

Article

Promotion of Color Sorting in Industrial Systems Using a Deep Learning Algorithm

Ivana Medojevic ^{1,*}, Emil Veg ², Aleksandra Joksimovic ² and Jelena Ilic ³

¹ Department of Engineering of Biotechnical Systems, Faculty of Mechanical Engineering, University of Belgrade, 11120 Belgrade, Serbia

² Department of Theory of Mechanisms and Machines, Faculty of Mechanical Engineering, University of Belgrade, 11120 Belgrade, Serbia

³ Department of Physics and Electrical Engineering, Faculty of Mechanical Engineering, University of Belgrade, 11120 Belgrade, Serbia

* Correspondence: imedojevic@mas.bg.ac.rs

Abstract: Color sorting is a technological operation performed with the aim of classifying compliant and noncompliant agricultural products in large-capacity industrial systems for agricultural product processing. This paper investigates the application of the YOLOv3 algorithm on raspberry images as a method developed for the detection, localization, and classification of objects based on convolutional neural networks (CNNs). To our knowledge, this is the first time a YOLO algorithm or CNN has been used with original images from the color sorter to focus on agricultural products. Results of the F1 measure were in the 92–97% range. Images in full resolution, 1024 × 1024, produced an average detection time of 0.37 s. The impact of the hyperparameters that define the YOLOv3 model as well as the impact of the application of the chosen augmentative methods on the model are evaluated. The successful classification of stalks, which is particularly challenging due to their shape, small dimensions, and variations, was achieved. The presented model demonstrates the ability to classify noncompliant products into four classes, some of which are appropriate for reprocessing. The software, including a graphic interface that enables the real-time testing of machine learning algorithm, is developed and presented.

Keywords: machine learning; deep learning; YOLO; color sorting; inspection; agriculture



Citation: Medojevic, I.; Veg, E.; Joksimovic, A.; Ilic, J. Promotion of Color Sorting in Industrial Systems Using a Deep Learning Algorithm. *Appl. Sci.* **2022**, *12*, 12817. <https://doi.org/10.3390/app122412817>

Academic Editor: Wan-Soo Kim

Received: 28 October 2022

Accepted: 11 December 2022

Published: 13 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since digitalization of agriculture contributes to increased efficiency in primary production processes [1–3], current technologies in industrial processing systems demand innovation. Serbia is one of the three greatest raspberry producers in the world; therefore, any loss during processing is a global-level loss. Raspberry fruit can be frozen, sorted, and processed as whole, whole and broken, crumbled, or used as raw material for puree production. As for the cold processing of raspberry, the requirements for machine inspection concerning extra class are as follows: the fruits must be whole, clean, fresh, and seemingly healthy, without any defects and foreign materials. In addition, they must be uniform in maturity level and firm enough to withstand handling and transport. Certain deviations, up to 5%, are permissible. For premium-class raspberries, some deviations are tolerated regarding fruit development and size (up to 10%); however, fruits must not be over-ripe.

Industrial systems for machine inspection of agricultural products are mostly standardized. Each machine has its control system and algorithm to define the criteria and perform the selection of products. Output accuracy of the color sorter ranges commonly from 91 to 98%. One operator services the machine and generates recipes for each product being processed separately. When generating a recipe, a certain amount of product is allowed to pass through the machine, screened with available optical devices. In general, the external quality of fruits and vegetables is assessed using the physical characteristics

of color, texture, size, shape, and visual defects [4–6]. The operator obtains the output parameters that refer, for the most part, to the color and shape. On the basis of differences in parameter values for compliant and noncompliant products, critical criteria and their values are selected and adjusted, i.e., a recipe is generated. Noncompliant products are products whose physical characteristics do not conform to the Class Specification Regulations, or that contain foreign materials found during sorting, such as soil, wood, plastics, glass, metal, leaves, animal and insect parts, etc.

The promotion of machine vision was contributed to mostly by deep learning algorithms. During the investigation of the latest research in the machine vision field, convolutional neural networks seemed to be the best approach for resolving some of the aforementioned issues during color sorting. On the other hand, with many training images, algorithms can learn quickly and, after some time, the efficiency and accuracy of overall human capabilities may be calculated.

Although there are no scientific papers directly related to the application of machine and deep learning on machines for color sorting and inspection in industrial production, a considerable number of research studies indicate the potential for inspection and assessment of fruits [7–9], vegetables, the quality of cereals [10,11], legumes, detection of bruises and diseases [12,13], and in primary agricultural production [14–17]. Initial investigations of the YOLO (you only look once) [18] algorithm implementation, as a widely used algorithm that can operate in real time, were related to autonomous vehicles and detection of objects and pedestrians [19,20] or insect classification [21]. Very soon, these algorithms were applied in other areas [22–25], including in fruit and vegetable recognition [26,27]. The closest comparison to our research is found in [28]. Here, the authors designed databases of raspberry images which are ready for industrial inspection. The focus of this paper is evaluating the largest amount of fruit possible and not a single sample. Therefore, the difference from our dataset is that our data come directly from the machine referred to as the color sorter, where raspberries or other fruits are mostly evenly distributed on the belt. For the future, it could be interesting to train those datasets with our proposed flow and web application.

YOLO is most commonly used for application in real time and combines object localization and classification into a regression problem. Test results on the largest datasets and comparisons with previous algorithms such as RCNN are presented in the work by Redmon et al. [18], as well as on their official website, providing detailed analyses (<https://pjreddie.com/darknet/yolo/> (accessed on 10 June 2021)).

YOLOv3 is also good enough for small object detection with Darknet-53, which possesses several improvements from Darknet-19 [29]. Regarding diseases or bruises in agricultural products, some authors have improved YOLOv3 by modifying or adding a particular network layer [13,30]. However, there have not been a larger number of investigations of agricultural product detection of any size in industrial processing plants.

Liu et al. [14] presented the idea of an algorithm for tomato detection and segmentation in a greenhouse. In addition, a comparison was made with several algorithms, including YOLO. Both algorithms, in addition to the proposed YOLO, produced excellent results under the same conditions.

Fruit detection in the orchard in real time represents one of the essential methods in growth stage and yield assessment. Size, color, fruit density on the crown, and other characteristics of fruits and vegetables change during growth and ripening. The YOLO algorithm can be adapted to various growth stages of agricultural products, for example, in apple detection in an orchard [16]. Traditional methods mainly detect a specific condition using a single model. A modified/improved YOLO model is proposed for apple detection in the orchard, at different growth stages, with fluctuating illumination, complex backgrounds, apples located on top of one another, and apples partially covered by branches or leaves.

The aim of this paper is to find a new approach to optimize the number of the criteria and complexity of the detection algorithm while, at the same time, creating the definition of a good product. As mentioned, deep learning algorithms produce excellent results in the

recognition of different types of objects, and the challenge in industrial processing of agricultural products is that automatic inspection must be performed in real time. This was the main reason for choosing the YOLOv3 algorithm over its previous versions and other deep learning algorithms. In addition to YOLOv3, both YOLOv4 [31] (April 2020) and YOLOv5 (May 2020) were subsequently developed. Although some investigations have shown that the latest versions achieve somewhat better results compared to YOLOv3 [32–34], the authors agreed to use it for initial investigation because the problem dynamics are not complex. In the case of YOLOv3, the input data include the entire image, and the algorithm itself detects a set of common characteristics obtained as training data and classifies them; raspberry variations, in the form of color and shape, are not complex, although a stalk represents a special challenge. The authors concluded that implementation of deep learning algorithms in industrial systems for various products' inspection is the next logical step, and research results indicated that there is a theoretical possibility of their application.

The remainder of the paper is organized as follows: Section 2 presents the detailed methodology, followed by a visual representation of the flow of data. Section 3 first presents the results, then a detailed discussion, and the developed web application is described and proposed. Finally, Section 4 provides a final discussion and conclusions about results.

2. Materials and Methods

Input data used as a basis for investigation are digital images of raspberry obtained by screening with an Optyx 3000 color sorter during industrial processing. The system for color sorting and inspection is part of an automatic line for inspection and packaging, of 3 t/h capacity, in the ITN Eko Povlen plant, Serbia. Color sorter and the schematic process of color sorting are represented in Figure 1.

