applied sciences

MDPI

Article

Influence of Training Parameters on Real-Time Similar Object Detection Using YOLOv5s

Tautvydas Kvietkauskas [1] and Pavel Stefanovič [2,*]

[1] Department of Information Technology, Vilnius Gediminas Technical University, 10223 Vilnius, Lithuania
[2] Department of Information Systems, Vilnius Gediminas Technical University, 10223 Vilnius, Lithuania
* Correspondence: pavel.stefanovic@vilniustech.lt

Abstract: Object detection is one of the most popular areas today. The new models of object detection are created continuously and applied in various fields that help to modernize the old solutions in practice. In this manuscript, the focus has been on investigating the influence of training parameters on similar object detection: image resolution, batch size, iteration number, and color of images. The results of the model have been applied in real-time object detection using mobile devices. The new construction detail dataset has been collected and used in experimental investigation. The models have been evaluated by two measures: the accuracy of each prepared model has been measured; results of real-time object detection on testing data, where the recognition ratio has been calculated. The highest influence on the accuracy of the created models has the iteration number chosen in the training process and the resolution of the images. The higher the resolution of the images that have been selected, the lower the accuracy that has been obtained. The small iteration number leads to the model not being well trained and the accuracy of the models being very low. Slightly better results were obtained when the color images were used.

Keywords: YOLOv5s; real-time object detection; construction details dataset; similar objects

Citation: Kvietkauskas, T.; Stefanovič, P. Influence of Training Parameters on Real-Time Similar Object Detection Using YOLOv5s. *Appl. Sci.* **2023**, *13*, 3761. https://doi.org/10.3390/app13063761

Academic Editors: Junchi Yan and Minghao Guo

Received: 19 February 2023
Revised: 13 March 2023
Accepted: 14 March 2023
Published: 15 March 2023

1. Introduction

Artificial intelligence has become one of the main areas of interest in recent decades. A large amount of data is collected in various fields that can be used to train models to solve different problems, such as creating recommendation models; personalizing systems for our needs; developing the Internet of things; and securing our devices, etc. The field is broad, but image recognition has received special attention because of its wide applicability in practice. In medicine, image recognition is widely used for various diagnoses of diseases—for example, for different types of cancer [1–3]. In the military, drones are used to patrol and recognize objects, such as smugglers [4]. In autonomous vehicles, image recognition helps to analyze road activity and recognize traffic signs [5,6]. Real-time object detection also helps in agriculture, for example, by identifying and preventing various plant diseases [7–9] that help to save a harvest or implement the new technologies that today lead to the success of modern agriculture. Today, common computer vision tasks include image classification and localization, object detection, semantic segmentation, instance segmentation, and classification. In this paper, real-time object detection was analyzed.

Currently, one of the main problems of object detection is the collection and preparation of data and their use in training models, which largely determines the accuracy of the trained model. Despite the different fields of application, the problem of detecting similar objects is often encountered, which is attempted to be solved by applying various methods and algorithms of object detection [10]. Detecting similar objects involves problems of color, size, curvature, or other similar features. As a result, the accuracy of the models deteriorates, so it is important to fully examine the effectiveness of object detection algorithms and data preparation for it [11].

Object detection is one of the main topics in the field of artificial intelligence. Many researchers are studying and trying to obtain more accurate recognition results, for example, in self-service checkouts for item recognition [12] and in medicine for the recognition of certain diseases [13,14]. However, some researchers focus on objects that are large enough and have different appearance characteristics [15], whereas others focus on small objects [16]. During experiments, large and different objects are usually divided into five or more classes, while small objects are divided into 1–2 classes. The biggest challenge is when the objects are very similar in color, shape, size, and other appearance characteristics. For example, apples are similar in shape but different in variety, so it is important to pay attention to the color. However, the model has distinguished between 10 and more different classes of very similar and small objects.

Many object detection models have been developed and published publicly, but generally not many of them focus on similar object detection. The novelty of this article is that the main focus is on the real-time detection of similar objects using the mobile device. Such a method of object detection leads to different challenges. For this purpose, the new similar object dataset of the construction details has been prepared and analyzed. In total, the detection of 36 classes (construction details) has been analyzed. In this paper, similar data are considered objects that, in some cases, can look identical. For example, the construction details can look identical, depending on the angle of the camera in real-time object detection. An example of similar objects is presented in Figure 1. From the top view, these details look the same.

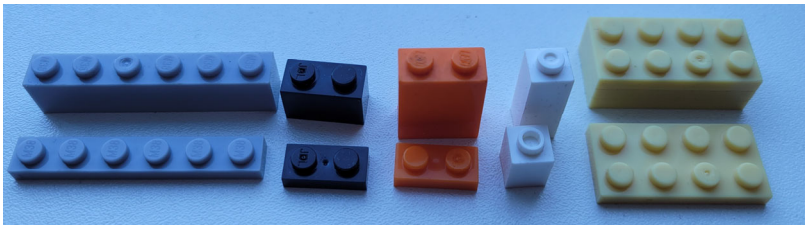

Figure 1. Example of similar objects.

It is obvious that object detection results depend on various factors, such as the selected object detection algorithm and its hyperparameters, as well as image properties that have been used to train algorithms. In this paper, the influence of training parameters has been analyzed for one of the new and still not fully investigated object detection algorithms, YOLOv5, which has also been analyzed. An analysis of related works [17,18] has shown that, usually, other similar research focuses only on a few Yolov5 hyperparameters, such as the learning rate, momentum, and weight selection, but other hyperparameters usually have not been changed. The main reason is that, in object detection tasks, the training of each model usually takes at least a few days. Therefore, in this research, we select the best learning hyperparameters that have been used in related works and focus on parameters related only to the training dataset: the image color, batch size, image resolution, and learning time. In our case, when the dataset has a large number of classes and the dataset items have similar features, it is important to investigate the dataset parameters and which parameters have the highest influence in such a case. The models have been trained and tested in real-time object detection using a mobile device. For this reason, the smaller version of YOLOv5 has been used, YOLOv5s, that is suitable for such types of tasks.

