International Journal of Agricultural and Natural Sciences
E-ISSN: 2651-3617 14(3): 232-246, 2021
Research Article

REAL TIME PEST DETECTION USING YOLOv5

Eray Önler*

Tekirdag Namik Kemal University, Faculty of Agriculture, Department of Biosystem Engineering, Tekirdağ, Turkey

**Corresponding Author*
E-mail: erayonler@nku.edu.tr

(Received 10th May 2021; accepted 15th December 2021)

ABSTRACT. Depending on the increasing population and nutritional needs, we should develop new methods and systems in agricultural production that take environmental issues into account and ensure efficiency and sustainability. Inappropriate pest control methods can result in 70% of yield loss. The caterpillar is a pest that can be invasive and can damage yield by eating the leaves, shoots, fruit and flower parts of plants and trees. Pesticide spraying is the most preferred pest control method due to its speed of action and scalability. However, due to the increasing environmental and health awareness, less pesticide use is required. One of the important methods of reducing pesticide usage is to spray only the places where they are needed. In order to perform spot spraying, first of all, the location of the pest must be determined. It is possible to detect pests using computer vision methods. In the study, we developed an object detection system to detect the thistle caterpillar (*Vanessa cardui*), which is encountered in Turkey and can cause damage to sunflower cultivation, in real time via video using the YOLOv5 object detection architecture. For this purpose, we used 2416 images taken under different lighting and background conditions. We trained the object detection system in two different ways using transfer learning and learning from scratch methods and compared the results. Results indicate that the system is functional and being able to correctly detect the thistle caterpillar at 65 FPS.

Keywords: *Deep Learning, Pest Detection, Real Time Detection, YOLOv5*

INTRODUCTION

Inappropriate pest control methods can result in 70% of yield loss. Agricultural pests can cause 33% of potential damage. In addition, loss of quality can cause 15% economic loss. [4]. Pests and diseases cause 20-40% loss of agricultural products worldwide. An economic loss of 220 billion USD occurs due to plant diseases and 70 billion USD due to pests [3]. Changes that occur as a result of climate change increase the incidence of diseases and pests and cause them to be seen in places that have not been seen before. [12]. The caterpillar is a pest that can be invasive and can damage yield by eating the leaves, shoots, fruit and flower parts of plants and trees. [2, 22]. There are many caterpillar species that are the main pests in arable farming, horticulture and fruit production [1, 10, 20]. Pesticide spraying is the most preferred pest control method due to its speed of action and scalability [13]. However, due to the increasing environmental and health awareness, less pesticide use is required. One of the important methods of reducing pesticide usage is to spray only the places where they are needed. It is known that pesticide spraying costs can be reduced by 90% with spot spraying applications [16]. In order to perform spot spraying, first of all, the location of the pest must be determined. Manual methods are commonly used for the detection of pests, which are labor dependent and therefore highly error-prone [7]. However, due to the developing technology, it is possible to detect pests using image processing methods.

Deep neural networks are widely used in computer vision applications due to their automatic feature extraction and ability to extract complex relationships [5].

Technological developments in GPUs (Graphical Processing Unit) have been able to train deeper artificial neural networks that can process more data. Deep neural networks, which proved their accuracy in object classification, have also been used in object detection algorithms. The most widely used object detection algorithms are divided into classification based object detectors (two-stage detectors) and regression based object detectors (one-stage detectors). Two-stage object detectors have higher object detection accuracy than single-stage object detectors, but they are slower in terms of inference speed [11].

In this study, we aimed to detect the thistle caterpillar in real time from digital image/video by using the YOLOv5 object detection architecture which is based on the one-stage object detector YOLO (You Look Only Once) [14] method. YOLOv5 versions with different sizes, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x architectures, were compared in terms of mAP (mean average precision) and inference speed with agricultural damage-causing thistle caterpillar images using transfer learning and training from scratch.

The results show that the developed system is functional and can detect the spiny caterpillar. We observed that the object detection system we developed can detect objects at 65 FPS with the use of the Tesla K80 12 GB (Gigabyte) GPU in the Google COLAB development environment. In addition, it has been observed that the model can learn faster when transfer learning is used.

MATERIAL AND METHODS

Our research approach consists of 5 consecutive steps (Fig. 1). First, we collected a dataset of thistle caterpillar images that we will use in the training and validation of the object detection system. Then, we pre-processed this entire dataset by annotation. We trained the YOLO object detection models with the dataset, which we split for training. Using the dataset we split for validation, we validated the real-time detection performance of the trained YOLO models and evaluated the results.

Fig. 1. *Research approach*

Dataset Collection

To train and validate the object detection system, we used open-source images of thistle caterpillars captured on soybeans (Fig. 2). The dataset we used was divided by the owner to include 1934 images for training and 241 images for testing and 241 images for validation [18]. In the study, we combined the test and validation folders in the original dataset and used them in the validation of the trained models. After training and testing different model variations, we identified the most successful model in terms of mAP. To test the generalizability of this model and whether it can be used to detect different caterpillar species, we used images of cabbage caterpillars (*Pieris brassicae)* taken in September 2021, which were encountered in a cabbage-growing farm in the Muratbey District of Catalca - Istanbul - Turkey

Fig. 2. *Images of Thistle Caterpillars we use in training and validation of object detection models*

Data Annotation

We subjected the thistle caterpillar images, which we divided into different folders for training and validation, to data annotation in YOLO format. We performed the annotation manually with the free data annotation app makesense.ai (Fig. 3).

