLOv5m, and YOLOv5s. In the comparison between each other, YOLOv5s has a minimum model size.

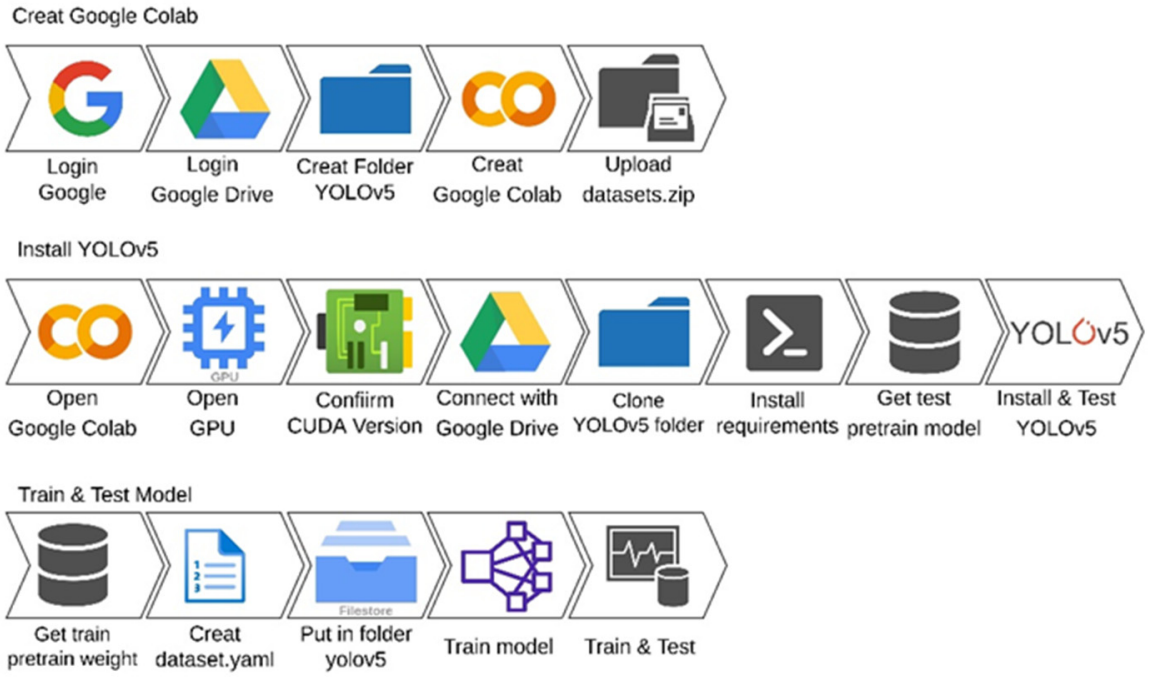

Figure 4. Implementation of YOLOv5 for model training and testing with Google Colab.

2.3. YOLO Algorithms

You Only Look Once (YOLO) is a typical one-stage object detection algorithm. It is formulated as a regression problem, from which the bounding boxes and class probabilities can be predicted directly from full images in one evaluation [50]. The first generation of YOLO, YOLOv1, was proposed in 2016 by Joseph Redmon [51]. Compared to the two-stage object detection methods, YOLO series have significant improvements in detection speed and model size. In YOLO series, Darknet is the deep learning framework adopted for YOLO v1~v4, while the framework of YOLOv5 is PyTorch. YOLOv1, inherited from GooLeNet, considers the objection detection as a regression problem. The processing speed is fast, but the recognition accuracy is less than the two-stage algorithms. In particular, the recognition outcomes of small objects need a certain degree of improvement. Based on YOLOv1, YOLOv2 has shown some progress on accuracy and processing speed with the use of Darknet-19 framework. However, the recognition of small objects is still unsatisfactory. In YOLOv3, a Neck module is added for feature fusion, the output dimension is increased to three, and the recognition of small objects is better. YOLOv4, combined with many technical studies and experimental results, provides better accuracy of object detection. YOLOv5 is a newly developed algorithm, which has a much smaller model size than the aforementioned algorithms [52].

Backbone, Neck and Head are the main modules in YOLO. To concisely explain the key differences in the YOLO series, the scheme diagrams of YOLOs are shown in Figure 5. In Figure 5a, it can be seen that YOLOv1 basically uses the traditional convolutional layer and a fully connected network to extract features. On the other hand, the feature extraction of YOLOv2 is performed with a Darknet network. In YOLOv3~YOLOv5, a certain feature fusion is added that can integrate related information extracted from a group of images without losing data. The feature extraction and fusion methods of YOLOv3~YOLOv5 are shown in Figure 5b [53–55]. The function of FOCUS is mainly to increase the speed, where the image will be sliced and rearranged. SPP, known as the spatial pyramid pooling method, solves the problem of picture distortion caused by image deformation stretching or cropping, that greatly improves the speed of generating candidate boxes and the computational cost reduction. In addition, the use of FPN (Feature Pyramid Network)

and PAN (Path Aggregation Network) will complete the feature fusion of high and low layers, so that object detection can be improved, especially in the detection of small objects.