The structure of this paper is as follows. In Section 2, the related works are reviewed. In Section 3, the latest YOLO group algorithms have been overviewed. The new dataset and the experimental investigation are presented in Section 4. In Section 5, the discussion and limitations of the research are presented. Section 6 concludes the paper.

2. Related Works

An analysis of related works has shown that there are no public investigations where the construction details have been analyzed, but other research where small object detection

has been analyzed has been overviewed. Faster R-CNN, SSD, and YOLO object detection algorithms are the most well-known in the field of image recognition [19]. Researchers carry out studies and tests to compare algorithms. These models can be designed to recognize one specific object, for example, a person, or different objects with similar data features, such as small pills of different shapes. Correct identification is required to ensure that patients receive safe treatment with medications. Research by Tan et al. [20] used three object detection algorithms in a real-time pill recognition study: Faster R-CNN, SSD, and YOLOv3. In this study, the accuracy and speed of the pattern recognition were examined. The dataset was created from images that were prepared by the researchers. The images were taken with a 12-MP camera connected to a computer. The pills were randomly placed on both sides of the image board and 5131 images were taken. The dataset contains a total of 261 pills, of which 70 were capsule pills. All training parameters of the three algorithms were changed to 64 batches, 16 subdivisions, 0.001 learning rate, 0.9 momentum, and 0.0001 decay. Accuracy and speed results were acquired after training and testing. Based on these results, it was concluded that YOLOv3 is faster than SSD and Faster R-CNN. According to the mAP indicator, Faster R-CNN seems to be the best (82.89%); however, the detection speed is only 17 frames per second. The SSD-based model demonstrated an average performance of 32 frames per second and 82.71% mAP. Compared to recent models, the YOLOv3 achieves only 80.69% mAP, but is capable of greatly enhancing the detection rate and achieving real-time performance, with 51 frames per second. Therefore, it was concluded that the YOLO group model would be suitable for real-time pill recognition as it can identify pills quickly and with reasonable accuracy.

Alkentar et al. [21] describe and present an experimental investigation related to the performance of YOLOv3, SSD, and Faster R-CNN algorithms for drone recognition. Detecting drones is a big challenge because drones come in different shapes and sizes, move at different speeds, and always have a variety of backgrounds. Drones are not included in common datasets, so specific datasets had to be collected to evaluate the performance of deep learning detection and recognition algorithms. In the training phase of each model, 2664 images were used with different drone models and 12,000 iterations. During the testing phase, videos were used with a frame resolution of 1280×720 and a total of 6032 frames, of which 4752 frames were with drones of various models. The background of the images in the videos varied. This variety consisted of clear, cloudy skies, trees, and mountains. The drone recognition models were tested by changing the dimensions of the input video images to match the input dimensions of CNN. In this way, eight models were created. According to the results obtained, pattern recognition speed increases when networks with small input dimensions are used, but significantly higher mAP rates were obtained when the network input dimensions used are close to those of the image dimensions. Research results have shown that using networks with input dimensions larger than the input images is impractical and produces worse results. A study of drone recognition with the SSD, Faster R-CNN, and YOLOv3 algorithms concluded that Faster R-CNN showed the best results. However, this algorithm is slow in frame processing and is not practical for use in real-time object detection. Speed can be increased by using a smaller network input size, but with this way the accuracy would be lower. On the other hand, the YOLO group algorithm is the best option for real-time image processing in terms of processing speed. In contrast, the SSD gave a very high false positive rate compared to the other models (677 for SSD, 0 for all other models) and is therefore not suitable.

Many object recognition algorithms have been released recently, but there is not much material that specifically compares the latest algorithms, such as YOLOv5 focused on traffic objects. In the manuscript by Naftali et al. [22], the investigation of the SSD MobileNetv2, YOLOv3, YOLOv4, and YOLOv5 algorithms has been carried out for the detection of objects at the street level in real-time. The research dataset consisted of 3169 images containing 24,102 annotations. Five classes were distinguished that, respectively, included cars (16,446 annotations), traffic lights (4790), crossings (1756), trucks (761), and motorcycles (349). This dataset was divided into training (2010), validation (586), and testing (573) datasets.

Augmentation has been applied to the training dataset. The dataset was also pre-processed by rescaling each image to 416 × 260 pixels while maintaining the 16:10 aspect ratio. HSV scaling (−25 to 25), noise enhancement (up to 5% of pixels), and cut-out (3 cells of 10% each) have also been applied. During the training phase, for the YOLO group algorithms, 100 epochs were set with the stochastic gradient descent (SGD) optimizer. Meanwhile, 32,000 training steps were set for the SSD MobileNetv2 FPN-lite. After training and testing the resulting models, the comparison results were obtained. The results have shown that YOLOv5l is the most accurate algorithm. However, the mAP rates were not exceptionally significant compared to the other YOLO models. The research results have shown that YOLOv4 performed slightly worse than YOLOv5l and YOLOv3 in terms of F1 score, precision, and mAP. Additionally, YOLOv4 is the slowest of all models. Meanwhile, YOLOv5 outperforms other YOLO versions in terms of mAP@.5 and inference time. Compared to all object detection algorithms evaluated in this experiment, SSD MobileNetv2 FPN-lite performed the worst in mAP@.5, with only 0.315. However, it is the fastest algorithm in this experimental investigation, with an inference time of only 6.3 milliseconds. The second fastest object detection algorithm is YOLOv5s; it showed an inference time of 8.50 milliseconds, an F1 score of 0.579, and a mAP@.5 of 0.530, which is only 11% worse, and mAP@.5:95 is 17% worse than YOLOv5l, which is the most accurate model in this experiment. YOLOv5l is the most accurate algorithm compared to others, while SSD MobileNetv2 FPN-lite is one of the fastest. However, further analysis showed that YOLOv5s is an ideal algorithm for the detection of street-level objects in real-time self-driving cars as it provides relatively accurate results in a short time.