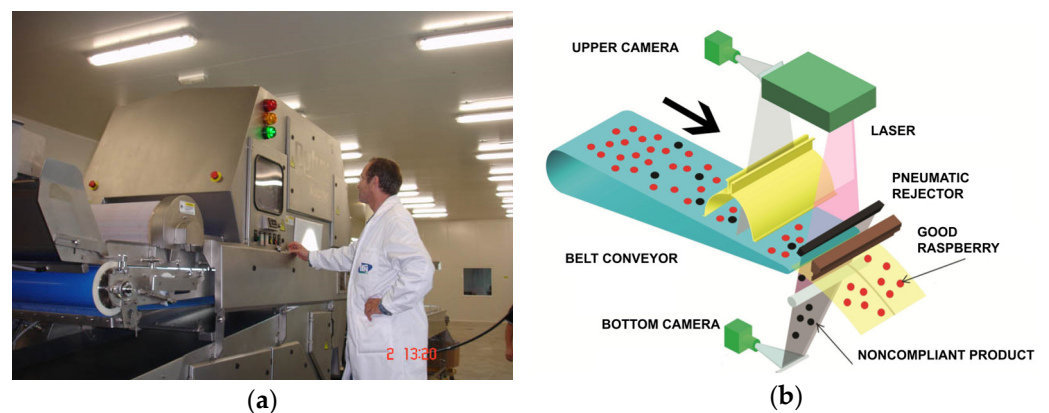


Figure 1. (a) Optyx 3000 color sorter in plant; (b) schematic representation of the process of color sorting.

The system is screening images of the resolution 1024×1024 pixels in RGB format, and the screening speed is 4000 Hz. The machine recognizes compliant bioproducts and detects noncompliant bioproducts based on prior generated recipe. The recipe contains criteria that define the compliant product, while the color sorter, by means of a pneumatic ejector, eliminates products that do not meet the set criteria. Figure 2 summarizes processes after dataset collected.

Raspberry images of the resolution 224×224 pixels, saved in .jpg format, were used for input data. Input training images need not be in a specific resolution, it is only required that the image size is divisible by 32 [18]. Color components remained in RGB format. As reported in [35], many works related to fruit classification using detection algorithms selected an RGB camera as primary sensory detector.

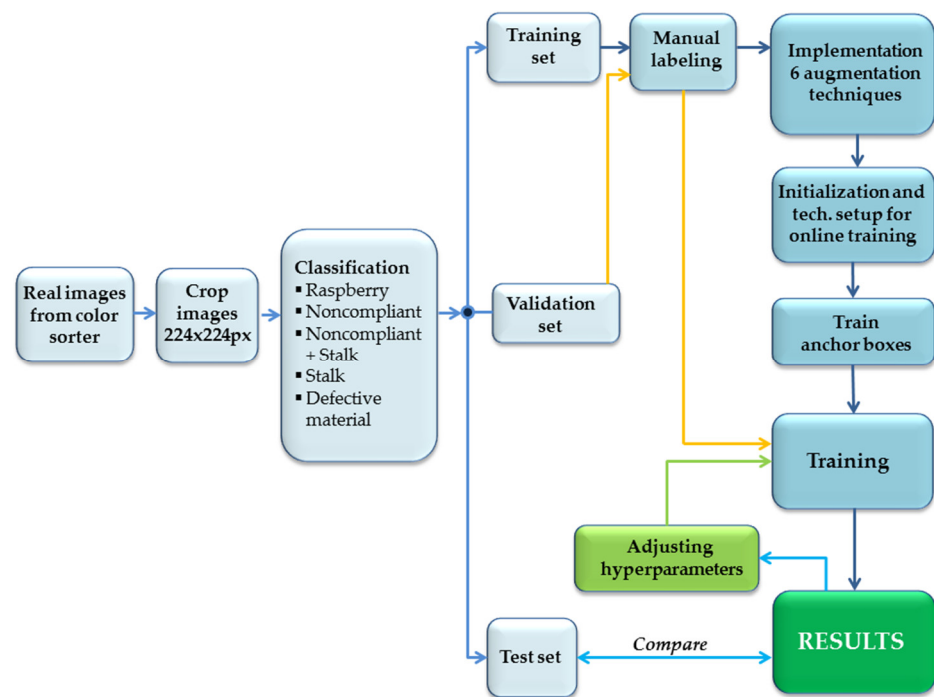


Figure 2. Overview of the performed methods. Training and evaluation pipeline.

Classification was performed within the dataset. An example is provided in Figure 3. Five possible classes were defined: raspberry (compliant raspberry) example is provided in Figure 3a, noncompliant (noncompliant/moldy raspberry), noncompliant + stalk (noncompliant/moldy raspberry with a stalk) and stalk were provided in Figure 3b, and defective material (contaminated material) is shown in Figure 3c.

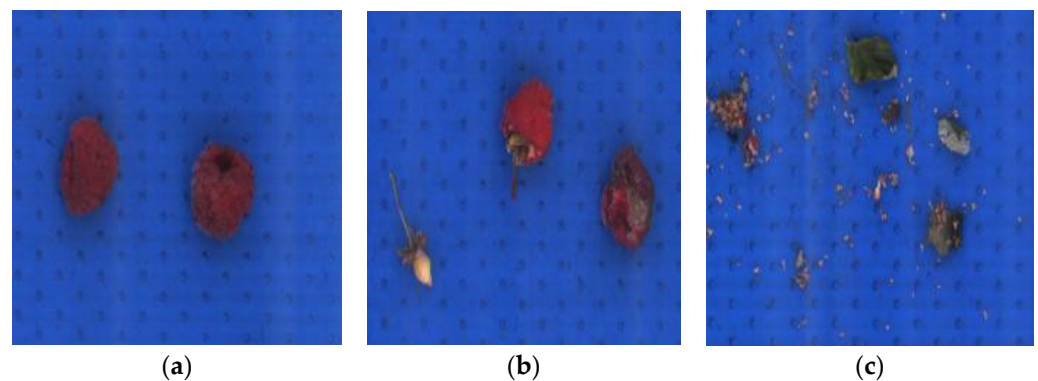


Figure 3. (a) Compliant raspberry (class: raspberry), (b) noncompliant raspberries (class: stalk, noncompliant, noncompliant + stalk), (c) defective materials (class: defect. mat.).

The original set contained 99 images of compliant raspberries and 66 images of non-compliant raspberries, as well as defective material. After numbering the images, the set was divided into the training set, validation set, and test set. The validation set included 20 randomly selected images, and the test set included 20 selected representative images containing a specific number of possible appearance variations of compliant and noncompliant raspberries, as well as defective material. A total of 55 objects of interest were included.

Data augmentation was performed with the training set, using techniques such as brightness level, rotation, blurring, gamma correction, mirror effect, and noise (Figure 4). The final set size was 4000 images (Table 1).

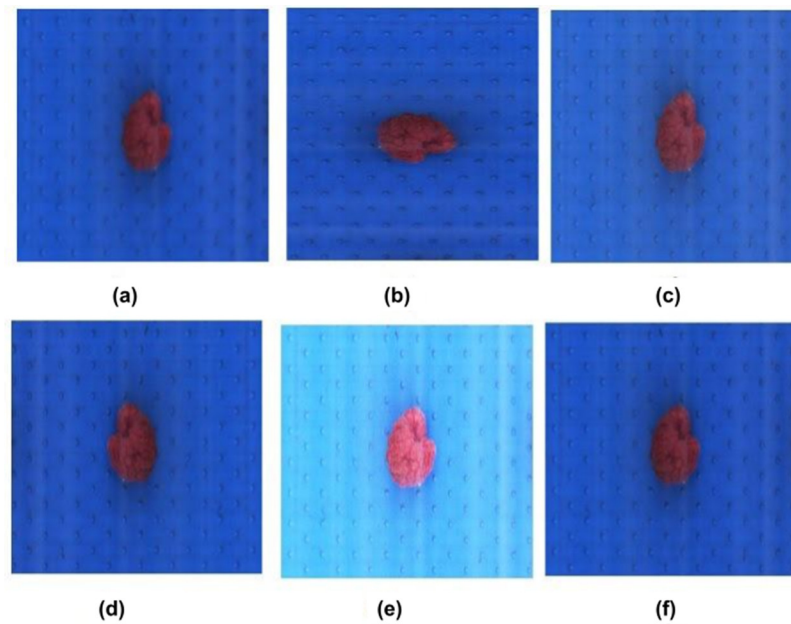


Figure 4. Image augmentation methods: (a) blur; (b) rotation; (c) gamma correction; (d) mirror effect; (e) brightness level; (f) noise.