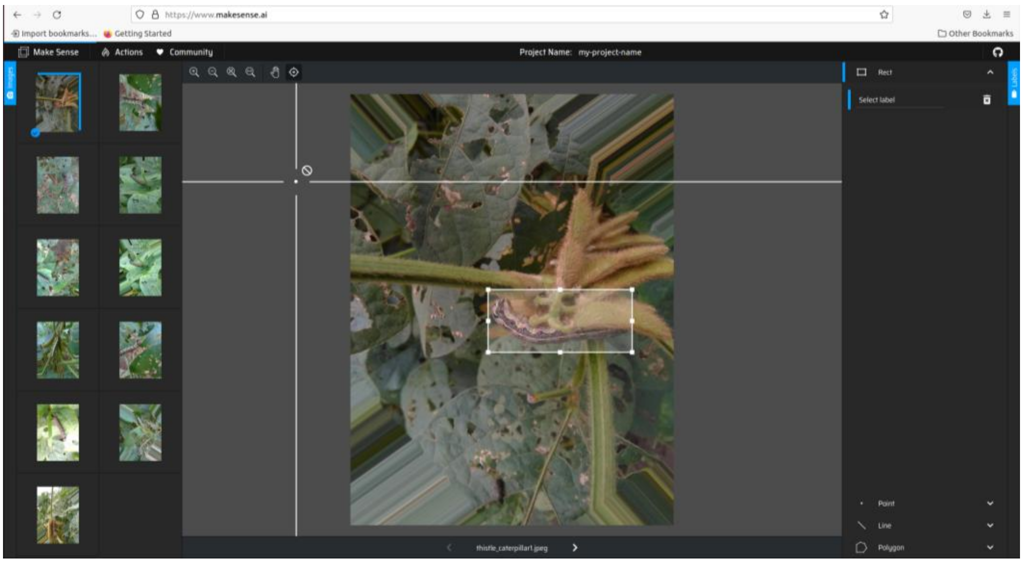

Fig. 3. *Image Annotation App*

The annotation process consists of marking the object to be detected on each image by taking it into a rectangle. More than one object can be marked on the image. Information about the objects marked on each image is stored in a text file with the same name as the related image. This text file has one line per object. In each line in the YOLO annotated file, the data includes the class of the marked object, the center of the drawn rectangle on the x and y axis, and the width and height of the rectangle. The coordinates of the rectangle are normalized between 0 and 1 to be independent of the image size. Thus, since each image will be evaluated according to its size, images with different sizes can be used in the object detection system. Figure 4 shows the content of the sample annotation file of an image. For each image, we created a text file with the same name as the image, which contains the annotated information.

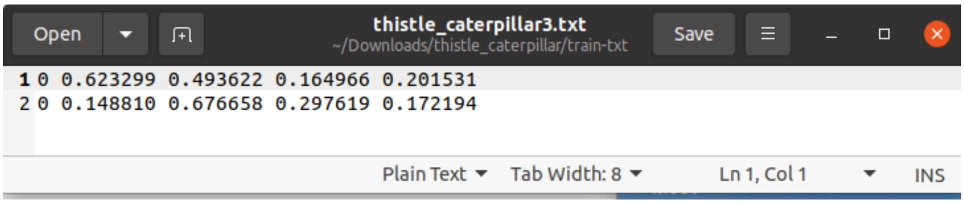

Fig. 4. *Sample annotation file for an image*

YOLOv5

YOLOv5 is a one-stage object detection architecture. One-stage object detection architectures treat object detection as a regression problem. It estimates the class probability and the coordinates of the bounding box that will contain the object in a single step on the input image [19]. Like other one-stage object detection architectures (SSD, YOLOv3, YOLOv4, RetinaNet etc.), it consists of three basic parts: backbone, neck and head (Fig. 5). The head layer is also called the YOLO layer.

The task of the model backbone is to reveal the distinctive features from the given image. CSPNet (Cross Stage Partial Networks) structure is used as the model backbone in YOLOv5 [24]. In large-size neural network backbones, the gradient information is copied in updating the layer weights and most of them have to be learned over and over again. This is a situation that negatively affects model size and inference speed. In the CSPNet approach, the feature map in the base layer is divided into two, some of them reach the transition layer through the dense block, and the other part is directly combined with the transition layer. In this way, not only the model size is reduced, but also the inference speed increases [25].