Related work has shown that object recognition accuracy is strongly influenced by the parameters set in the training phase. The values of the different hyperparameters of the algorithms also have a significant impact. Hyperparameter values that achieve good accuracy in detecting, for example, apples, will not achieve the same accuracy in drone recognition. To obtain high accuracy, a large dataset must be prepared. The more different pictures with similar objects, the better the model will learn and be able to recognize them more accurately. To improve the recognition accuracy, images must contain objects with different backgrounds and at different positions. In addition, most studies compare YOLOv3 with other models, and there are few non-scientific sources that compare the latest algorithms such as YOLOv5, YOLOv6, and YOLOv7. In the manuscript by Dlužnevskij et al. [23], experimental research has been performed to investigate the efficiency of YOLOv5 using a mobile device with real-time object detection tasks. The main focus of the research carried out has been on four YOLOv5 model sizes: small, medium, large, and extra-large. In the experimental investigation, the original COCO dataset was reduced to fit the requirements of mobile devices. The results of the experimental investigation have shown that the performance of the model is highly influenced by the hardware architecture and the system in which the model is used.

3. Review of YOLO Group Algorithms

As mentioned above, many different object detection algorithms exist and are still being developed. Analysis of related work has shown that older algorithms such as RCNN, Faster RCNN, and SDD are still used in various tasks, and some modifications of these algorithms have been proposed today [24,25]. In recent years, new algorithms for object detection have been developed, for example, end-to-end object detection using transformers [26], residual neural networks [27], etc. Related work has shown that in real-time object detection, the YOLO group algorithms allow for obtaining promising results, especially when YOLOv5 has been released. These days, the newest YOLO group algorithms are YOLOv6 and YOLOv7, but these two algorithms are still in development and have not been officially released, so there are different issues with using them. In the YOLOv6 algorithm [28], a hybrid channel technique has been used to create a more efficient decoupled head. In this case, the number of middle 3 × 3 convolutional layers has been limited to one. The width multipliers for the backbone and neck are used to scale the breadth of the head. These changes significantly minimize the processing costs to achieve a shorter

inference delay. Anchor-free detectors are distinguished by their superior generalization capabilities and ease of decoding prediction results; therefore, its post-processing time is significantly reduced. Extended efficient layer aggregation networks (ELANs) have been extended and renamed E-ELAN in the YOLOv7 algorithm [29]. The main advantage of ELAN was that it allowed a deeper network to learn and converge more successfully by managing the gradient path. The architecture of the computing block is significantly altered via E-ELANs, whereas the architecture of the transition layer remains unchanged. It uses expand, shuffle, and merge approaches to improve the learning capabilities of the network without altering the original gradient path. In this case, the strategy is to use group convolution to increase the channel and the number of computational blocks that apply the same group parameter and channel multiplier to all computational blocks in a computational layer. The feature map computed by each computational block is shuffled and concatenated. As a result, the number of channels in each feature map group will be equal to the number of channels in the original architecture. Finally, combine these feature map groups. E-ELANs have also improved the ability to learn more diverse features. In this manuscript, we use the stable version of the YOLO group that has not yet been fully investigated, YOLOv5.

YOLOv5 differs from previous versions of YOLO algorithms [30] as follows: it uses the PyTorch framework, rather than Darknet, and uses CSPDarknet53 as the backbone. The backbone solves the problem of repetitive gradient information in large backbones and integrates gradient shifting into a feature map that reduces inference speed, increases accuracy, and reduces model size by reducing parameters. The YOLOv5 architecture uses the Path Aggregation Network (PANet) as a neck to increase the flow of information. PANet uses a new feature pyramid network (FPN) that consists of several layers from bottom to top and from top to bottom. This improves the dispersion of low-level features in the model. PANet improves location in the lower layers, increasing the accuracy of the object's location. The head of YOLOv5 is the same as that of YOLOv3 and YOLOv4, which generates three different feature map outputs to achieve multiscale prediction. This helps to effectively increase the prediction of small and large objects in the model. The image is passed to CSPDarknet53 for feature extraction and then passed back to PANet for feature fusion. Finally, the output layer generates the results. The *Focus* layer evolved according to the YOLOv3 structure (Figure 2). This involves replacing the first three YOLOv3 layers and creating a YOLOv5 layer.

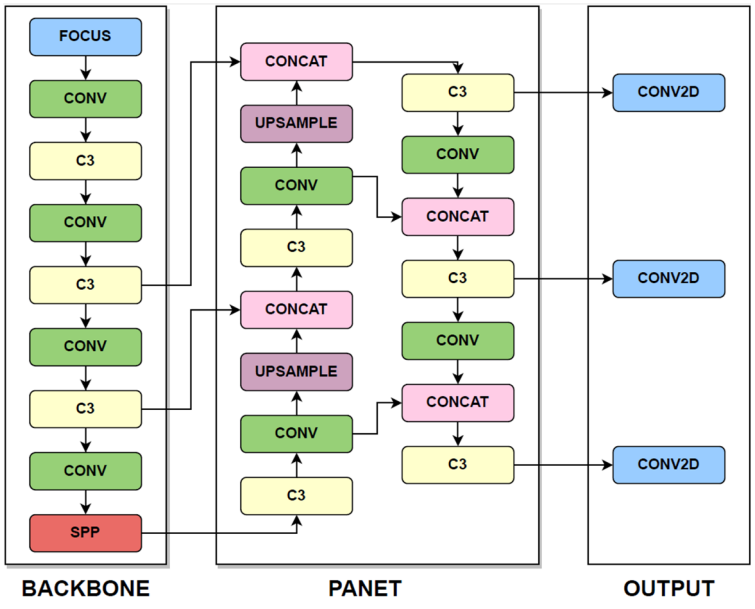

Figure 2. Architecture of YOLOv5.

