Multimedia Tools and Applications (2024) 83:27921–27934
https://doi.org/10.1007/s11042-023-16578-1

Improving the generalization capability of YOLOv5 on remote sensed insect trap images with data augmentation

Jozsef Suto[1,2]

Received: 24 March 2023 / Revised: 6 July 2023 / Accepted: 18 August 2023 /
Published online: 28 August 2023

Abstract

In agricultural pest management, the traditional insect population tracking in the case of several insect types is based on outsourced sticky paper traps that are checked periodically by a human operator. However, with the aid of the Internet of Things technology and machine learning, this type of manual monitoring can be automated. Even though great progress has been made in the field of insect pest detector models, the lack of sufficient amount of remote sensed trap images prevents their practical application. Beyond the lack of sufficient data, another issue is the large discrepancy between manually taken and remote sensed trap images (different illumination, quality, background, etc.). In order to improve those problems, this paper proposes three previously unused data augmentation approaches (gamma correction, bilateral filtering, and bit-plate slicing) which artificially enrich the training data and through this increase the generalization capability of deep object detectors on remote sensed trap images. Even with the application of the widely used geometric and texture-based augmentation techniques, the proposed methods can further increase the efficiency of object detector models. To demonstrate their efficiency, we used the Faster Region-based Convolutional Neural Network (R-CNN) and the You Look Only Once version 5 (YOLOv5) object detectors which have been trained on a small set of high-resolution, manually taken trap images while the test set consists of remote sensed images. The experimental results showed that the mean average precision (mAP) of the reference models significantly improved while in some cases their counting error was reduced to a third.

Keywords Insect detection · YOLOv5 · Faster R-CNN · Remote sensing · Data augmentation

✉ Jozsef Suto
suto.jozsef@inf.unideb.hu

[1] Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen, Kassai Street, 26, 4028 Debrecen, Hungary

[2] Eközig Zrt, Köntösgát sor, 1-3, 4031 Debrecen, Hungary

1 Introduction

A general goal of vegetable and fruit growers is to achieve high crop yield. Since crop yields are strongly affected by insect pests which may damage crops, farmers use insecticides against them at scheduled times without taking into consideration the size of pest population [1]. Spraying is the main control strategy against several insect pests. As an example, approximately 70% of insecticide treatments applied against codling moth in apple orchards [2]. Instead of the periodical spraying, a more optimized solution would be to use insecticides only when the insect pest population size exceeded the economic threshold. To realize such a spraying strategy, an exact pest population forecast is necessary. Such a forecast has not only a significant environmental (e.g. less amount of insecticides) but also economic effects (e.g. saving money, manpower, etc.) because growers can apply insecticides at the right time to defend their crops.

To acquire quantitative information for pest density prediction, different types of traps can be used such as light and pheromone-based [3, 4]. In the case of the pheromone traps (this article focuses on that), the pheromone substance attracts male insects to the trap and when the pest enters the trap it remains stuck on the sticky paper. Sticky papers are then periodically changed and inspected by an expert who counts the number of insects found on them. This type of manual or "conventional" insect monitoring has several well-known disadvantages (e.g. requires a skilled person, time consuming, expensive, etc.) which have been mentioned in several articles [5–7]. In addition, the manual insect counting does not provide continuous feedback therefore the insect pest population monitoring has a low temporal resolution. However, the temporal resolution is significant because if the number of cached insects cannot be obtained in time, it is impossible to take quick intervention [8]

Due to the drawback of manual insect counting, researchers and their industrial partners turned towards smart solutions. Recently, several embedded system-based automatized traps, or Internet of Things (IoT) systems (edge devices plus the server side) have been developed with the support of machine learning for insect counting [9, 10]. Some of them provide real-time data while others provide off-line data for more precise treatments and interventions. In this article we will refer to the image capture device which is inside the trap as sensing device.