Table 1. Table with input data.

Training Set	Compliant Raspberries	Noncompliant Raspberries	Total Raspberries
Number of original images	79	46	125
Brightness: +10, +20, +50, +100, 150, −10, −20, −50, −100	711	414	1125
Gamma: 0.5; 0.7; 0.9; 1.2; 1.5; 1.7; 2.0	553	322	875
Gaussian noise: 0.2; 0.5; 1.0; 1.5; 2.0	395	230	625
Rotation: 90 cw, 90 ccw, 180	237	138	375
Mirror	158	92	250
Blur: 0.2; 0.4; 0.6; 0.8; 1.0	395	230	625
Total		4000	

These techniques are ideal for the training set augmentation of the research subject and for other practical reasons. During the inspection process, the object is not always in the same position. In particular, the raspberry does not have a regular shape, so it can occupy several different positions due to shape variations. Illumination that is used by the color sorter is extremely strong; however, in real production some of the lamps may fail. Therefore, there is a possibility of reduced brightness level until the lamps are replaced, which can take several days. The image blurring technique Gaussian blur was utilized for several reasons. During agricultural products' inspection on the color sorter, there is a possibility—albeit very unlikely—that the glass, located above the conveyor belt but below the camera for product screening, may become blurred or dirty. The reasons for this may range from temperature changes in the plant to uncontrolled product motion along the belt at high speeds of fruit transport. Therefore, this technique has found its application for that purpose.

Further, manual labelling of the original training images, images augmented by rotation, and mirror effect were executed. The program `LabelImg` (<https://tzutalin.github.com/heartexlabs/labelImg/> (accessed on 10 May 2021)) was used for this purpose because it has the capacity of image labelling and automatic saving in the format demanded by Yolo and PascalVOC [36]. The program stores an image and a text file with the coordinates of the labelled object of interest and its assigned class.

A php script was created for other sets of augmented images. This script automatically copies the text file from the original training images, because coordinates of the object of

interest are identical. Only image exposures were changed, not the object position in the image, e.g., in the case of rotation and mirror effect.

The first step for initialization and technical setup was to create an account on a virtual machine and to install YOLOv3 environment. Here, YOLO implementation is based on Python programming language (<https://github.com/ultralytics/yolov3> (accessed on 24 August 2021)). A virtual machine, “Deep Learning”, was used for the needs of algorithm training, leased on the Google Cloud platform (<https://cloud.google.com> (accessed on 1 September 2021)). A virtual machine was used instead of a classic computer system for the following reasons:

- convolutional networks’ training on computers with average graphics card is slow, and even unusable;
- virtual machine comes with preinstalled software, which shortens the environment time-alignment;

User can configure multiple items such as: number of processors, graphics card type, disk size, preinstalled software, region of server location, etc. Configuration of the training computer used for research involves the following:

- GPUs: 1 × NVIDIA Tesla K80;
- CPU platform: Intel Haswell, 4 vCPUs, 15 GB memory;
- SSD hard disk.

Repository on all configurations is available online at any time at the following link: <https://github.com/IvaMark/YoloV3-Raspberry/> (accessed on 1 September 2020). The total of 9 trainings were executed with different hyperparameters to determine the best performances.

The following hyperparameters were changed: number of epochs, batch size, and learning rate. The basic ideas and guidelines in the choice of these hyperparameters are defined by Le Cun in his works [37,38]. Figure 5 shows a network used by YOLOv3 and a method of calculating the coordinates.

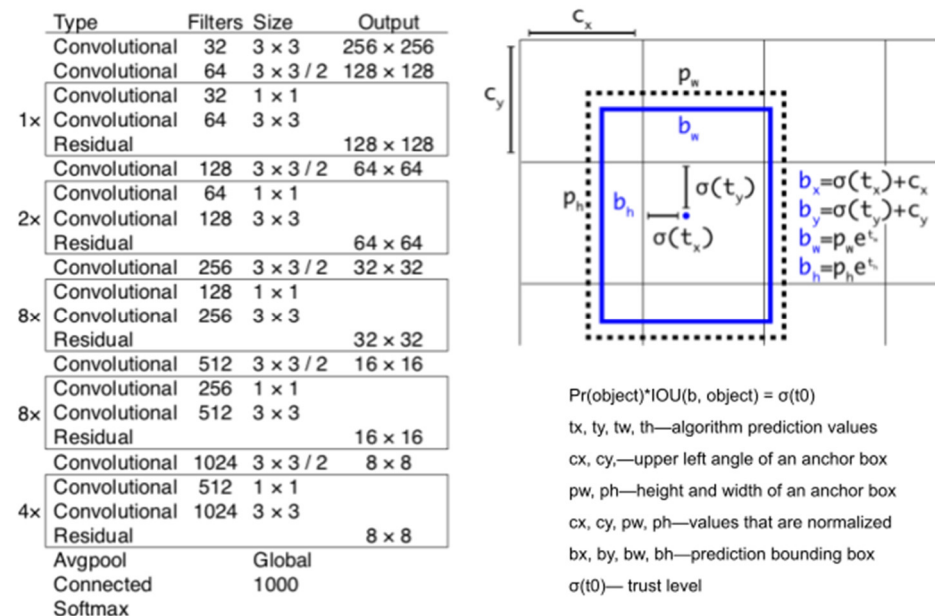


Figure 5. YOLOv3 network, DarkNet 53 and calculation of the coordinates adapted from Ref. [18]. 2018, Joseph Redmon and Ali Farhadi.

Detection on different layers solves defects of detecting small objects in the image. Upsampling layers concatenated with previous layers retains tiny characteristics that assist in detecting small objects. Figure 6 represents a simplified depiction of YOLOv3 network in order to show scaling and number of anchor boxes used.

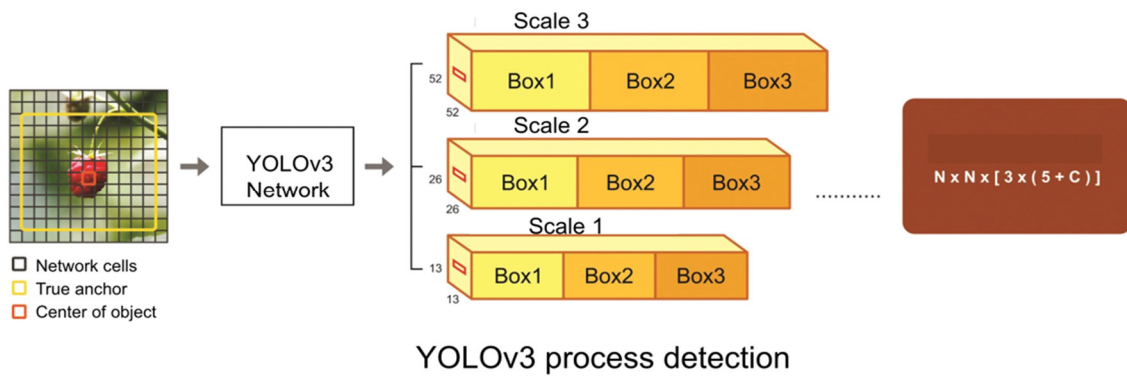


Figure 6. YOLOv3 network, simplified representation.

In trainings 1, 3, 4, 5, 6, and 7, pre-defined dimensions of Anchor boxes trained on the COCO dataset [39] were used, and in trainings 2, 8, and 9, new anchor boxes were trained using k-means algorithm on the original training set. The evaluation of the results was performed through the value of F-measure, GIoU, mAP, and test loss. The training time, detection time, and individual influence of augmentation techniques are shown and compared in tables and graphics.

Table 2 shows the parameters used for each training. It can be noted that hyperparameters for training 1 and 3 are identical; however, the difference lies in subsequent revision, where a series of errors was observed related to the incorrect assignment of a class to an object. Thus, a modified training test was subsequently used.