Conv stands for convolutional layer, and C3 is made up of three convolutional layers and a module cascaded by various bottlenecks. Spatial pyramid pooling (SPP) is a pooling layer that is used to remove the limitation of a fixed network size. Upsample is used to increase the fusion of the previous layer at the nearest node. Concat is a clipping layer and is used to clip the previous layer. The last three Conv2d are detection modules used in the network head.

4. Experimental Investigation

4.1. Dataset

To perform an experimental investigation of similar objects in real-time object detection, a new construction detail dataset has been collected. In total, 355 color images of 36 different sizes, shapes, and color construction details were taken (Figure 3). Overall, the three datasets have been formed and used in the experimental investigation: the first dataset consisted of randomly selected 117 color images of 16 classes; the second dataset is the same 117 images, but, in addition, using the Python PIL framework, the shades were changed from RGB to greyscale; the third dataset is a whole collected dataset of 355 color images of 36 classes.

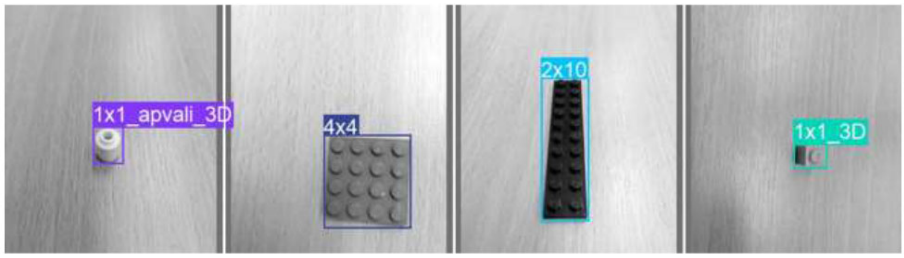

Figure 3. Sample of construction detail dataset.

Smaller datasets have been used to find the influence of image size, batch size, iteration number, and color on the accuracy of the object detection model. The reason is that, by using the dataset augmentation in the training process, the dataset becomes very large, and this costs a lot of recurses. The entire dataset has been used for a deeper analysis after the best parameters of the training process have been found in the primary research. The workflow of the experimental investigation performed is presented in Figure 4.

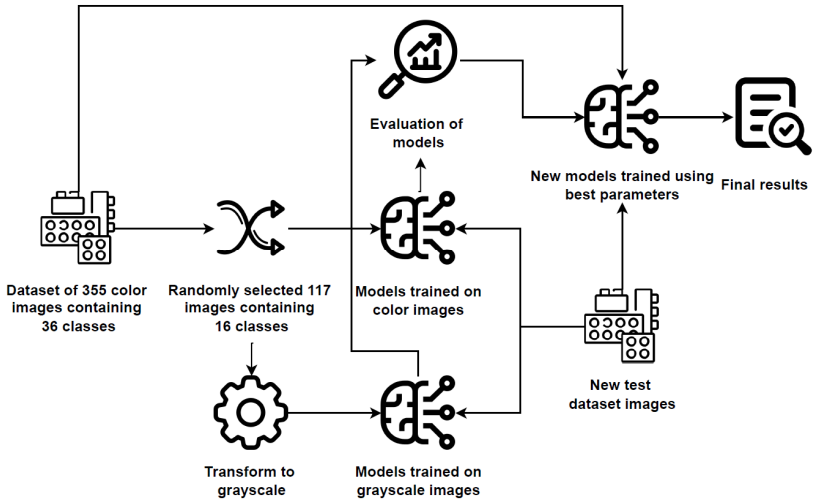

Figure 4. Scheme of experimental investigation performed.

To test the trained model in real-time object detection, random images of construction details were fed to the model. In the image, some details have been presented at the same

time (Figure 5). It is worth mentioning that the color of the test images has been selected by taking into account the color of the training dataset images.

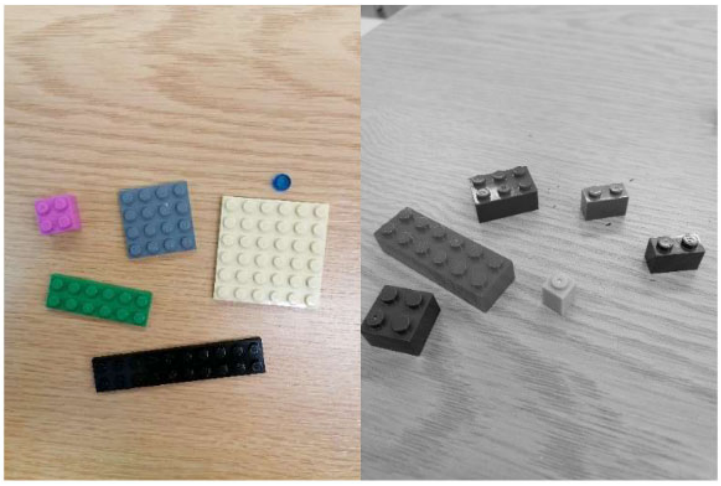

Figure 5. Example of a test dataset.

4.2. The Results of Experiments

Similarly, as in any phase of the training model, many parameters could influence the quality of the model. Therefore, finding the best parameters takes a lot of time and resources. YOLOv5s contains 28 hyperparameters that can be used in the training process. In this experimental investigation, the same hyperparameters as in the related work have been used, which have been obtained using the COCO dataset. In primary research, the influence of the batch size (16, 32, 64, 128), image size (320, 640, 1280), iteration (100, 300, 600), and color has been investigated. To improve the accuracy of the model, mosaic augmentation of the dataset has also been used. To train the YOLOv5s model, the *Google Colab Pro* (GPU: Tesla P100 16 GB, RAM: 24 GB, CPUs: 2 × vCPU) has been used.