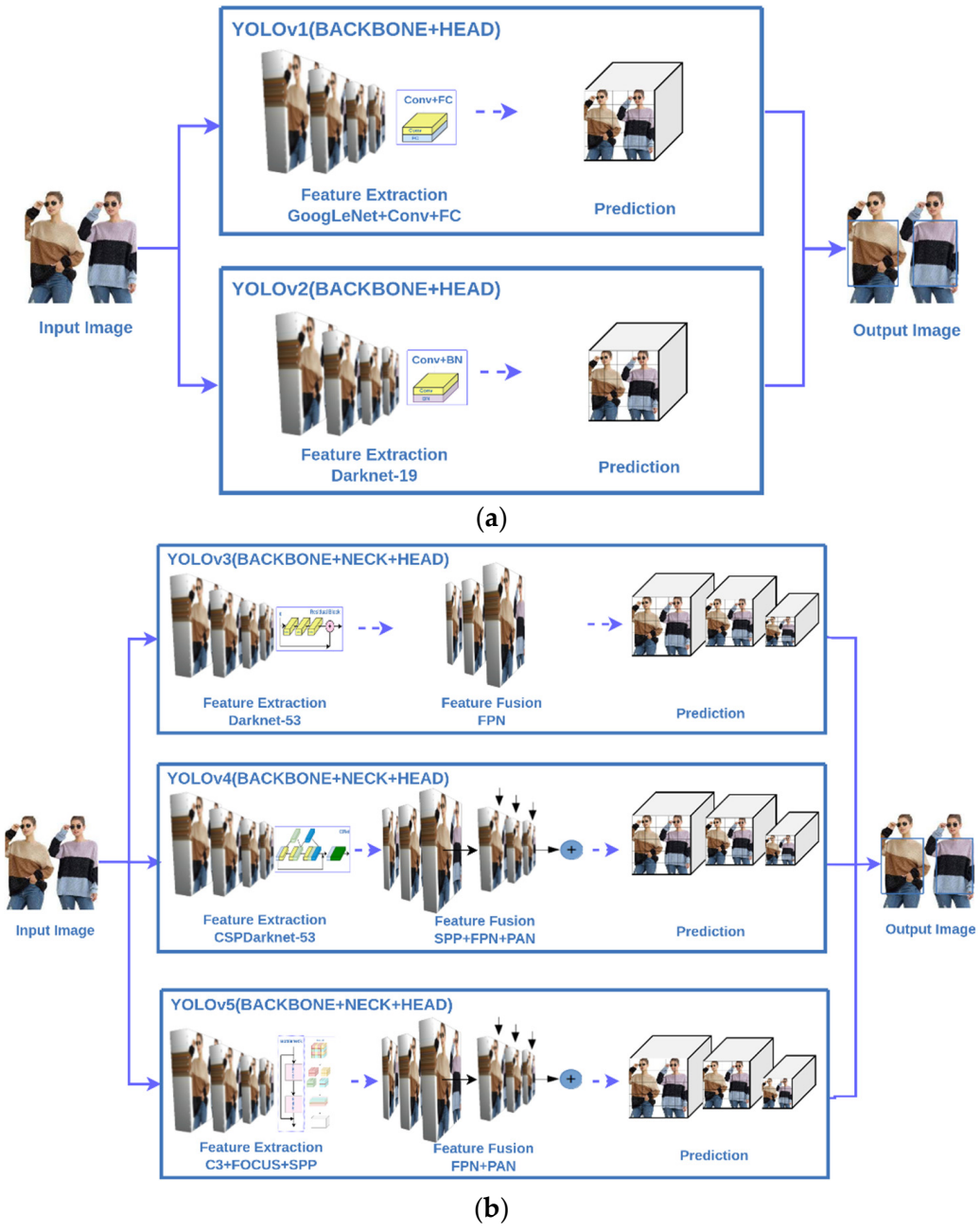

Figure 5. Architecture of YOLO: (**a**) YOLOv1~YOLOv2, (**b**) YOLOv3~YOLOv5.

3. Results

3.1. Integrated Develepmental Environment

The whole scheme of developed environment for YOLOv5s is shown in Figure 6. Google Colab is used to build the experimental environment, where the operating system is Xubuntu [56]. Xubuntu is an elegant and easy to use operating system, coming from Xfce, which is a stable, light and configurable desktop environment. In addition, the virtual GPU is Nvidia Tesla K80 with CUDA 11.2 and OpenCV 4.1.2. PyCharm is a Python based development environment providing many essential tools for Python developers, tightly integrated to create a convenient environment to access the command line, connect to a database, create a virtual environment, and manage your projects [57]. For the image

labelling, LabelImg is a graphical image annotation tool, written in Python and using PyQt for its graphical interface. Annotations are saved as XML files in PASCAL VOC format. Moreover, it also supports YOLO and CreateML formats [58,59]. During the labelling process, the area of object and its class belonging will be determined. The information format is selected to fit the YOLO format. This information contains the class, x-y coordinate, and length-width of objects, as shown in Figure 7.

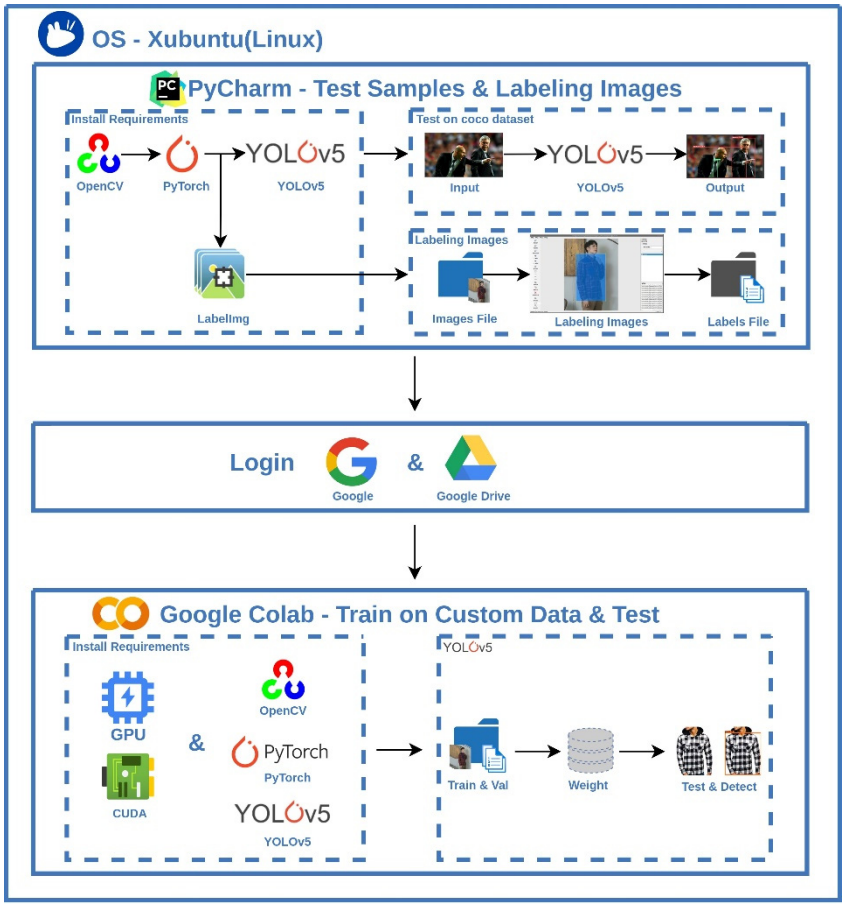

Figure 6. The scheme of the proposed YOLOv5s deep learning environment.

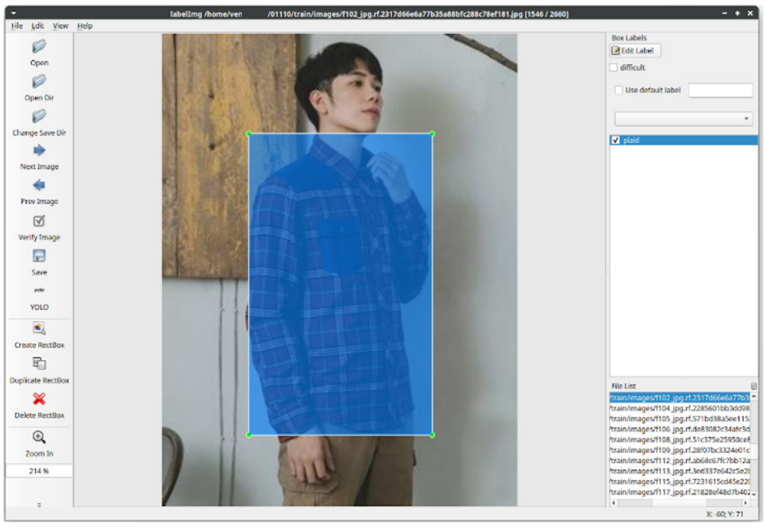

Figure 7. Outcome of image labelling.