Table 2. Training parameters.

	Number of Epochs	Learning Rate	Batch Size	Anchors
Training 1	100	0.001 with default alignment	64	COCO
Training 2	150	0.001 with default alignment	64	trained
Training 3	100	0.001 with default alignment	64	COCO
Training 4	130	0.001 with default alignment	32	COCO
Training 5	130	0.001 with default alignment	128	COCO
Training 6	110	0.1 with default alignment	64	COCO
Training 7	110	initial 0.01, 0.001 after $\simeq 60$ epochs (3700 iterations), 0.0001 after $\simeq 25$ epochs (1500 iterations), 0.00001 after $\simeq 25$ epochs (1500 iterations)	64	COCO
Training 8	150	initial 0.001, change after $\simeq 60, \simeq 25, \simeq 25$ epochs	32	trained
Training 9	150	0.001 with default alignment	128	trained

Result evaluation was carried out through the value of F1 measure, GIoU, mAP, and test loss. The accuracy of predicted localization is affected by IoU, GIoU, confidence, and mAP, whilst classification is affected by F1 measure and classification. The model’s total performance, i.e., output errors, are represented by train and test loss.

2.1. F1 Measure

F1 measure is a measure most commonly used to define the success of a model [40]. It is calculated based on recall (R) and precision (P):

R, recall, is the relation between correctly classified objects (TP: true positive) and the total number of objects in that class.

$$R = TP / (TP + FN) \tag{1}$$

where TP (true positive) are correctly positively classified objects of interest, and FN (false negative) are incorrectly negatively classified objects of interest.

P, precision, is a measure that indicates the relation between true positives in a certain class and objects classified in that class, and is calculated according to the following formula:

$$P = TP / (TP + FP) \quad (2)$$

where FP (false positive) denotes incorrectly positively classified objects of interest.

And the F1 measure can be further calculated using the formula:

$$F1 = 2 * R * P / (P + R) \quad (3)$$

2.2. Mean Average Precision—mAP

Average precision (AP) refers to the area under the precision/recall (P/R) curve for each class. Mean value of average precision of individual classes is mean average precision and is defined in more detail in [41]. In the original YOLO work [42], authors report that they do not make the distinction between AP and mAP.

2.3. Generalized Intersection over Union—GIoU

In neural networks, any loss function must be differentiable so as to execute learning by the backpropagation method. IoU has value 0 in cases when the predicted bounding box does not intersect with the real box; therefore, there is no derivative. On the other hand, GIoU is always differentiable, and its range is from -1 to 1 . It covers the smallest possible area between the predicted and the real box. Therefore, it measures how close the predicted shape is to the real one, although they do not intersect, because some boxes are farther, and others are closer. An example is provided in Figure 7.

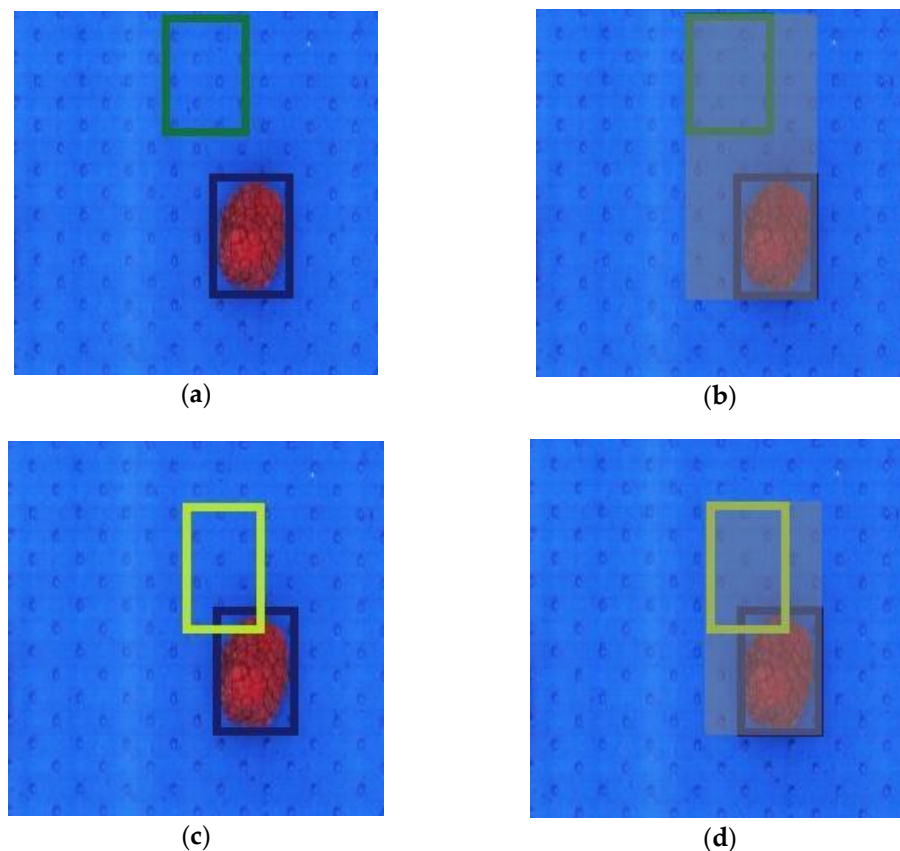


Figure 7. The cases when the predicted box (a) does not coincide at any point with real box: (c) has intersection with real box: (b,d) area C that serves to define how close the predicted box is to the real one.

Input: two arbitrary shapes: $A, B \subseteq S \in \mathbb{R}^n$. Output: GIoU
 A, B represent the predicted and real bounding box. C is the smallest convex envelope that includes A and B .

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \tag{5}$$

2.4. Total Error on the Training and Test Set—Train and Test Loss

The loss function is one of the criteria for evaluating model performances. As mentioned above several times, the primary goal is to minimize the value of this function and, through the training itself, to find the predicted values of weight factors, which coincide with real output values. The loss function, i.e., the total error of model, is defined in YOLO as the sum of coordinate prediction errors $Error_{coor}$, trust level errors $Error_{IoU}$, and classification errors $Error_{clc}$:

$$Loss = Error_{coor} + Error_{IoU} + Error_{clc} \tag{6}$$

Finally, a comparison of training time and detection time, as well as the effect of each augmentation technique, was performed.

3. Results and Discussion

Table 3 summarizes the results of all eight trainings. The F1 measures and GIoU demonstrated the best values during the second training, after using the trained anchor boxes (anchors = 24,26; 29,46; 31,80; 38,60; 42,45; 45,66; 46,81; 64,75; 66,42.). These anchor boxes were chosen and trained using the k-means algorithm on the original training set. The best values of mAP = 0.996 and test loss = 0.138 were obtained during the fifth training, after the batch size was augmented to 128. Some training employed the dimensions of anchor boxes trained on the COCO dataset [37] (anchors = 10,13; 16,30; 33,23; 30,61; 62,45; 59,119; 116,90; 156,198; 373,326).

Table 3. Summarized training results.

Training	GIoU	Train Loss	Precision	Recall	mAP	F1	Test Loss
1	0.0732	0.142	0.866	0.995	0.992	0.923	0.157
2	0.0577	0.122	0.941	0.995	0.993	0.967	0.142
3	0.0733	0.136	0.871	0.997	0.995	0.928	0.163
4	0.0624	0.11	0.899	0.998	0.994	0.943	0.145
5	0.0635	0.119	0.899	0.998	0.996	0.944	0.138
6	0.0643	0.114	0.884	0.996	0.994	0.935	0.149
7	0.0647	0.116	0.899	0.996	0.994	0.942	0.14
8	0.0635	0.14	0.913	0.998	0.996	0.953	0.163

Figure 8 shows examples of the results for detection, localization, and classification by YOLOv3 algorithm implementation.

To assess the effects of each technique applied for training set augmentation on the model performances, a model was designed (training 9) with new combinations of hyperparameters based on previous trainings: initial learning rate was 0.001; with first reduction, the learning rate was 0.0001 after 3200 iterations, and the learning rate after the second reduction was 0.00001 after 3600 iterations, with a batch size of 128, and with trained anchor boxes. The results are displayed in a graphical (Figure 9) and tabular (Table 4) form in the next section.