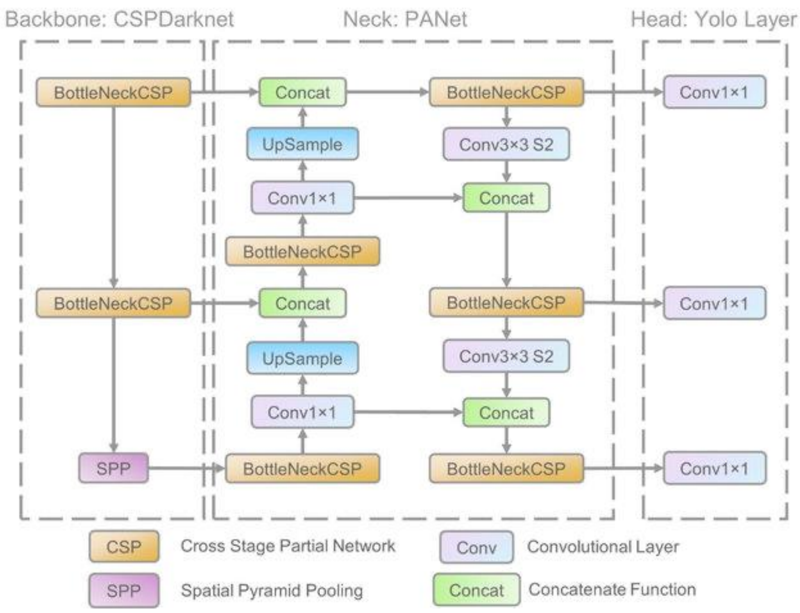

Fig. 5. *YOLOv5 Architecture [25]*

The model neck is used to create feature pyramids. With feature pyramids, the model can be generalized for different dimensions of the object (Fig. 6). Thus, images of the same object in different sizes and scales can be detected. YOLOv5 is using the PANet (Path Aggregation Network) feature pyramid. In PANet feature transfer, it creates an information shortcut that will allow localization signals from lower layers to reach the top feature layers without being lost. For this, an additional bottom-up path augmentation is added to the classical FPN (Feature Pyramid Network) [9, 21].

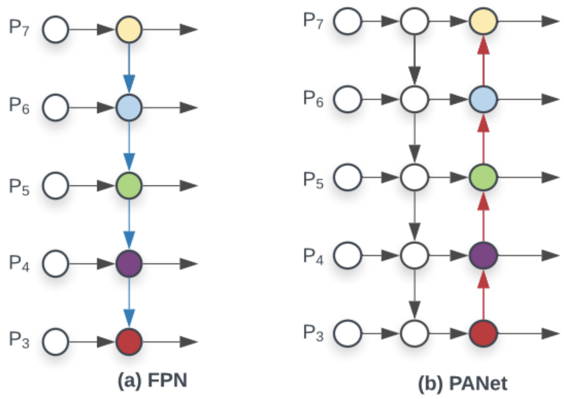

Fig. 6. *Two Types Feature Pyramids [21]*

The model head is also called the YOLO layer and is the step in which the object and object position are predicted. The YOLO layer outputs vectors containing class probability, confidence score and bounding box coordinates.

In the study, we used 4 versions of YOLOv5, which are named according to model size and complexity, as small, medium, large and extra large. The results obtained when these models are trained on the COCO dataset [6] are shown in Fig. 7 COCO is a dataset of 80 classes and 330k images used as a benchmark to compare object detection architectures.

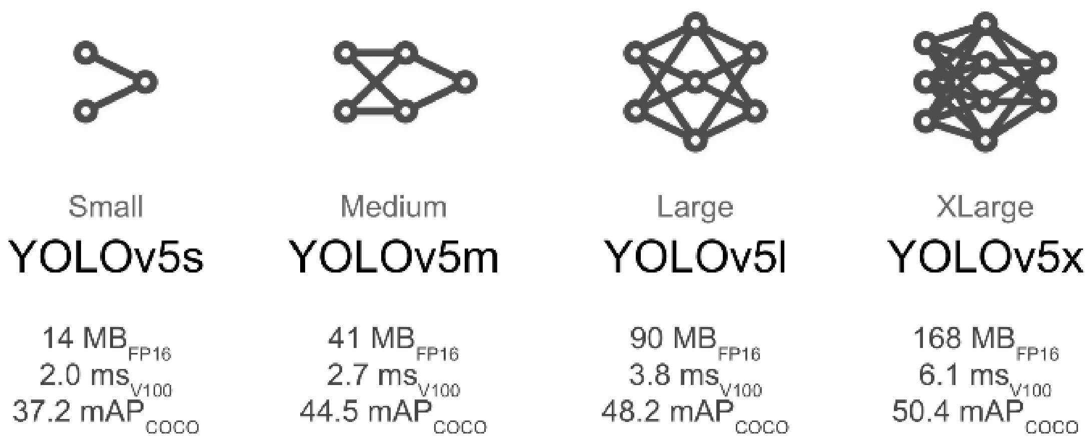

Fig. 7. *Yolov5 versions of different complexity [23]*

Transfer Learning

Machine learning methods based on the use of neural networks are very data hungry. Especially in applications based on computer vision, collecting a large number of images and annotating them all is a time-consuming and costly process. With transfer learning, it has become possible to transfer basic and common features learned by models trained on another domain to other models. Thus, while transferring low-level features, learning only higher-level features in the new model can be achieved with relatively less data [26]. For this purpose, transfer learning was applied using models trained on the COCO dataset by using their pre-trained weights.

Performance Evaluations

We used mAP@IoU=0.5 (IoU: Intersection over Union), mAP@IoU=0.5:0.95, Precision, Recall and Inference time to compare the performance of the models. IoU is a fundamental metric used to compare object detection systems [15]. The relationship between the ground truth bounding box annotated by us and the bounding box predicted by the model is examined. It is calculated by dividing the intersection sets of these two bounding boxes by their union sets (Fig. 8).

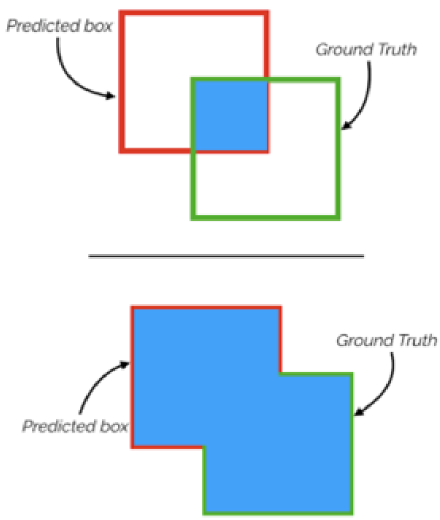

Fig. 8. *IoU calculation [8]*

It is calculated as TP (True Positive) if IoU is greater than the defined threshold value, and as FP (False Positive) if it is small. Precision, Recall and mAP performance metrics were calculated using the Eqn. 1,2 and 3 respectively with the obtained TP, TN (True Negative), FP and FN (False Negative) values. At mAP@IoU=0.5, the threshold value is 0.5, and for mAP:IoU=0.5:0.95, the threshold value has taken 10 different values between 0.5 and 0.95 in steps of 0.05.