3.2. Dataset

In this paper, the image datasets are collected from five open resources. For example, DeepFashion is a dataset of 50 classes fashion clothes, where the total number of images is over 80 million [60]. In addition, DeepFashion2 contains 49.1 million images of 13 classes of clothes [61]. Moreover, image samples are also collected from Google pictures and the sites of web fashion shops. The initial amount of clothing samples is 4455. With additional data augmentation, the number of total image samples used for this research is 5141, shown in Figure 8. The image samples are categorized into five groups: plaid; plain; block; horizontal; and vertical. In Figure 8, the number of each category is indicated, such as plaid having 1024 samples, etc.

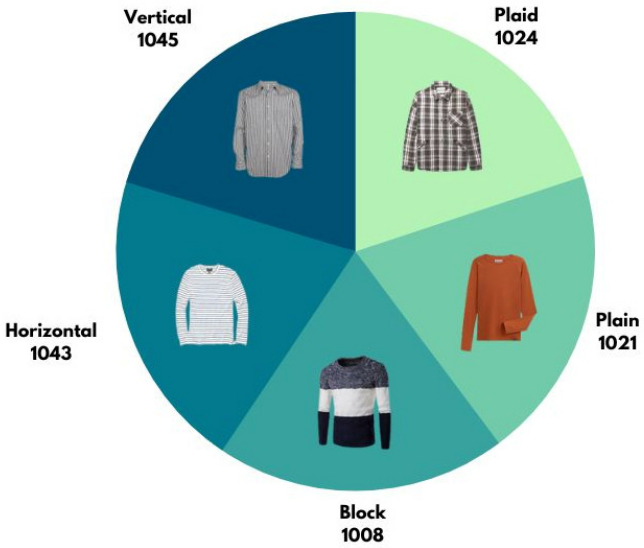

Figure 8. Image samples of each category of shirts.

The collected image samples will be divided into training set, validation set, and test set. Training set is used to train the learning model. Validation set is used to adjust and select models. Finally, the model evaluation is performed with the test set. According to the 60:10:30 ratio, the number of samples of each category in the stage of training, validation, and testing are shown in Figure 9.

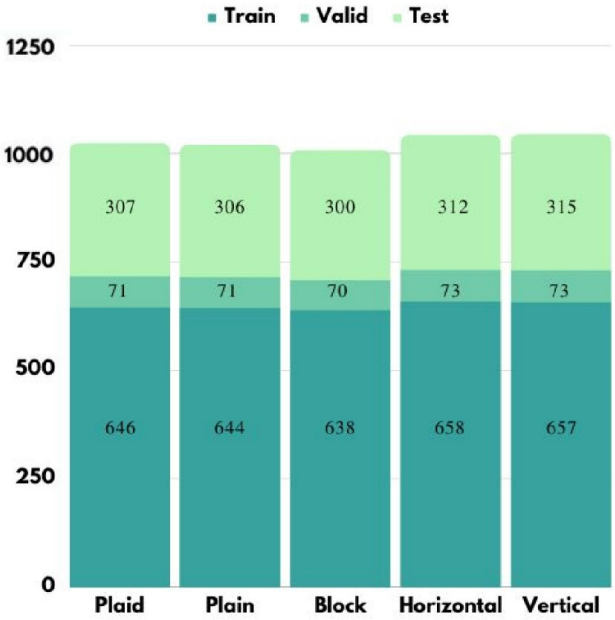

Figure 9. Image samples of each category in the process of training, validation, and testing.

3.3. Integration Testing Results

The YOLOv5 is mainly composed with four modules, i.e., Mosaic augmentation, Backbone, Neck, and Head. The corresponding experimental outcomes are illustrated in the following.

Image augmentation creates new training examples out of existing training data. It is not easy to truly capture an image for every real-world scenario. Thus, adjusting existing training data to generalize to other situations allows the model to learn from a wider array of situations. The idea behind Mosaic is simply taking four images and randomly combining them into a single image, as shown in Figure 10.

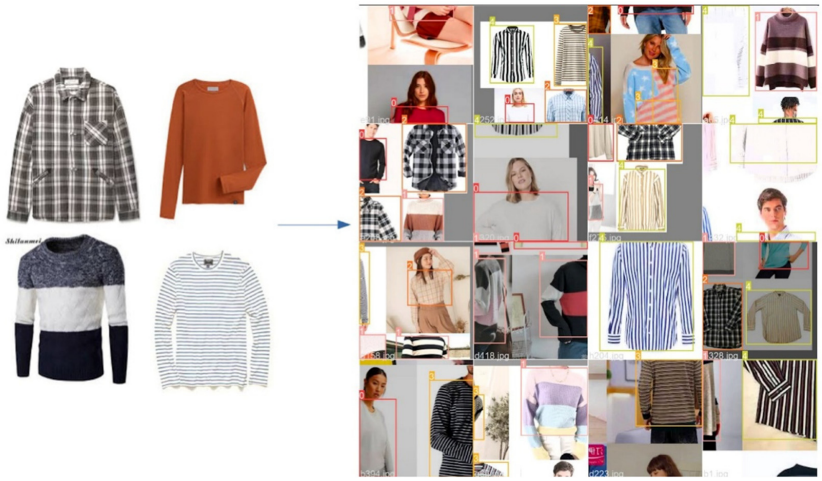

Figure 10. Mosaic image augmentation.

In object detection, bounding boxes are usually used to describe the spatial location of an object. For solving the problem of objects overlapping in an image, YOLO uses different sizes of rectangles to the anchor boxes. Basically, for each anchor box, one can calculate which object's bounding box has the highest overlap divided by non-overlap. During the training process, the predicted bounding box is iteratively compared to the ground-truth to generate the optimal bounding rectangle of an object, shown as Figure 11. In Figure 11a, each possible class is represented with rectangles of the same color. The optimal bounding box of the object is shown in Figure 11b.

(**a**) (**b**)

Figure 11. Anchor boxes: (**a**) candidates during the processing, (**b**) final choice.