Beyond the remote sensing devices, an accurate insect-counting method is also needed. Since insect counting can be seen as a special object detection problem, researchers realized that the state-of-the-art one or two stages deep object detectors could be used efficiently for this task [6, 11, 12]. Zhong et al. [1] were among the first researchers to apply the You Only Look Once (YOLO) two-stages object detector model. In their work the first version of YOLO played an object proposal role while the object classifier was a Support Vector Machine (SVM). Their reason for this unusual model pairing was the relatively small dataset that was available for them. With this approach, they measured 92.5% counting accuracy (correctly detected objects per all objects) on their test images. Later, Hong et al. [13] investigated the accuracy and inference time of more deep object detectors including Faster Region-based Convolutional Neural Network (R-CNN) and the Single Shot Multibox Detector (SSD) on manually collected sticky trap images. Beyond the trap images, the authors added 168 photos to their dataset to increase the number and type of negative samples in the "unknown" class. Not surprisingly, their investigation showed that the Faster R-CNN model had the highest (90.25%) mean average precision (mAP) and the longest decision time while the SSD detector was the fastest, but its mAP was only 76.86%. The authors also mentioned that the object detector needs be updated with remote sensed trap images which include various environmental light effects to be robust. Li et al. [14] compared the Faster R-CNN, Mask R-CNN and YOLOv5 on some selected insect categories of the Baidu AI insect detection and IP102 datasets. Their experimental results showed that the YOLOv5 is the recommended model in the case of the Baidu AI insect detection dataset because its accuracy was above 99% while Faster R-CNN's and Mask R-CNN's reached approximately 98%. They explained this result with the homogeneous background of Baidu AI images.

Even though the earlier results are very encouraging, in most cases the images were not remote sensed trap images. For illustration, Fig. 1 shows a manually taken sticky paper image and a remote sensed trap image where the differences are clearly visible.

The lack of remote sensed trap images is a general issue in the field of insect pest counting [10, 15]. For example, Li et al. [14] noted that the weakness of their model lies in the lack of high adaptability because the images used for model training are significantly different from trap images taken by a remote sensing device. Cardoso et al. [16] claimed that there are significant differences between manually taken trap images in controlled environment and remote sensed images. For a bigger training dataset, Diller et al. [17] added manually taken images of sticky papers to their remote sensed images. Their laboratory environment consisted of the trap's house, a camera and the yellow sticky-board populated with insects. With this approach, several hundred images have been generated where images were collected under controlled conditions, with a significant contrast between the background and the target insect.

In order to artificially increase the training set size, researchers turned to data augmentation techniques for help. Data augmentation has been an important component of the learning chain especially in those cases where the available training dataset is limited like in insect pest detection. Year by year, more data augmentation approaches appear in the practice. Many of them also have been used in pest detection related articles [18]. Several articles can be mentioned where the authors applied image rotation, scaling, flip, translation, brightness adjustment, and noise pollution on the training images [1, 4, 17, 19, 20]. The authors of the [21] review categorized data augmentation techniques into five classes including geometric transformations, noise injection, color space transformations, over sampling, and Generative Adversarial Network-based augmentation. Since moths are caught in different poses, the insect counter model needs to be rotation invariant. Moreover, to handle size differences the model also needs to be scale invariant. The geometric augmentation methods help to handle pose discrepancies of caught insects. On the other hand, photometric augmentation such as brightness and contrast adjustment help to handle

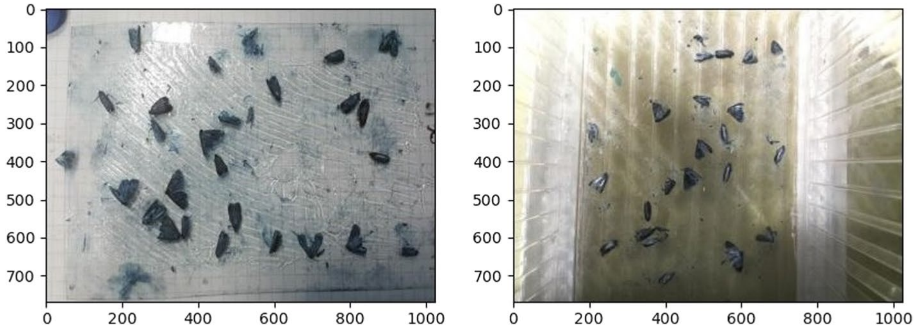


Fig. 1 Manually taken sticky paper trap image (left) and remote sensed trap image (right)

the texture difference of insects. Beyond the above-mentioned techniques, there are some additional augmentation strategies such as random erase, mixup or mosaic augmentation that are also taken into consideration in modern object detectors [22]. From the insect detection point of view, mosaic augmentation is especially useful because it helps to better handle the well-known “small object detection problem”. Its idea is to compose a new training image from four other images in specific ratios.