$$Precision = \frac{TP}{TP+FP} \qquad \textbf{Eqn.1}\ [8]$$

$$Recall = \frac{TP}{TP+FN} \qquad \textbf{Eqn.2}\ [8]$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \qquad \textbf{Eqn.3}\ [8]$$

N: *Number of queries,* ***AP:*** *Average precision*

Training

We used 1934 images for training each model variation of the object detection system. In the models, we used the LR (Learning Rate) as 0.01, the photo size as 640 px (pixel), the batch size as 16, and the number of epochs as 40. We used SGD (Stochastic Gradient Descent) [17] as the optimization algorithm. We trained the models on Google Colab using the Tesla K80 12GB GPU. Table 1 shows the number of layers of the models and the number of trained parameters.

Table 1. *Information on Model Variations Used in Training*

Model	Number of Layers	Number of Parameters
YOLOv5x	476	87.178.694
YOLOv5l	392	46.600.566
YOLOv5m	308	21.037.638
YOLOv5s	224	7.053.910

Validation

For the validation of the trained model, we used a dataset of 482 images. We performed the validation of all the trained models on Google Colab using Tesla K80 12 GB GPU. As in the training phase, we used the image size as 640 px.

Testing Generalization of the Model

We used cabbage caterpillar images to test whether object detection models trained with thistle caterpillar images can be generalized to caterpillar detection. We used a total of 19 images of cabbage caterpillars encountered in September 2021 in a cabbage-growing farm in the Muratbey District of Catalca - Istanbul - Turkey. There are 25 cabbage caterpillar instances in total on these 19 images. We used Google Colab Tesla K80 12 GB GPU for object detection on the image. The image size was taken as 640 px, in which the models were trained.

RESULTS AND DISCUSSION

Results without Transfer Learning

In models trained without transfer learning, we analyzed the Precision, Recall, mAP@IoU:0.5 and mAP@IoU:0.5:0.95 metrics as a result of the training. Fig. 9 shows the results of four different model variations. Since the training of the models was started from scratch, it is seen that there are fluctuations in all model metrics, especially during the first 20 epochs of training. We think that these fluctuations can be reduced by increasing the number of epochs. We also observed strong decreases in mAP values in YOLOv5x and YOLOv5l models after the 20th epoch. This is because these models require longer epoch numbers or more data as they are more complex.