The YOLO Backbone is a convolutional neural network that pools image pixels to form features at different granularities. The Backbone in the deep learning architecture basically acts as a feature extractor. The Neck is a subset of the bag of specials, and it basically collects feature maps from different stages of the backbone. In simple terms, it is a feature aggregator. This is the part of the network that makes the bounding box and class prediction. The Head is also known as the object detector. Basically, the Head can be used to find the region where the object might be present, but with no information about which object it is. The corresponding outcomes are shown in Figures 12 and 13. Furthermore, some testing results for the trained YOLOv5s model are shown in Figure 14. To validate the feasibility of the constructed YOLO learning architecture, some two- and one-stage learning algorithms are adopted for performance comparisons, as shown in Tables 2 and 3. In Table 2, average precision (AP), mean average precision (mAP), model size, and frame per second (FPS) are addressed. The metrics of precision, recall, and F1-score are shown in Table 3. In the training and testing processes, the losses and performance metrics of each epoch are shown in Figure 15, where the number of epochs is 100. From the outcomes in Tables 2 and 3, the YOLOv5s has better performance in all metrics of interest. Furthermore, the metrics in the training and testing processes of YOLOv5s with 300 epochs are shown in Figure 16. In Tables 2 and 3, both the metrics for performance validations of YOLOv5s with 100 epochs and 300 epochs are included. Efficiency is generally improved with more epochs; however, the model size is slightly increased.

Figure 12. Feature extraction and fusion by Backbone and Neck.

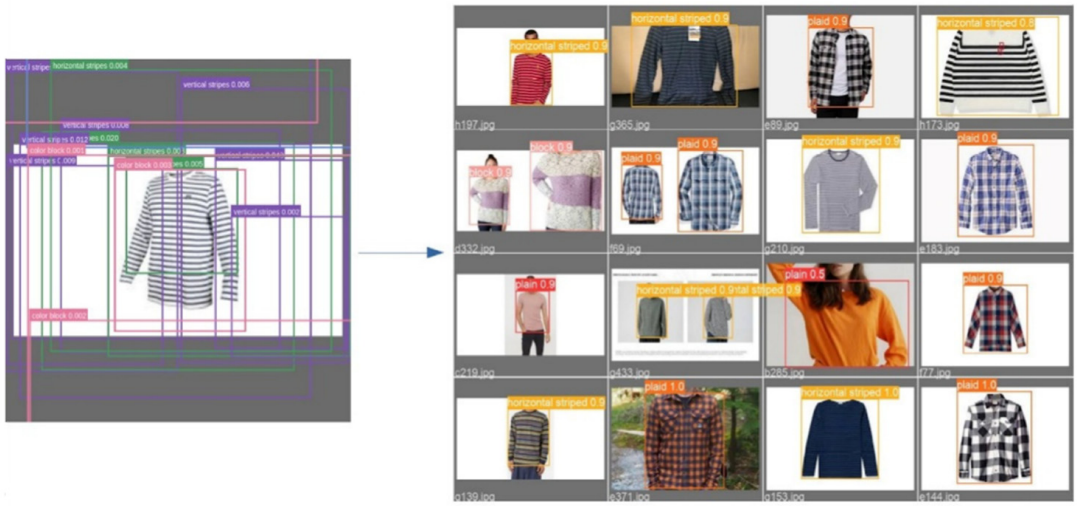

Figure 13. Object prediction by Head.

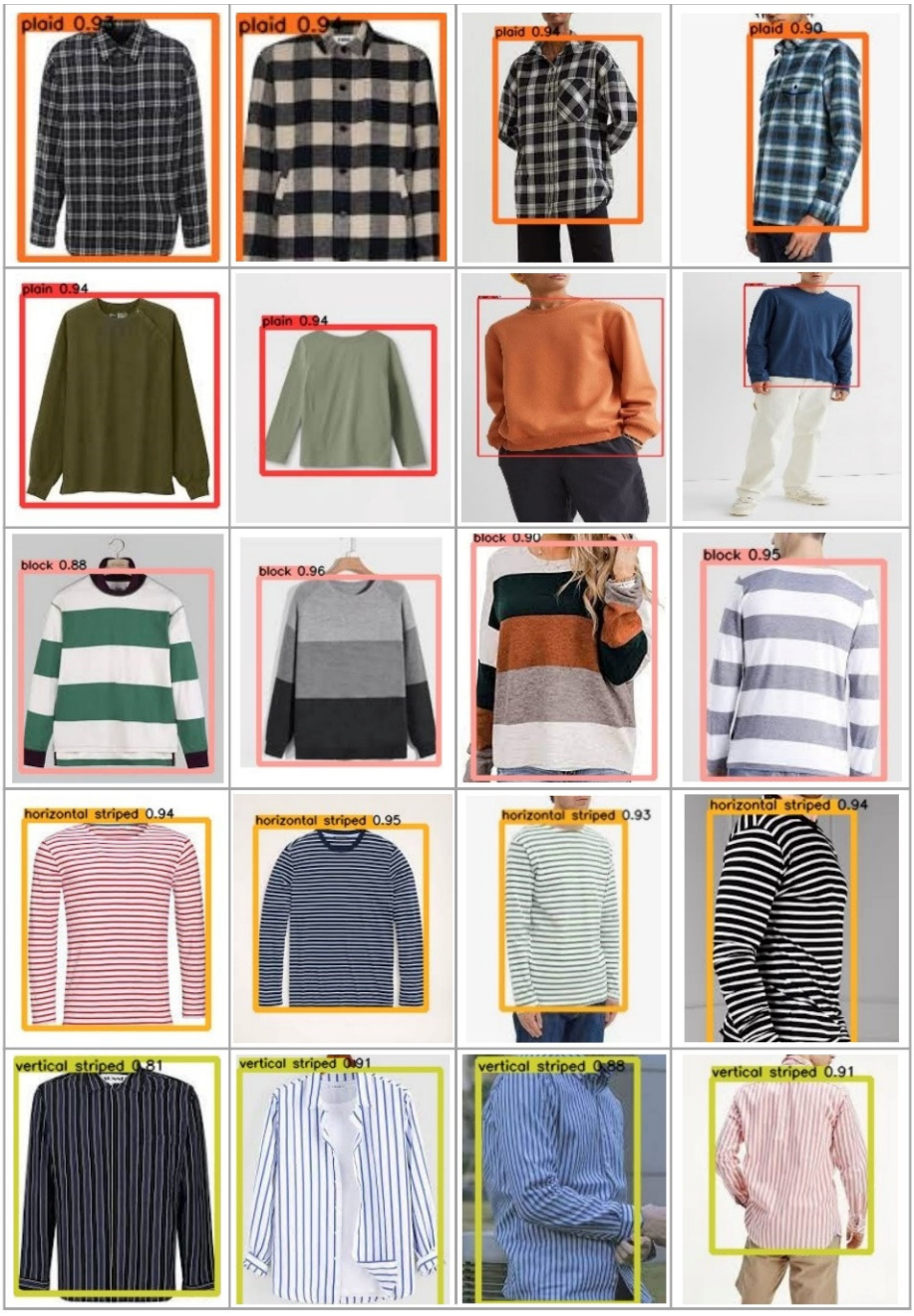

Figure 14. Testing results (top to bottom: plaid, plain, block, horizontal, vertical).