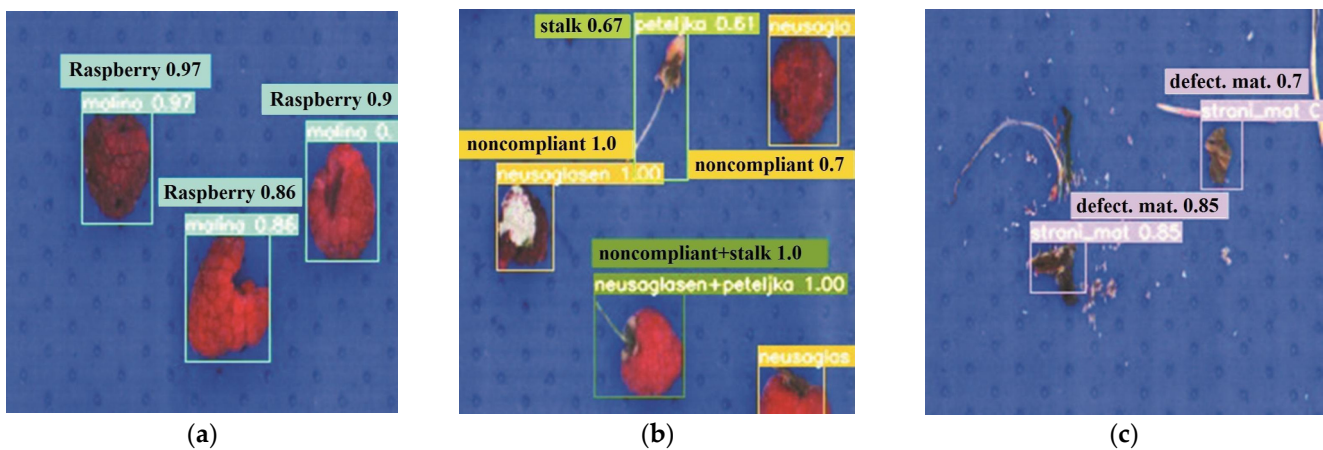


Figure 8. Example of the results obtained in training 3: (a) compliant raspberry, (b) class of non-compliant raspberry and stalk, (c) defective material.

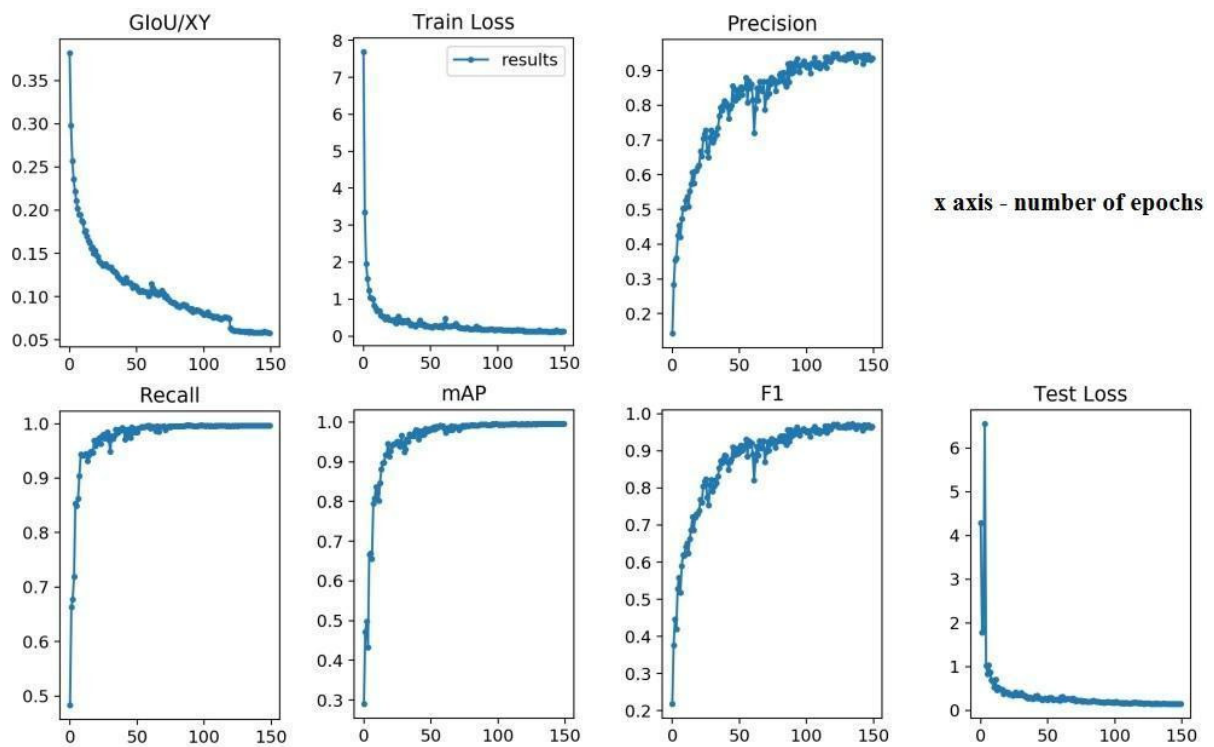


Figure 9. Graphical representation of the results for training 9.

Table 4. Summarized results for effects of other applied augmentation techniques.

Training	Detection Time (in Hours)	GIoU	Train Loss	mAP	F1	Test Loss
Original set—training 9	9.061	0.0579	0.125	0.995	0.965	0.146
Without brightness set	6.474	0.0692	0.156	0.992	0.953	0.183
Without gamma set	7.074	0.0728	0.16	0.991	0.945	0.192
Without rotation set	8.175	0.047	0.092	0.995	0.981	0.125
Without mirror effect set	8.685	0.057	0.109	0.994	0.975	0.143
Without blur set	7.607	0.069	0.161	0.991	0.933	0.206
Without noise set	8.094	0.065	0.142	0.994	0.97	0.156

Based on the table of obtained values (Table 4), it can be concluded that elimination of data produced by the rotation technique yields better results for test loss = 0.125,

F1 = 0.981, and GIoU = 0.047, whereas the highest positive effect was achieved by the blurring technique, and better results were also produced by other techniques.

3.1. Training and Detection Time

Table 5 presents the average network training time with training data, as well as the average time required to detect a single test image for the aligned parameters of the aforementioned trainings. Figure 10 shows a diagram with the comparative average detection time in each figure of the trainings 1–9.

Table 5. Average network training time for each training and test image.

Training	1	2	3	4	5	6	7	8	9
Network training	3.48 h	9.15 h	3.5 h	7.788 h	7.754 h	4.89 h	4.88 h	9.141 h	9.061 h
Test images	0.045 s	0.047 s	0.056 s	0.043 s	0.043 s	0.042 s	0.044 s	0.042 s	0.044 s