The above-mentioned data augmentation techniques try to handle scale invariance, texture differences, and “the small object detection” problems. However, they do not handle well the significant quality and illumination discrepancies between manually taken (high-resolution) and remote sensed trap images. To “attenuate” those differences we introduce three new data augmentation approaches. Namely, gamma correction, image compression, and bilateral image filtering-based data augmentations. The efficiency of the proposed augmentation approaches is demonstrated on the YOLOv5 and Faster R-CNN (with ResNet50 backbone) models which are trained on manually taken high resolution trap images and tested on remote sensed trap images. The experimental results clearly show that the performance of both models improve significantly if the proposed augmentation techniques are also used for data enrichment. In the case of the YOLOv5 had a spectacular improvement where the gamma correction-based augmentation approach increased the mean average precision (mAP) from 0.887 to 0.934 while its counting error decreased from 3.29 to 1.07.

2 Materials and methods

2.1 Object detector

Automated insect pest counting can be seen as object detection problem which is an outstanding subject in computer vision. Most object detection problems involve detecting visual object categories like faces, humans, vehicles, etc. For those tasks, we can apply more possible detector algorithms that belong into three categories: traditional computer vision-based like the Viola-Jones [23], two-stage deep learning-based like the Fast and Faster R-CNN [24, 25], and the single-stage deep learning-based methods like the members of the YOLO model family. Since the appearance of R-CNN (in 2014), the CNN-based object detection has started to evolve at an unprecedented rate [26]. In the next years several single and multi-stage deep object detector models have been developed. At now, the lates YOLO models are considered as the state of the art due to their fast inference and accurate localization capabilities. In this paper we used the YOLOv5 as object detector. Those fasts motivated us to use the Faster R-CNN and the YOLOv5 object detectors in this work as reference models.

The YOLOv5 has been released on GitHub by Glen Jocker (Ultralytics) in 2020. YOLOv5 offers more object detector architectures including the YOLOv5s (small), YOLO5m (medium), YOLOv5l (large) and YOLOv5x (extra-large) models. All of them are pre-trained on the Microsoft COCO dataset [27]. The main difference between the models is the number of convolutional layers. Since insect pest counting takes place in the sensing device in some systems, the inference time (due to the battery lifetime) and the computational resource requirement (e.g. available physical memory) is a critical factor. Therefore, we have chosen the YOLOv5s as object detector which is the smallest member of the YOLO family after the nano architecture.

2.2 Data augmentation

In machine learning, a generally accepted fact is that the (relevant) data growth contributes to the validation performance increase of the machine learning model. In addition, deep models demand a huge amount of data to fine-tune weights. Therefore, data augmentation or in other words artificial data enrichment is an important component of the learning chain because it helps to extend the available training dataset. Its importance increases even more in those cases where the available training dataset is limited, like in insect pest detection.

Remote sensed trap images acquired in the fields are affected by a wide variety of illumination conditions due to day-cycle light, weather conditions, and landscape elements that cause shadows [28]. In order to attenuate the illumination differences between the manually taken high resolution images and remote sensed images we used gamma correction (also called as power law transform). Gamma correction transforms the input image pixelwise according to formula (1) where *f(x,y)* is the scaled input pixel (range from 0 to 1) at coordinate *(x, y)*, and c is the gain. Both c and γ are positive numbers.

$$f^{*}(x,y) = cf(x,y)^{\gamma} \tag{1}$$

Remote image capturing is also affected by oscillations due to the wind, which may result worse image quality due to motion blur. In addition, there is a significant image quality difference between remote sensed and manually taken images due to the different spectrum sensitivity, field of view, focusing, etc. of cameras. Those models that use high quality images for training tend to achieve worse results since they cannot deal with such variability [15]. To try to compensate the image quality difference, we introduced bit-plane slicing and image smoothing as two additional augmentation techniques. In the RGB color representation the value of all channels is stored in 8-bit. This 8-bit can be considered as eight 1-bit planes where the lower order planes (least significant bit positions) carry the subtle intensity details of the image. Decomposing an image into bit-planes useful for investigating the importance of all bit positions. Leaving the least significant bit positions (LSB) of the original representation can be seen as image compression where we keep the “main features” of the image while the fine details will be removed. Generally, removing the content of the last k LSB positions will not degrade the appearance of the original image significantly.

To mimic the possible motion blur, the manually taken images have been smoothed. Blurring removes fine details from the original image which can be beneficial in the case of high-resolution images because too much detail may lead to over segmentation [15]. The degree of blurring is determined by the size of the kernel and the values inside it. However, using a simple averaging or Gaussian kernel strongly damages edges. Therefore, we applied bilateral filtering where the kernel takes into consideration not just the spatial but also the intensity relationship between neighbor pixels. Both relationships are modelled by Gaussian distributions. Changing their standard deviation controls the smoothing effect of the filter. More information about the bilateral filtering can be found in [29].