The results of the primary research using 117 color images are presented in Figure 6 and Table 1. As we can see, the highest accuracy for all models is equal to 0.9950. The lowest accuracy has been obtained when the iteration number is equal to 100.

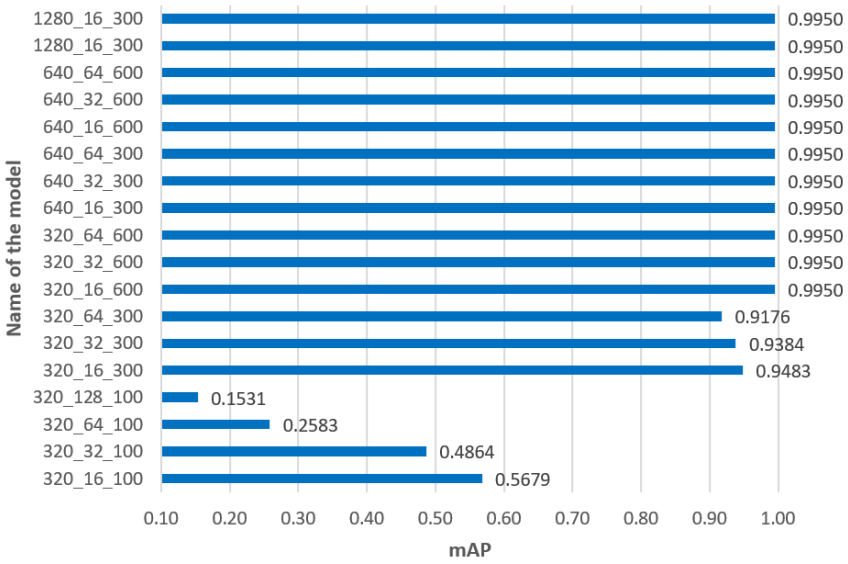

Figure 6. Accuracy results of the model trained on 117 color images of 16 classes.

Table 1. Testing results of model trained on 117 color images of 16 classes.

Image Resolution	Batch Size	Number of Iterations	Correctly Detected	Recognition Ratio
320	16	100	22	0.1946
	32		2	0.0177
	64		0	0
	128		0	0
	16	300	**69**	**0.6106**
	32		**75**	**0.6637**
	64		54	0.4778
	16	600	**69**	**0.6106**
	32		64	0.5663
	64		62	0.5486
640	16	300	**69**	**0.6106**
	32		62	0.5486
	64		62	0.5486
	16	600	63	0.5575
	32		41	0.3628
	64		60	0.5309
1280	16	300	59	0.5221
		600	54	0.4778

All models have been tested in real-time object detection and divided into models trained in color and greyscale images. In total, 21 images have been fed to the models, which contain a total of 113 construction details in the images. As we can see in Table 1, when the resolution of the training images is equal to 320, the best recognition ratio (0.6637) has been obtained using batch size 32, and the iteration number is equal to 300. Using the same image resolution, the recognition ratio (0.6106) is high when the batch size is equal to 16 and the iteration number is equal to 300, and also, respectively, when the batch size is equal to 16 and the iteration number is equal to 600. The worst results have been obtained when the iteration number is equal to 100 and with any batch size chosen. In this case, some models do not recognize either of the construction details. Therefore, in other models, training with higher image resolution and experimental investigation have not been performed using this iteration number.

The models trained using the images with resolutions equal to 640, the best recognition ratio (0.6106) has been obtained when the batch size is equal to 16 and iteration number is equal to 300. In other cases, the results of the recognition ratio are slightly lower (the highest recognition ratio is equal to 0.5221). When the image resolution is set to 1280, the results of the overall recognition ratio are far from the results with a smaller image resolution. It should be mentioned that in some cases, when the batch size has been selected to 64 or 128 and the image resolution is high, experiments have not been performed because of the limitation of the environment.

The results of primary research using 117 greyscale images are presented in Figure 7 and Table 2. As for models trained on color images, the lowest accuracy has been obtained when the iteration number is equal to 100. Higher accuracy (0.9950) was obtained when the number of iterations increased.

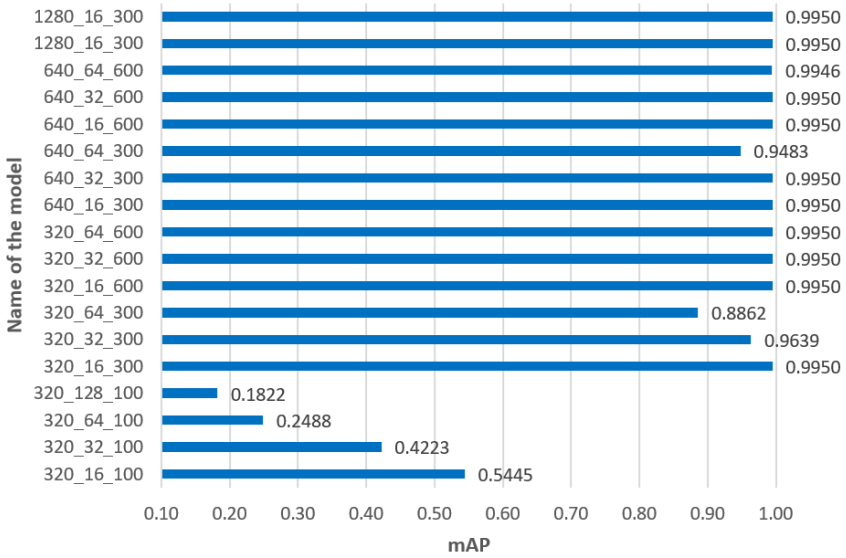

Figure 7. Accuracy results of model trained on 117 greyscale images of 16 classes.