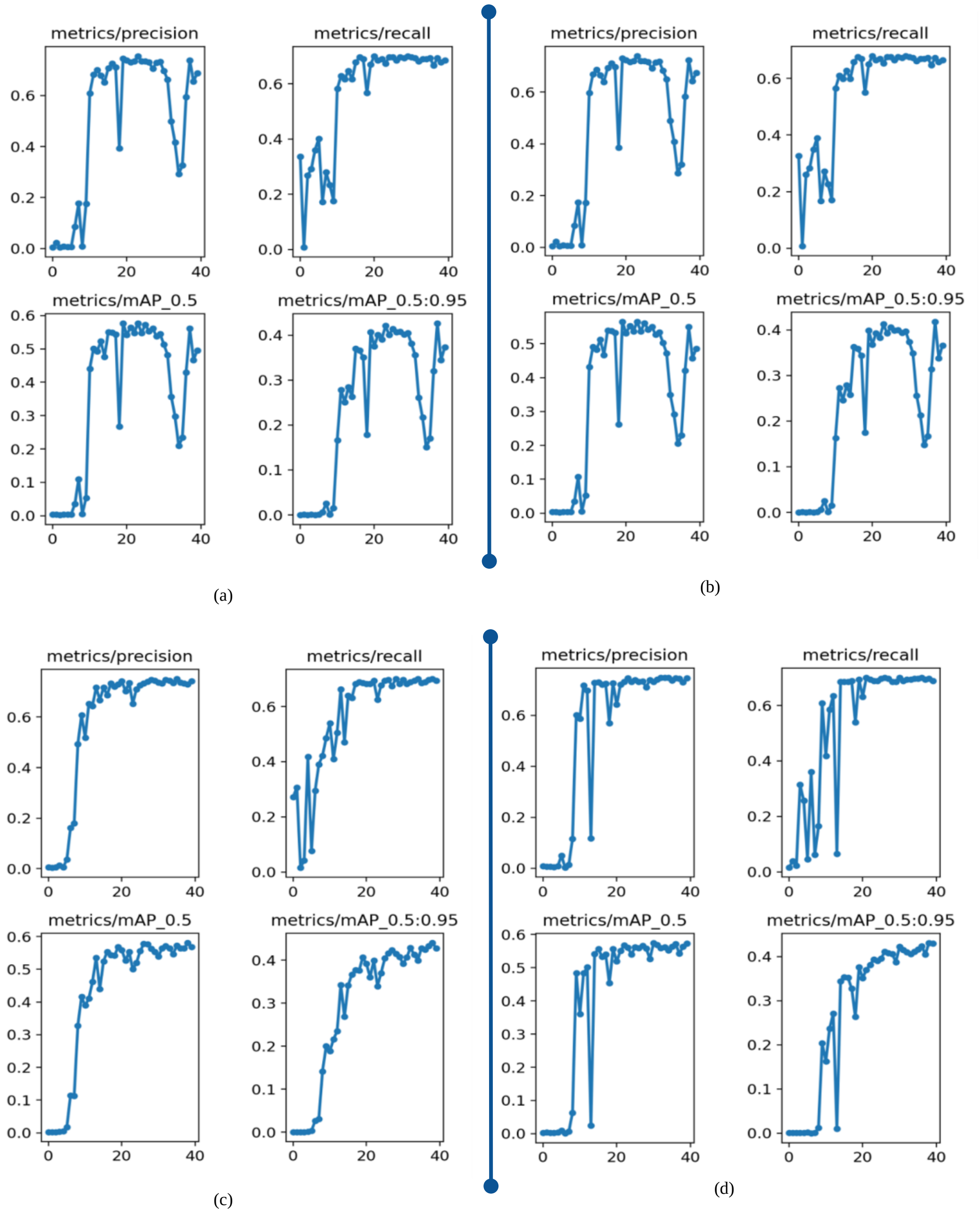

Fig. 9. *Precision, Recall, mAP@IoU:0.5 and mAP@IoU:0.5:0.95 results without transfer learning. (a)YOLOv5x, (b)YOLOv5l, (c)YOLOv5m, (d)YOLOv5s*

After the training of the models was completed, we tried the versions with the best results on the validation dataset. Validation dataset consists of 482 images. Trial results can be seen in Table 2. The highest model size belongs to YOLOv5x as expected. The highest mAP values were obtained in YOLOv5s. The fastest detection time belongs to YOLOv5s with 18.3ms.

***Table 2.** Validation dataset results without transfer learning*

Model	Precision	Recall	mAP @IoU :0.5	mAP @IoU :0.5:0.95	Inference Time	FPS	Model Size
YOLOv5x	0.306	0.574	0.208	0.123	131.8ms	7.5	116.9MB
YOLOv5l	0.428	0.554	0.304	0.231	71.3ms	14	89.3MB
YOLOv5m	0.563	0.613	0.437	0.271	44.5ms	22	40.4MB
YOLOv5s	0.629	0.617	0.482	0.285	18.3ms	55	13.7MB

***MB:** Megabyte, **ms:** millisecond, **FPS:** frame per second*

Results with Transfer Learning

In models trained using transfer learning, we analyzed the Precision, Recall, mAP@IoU:0.5 and mAP@IoU:0.5:0.95 metrics as a result of the training. Fig. 10 shows the results of four different model variations. When we compare the training results using transfer learning with the training results without transfer learning, the first thing that stands out is that the transfer learning models have much more stable metric measurements and converge to better results faster. In particular, the fact that the YOLOv5x and YOLOv5l models remained in an upward trend in mAP measurements indicates that training should be continued with more epochs.