2.3 Evaluation metrics

In insect pest detection and counting, an important question is: how to measure the accuracy of the algorithm? In 2016, the automatized insect counting with deep object detectors was a relatively new research field and there was no standard protocol for the evaluation of insect counting algorithms [30]. Therefore, researchers adopted metrics from other fields of computer vision such as pedestrian detection. In computer vision a generally accepted performance metric of object detectors is the mAP. It is equal to the average precision (AP) metric across all classes in the dataset where AP is equal to the area under the precision-recall curve.

To construct the precision-recall curve, we need to have information about the true and false detections (proposed bounding boxes). The correctness of a detection can judged with the Intersection-over-Union (IoU). IoU is a ratio of the overlapping area between the ground truth bounding box and the predicted bounding box and the area of their union. In most studies if the IoU value is equal or higher than 0.5 the proposed bounding box is considered as true positive otherwise the proposed box is false positive [19, 28, 31]. In this paper we also used the mAP as the primary metric with 0.5 IoU threshold value to analyze the performance change of the YOLO model.

Beyond mAP, some other metrics are also available to better describe the counting method's performance [10]. In this work, we used a simple error function (2) over all test images (N) where $\boldsymbol{c^p}$ is the predicted number of caught insects (for all test images) while $\boldsymbol{c^r}$ is the real number of insects (ground truth boxes). The (2) formula can be seen as the average counting error of the algorithm. Although this can be positively influenced by the same number of false negative and false positive detection, and it does not give any information about the localization accuracy of the insect counter model, but it is a much clearer indicator for the final user.

$$e(\boldsymbol{c}^p, \boldsymbol{c}^r) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left| c_i^p - c_i^r \right| \tag{2}$$

3 Results and discussion

3.1 Model settings

In this work, we used the freely available Faster R-CNN (two-stages) and the YOLOv5s (single-stage) object detector models to investigate the efficiency of the proposed data augmentation methods. Both have been trained with stochastic gradient descent (SGD) method where the minibatch size was 16 (aligned to the available GPU memory), the momentum was 0.9 while the weight decay was 0.0001. The initial learning rate was set to be 0.001. As stop condition, we applied the "no improvement in 20 epochs". The maximum number of epochs has been set to 500 in the case of YOLO while it was 100 in the case of Faster R-CNN. The experiments have been executed on a notebook with AMD Ryzen 9 5900HX 3.3GHz processor, 24 GB physical memory and Nvidia GeForce RTX 3060 GPU.

3.2 Manually taken and remote sensed trap images

The core of our training dataset was 175 manually taken, high resolution trap images. Those images have been acquired by the Eközig Company, Debrecen, Hungary. The image capture circumstances were different. A few of them have been taken indoor while others are in the field. The image capture camera was also not uniform. In most cases, the image capture device was a smartphone camera, but professional cameras also have been used.

The number of remote sensed test images is 36. In this case, the data capture framework was the same. All the test images have been taken with a particular sensing device dedicated to remote insect pest monitoring. It consists of a plug-in board, a Raspberry

Pi Zero W, and a Raspberry Pi Camera v2. A sample image about the sensing device can be seen in Fig. 2. In brief, the sensing device turns on every day during the observation period at 10:00 o'clock for sufficient light and lower daily temperature. At start up, the controller software synchronizes the real-time clock (RTC) and generate a timestamp according to it. Setting the time accurately is important to track the date of capture. After that the Raspberry Pi controller unit takes an image of sticky paper (the paper is located at the bottom of the trap house) which can be transmitted toward the server side through the Long-Term Evolution (LTE) network. At the end, the controller software sets up the next "wake up" time and starts the shutdown process. Additional details about the sensing device can be found in [32].

3.3 Data augmentation with the proposed methods

At first the original 175 manually taken training images have been augmented with gamma correction. In the case of $\gamma < 1$, the (1) mapping function will transform a narrow range of small intensities into a wider range of output intensities. On the other hand, the effect will be the opposite if $\gamma > 1$. Since we need both approaches, the used γ interval starts from 0.4 and goes until 2.8 with 0.2 step size (except 1.0). The visualization of the mapping functions can be seen on Fig. 3.

Gamma correction working on a single bit plane so it can be applied independently on the different channels of a colour image. Since, the RGB colour histogram of the trap images showed similar intensity distributions in all channels, we applied the same mapping function for each channel. The effect of gamma correction on a sample trap image can be seen on Fig. 4. The gamma correction-based data augmentation produced 2100 new training sample from the initial 175 trap images.