Table 2. Testing results of model trained on 117 greyscale images of 16 classes.

Image Resolution	Batch Size	Number of Iterations	Correctly Detected	Recognition Ratio
320	16	100	6	0.0531
	32		0	0
	64		0	0
	128		0	0
	16	300	55	0.4867
	32		38	0.3362
	64		40	0.3539
	16	600	47	0.4159
	32		49	0.4336
	64		50	0.4424
640	16	300	48	0.4247
	32		51	0.4513
	64		**65**	**0.5752**
	16	600	52	0.4601
	32		45	0.3982
	64		41	0.3628
1280	16	300	41	0.3628
		600	46	0.4070

In Table 2, results of models trained on greyscale images has been presented. In Table 2, results of models trained on greyscale images has been presented. The same as when using the color images, the worst recognition ratio results have been obtained when the iteration number is equal to 100. Almost all models do not recognize any construction details. The best results of the recognition ratio (0.4867) are achieved when the image resolution is equal to 320, the batch size is equal to 16, and the iteration number is equal to 300. In this case, the results are significantly lower compared to models trained on color images. The results using the highest image resolution do not differ from the results of the models trained on the color images. The highest recognition ratio (0.4070) has been obtained when the

batch size is equal to 16 and the iteration number is equal to 600. Of all trained models that used greyscale images, the highest recognition ratio (0.5752) was obtained when the image resolution was equal to 640, the batch size was equal to 64, and the iteration number is equal to 300.

To conclude the primary research results, the parameters of four models (with highest accuracy) that obtain the highest recognition ratio have been selected for a deeper analysis (cells highlighted in orange and green in Tables 1 and 2). The four new models have been trained using the full dataset, 355 color images of 36 classes. The accuracy results of the model are presented in Figure 8. As we can see, the accuracy becomes slightly higher compared to the results of the primary research, even though the number of classes increased. The highest accuracy (0.9952) has been obtained when the image resolution is equal to 600, the batch size is equal to 16, and the iteration number is equal to 600.

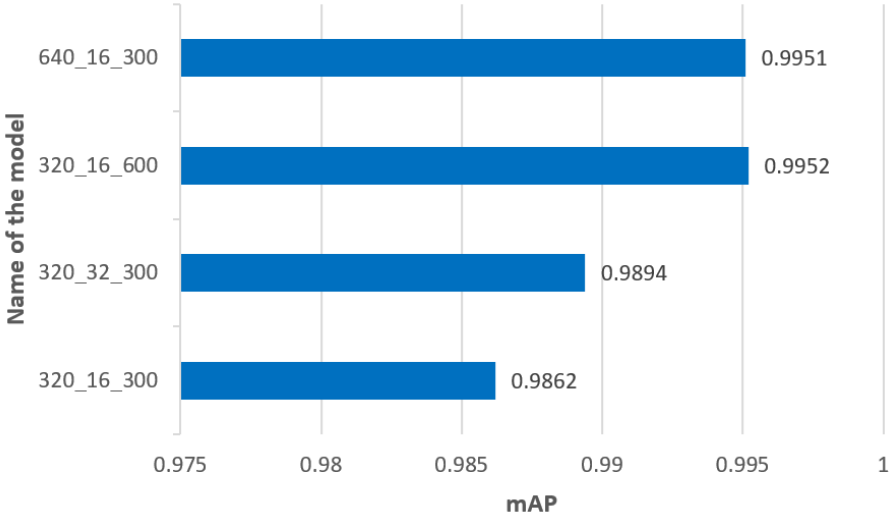

Figure 8. Accuracy results of the model trained on 355 color images of 36 classes.

In Table 3, the influence of the training parameters is presented using the entire dataset. All models have been tested using 39 new images, where a total of 309 construction details have been taken. As we can see (Figure 8), the accuracy of model 320_16_300 is equal to 0.9894, but the recognition ratio (0.5242) is the highest of all models.

Table 3. Testing results of model trained on 355 color images of 36 classes.

Image Resolution	Batch Size	Number of Iterations	Correctly Detected	Recognition Ratio
320	16	300	162	0.5242
320	32	300	137	0.4433
320	16	600	142	0.4595
640	16	300	155	0.5016

In Figure 9, the box plot of each model's result is presented. The box plot shows the results of the recognition ratio distribution for each class. As we can see, the highest recognition ratio was obtained by Model 320_16_300, where most of the recognition ratio is higher than the median. A similar distribution is in model 320_16_600, but in this case, five construction details have not been recognized (0% accuracy). The recognition ratio of the other models is lower. The majority of the recognition ratio value in model 600_16_300 is lower than the median, and many construction details have not been recognized at all.