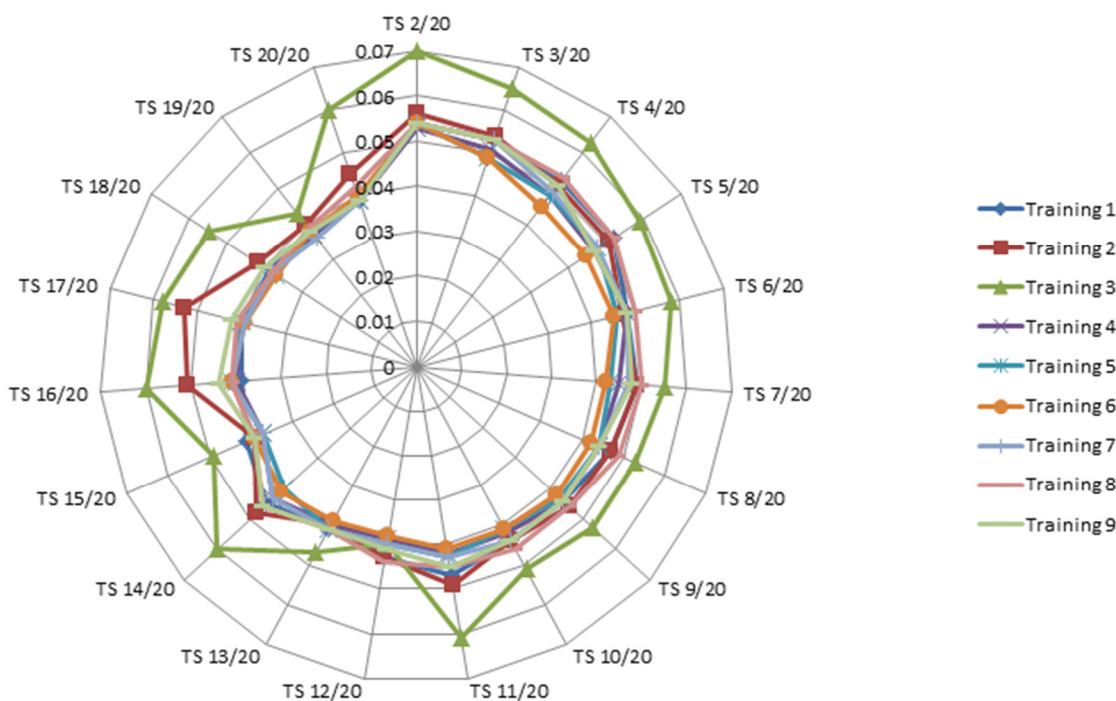


Figure 10. Comparative representation of detection time, in seconds, in 20 test images for all trainings.

Detection time was also measured in full-resolution images (1024×1024) obtained from an Optyx 3000 color sorter. The average time amounted to 0.37 s per image. The maximum conveyor belt speed in an Optyx 3000 color sorter is 3 m/s, i.e., ≈ 0.33 s duration for one image captured, considering that one pixel screened is 1 mm. If YOLOv3 detection time in a full-resolution image (1024×1024) is ≈ 0.37 s, it can be concluded that this algorithm could be applied on these machines for its stated purpose. It should be noted that network training time depends on the processing unit used for training. Network training can be executed through GPU (graphics processing unit), where graphics card power is utilized, or CPU (central processing unit), where processor power is used. Training of convolutional neural networks is more convenient through GPU. As mentioned at the beginning of this paper, the power of the NVIDIA Tesla K80 graphics card was used, and this represents a middle class for training of deep learning algorithms. For example, when a deep learning virtual machine is created on Google Cloud, this card is set as default. The training set for this research did not require extensive computer resources; however, network training time would certainly be shorter if a more powerful graphics card was used.

Furthermore, the number and size of data in the training set, in addition to the above-mentioned hyperparameters, affect the length of training duration.

3.2. Web Application

After the results were obtained, a simple web application was created in the form of a graphical interface for training weight coefficients for new agricultural products. The application contains predefined weight factors trained using the YOLOv3 algorithm for raspberries, as shown in Figure 11.

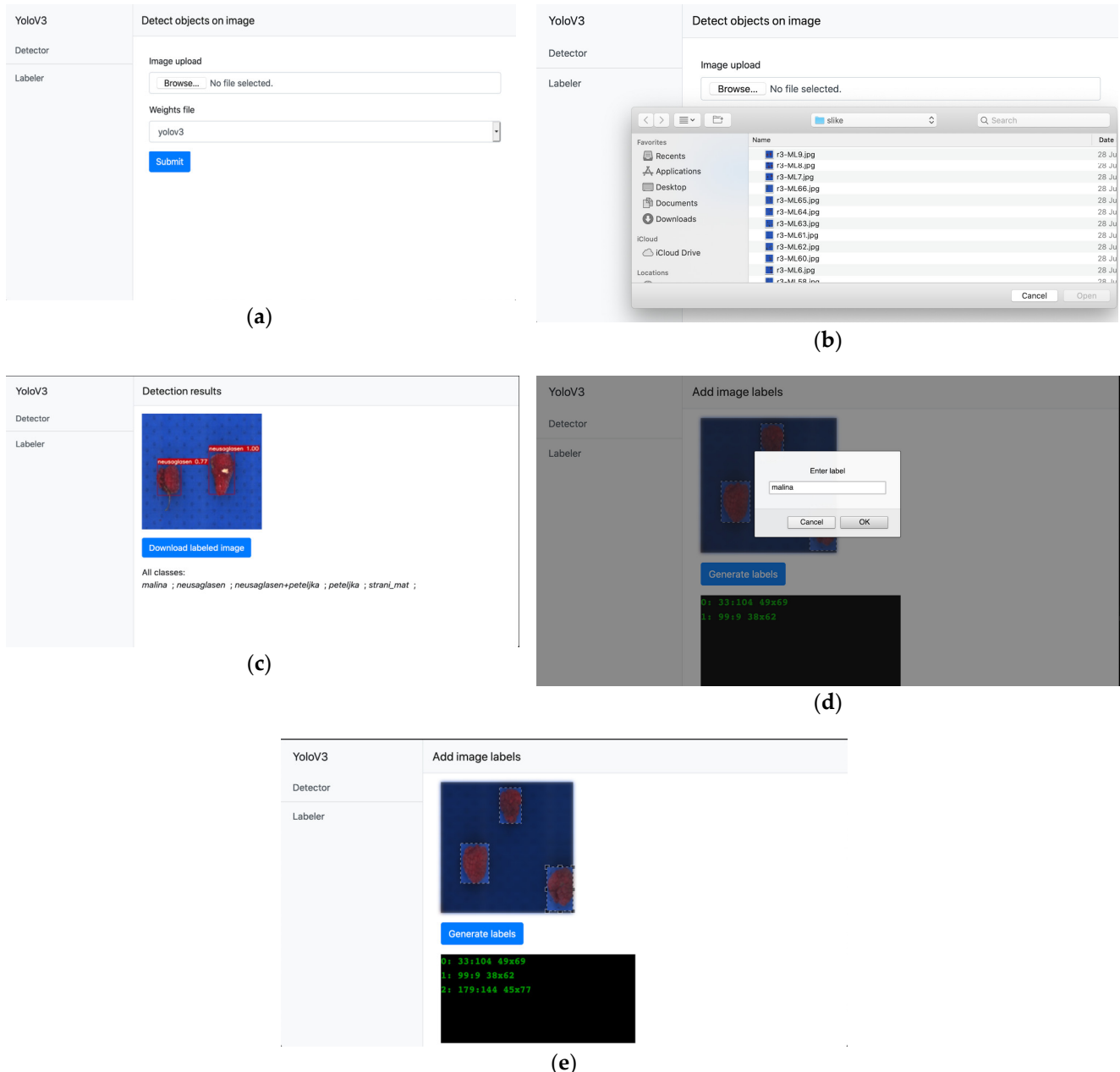


Figure 11. (a) Front page of the application for the selection of weight coefficients and images, (b) inserting an image, (c) detection results, (d) creating a new class, (e) labeling objects of interest.

The application is open source and was set up on a GitHub development platform at the following link: <https://github.com/ultralytics/yolov3> (accessed on 24 August 2021). Image detection, by using trained weight factors, is executed through a command-line interface. After starting, a selection of images from a default input folder is produced, as

shown in Figure 11b. A background algorithm with predefined weight factors performs the labeling of objects in the image (Figure 11c) and saves the labeled image in a default output folder. The input and output folder can be forwarded as a command argument. Web application is written in the Flask web framework based on the Python programming language. The original application code and configuration training are stored in the GitHub repository at the following link: <https://github.com/IvaMark/YoloV3-Raspberry>. The application is packed in a Docker container; therefore, the user does not need to install the application and dependencies manually. The only requirement is the installation of the Docker software (<https://www.docker.com/> (accessed on 1 August 2021)).

Once Docker is successfully installed, the GitHub repository needs to be cloned and brought to the docker-application directory. Further, the following commands must be executed in order to create applications in the local environment:

- `docker-compose build`
- `docker-compose up -d`

The application is currently available on the web through the following address: <http://localhost:43311/> (accessed on 20 September 2021).

In addition to weight factors trained on the raspberry example, image detection can be executed by using weight factors trained on the COCO dataset (330,000 images, 80 classes) created for YOLOv3, YOLOv3-spp, and YOLOv3-tiny. It is possible to add a larger number of predefined parameters and, therefore, to extend the assortment of agricultural products that can be tested. The next possibility for the application is for the selection of the predefined classes' representation for specific agricultural products. The developed application represents a beta version that can be adapted to various agricultural products.