In the second step, the bit-plane-based data augmentation has been performed on the original 175 trap images. As can be seen on Fig. 5, even the two MSB bit positions carry

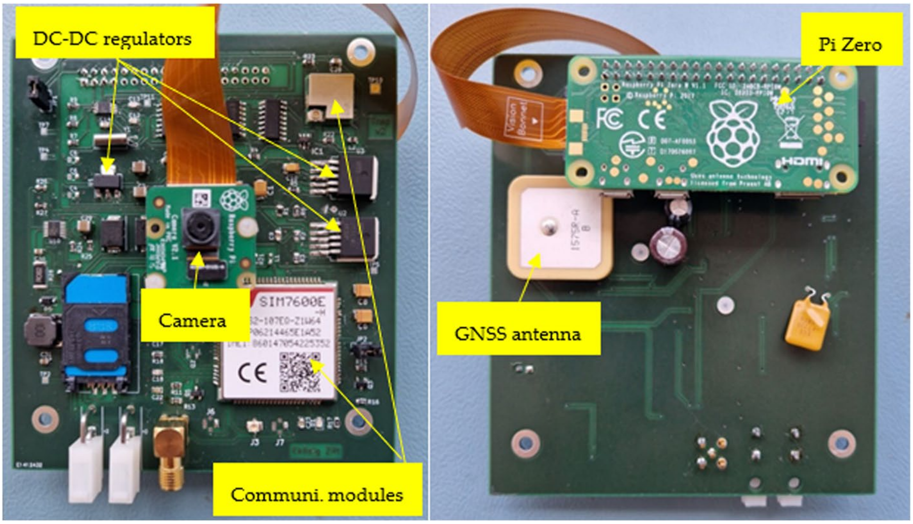


Fig. 2 Front (left) and back (right) sides of the remote sensing device

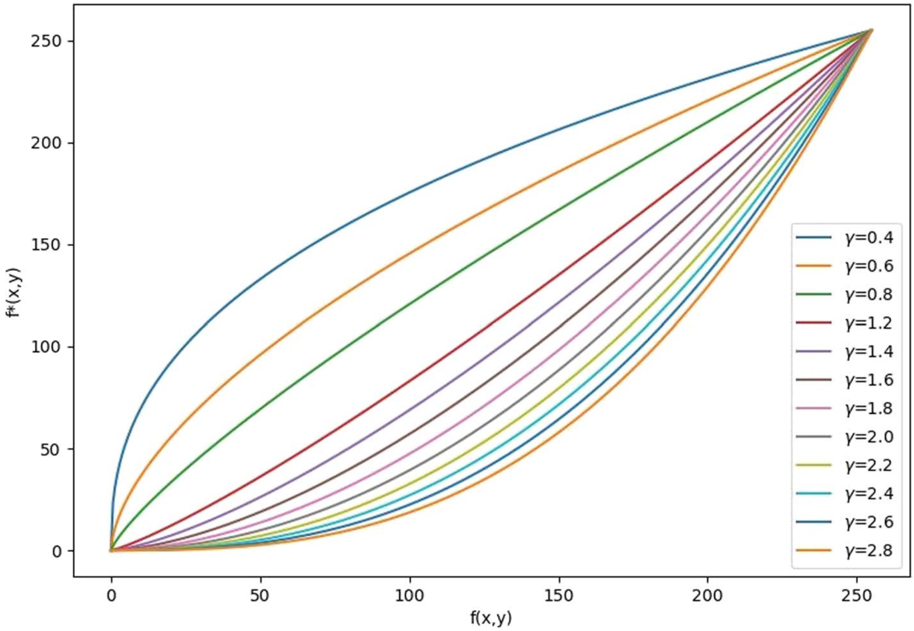


Fig. 3 Mapping functions of gamma correction

a huge amount of information about the content of the image. Taking into consideration additional least significant bit positions the difference between the original and the bit reduced image is almost invisible to the human eye. With this type of augmentation 1050 new training sample have been generated from the original trap images.