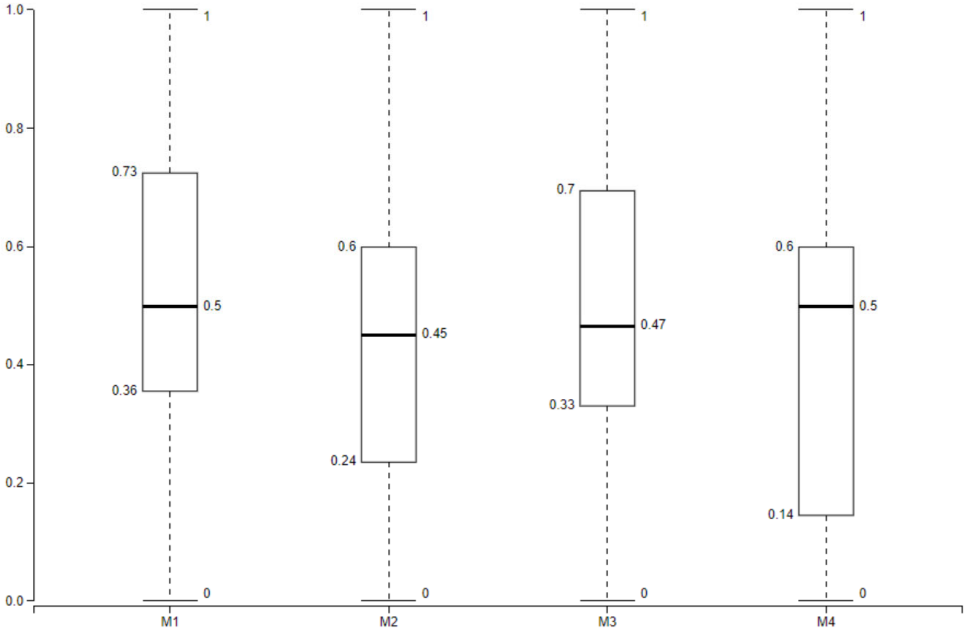

Figure 9. The box plot of the results of the test dataset for each model.

The detail recognition ratios of each class are presented in Table 4. The results of the four best models were compared. As we can see, only two details were recognized correctly by all models and two construction details have not been recognized by any of the trained models. The models 320_16_300 and 320_16_600 recognize, respectively, four and five construction details with 100% accuracy. Model 640_16_300 has shown the worst performance, not recognizing a total of eight out of thirty-six construction details.

Table 4. Testing results for each class for all trained models.

Label of The Construction Detail	Recognition Ratio of Each Class			
	320_16_300	**320_32_300**	**320_16_600**	**640_16_300**
6 × 6	1	0.5	1	1
2 × 6	0.75	0.88	0.75	1
2 × 3	0.33	0.33	0.5	0.5
1 × 4	0.69	0.62	0.69	0.54
2 × 2	0.8	0.2	0.4	0.6
8 × 16	1	1	1	1
4 × 6	1	1	1	0
1 × 2	0.39	0.28	0.39	0.33
1 × 2_3D	0.58	0.58	0.5	0.42
2 × 3_3D	0.7	0.6	0.7	0.6
2 × 6_3D	0.3	0.5	0.6	0.6
2 × 2_3D	0.57	0.57	0.43	0.29
1 × 4_3D	0.46	0.54	0.38	0.54
1 × 4_trapeze	0.82	0.73	0.82	0.73
grid	0.92	0.68	0.56	0.92
1 × 1_square	0.5	0.5	0.33	0.83

Table 4. *Cont.*

Label of The Construction Detail	Recognition Ratio of Each Class			
	320_16_300	320_32_300	320_16_600	640_16_300
2 × 4	0.44	0.31	0.38	0.56
2 × 16	1	0	0.5	0
1 × 1_3D_square	0.5	0.25	0.25	0.5
1 × 12	0	1	1	0
2 × 10	0.5	0	0.5	0
4 × 12	0	0	0	0
3 × 3	0.6	0.4	0.2	0
4 × 4	0.5	0.5	0.5	0.5
2 × 4_3D	0.38	0.38	0.38	0.5
1 × 3	0.31	0.08	0.31	0.46
1 × 1_round_flat	0.6	0.6	0.6	0.6
1 × 1_cilindre	0.33	0.3	0.37	0.37
1 × 6_3D	0	0	0	0
1 × 6	0.5	0	1	0
1 × 1_trapeze	0.44	0.33	0.22	0.22
1 × 1_round	1	1	1	1
1 × 3_3D	0.22	0.22	0.11	0.11
1 × 2_rectangle_flat	0.5	0.39	0.39	0.44
1 × 2_trapeze	0.27	0.18	0.18	0.18
1 × 2_rectangle_knob	0.58	0.5	0.33	0.67

5. Discussion

Real-time similar object detection has many different issues that usually depend on various factors, such as object detection algorithm, selection of hyperparameters, the specific dataset used in the training process, etc. It is obvious that it is not possible to take into account all factors, but the results of this research show where attention needs to be maintained and where to focus on in future research. The main problem in object detection is that training the models takes time, so performing a lot of different angle experiments is confusing. In such cases, the experiments are usually split into parts and investigated into smaller scopes. It is obvious that not all cases have been taken into account in these experiments, but the obtained results show where we need to pay attention to similar object detection in real-time. It is possible to single out the main problems encountered in all models prepared in this paper. The main problem noticed in this research is that the focus has been on a dataset of similar objects, which leads to a higher incorrect recognition ratio. Some construction details have similar features, for example, the same construction detail but different color; the rounded construction detail and not rounded; the same construction detail but different height, etc. Much depends on the angle when the object detection in real-time is performed. Some problems of real-time object detection are presented in Figure 10. In this experimental investigation, the analysis of other YOLOv5 models such as m, s, l, and x has not been performed, nor has it been investigated how the models affect the recognition results. The main focus has been on the YOLOv5s model, which usually shows the best performance for real-time object detection using the mobile environment.

Figure 10. Example of a test dataset in real-time object detection.

6. Conclusions

In this paper, the influence of training parameters on real-time object detection has been investigated. The newly collated construction details dataset of similar objects has been collected and analyzed. The main problem of such an analysis is that the construction details are similar in their features (similar shape, color, size, etc.), so it is important to choose the right training parameters that allow the higher accuracy of the model to be obtained in the training and testing phases. The hyperparameters of the YOLOv5s algorithms have not been taken into account. However, the best hyperparameters are used based on the analysis of related works.

The experimental results have shown that the highest influence on the accuracy of the model has been the selected iteration number. The small iteration number is not suitable for this kind of object detection because the models are not trained well at all. The color images allow for slightly better results compared to the greyscale images. The higher resolution of the images does not increase the accuracy of similar object detection, and for such a kind of analysis, a huge environment and lots of resources need to be implemented. The optimal resolution of the images should be 320 or 640 and the batch size equal to 16 or 32. In future work, an experimental investigation should be improved by trying different data-augmentation methods and focusing on YOLOv5s hyperparameters.