Table 2. Performance comparisons of different deep learning algorithms.

	AP (%)					mAP (%)	Model Size	FPS
	Plaid	Plain	Block	Horizon	Vertical			
Faster R-CNN	87.0	93.3	100	85.9	85.7	90.3	175.5	7
YOLOv3-tiny	89.7	95.7	90.0	92.0	91.3	91.7	33.4	28.5
YOLOv4-tiny	98.9	98.0	94.1	93.0	97.0	96.2	23.1	24.5
YOLOv5s (100 epochs)	99.3	99.1	99.0	97.3	94.9	98.4	14	40
YOLOv5s (300 epochs)	98.5	99.4	99.3	99.4	96.6	99.1	14.4	40

Table 3. Performance comparisons of different deep learning algorithms (cont'd).

	Precision	Recall	F1-Score
Faster R-CNN	0.91	0.89	0.90
YOLOv3-tiny	0.93	0.85	0.89
YOLOv4-tiny	0.90	0.95	0.93
YOLOv5s (100 epochs)	0.98	0.96	0.97
YOLOv5s (300 epochs)	0.97	0.98	0.97

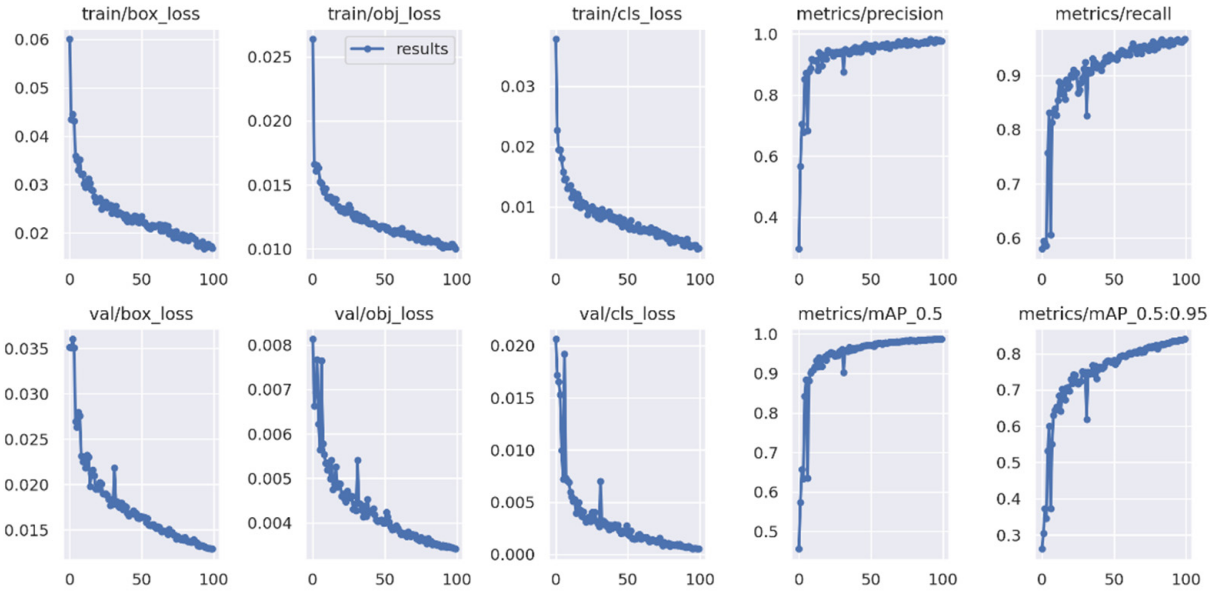

Figure 15. Integration results of YOLOv5s (100 epochs).

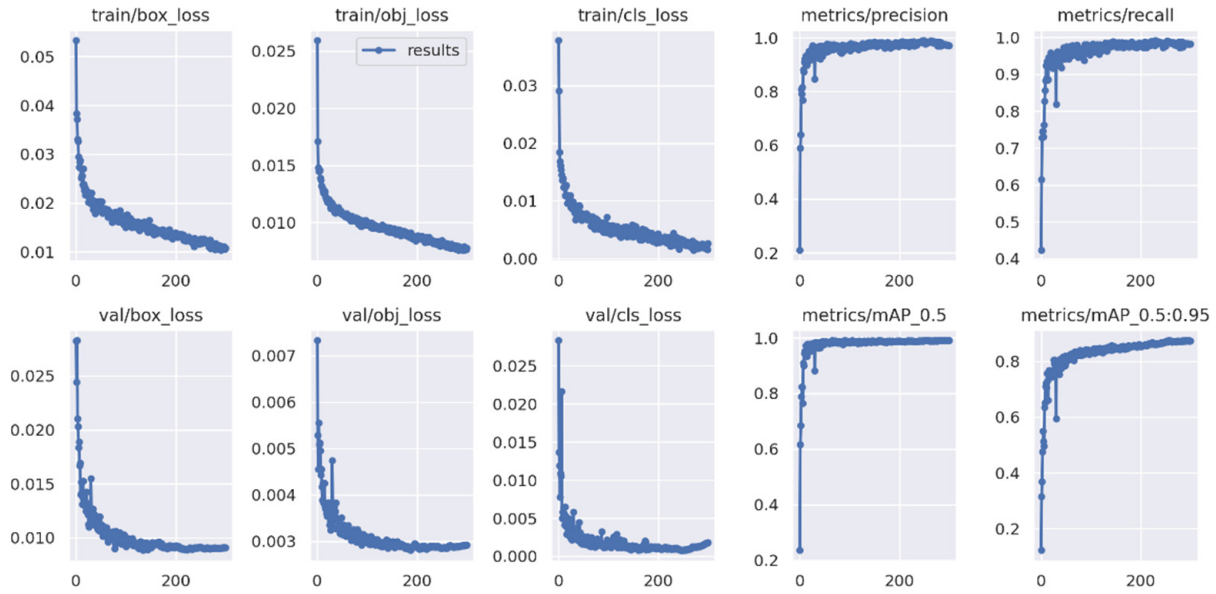

Figure 16. Integration results of YOLOv5s (300 epochs).

4. Discussion

In this paper, one of the main concerns is to concisely demonstrate the essential steps to build a model training and testing environment on Google Colab. The authors sincerely

hope that the guidance can help readers without the need for expensive computation supports. Based on the authors' experience, the following comments could be helpful. For instance, while getting access to Google Colab, the selected GPU in the virtual machine must be enabled. Your own Google Drive is connected, and the file paths are correct. Furthermore, the version of each adopted software program like the operating system and framework must be compatible with each other. In the stage of building the YOLO dataset in the cloud environment, the self-defined storage path needs to be identified. The labelling outputs must meet the YOLO format. In practice, the filename of images (.jpg) and the labels (.txt) must be consistent. Furthermore, the user needs to install application programs according to the content shown in requirements.txt. To avoid careless mistakes, it is suggested that a small number of epochs is first chosen. After the correctness of all settings has been confirmed, a bigger number of epochs can be then used in the process of model training.