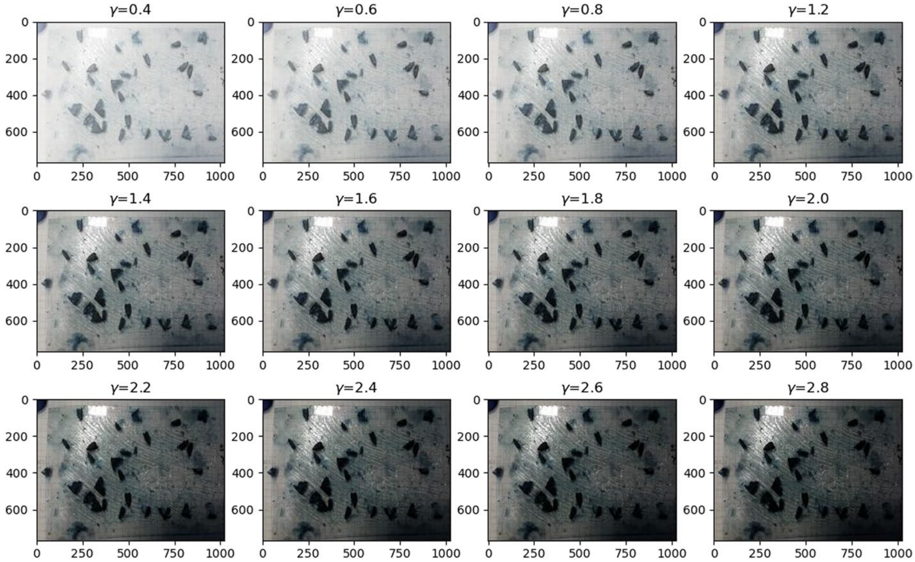


Fig. 4 The effect of gamma correction on a sample image

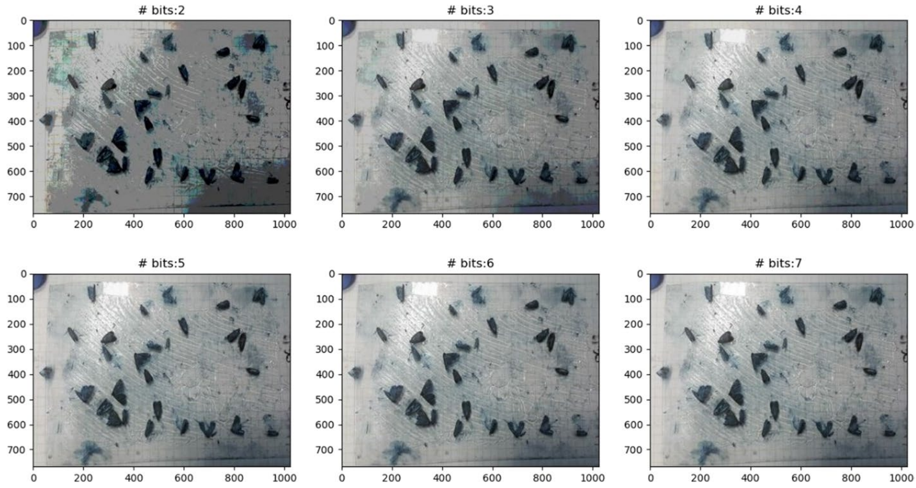


Fig. 5 Bit reduced versions of a sample image

The last data augmentation has been performed with the bilateral filtering. The bilateral kernel has three parameters: kernel size, standard deviation of the spatial distribution (σ_s), and the standard deviation of the intensity distribution (σ_c). The value of those parameters is application dependent. For simplicity, we tried to minimalize the number of parameter combinations. According to our preliminary investigations, the used kernel size was fix 31×31 pixels while $\sigma_s \in \{5, 10\}$ and $\sigma_c \in \{50, 100, 150\}$. The effect of bilateral filtering on a sample image can be seen on Fig. 6.

The effect of the above-described data augmentation methods on the YOLOv5's and Faster R-CNN's mAP and loss (2) are summarized in Tables 1 and 2 respectively. As references, both models also have been trained on the original manually taken trap images without the proposed augmentation methods. It is worth mentioning again that the model's training chain already incorporates geometric and photometric augmentations. When one of the proposed data augmentations is used (in addition to the original trap images) the newly generated images were also a part of the training set.

The results in Tables 1 and 2 show that, all proposed image augmentation techniques increase the model's mAP value and decrease (or do not modify) their counting error. Out of the three augmentations techniques, the gamma correction-based was the most efficient for both models. Although the improvement tendency for each metric was similar for both models, but the magnitude of improvement was more spectacular on the YOLOv5. The gamma correction-based augmentation increased the YOLOv5's mAP value from 0.887 to 3.29 and reduced the counting loss from 3.29 to 1.07. This is a huge improvement because it means that the model's average insect count prediction differs from the true insect count by approximately one. A visual demonstration of this high detection accuracy can be seen on Fig. 7 where all caught insects have been correctly localized and recognized in both images.