Author Contributions: Conceptualization, T.K. and P.S.; methodology, P.S.; validation, T.K.; formal analysis, T.K. and P.S.; data curation, T.K. and P.S.; writing—original draft preparation, T.K. and P.S.; writing—review and editing P.S.; visualization, P.S.; supervision, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Arunkumar, N.; Mohammed, M.A.; Abd Ghani, M.K.; Ibrahim, D.A.; Abdulhay, E.; Ramirez-Gonzalez, G.; de Albuquerque, V.H.C. K-means clustering and neural network for object detecting and identifying abnormality of brain tumor. *Soft Comput.* **2019**, *23*, 9083–9096. [CrossRef]
2. Welikala, R.A.; Remagnino, P.; Lim, J.H.; Chan, C.S.; Rajendran, S.; Kallarakkal, T.G.; Barman, S.A. Automated detection and classification of oral lesions using deep learning for early detection of oral cancer. *IEEE Access* **2020**, *8*, 132677–132693. [CrossRef]
3. Bozaba, E.; Solmaz, G.; Yazıcı, Ç.; Özsoy, G.; Tokat, F.; Iheme, L.O.; Çayır, S.; Ayaltı, S.; Kayhan, C.K.; İnce, Ü. Nuclei Detection on Breast Cancer Histopathology Images Using RetinaNet. In Proceedings of the 2021 29th Signal Processing and Communications Applications Conference (SIU), Istanbul, Turkey, 9–11 June 2021; pp. 1–4.
4. Ajakwe, S.O.; Ihekoronye, V.U.; Akter, R.; Kim, D.S.; Lee, J.M. Adaptive drone identification and neutralization scheme for real-time military tactical operations. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022; pp. 380–384.
5. Hnewa, M.; Radha, H. Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques. *IEEE Signal Process. Mag.* **2020**, *38*, 53–67. [CrossRef]
6. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
7. Roy, A.M.; Bose, R.; Bhaduri, J. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Comput. Appl.* **2022**, *34*, 3895–3921. [CrossRef]
8. Tseng, G.; Sinkovics, K.; Watsham, T.; Rolnick, D.; Walters, T.C. Semi-Supervised Object Detection for Agriculture. In Proceedings of the 2nd AAAI Workshop on AI for Agriculture and Food Systems, Washington, DC, USA, 13–14 February 2023.
9. Valdez, P. Apple defect detection using deep learning based object detection for better post harvest handling. *arXiv* **2020**, arXiv:2005.06089.
10. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [CrossRef]
11. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
12. Kim, H.; Kim, D.; Ryu, G.; Hong, H. A Study on Algorithm Selection and Comparison for Improving the Performance of an Artificial Intelligence Product Recognition Automatic Payment System. *Int. J. Adv. Cult. Technol.* **2022**, *10*, 230–235.
13. Yahalomi, E.; Chernofsky, M.; Werman, M. Detection of distal radius fractures trained by a small set of X-ray images and Faster R-CNN. In *Intelligent Computing: Proceedings of the 2019 Computing Conference*; Springer International Publishing: Cham, Switzerland, 2019; Volume 1, pp. 971–981.
14. Zhu, G.; Piao, Z.; Kim, S.C. Tooth detection and segmentation with mask R-CNN. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19–21 February 2020; pp. 070–072.
15. Uzkent, B.; Yeh, C.; Ermon, S. Efficient object detection in large images using deep reinforcement learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 1824–1833.
16. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [CrossRef]
17. Isa, I.S.; Rosli, M.S.A.; Yusof, U.K.; Maruzuki, M.I.F.; Sulaiman, S.N. Optimizing the Hyperparameter Tuning of YOLOv5 for Underwater Detection. *IEEE Access* **2022**, *10*, 52818–52831. [CrossRef]
18. Mantau, A.J.; Widayat, I.W.; Adhitya, Y.; Prakosa, S.W.; Leu, J.S.; Köppen, M. A GA-Based Learning Strategy Applied to YOLOv5 for Human Object Detection in UAV Surveillance System. In Proceedings of the 2022 IEEE 17th International Conference on Control & Automation (ICCA), Naples, Italy, 27–30 June 2022; pp. 9–14.
19. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv:1905.05055. [CrossRef]
20. Tan, L.; Huangfu, T.; Wu, L.; Chen, W. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 324. [CrossRef] [PubMed]
21. Alkentar, S.M.; Alsahwa, B.; Assalem, A.; Karakolla, D. Practical comparation of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection. *J. Eng.* **2021**, *27*, 19–31. [CrossRef]
22. Naftali, M.G.; Sulistyawan, J.S.; Julian, K. Comparison of Object Detection Algorithms for Street-level Objects. *arXiv* **2022**, arXiv:2208.11315.
23. Dlužnevskij, D.; Stefanovic, P.; Ramanauskaite, S. Investigation of YOLOv5 efficiency in iPhone supported systems. *Balt. J. Mod. Comput.* **2021**, *9*, 333–344. [CrossRef]
24. Pramanik, A.; Pal, S.K.; Maiti, J.; Mitra, P. Granulated RCNN and multi-class deep sort for multi-object detection and tracking. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**, *6*, 171–181. [CrossRef]
25. Han, G.; Huang, S.; Ma, J.; He, Y.; Chang, S.F. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 780–789. [CrossRef]
26. Zhen, P.; Gao, Z.; Hou, T.; Cheng, Y.; Chen, H.B. Deeply tensor compressed transformers for end-to-end object detection. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 4716–4724. [CrossRef]

27. Han, Y.; Wang, L.; Cheng, S.; Li, Y.; Du, A. Residual dense collaborative network for salient object detection. *IET Image Process.* **2023**, *17*, 492–504. [CrossRef]
28. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Wei, X. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
29. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
30. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef] [PubMed]