In practice, during the validation process, a better model with related parameter settings can be obtained. Then the selected model will be used for performance testing. Using the three-category dataset division, train–validation–test, the problem of overfitting can be avoided. Furthermore, the generalized errors of the corresponding machine learning can be reduced. In real-time applications, YOLO can also be workable with embedded devices, such as Raspberry Pi, Nvidia Jetson TX2, and Arduino UNO [62–66]. The implementation of state-of-art lightweight algorithms, like the YOLOv5s addressed in this paper, is an emerging topic in deep learning. In this paper, a complete and easy to use open-source environment has been built. In the future, it can afford the needs of enhanced applications, such as the addition of color recognition into the recognition of clothing styles. Furthermore, the generative adversarial network (GAN) can be considered to improve the capability of small-object detection of YOLOs.

In summary, the essential concepts of the object detection algorithms are highlighted in the following. The key steps of the object detection mainly include the area identification and category classification. Two-stage object detection algorithms consider these steps as two unique execution processes, while the one-stage algorithms merge these two steps into one process. In general, a high recognition accuracy of two-stage detection algorithms can be obtained, but the whole process is generally time-consuming. In the one-stage algorithms, a regression-based classification is used such that the detection time can be reduced without losing detection accuracy much. Due to the high complexity of two-stage algorithms, the computing cost of the relevant model training increases as well. R-CNN, Fast R-CNN, and Faster R-CNN are typical two-stage object detection algorithms. In R-CNN, selective search algorithm is used to locate the candidate regions in images for the consequent feature extraction. Basically, this process is very time-consuming as a result of repeated feature extraction in overlapping areas. Since each candidate region will be processed by the convolutional neural network, the computation burden is increased along with the amount of increasing candidate regions. In Fast R-CNN, only one iteration of the CNN computation is required for the feature extraction of input images so that the computation cost is reduced. In Faster R-CNN, to further improve the computation efficiency, a RPN network is considered to replace the selective search method, thus the detection accuracy and speed is increased.

In the YOLO series, the input images are resized into a format of 448 $\times$ 448, and then sliced into several 7 $\times$ 7 grids. Following the CNN feature extracting, non-maximum suppression (NMS) is used to filter out less confident bounding boxes. In YOLOv1, input images are first sliced into grid cells. In each grid cell, only two bounding boxes are used for class prediction. Through the subsampling process, the image in the last layer is much smaller and makes object recognition become difficult. Alternatively, anchor boxes are used for the prediction of bounding boxes in YOLOv2. To improve the recognition capability for small objects, a passthrough layer is added in the second to last CNN layer. In YOLOv3, a Neck module is adopted for multi-scale feature fusion. Thus, the recognition efficiency of small objects is significantly improved. In YOLOv4, a cross stage partial network (CSPNet)

is used for feature extraction. In addition, the use of FPN and PAN will complete the feature fusion, so that the efficiency of object detection can be improved, especially in the detection of small objects. Finally, in YOLOv5, the addition function of FOCUS is mainly to increase the detection speed, and the SPP greatly improves the speed of generating candidate boxes and the computational cost reduction.

5. Conclusions

In this paper, a lightweight learning algorithm, YOLOv5s, is considered for the recognition of clothing styles. YOLOv5s is a one-stage objection method that the superiority of detection speed and model size is the main concern. An open-source integration development environment on Google Colab is built for model training, validation, and testing. The image samples are collected from either datasets of fashion clothes or Web searching of online clothing shops. The integrated process about how to build a free computing environment is concisely explained. The readers who are interested in deep learning may be easily able to build their own environments in various applications. Experimental results illustrate that the one-stage object detection algorithm YOLOc5s has the benefits in many metrics, such as mAP, precision, recall, F1-score, model size, and FPS.

Author Contributions: Conceptualization, Y.-H.C. and Y.-Y.Z.; methodology, Y.-H.C. and Y.-Y.Z.; software, Y.-Y.Z.; validation, Y.-H.C. and Y.-Y.Z.; formal analysis, Y.-H.C. and Y.-Y.Z.; investigation, Y.-H.C. and Y.-Y.Z.; data curation, Y.-Y.Z.; writing—original draft preparation, Y.-H.C. and Y.-Y.Z.; writing—review and editing, Y.-H.C. and Y.-Y.Z.; visualization, Y.-H.C. and Y.-Y.Z.; supervision, Y.-H.C. and Y.-Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Joshua, J.; Hendryli, J.; Herwindiati, D.E. Automatic License Plate Recognition for Parking System Using Convolutional Neural Networks. In Proceedings of the 2020 International Conference on Information Management and Technology (ICIMTech), Bandung, Indonesia, 13–14 August 2020; pp. 71–74.
2. Latha, R.S.; Sreekanth, G.R.; Rajadevi, R.; Nivetha, S.K.; Kumar, K.A.; Akash, V.; Bhuvanesh, S.; Anbarasu, P. Fruits and Vegetables Recognition Using YOLO. In Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25 January 2022; pp. 1–6.
3. Jia, D. Intelligent Clothing Matching Based on Feature Analysis. In Proceedings of the 2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, China, 15–16 January 2022; pp. 653–656.
4. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
5. Han, W.; Jiang, F.; Zhu, Z. Detection of Cherry Quality Using YOLOV5 Model Based on Flood Filling Algorithm. *Foods* **2022**, *11*, 1127. [CrossRef] [PubMed]
6. Fan, Y.; Zhang, S.; Feng, K.; Qian, K.; Wang, Y.; Qin, S. Strawberry Maturity Recognition Algorithm Combining Dark Channel Enhancement and YOLOv5. *Sensors* **2022**, *22*, 419. [CrossRef] [PubMed]
7. Mathew, M.P.; Mahesh, T.Y. Leaf-Based Disease Detection in Bell Pepper Plant Using YOLO V5. *Signal Image Video Process.* **2022**, *16*, 841–847. [CrossRef]
8. Safonova, A.; Hamad, Y.; Alekhina, A.; Kaplun, D. Detection of Norway Spruce Trees (Picea Abies) Infested by Bark Beetle in UAV Images Using YOLOs Architectures. *IEEE Access* **2022**, *10*, 10384–10392. [CrossRef]
9. Qi, J.; Liu, X.; Liu, K.; Xu, F.; Guo, H.; Tian, X.; Li, M.; Bao, Z.; Li, Y. An Improved YOLOv5 Model Based on Visual Attention Mechanism: Application to Recognition of Tomato Virus Disease. *Comput. Electron. Agric.* **2022**, *194*, 106780. [CrossRef]
10. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [CrossRef]
11. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision–ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37, ISBN 978-3-319-46447-3.