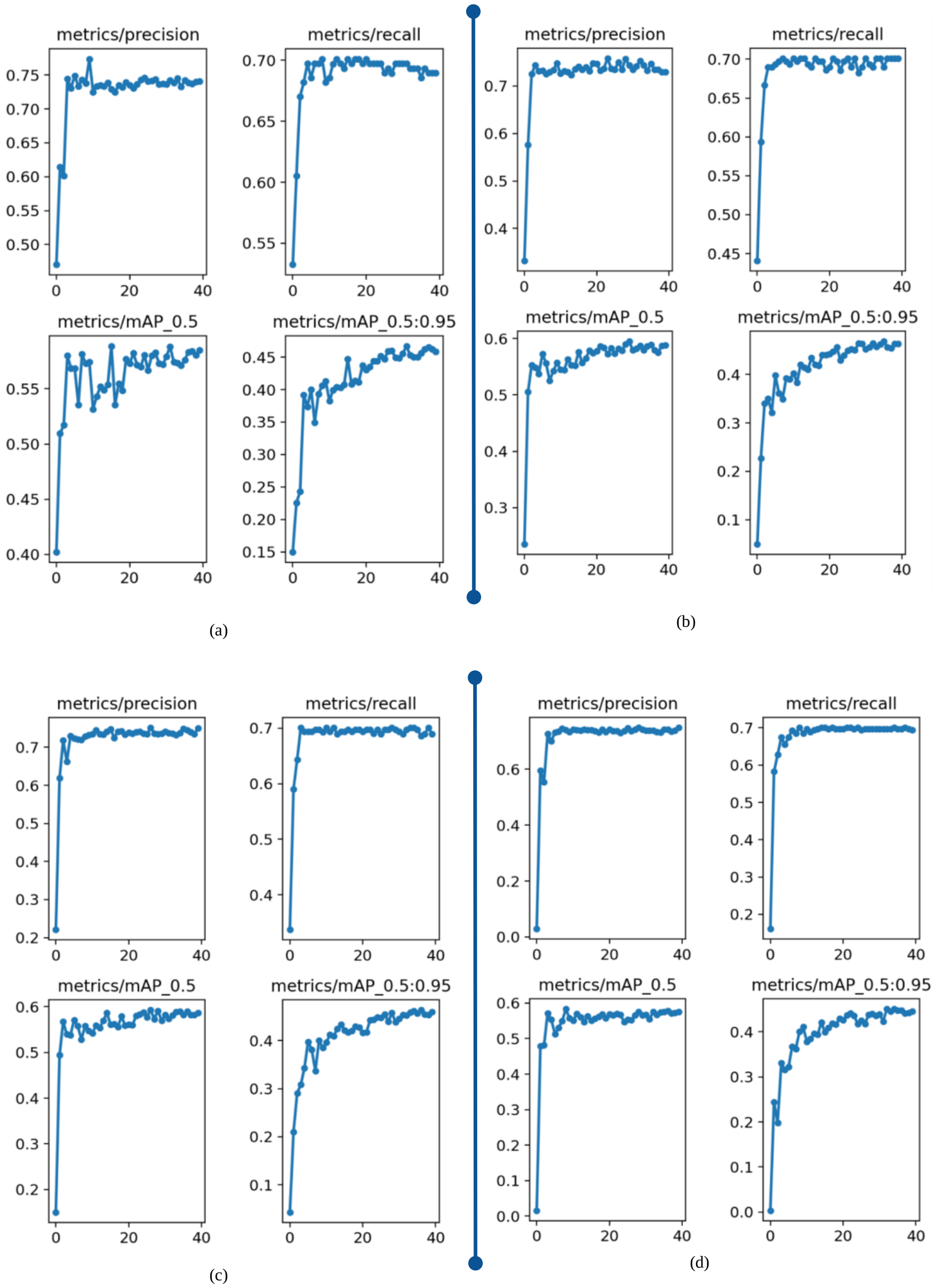

Fig. 10. *Precision, Recall, mAP@IoU:0.5 and mAP@IoU:0.5:0.95 results with transfer learning. (a)YOLOv5x, (b)YOLOv5l, (c)YOLOv5m, (d)YOLOv5s*

After the training of the models was completed, we tried the versions with the best results on the validation dataset. Validation dataset consists of 482 images. Trial results can be seen in Table 3. The highest model size belongs to YOLOv5x as expected. The highest mAP values were obtained in YOLOv5m. The YOLOv5m's detection time is close to real-time detection, with 43.4ms. The fastest detection time belongs to YOLOv5s with 15.3ms.

Table 3. *Validation dataset results with transfer learning*

Models	Precision	Recall	mAP @IoU :0.5	mAP @IoU :0.5:0.95	Inference Time	FPS	Model Size
YOLOv5x	0.743	0.697	0.588	0.464	132.9ms	8	116.9MB
YOLOv5l	0.747	0.690	0.589	0.463	70.3ms	14	89.3MB
YOLOv5m	0.746	0.686	0.591	0.464	43.4ms	23	40.4MB
YOLOv5s	0.740	0.697	0.575	0.451	15.3ms	65	13.7MB

Testing Model as General Caterpillar Detector

Among the models trained and validated with and without transfer learning, the most successful one was the YOLOv5m model, which we trained with transfer learning. This model has been examined in terms of real-time object detection with images, which we captured in a cabbage-growing farm in the Muratbey District of Catalca - Istanbul - Turkey, have not been used in training and validation classes before. Here, we aimed to test whether the model could be used as a general caterpillar detector.

We observed that the model could even detect a different caterpillar species under different background, plant species and lighting conditions. It was able to detect more than one object at the same time and when light comes from different angles. We used 19 images in total for testing generalization of the model. 19 of the 25 caterpillars in these images were detected. The object detection system we designed made 3 FP, 6 FN detections.