The combined use of augmentation approaches brought another interesting result. Even though the size of the training set was bigger after the combined use of the bilateral and gamma augmentations, but the mAP and the error count of the models degraded compared to the solo use of gamma correction-based augmentation. In the case of the YOLOv5, the mAP was 0.919 when all the three augmentation approaches have been

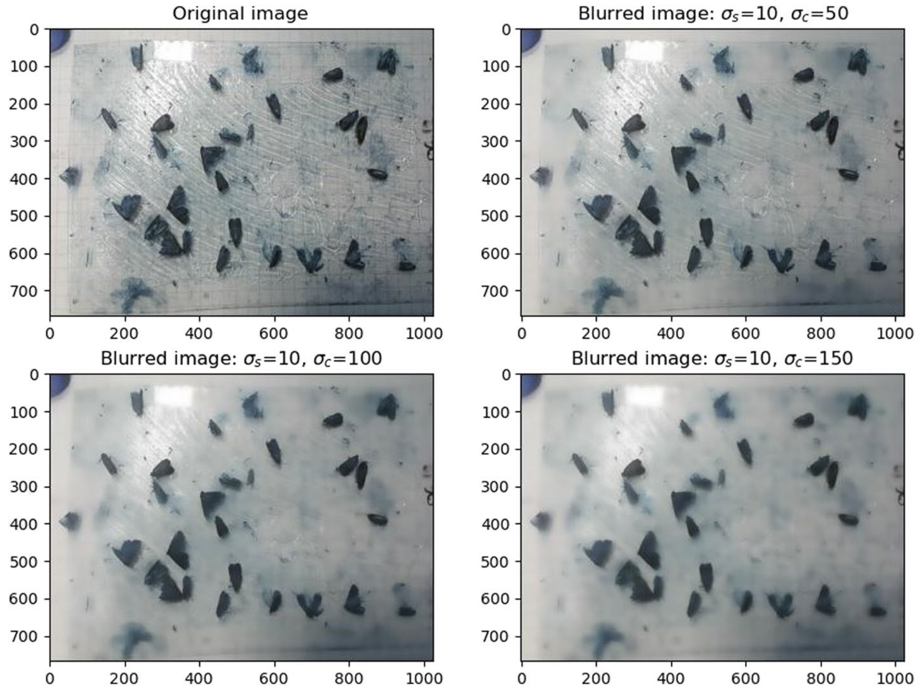

Original image
Blurred image: σ_s=10, σ_c=50
Blurred image: σ_s=10, σ_c=100
Blurred image: σ_s=10, σ_c=150
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

Fig. 6 Bilateral filtering of a trap image with three different filter parameters

applied which is significantly smaller than the mAP of the model with the gamma augmentation only. On the other hand, the counting error was only 1.0 which is the smallest out of all augmentation approaches.

Although the increased training set makes the model more restrictive against overfitting, but our results showed that the involvement of more and more data augmentation methods does not guarantee performance improvement. In the case of excessive augmentation, the original training set will be only a small portion of the augmented set where a huge number of images do not contain useful information, but they may add extra noise. Finally, we can also observe a negative correlational relationship between mAP and counting loss (Fig. 8).

Table 1 The effect of the proposed data augmentation methods on the YOLOv5's performance

Augmentation method	mAP (0.5 IoU)	Counting error (2)	# training samples
-	0.887	3.29	175
Gamma correction	0.934	1.07	2275
Bit-plane slicing	0.899	2.0	1225
Bilateral filtering	0.922	1.36	1225
Bilateral + Gamma	0.92	1.5	3325
All	0.919	1.0	4375

Table 2 The effect of the proposed data augmentation methods on the Faster R-CNN's performance

Augmentation method	mAP (0.5 IoU)	Counting error (2)	# training samples
-	0.863	3.82	175
Gamma correction	0.893	2.55	2275
Bit-plane slicing	0.868	3.82	1225
Bilateral filtering	0.874	3.0	1225
Bilateral + Gamma	0.878	2.86	3325
All	0.887	2.64	4375