13. Ontor, M.Z.H.; Ali, M.M.; Hossain, S.S.; Nayer, M.; Ahmed, K.; Bui, F.M. YOLO_CC: Deep Learning Based Approach for Early Stage Detection of Cervical Cancer from Cervix Images Using YOLOv5s Model. In Proceedings of the 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 21 April 2022; pp. 1–5.
14. Shah, R.; Shastri, J.; Bohara, M.H.; Panchal, B.Y.; Goel, P. Detection of Different Types of Blood Cells: A Comparative Analysis. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23 April 2022; pp. 1–5.
15. Reddy, J.S.C.; Venkatesh, C.; Sinha, S.; Mazumdar, S. Real Time Automatic Polyp Detection in White Light Endoscopy Videos Using a Combination of YOLO and DeepSORT. In Proceedings of the 2022 1st International Conference on the Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 6 May 2022; pp. 104–106.
16. Sha, M.; Wang, H.; Lin, G.; Long, Y.; Zeng, Y.; Guo, S. Design of Multi-Sensor Vein Data Fusion Blood Sampling Robot Based on Deep Learning. In Proceedings of the 2022 2nd International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 18 March 2022; pp. 46–51.
17. Gupta, S.; Chakraborti, S.; Yogitha, R.; Mathivanan, G. Object Detection with Audio Comments Using YOLO V3. In Proceedings of the 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 9 May 2022; pp. 903–909.
18. Htet, S.M.; Aung, S.T.; Aye, B. Real-Time Myanmar Sign Language Recognition Using Deep Learning. In Proceedings of the 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russian, 16 May 2022; pp. 847–853.
19. Youssry, N.; Khattab, A. Accurate Real-Time Face Mask Detection Framework Using YOLOv5. In Proceedings of the 2022 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Cairo, Egypt, 6 June 2022; pp. 1–6.
20. Liu, C.-C.; Fuh, S.-C.; Lin, C.-J.; Huang, T.-H. A Novel Facial Mask Detection Using Fast-YOLO Algorithm. In Proceedings of the 2022 8th International Conference on Applied System Innovation (ICASI), Nantou, Taiwan, 22 April 2022; pp. 144–146.
21. Kolpe, R.; Ghogare, S.; Jawale, M.A.; William, P.; Pawar, A.B. Identification of Face Mask and Social Distancing Using YOLO Algorithm Based on Machine Learning Approach. In Proceedings of the 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 25 May 2022; pp. 1399–1403.
22. Sharma, R.; Sharma, A.; Jain, R.; Sharma, S.; Singh, S. Face Mask Detection Using Artificial Intelligence for Workplaces. In Proceedings of the 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 25 May 2022; pp. 1003–1008.
23. Priya, M.V.; Pankaj, D.S. 3DYOLO: Real-Time 3D Object Detection in 3D Point Clouds for Autonomous Driving. In Proceedings of the 2021 IEEE International India Geoscience and Remote Sensing Symposium (InGARSS), Ahmedabad, India, 6 December 2021; pp. 41–44.
24. Mostafa, M.; Ghantous, M. A YOLO Based Approach for Traffic Light Recognition for ADAS Systems. In Proceedings of the 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 8 May 2022; pp. 225–229.
25. Toheed, A.; Yousaf, M.H.; Rabnawaz; Javed, A. Physical Adversarial Attack Scheme on Object Detectors Using 3D Adversarial Object. In Proceedings of the 2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2), Rawalpindi, Pakistan, 24 May 2022; pp. 1–4.
26. Amrouche, A.; Bentrcia, Y.; Abed, A.; Hezil, N. Vehicle Detection and Tracking in Real-Time Using YOLOv4-Tiny. In Proceedings of the 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), Mostaganem, Algeria, 8 May 2022; pp. 1–5.
27. Miekkala, T.; Pyykonen, P.; Kutila, M.; Kyytinen, A. LiDAR System Benchmarking for VRU Detection in Heavy Goods Vehicle Blind Spots. In Proceedings of the 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 28 October 2021; pp. 299–303.
28. Athala, V.H.; Haris Rangkuti, A.; Luthfi, N.F.; Vikri Aditama, S.; Kerta, J.M. Improved Pattern Recognition of Various Traditional Clothes with Convolutional Neural Network. In Proceedings of the 2021 3rd International Symposium on Material and Electrical Engineering Conference (ISMEE), Bandung, Indonesia, 10 November 2021; pp. 15–20.
29. Rangkuti, A.H.; Hasbi Athala, V.; Luthfi, N.F.; Vikri Aditama, S.; Aslamiah, A.H. Content-Based Traditional Clothes Pattern Retrieval Using Convolutional Neural Network. In Proceedings of the 2021 3rd International Symposium on Material and Electrical Engineering Conference (ISMEE), Bandung, Indonesia, 10 November 2021; pp. 9–14.
30. Rizki, Y.; Medikawati Taufiq, R.; Mukhtar, H.; Apri Wenando, F.; Al Amien, J. Comparison between Faster R-CNN and CNN in Recognizing Weaving Patterns. In Proceedings of the 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 19 November 2020; pp. 81–86.
31. Shubathra, S.; Kalaivaani, P.; Santhoshkumar, S. Clothing Image Recognition Based on Multiple Features Using Deep Neural Networks. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 166–172.
32. Li, Y.; He, Z.; Wang, S.; Wang, Z.; Huang, W. Multideep Feature Fusion Algorithm for Clothing Style Recognition. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5577393. [CrossRef]
33. Yang, M.; Yu, K. Real-Time Clothing Recognition in Surveillance Videos. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2937–2940.