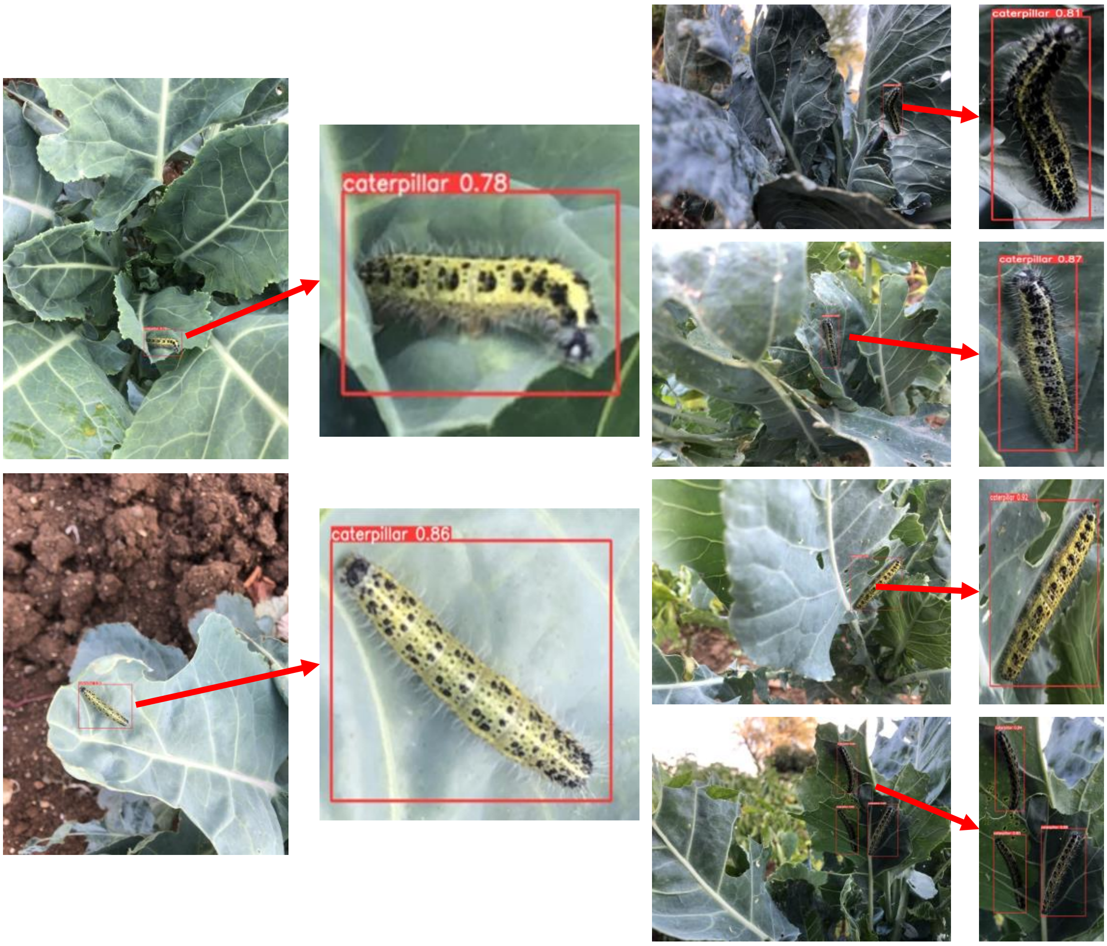

Fig. 11. *Detection results of YOLOv5m (trained with transfer learning) on different caterpillar species*

Discussion

Fig. 11 shows that our object detection system can be used even in different plant and caterpillar species from the training and validation set of YOLOv5m, one of the models trained with transfer learning.

As we can see in Fig. 12, the problems encountered in object detection in the developed model can be divided into three parts. These problems are the inability to detect the object due to blur and occlusion of the object on the image, and the examples that are incorrectly detected due to the similarity of the shape. We used 2416 images for training and validation in the study. We found that models trained in 40 epochs, 16 batches, and 640 px image sizes using the YOLOv5 architecture were able to achieve high mAP (59%) at real-time inference speeds. These values obtained are comparable to studies which are using two-stage object detection architectures, whose detection accuracy is higher than the one-stage architectures we use [8]. Increasing the images similar to the object detection errors we encounter in the dataset, as well as using the data augmentation methods such as image blur will increase the object detection accuracy. It has been observed that mAP can be increased with appropriate data augmentation methods in other YOLO based studies [7].

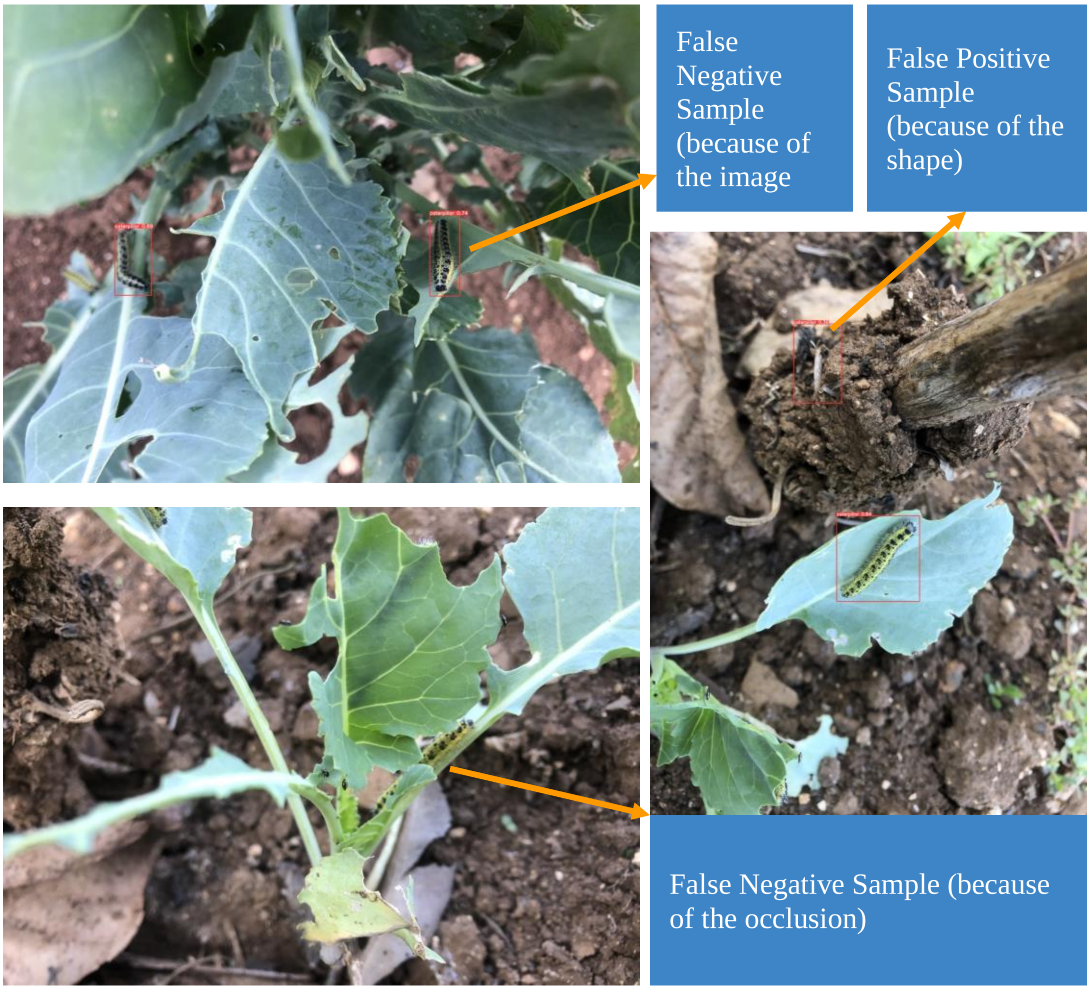

Fig. 12. *Common problems we encounter in our object detection system*

CONCLUSIONS

In this study, we have developed an object detection system that can detect caterpillars over digital images/videos by using the YOLOv5 single-stage object detection architecture. We used a public dataset consisting of images taken in realistic outdoor environments. We performed the training, validating and testing of the system on Google Colab using Tesla K80 12GB GPU.