3.3. Discussion

The information and results provided above present the theoretical possibility of the YOLOv3 algorithm's real application in industrial production for the needs of agricultural products' sorting and inspection.

As mentioned, the literature considered does not include works directly related to the subject of this research. To our knowledge, this is the first time that research employs real images from a color sorter, and we intended to initially contribute and open the theme for future research and application of deep learning algorithms in the industrial conditions of inspection.

In [14,32,33,43], only one or two classes were used as an output [44]. In relation to the more complex variations of the object of interest and the number of classes in our work, it can be concluded that the results obtained in this research are more significant because 5 classes are defined in our work and the results are promising: F1 as a measure of the proposed algorithm amounted to 92–97%, and the machine's current accuracy is 91–98%.

Research study [16] defines multiple apple classes according to growth stages. Results of F1 measure indicated that the number of classes affects model output performances. On the other hand, the training set in this paper involves compliant raspberries with different ripeness levels, but they are defined as a single class, and there were no errors of this type on the test set. It should be noted that in their study, YOLOv3 is modified by the *DenseNet* method, the aim of which is to better preserve and use significant characteristics with the YOLO network; therefore, this network produced better results in their research compared to the standard YOLOv3 network, whilst our work did not employ additional promotion of the network.

The same group of researchers used the above-mentioned improved version of YOLOv3 to detect anthracnose apples [13]. This study can represent the confirmation of the success of the YOLO algorithm in identifying infected fruit, as was carried out in this work, where only one disease is separated in three stages; however, our work is concerned with the examples of moldy raspberry, as well as different types of additional damages.

In [43], the authors highlighted the advantages of YOLOv5 in terms of lightweight and better real-time performance. However, YOLOv3 produced exactly the same results as YOLOv5 for F1, mAP, recall, and precision. A result of approx. 100% would have been achieved in our work also, if the aim had been to detect raspberry only. Here, we wanted to formulate a realistic scenario of agricultural products' industrial inspection and to highlight multiple challenges of detection, e.g., stalk detection.

In [45], the authors report that YOLO algorithms are continuously studied, and a table is shown containing the total number of created improved versions of different basic YOLO algorithms over a five-year period. Studies of the literature indicate small differences in precision results between YOLOv3 and later versions, but these are most evident in detection speed and the time needed for network training. In this context, if any other YOLO algorithm had been applied to our dataset, it would not have significantly affected our conclusions on potential applicability in industrial processing of agricultural products. Another reason for our belief that the application of CNN networks would produce good results in this field is that potential flaws of the application of networks such as CNNs include the fact that they do not encode the position and orientation of the object, and a great amount of training data is required, easily leading to a zero level with the use of augmentative methods such as rotation (or others), and to the easy processing of a large dataset in a short period of time. Furthermore, the industrial conditions of machine inspection are near-experimental: during product screening, in passing through a color sorter, the camera is always screening the product from the same distance, at the same angle, and the background does not change. At the very outset, the greatest challenges for detection algorithms were avoided.

The application of CNNs can represent the optimization of the process defining an acceptable product and a product intended for processing, such as a raspberry with a stalk. A major advantage of the YOLOv3 algorithm is the possibility of detection, localization, and classification in real time where the image is used as an input. It is certainly possible to create more easily the final model and weight factors with a specific number of images and for each defined class. The output accuracy would be equal or greater than the accuracy of current color sorters, which is presented in the work as theoretically possible but with additional advantages provided by deep learning algorithms. By developing the beta version of the proposed software, where trained weight factors can be stored, the desired agricultural product can be easily tested before it becomes the object of industrial color sorting.

4. Conclusions

Results of the work show that convolutional neural networks, specifically the YOLOv3 algorithm on the raspberry example, achieve accuracy and speed at a high level in detection, localization, and classification. In all 9 trainings, the model accuracy, according to F1 measure, ranged between 92 and 97%. The impact of hyperparameters defining the YOLOv3 model was assessed. Learning rate did not have significant influence on model performances. The training set size is assumed to be the reason. The number of epochs, trained anchor boxes, and the choice of batch size influenced performances and the classification precision of the model. It is also possible to obtain good results with default hyperparameters and anchor boxes. The effects of augmentation techniques on model performances are significant, according to the results. The augmentation technique that produced the greatest positive impact on the final defined model was blurring, while the rotation technique had a negative impact.

There are two significant findings. The first one is that the detection and classification of an object such as a stalk represents one of the significant results of YOLOv3 algorithm application due to its shape, small dimensions, and variations. Detection, localization, and classification of noncompliant raspberries that had various types of damage and shape, and were classified into a single class, can be also considered an outstanding result of the YOLOv3 algorithm implementation.

Since various results define how successful an algorithm is (e.g., precision, training time, speed detection, etc.), the results are not influenced only by the choice of the algorithm type but also by the hardware. The reliable result and algorithm choice can be confirmed only by direct use on the machine.

Authors agree that this method can be useful as a nondestructive method for detection, localization, and classification of agricultural products during machine inspection. Average detection time in full resolution images (1024×1024) amounts only to 0.37 s with the selected middle-class software components. With a better graphics card and overall system, this time can be additionally reduced. The application of this method can largely promote the accuracy of object identification, accelerate the efficiency of agricultural products' sorting, increase objectivity in decision making (i.e., machine learning), and ensure greater economic benefits of the entire process.

The paper also presents a developed model of software solution in the form of a graphical interface that allows for testing of the machine learning algorithms and potential application during real processing of agricultural products. Future development of the application functionality would include forwarding new images intended for training through online services, after which a new version of the weight factor could be sent to the application.

The next direction of development of this topic would involve investigation of the unsupervised learning algorithm application, where the possibility of processing a larger amount of data in real time would be increased. In general, it is necessary to set out and create solutions that apply deep learning in such systems of industrial production. A general improvement in the new area of deep learning and convolutional neural networks is evident, and essential for further investigation and application in smart and sustainable agricultural production, as well as for achieving greater precision and safety in the food industry.

Author Contributions: Conceptualization, I.M.; methodology, I.M.; software, I.M.; validation, I.M., J.I.; formal analysis, I.M., E.V. and A.J.; investigation, I.M.; resources, I.M., E.V. and A.J.; data curation, I.M., E.V. and A.J.; writing—original draft preparation, I.M.; writing—review and editing, I.M., E.V., A.J. and J.I.; visualization, I.M.; supervision, J.I. 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: Data sharing not applicable.

Acknowledgments: This research was supported by the Serbian Ministry of Science and Technological Development—project by contract 451-03-68/2022-14/200105 from 4 February 2022.

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

References

1. Rolandi, S.; Brunori, G.; Bacco, M.; Scotti, I. The Digitalization of Agriculture and Rural Areas: Towards a Taxonomy of the Impacts. Topical Collection “Smart Farming and Sustainability”. *Sustainability* **2021**, *13*, 5172. [[CrossRef](#)]
2. López-Morales, J.A.; Martínez, J.A.; Skarmeta, A.F. Digital Transformation of Agriculture through the Use of an Interoperable Platform. Special Issue Selected Papers from the 3rd Global IoT Summit. *Sensors* **2020**, *20*, 1153. [[CrossRef](#)] [[PubMed](#)]
3. Melnikov, Y.B.; Skvortsov, E.; Ziablitskaia, N.; Kurdyumov, A. Modeling of Territorial and Managerial Aspects of Robotization of Agriculture in Russia. *Mathematics* **2022**, *10*, 2540. [[CrossRef](#)]
4. Khoje, S. Appearance and characterization of fruit image textures for quality sorting using wavelet transform and genetic algorithms. *J. Texture Stud.* **2017**, *49*, 65–83. [[CrossRef](#)] [[PubMed](#)]
5. Markovic, D.; Mladenovic, N.; Simonovic, V.; Markovic, I.; Stevanovic, M.S. Fruit flow calculation on the rotating sizing machines. *Teh. Vjesn.* **2014**, *21*, 645–650.
6. Teena, M.; Manickavasagan, A.; Mothershaw, A.; Hadi, S.E.; Jayas, D.S. Potential of machine vision techniques for detecting fecal and microbial contamination of food products. A review. *Food Bioprocess Technol.* **2013**, *6*, 1621–1634. [[CrossRef](#)]