34. Bhatnagar, S.; Ghosal, D.; Kolekar, M.H. Classification of Fashion Article Images Using Convolutional Neural Networks. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; pp. 1–6.
35. Xiang, J.; Dong, T.; Pan, R.; Gao, W. Clothing Attribute Recognition Based on RCNN Framework Using L-Softmax Loss. *IEEE Access* **2020**, *8*, 48299–48313. [CrossRef]
36. Li, R.; Lu, W.; Liang, H.; Mao, Y.; Wang, X. Multiple Features with Extreme Learning Machines for Clothing Image Recognition. *IEEE Access* **2018**, *6*, 36283–36294. [CrossRef]
37. Yue, X.; Zhang, C.; Fujita, H.; Lv, Y. Clothing Fashion Style Recognition with Design Issue Graph. *Appl. Intell.* **2021**, *51*, 3548–3560. [CrossRef]
38. Tian, Q.; Chanda, S.; Kumar, K.C.A.; Gray, D. Improving Apparel Detection with Category Grouping and Multi-Grained Branches. *Multimed. Tools Appl.* **2021**, 1–18. [CrossRef]
39. Medina, A.; Méndez, J.; Ponce, P.; Peffer, T.; Meier, A.; Molina, A. Using Deep Learning in Real-Time for Clothing Classification with Connected Thermostats. *Energies* **2022**, *15*, 1811. [CrossRef]
40. Hidayati, S.C.; You, C.-W.; Cheng, W.-H.; Hua, K.-L. Learning and Recognition of Clothing Genres From Full-Body Images. *IEEE Trans. Cybern.* **2018**, *48*, 1647–1659. [CrossRef] [PubMed]
41. Dong, Q.; Gong, S.; Zhu, X. Imbalanced Deep Learning by Minority Class Incremental Rectification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1367–1381. [CrossRef] [PubMed]
42. Jain, S.; Kumar, J. Garment Categorization Using Data Mining Techniques. *Symmetry* **2020**, *12*, 984. [CrossRef]
43. Huang, F.-H.; Lu, H.-M.; Hsu, Y.-W. From Street Photos to Fashion Trends: Leveraging User-Provided Noisy Labels for Fashion Understanding. *IEEE Access* **2021**, *9*, 49189–49205. [CrossRef]
44. Donati, L.; Iotti, E.; Mordonini, G.; Prati, A. Fashion Product Classification through Deep Learning and Computer Vision. *Appl. Sci.* **2019**, *9*, 1385. [CrossRef]
45. Jo, J.; Lee, S.; Lee, C.; Lee, D.; Lim, H. Development of Fashion Product Retrieval and Recommendations Model Based on Deep Learning. *Electronics* **2020**, *9*, 508. [CrossRef]
46. Vijayaraj, A.; Vasanth Raj, P.T.; Jebakumar, R.; Gururama Senthilvel, P.; Kumar, N.; Suresh Kumar, R.; Dhanagopal, R. Deep Learning Image Classification for Fashion Design. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 7549397. [CrossRef]
47. Huang, C.-Q.; Chen, J.-K.; Pan, Y.; Lai, H.-J.; Lai, J.Y.; Huang, Q.-H. Clothing Landmark Detection Using Deep Networks With Prior of Key Point Associations. *IEEE Trans. Cybern.* **2019**, *49*, 3744–3754. [CrossRef]
48. Chun, Y.; Wang, C.; He, M. A Novel Clothing Attribute Representation Network-Based Self-Attention Mechanism. *IEEE Access* **2020**, *8*, 201762–201769. [CrossRef]
49. RCNN~YOLOv5. Available online: https://www.gushiciku.cn/dl/0aAQn/zh-tw (accessed on 21 May 2021).
50. Lin, C.-T.; Huang, S.-W.; Wu, Y.-Y.; Lai, S.-H. GAN-Based Day-to-Night Image Style Transfer for Nighttime Vehicle Detection. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 951–963. [CrossRef]
51. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
52. Zheng, J.; Sun, S.; Zhao, S. Fast Ship Detection Based on Lightweight YOLOv5 Network. *IET Image Process.* **2022**, *16*, 1585–1593. [CrossRef]
53. Huang, Y.-S.; Chou, P.-R.; Chen, H.-M.; Chang, Y.-C.; Chang, R.-F. One-Stage Pulmonary Nodule Detection Using 3-D DCNN with Feature Fusion and Attention Mechanism in CT Image. *Comput. Methods Programs Biomed.* **2022**, *220*, 106786. [CrossRef]
54. Yu, B.; Shin, J.; Kim, G.; Roh, S.; Sohn, K. Non-Anchor-Based Vehicle Detection for Traffic Surveillance Using Bounding Ellipses. *IEEE Access* **2021**, *9*, 123061–123074. [CrossRef]
55. Xie, F.; Lin, B.; Liu, Y. Research on the Coordinate Attention Mechanism Fuse in a YOLOv5 Deep Learning Detector for the SAR Ship Detection Task. *Sensors* **2022**, *22*, 3370. [CrossRef]
56. Vesth, T.; Lagesen, K.; Acar, Ö.; Ussery, D. CMG-Biotools, a Free Workbench for Basic Comparative Microbial Genomics. *PLoS ONE* **2013**, *8*, e60120. [CrossRef]
57. Singh, A.P.; Agarwal, D. Webcam Motion Detection in Real-Time Using Python. In Proceedings of the 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 10 March 2022; pp. 1–4.
58. Alon, H.D.; Ligayo, M.A.D.; Misola, M.A.; Sandoval, A.A.; Fontanilla, M.V. Eye-Zheimer: A Deep Transfer Learning Approach of Dementia Detection and Classification from NeuroImaging. In Proceedings of the 2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kuala Lumpur, Malaysia, 18 December 2020; pp. 1–4.
59. Kaufmane, E.; Sudars, K.; Namatēvs, I.; Kalniņa, I.; Judvaitis, J.; Balašs, R.; Strautiņa, S. QuinceSet: Dataset of Annotated Japanese Quince Images for Object Detection. *Data Brief* **2022**, *42*, 108332. [CrossRef]
60. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104.
61. Ge, Y.; Zhang, R.; Wang, X.; Tang, X.; Luo, P. DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5332–5340.

62. Liberatori, B.; Mami, C.A.; Santacatterina, G.; Zullich, M.; Pellegrino, F.A. YOLO-Based Face Mask Detection on Low-End Devices Using Pruning and Quantization. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 23 May 2022; pp. 900–905.
63. Sharma, H.; Das, S.; Mandal, P.; Acharya, A.; Kumar, P.; Dasgupta, M.; Basak, R.; Pal, S.B. Visual Perception Through Smart Mirror. In Proceedings of the 2022 Interdisciplinary Research in Technology and Management (IRTM), Kolkata, India, 24 February 2022; pp. 1–5.
64. Patil, H.D.; Ansari, N.F. Intrusion Detection and Repellent System for Wild Animals Using Artificial Intelligence of Things. In Proceedings of the 2022 International Conference on Computing, Communication and Power Technology (IC3P), Visakhapatnam, India, 7–8 January 2022; pp. 291–296.
65. Miao, Y.; Shi, E.; Lei, M.; Sun, C.; Shen, X.; Liu, Y. Vehicle Control System Based on Dynamic Traffic Gesture Recognition. In Proceedings of the 2022 5th International Conference on Circuits, Systems and Simulation (ICCSS), Nanjing, China, 13 May 2022; pp. 196–201.
66. Xu, X.; Zhang, X.; Zhang, T.; Shi, J.; Wei, S.; Li, J. On-Board Ship Detection in SAR Images Based on L-YOLO. In Proceedings of the 2022 IEEE Radar Conference (RadarConf22), New York, NY, USA, 21 March 2022; pp. 1–5.