We reached 65 FPS detection speed and maximum 59% mAP values with the parameters we trained. Moreover, our object detection system successfully detected different species of caterpillar. The accuracy of the object detection system can be improved by increasing the dataset size with new images added and data augmentation methods. Also, by adding the images of the caterpillar larvae to the dataset, it can be studied to detect the caterpillars in earlier periods. In addition, if different caterpillar species are divided into subgroups and annotated based on subgroup names, a multiclass detection model to detect caterpillar species can be developed.

REFERENCES

[1] Assis F A, Moraes J A, Assis G A, Parolin F J T. (2015): Induction of Caterpillar Resistance in Sunflower Using Silicon and Acibenzolar-S-Methyl. JAST. 17 (3) :543-550

[2] Cloyd, R. (2001): Caterpillar pests. Retrieved October 4, 2021, from Greenhouse Product News website:https://gpnmag.com/article/caterpillar-pests/

[3] FAO. (2019): FAO - News Article: New Standards to curb the global spread of plant pests and diseases. Retrieved October 4, 2021, from fao.org website: http://www.fao.org/news/story/en/item/1187738/icode/.

[4] Folnovic, T. (2021): Yield losses due to pests. Retrieved October 4, 2021, from Agrivi Blog website: https://blog.agrivi.com/post/yield-losses-due-to-pests

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015): Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[6] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L. (2014): Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014* (pp. 740–755). Cham: Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-10602-1_48

[7] Lippi, M., Bonucci, N., Carpio, R. F., Contarini, M., Speranza, S., & Gasparri, A. (2021): A yolo-based pest detection system for precision agriculture. *2021 29th Mediterranean Conference on Control and Automation (MED)*. IEEE. Retrieved from http://dx.doi.org/10.1109/med51440.2021.9480344

[8] Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., & Liu, W. (2019): PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access*, *7*, 45301–45312. https://doi.org/10.1109/access.2019.2909522

[9] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018): Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. Retrieved from http://dx.doi.org/10.1109/cvpr.2018.00913

[10] Mols, C. M. M., & Visser, M. E. (2007): Great Tits (Parus major) Reduce Caterpillar Damage in Commercial Apple Orchards. *PLoS ONE*, *2*(2), e202. https://doi.org/10.1371/journal.pone.0000202

[11] Murthy, C. B., Hashmi, M. F., Bokde, N. D., & Geem, Z. W. (2020): Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—a comprehensive review. *Applied Sciences*, *10*(9), 3280. https://doi.org/10.3390/app10093280

[12] Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2021): Recent advances in image processing techniques for automated leaf pest and disease recognition–A review.*Information Processing in Agriculture*, *8*(1), 27–51. https://doi.org/10.1016/j.inpa.2020.04.004

[13] Özkan, C. (n.d.): Course: Bitki Zararlıları İle Savaşım Yöntem ve ilaçları. Retrieved October 4, 2021, from Ankara Üniversitesi website: https://acikders.ankara.edu.tr/course/view.php?id=870

[14] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016): You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Retrieved from http://dx.doi.org/10.1109/cvpr.2016.91

[15] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019): Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Retrieved from http://dx.doi.org/10.1109/cvpr.2019.00075

[16] Rodriguez, D. (2019): Smart spot spray system saves money and reduces over-application. Retrieved October 4, 2021, from Trimble Agriculture website: https://agriculture.trimble.com/blog/smart-spot-spray-system-saves-money-and-reduces-over-application

[17] Ruder, S. (2016): An overview of gradient descent optimization algorithms. *Sebastian Ruder*. Retrieved from http://sebastianruder.com/optimizing-gradient-descent/index.html

[18] Saxena, Dr. R. R. (2020): Pest_Identification. Retrieved October 4, 2021, from Kaggle website: https://www.kaggle.com/drravirsaxena/pest-identification

[19] Soviany, P., & Ionescu, R. T. (2018): Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction. *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. Retrieved from http://dx.doi.org/10.1109/synasc.2018.00041

[20] Sulvai, F., Chaúque, B. J. M., & Macuvele, D. L. P. (2016): Intercropping of lettuce and onion controls caterpillar thread, Agrotis ípsilon major insect pest of lettuce. *Chemical and Biological Technologies in Agriculture*, *3*(1). https://doi.org/10.1186/s40538-016-0079-z

[21] Tan, M., Pang, R., & Le, Q. V. (2020): EfficientDet: Scalable and efficient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Retrieved from http://dx.doi.org/10.1109/cvpr42600.2020.01079

[22] UC Statewide IPM Program. (2014): Caterpillars - Pests of home, garden, landscape, and turf. Retrieved October 4, 2021, from UC Statewide IPM Program website: http://ipm.ucanr.edu/PMG/invertebrates/menu.caterpillars.html

[23] Ultralytics. (2021): Train Custom Data. Retrieved October 4, 2021, from Ultralytics Github website: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

[24] Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020): CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. Retrieved from http://dx.doi.org/10.1109/cvprw50498.2020.00203

[25] Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021): A forest fire detection system based on ensemble learning. *Forests*, *12*(2), 217. https://doi.org/10.3390/f12020217

[26] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., He, Q. (2021): A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109*(1), 43–76. https://doi.org/10.1109/jproc.2020.3004555