7. Zhang, Y.; Wang, S.; Ji, G.; Philips, P. Fruit classification using computer vision and feedforward neural network. *J. Food Eng.* **2014**, *143*, 167–177. [CrossRef]
8. Ortac, G.; Bilgi, A.S.; Tasdemir, K.; Kalkan, H. A hyperspectral imaging based control system for quality assessment of dried figs. *Comput. Electr. Agric.* **2016**, *130*, 38–47. [CrossRef]
9. Shankar, K.; Kumar, S.; Dutta, A.K.; Alkhayyat, A.; Jawad, A.J.M.; Abbas, A.H.; Yousif, Y.K. An Automated Hyperparameter Tuning Recurrent Neural Network Model for Fruit Classification. Special Issue Decision Making and Its Applications. *Mathematics* **2022**, *10*, 2358. [CrossRef]
10. Shrestha, B.L.; Kang, Y.M.; Yu, D.; Baik, O.D. A two camera machine vision approach to separating and identifying laboratory sprouted wheat kernels. *Biosyst. Eng.* **2016**, *147*, 265–273. [CrossRef]
11. Teimouri, N.; Omid, M.; Mollazade, K.; Rajabipour, A. A novel artificial neural networks assisted segmentation algorithm for discriminating almond nut and shell from background and shadow. *Comput. Electr. Agric.* **2014**, *105*, 34–43. [CrossRef]
12. Zhang, S.; Wu, X.; Zhang, S.; Cheng, Q.; Tan, Z. An effective method to inspect and classify the bruising degree of apples based on the optical properties. *Postharvest Biol. Technol.* **2017**, *127*, 44–52. [CrossRef]
13. Tian, Y.; Yang, G.; Wang, Z.; Li, E.; Liang, Z. Detection of Apple Lesions in Orchards Based on Deep Learning Methods of CycleGAN and YOLOV3-Dense. *J. Sens.* **2019**, *2019*, 7630926. [CrossRef]
14. Liu, G.; Mao, S.; Kim, J.H. A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. *Sensors* **2019**, *199*, 2023. [CrossRef]
15. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 3289801. [CrossRef]
16. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electr. Agric.* **2019**, *157*, 417–426. [CrossRef]
17. Kamilaris, A.; Prenafeta-Boldú, F.X. A review of the use of convolutional neural network. *J. Agric. Sci.* **2018**, *156*, 312–322. [CrossRef]
18. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. University of Washington. 2018. Available online: <https://arxiv.org/abs/1804.02767> (accessed on 15 May 2019).
19. Tajar, A.T.; Ramazani, A.; Mansoorizadeh, M. A lightweight Tiny-YOLOv3 vehicle detection approach. *J. Real Time Image Process.* **2021**, *18*, 2389–2401. [CrossRef]
20. Wang, K.; Liu, M. YOLOv3-MT: A YOLOv3 using multi-target tracking for vehicle visual detection. *Appl. Intell.* **2022**, *52*, 2070–2091. [CrossRef]
21. Genaev, M.A.; Komyshev, E.G.; Shishkina, O.D.; Adonyeva, N.V.; Karpova, E.K.; Gruntenko, N.E.; Zakharenko, L.P.; Koval, V.S.; Afonnikov, D.A. Classification of Fruit Flies by Gender in Images Using Smartphones and the YOLOv4-Tiny Neural Network. Special Issue Mathematical and Computational Methods. *Syst. Biol.* **2022**, *10*, 295. [CrossRef]
22. Jian, H.; Yaping, L.; Junyou, Z.; Shufeng, W.; Sixian, L. Target Fusion Detection of LiDAR and Camera Based on the Improved YOLO Algorithm. Special Issue Mathematics and Engineering. *Mathematics* **2018**, *6*, 213. [CrossRef]
23. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. Special Issue Artificial Intelligence for Smart Sensing, Test and Measurement. *Sensors* **2021**, *21*, 6519. [CrossRef] [PubMed]
24. Jiang, X.; Gao, T.; Zhu, Z.; Zhao, Y. Real-Time Face Mask Detection Method Based on YOLOv3. Section Computer Science & Engineering. *Electronics* **2021**, *10*, 837. [CrossRef]
25. Mujahid, A.; Awan, M.J.; Yasin, A.; Mohammed, M.A.; Damaševičius, R.; Maskeliūnas, R.; Abdulkareem, K.H. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. Special Issue Digital Image Processing and Analysis: Human and Computer Vision Applications. *Appl. Sci.* **2021**, *11*, 4164. [CrossRef]
26. Fei, S.; Yanping, Z.; Guanghui, W.; Pingzeng, L.; Yinfa, Y.; Linlu, Z. Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment. Special Issue Application of Image Recognition Processing Technology in Agricultural. *Agronomy* **2022**, *12*, 1638. [CrossRef]
27. Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V. Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415. Special Issue Mathematical Modelling in Science and Engineering. *Symmetry* **2022**, *14*, 148. [CrossRef]
28. Rincón, A.Q.; Mora, M.; Naranjo-Torres, J.; Fredes, C.; Valenzuela, A. Raspberries-LITRP Database: RGB Images Database for the Industrial Applications of Red Raspberries' Automatic Quality Estimation. *Appl. Sci.* **2022**, *12*, 11586. [CrossRef]
29. Nhat-Duy, N.; Tien, D.; Thanh, D.N.; Duy-Dinh, L. An Evaluation of Deep Learning Methods for Small Object Detection. *J. Electr. Comput. Eng.* **2020**, *2020*, 3189691. [CrossRef]
30. Liu, J.; Wang, X. Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. Section Technical Advances. *Plant Sci.* **2020**, *11*, 898. [CrossRef]
31. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. Computer Science, Computer Vision and Pattern Recognition. *arXiv* **2004**, arXiv:2004.10934.
32. Kuznetsova, A.; Maleva, T.; Soloviev, V. YOLOv5 versus YOLOv3 for Apple Detection. In *Cyber-Physical Systems: Modelling and Intelligent*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 349–358. [CrossRef]

33. 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. Special Issue Precision Management to Promote Fruit Yield and Quality in Orchards. *Agronomy* **2022**, *12*, 319. [[CrossRef](#)]
34. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Section Remote Sensors. *Sensors* **2022**, *22*, 464. [[CrossRef](#)] [[PubMed](#)]
35. Koiralaa, A.; Walsha, K.; Wanga, Z.; McCarthyb, C. Deep learning—Method overview and review of use for fruit detection and yield estimation. *Comput. Electr. Agric.* **2019**, *162*, 219–234. [[CrossRef](#)]
36. Everingham, M.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
37. LeCun, Y.; Bottou, L.; Orr, G.B.; Müller, K.R. *Efficient BackProp. Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7700. [[CrossRef](#)]
38. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
39. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2015**, arXiv:1405.0312.
40. Teimouri, N.; Omid, M.; Mollazade, K.; Mousazadeh, H.; Alimardani, R.; Karstoft, K. On-line separation and sorting of chicken portions using a robust vision-based intelligent modelling approach. *Biosyst. Eng.* **2018**, *167*, 8–20. [[CrossRef](#)]
41. Henderson, P.; Ferrari, V. End-to-End Training of Object Class Detectors for Mean Average Precision. In *Asian Conference on Computer Vision—ACCV*; Springer: Cham, Switzerland, 2016; pp. 198–213. [[CrossRef](#)]
42. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
43. Zhang, C.; Ding, H.; Shi, Q.; Wang, Y. Grape Cluster Real-Time Detection in Complex Natural Scenes Based on YOLOv5s Deep Learning Network. *Agriculture* **2022**, *12*, 1242. [[CrossRef](#)]
44. Zhang, C.; Kang, F.; Wang, Y. An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds. *Remote Sens.* **2022**, *14*, 4150. [[CrossRef](#)]
45. Peiyuan, J.; Daji, E.; Fangyao, L.; Ying, C.; Bo, M. A review of Yolo algorithm developments. In Proceedings of the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19, Chengdu, China, 9–11 July 2021; Volume 199, pp. 1066–1073. [[CrossRef](#)]