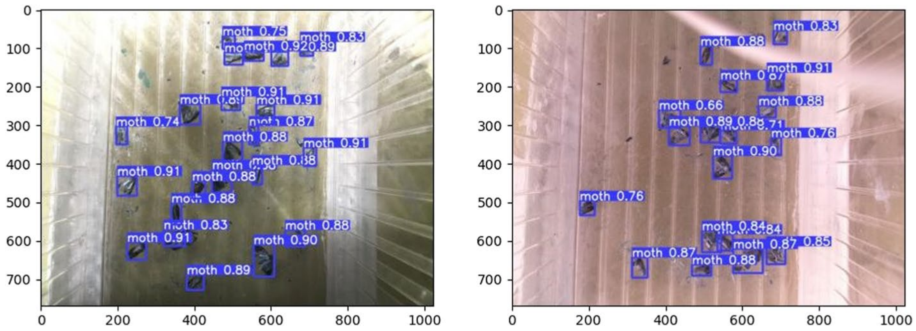
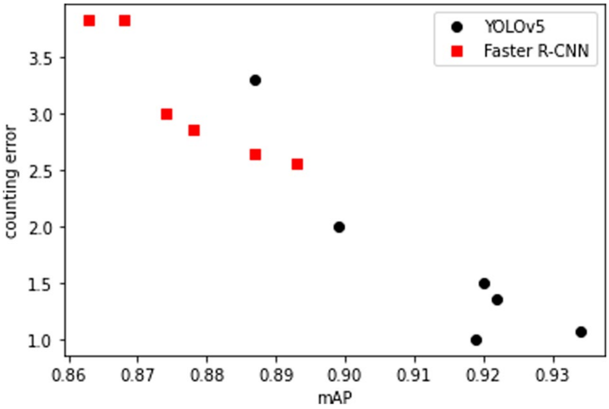


Fig. 7 Detected insects with the trained YOLOv5s on remote sensed trap images

Fig. 8 Scatter plot of mAP and counting error



4 Conclusion

In this paper we proposed three data augmentation methods to increase the training dataset and to try to attenuate the quality difference between the manually taken high resolution and the remote sensed insect trap images. To demonstrate that the proposed data augmentation approaches result further performance improvement of model's efficiency in addition to the well-known (e.g. geometric, mosaic, etc.) augmentation techniques, the YOLOv5s object detector model has been used. The change of the model's performance was measured with the mAP and the average counting error metrics. The experimental results on our trap images showed that each proposed data augmentation method increased the mAP and decreased the counting error. The most efficient augmentation approach was the gamma correction-based which increased the mAP of the model from 0.887 to 0.934 while it decreased the counting error from 3.29 to 1.07. The counting error decreased to the third which means a huge improvement. The highest mAP values were 0.934 and 0.893 with YOLOv5 and Faster R-CNN, respectively. In other similar works the authors achieved 0.886 mAP with Faster R-CNN (ResNet50) [13] or reported 0.762 and 0.812 mAP with Faster R-CNN and YOLOv5(s) [16]. Although the dataset in those articles was not the same as in our case but this comparison also indicates what high localization capability can be achieved with the proposed augmentation approaches.

Surprisingly, the combined usage of the three proposed augmentation approaches had not brought significant improvement neither in mAP nor in counting loss compared to the solo gamma-based augmentation. This observation raises questions because a generally accepted rule of thumb says that increasing the dataset size contributes to the increase of generalization capability of the model. However, this is not always the case as our results show. In our opinion, the efficiency of combined application of data augmentation techniques depends on the type of the problem and the ratio of the original and the augmented data size. Based on it, a small subset of the data augmentation techniques may achieve higher performance improvement than using many (relevant) augmentation techniques. Unfortunately, the most optimal combination of data augmentation techniques is not known in this research field. In order to get a clearer picture about it, further investigations are necessary.

Funding Open access funding provided by University of Debrecen. The author was supported by eKÖZIG Regionális Informatikai Szolgáltató Központ Zrt thanks to the project no. 2020-1.1.2-PIACI-KFI-2021-00249. Project no. 2020-1.1.2-PIACI-KFI-2021-00249 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development, and Innovation Fund, financed under the 2020-1.1.2-PIACI KFI funding scheme.

Data availability The datasets analysed during the current study are not publicly available due to the restriction of the Eközig Zrt.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI in scientific writing During the preparation of this work the author has not used AI tools.

References

1. Zhong Y, Gao J, Lei Q, Zhou Y (2018) A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors 18:1489. https://doi.org/10.3390/s18051489
2. Cirjak D, Miklecic I, Lemic D, Kos T, Zivkovic PI (2022) Automatic pest monitoring systems in apple production under changing climate conditions. Horticulturae 8:520. https://doi.org/10.3390/horticulturae8060520
3. Hoye TT, Arje J, Bjerge K, Hansen OLP, Iosifidis A, Leese F, Mann HMR, Meissner K, Melvad C, Raitoharju J (2020) Deep learning and computer vision will transform entomology. PNAS 118:e2002545117. https://doi.org/10.1073/pnas.200254511
4. Sun Y, Liu X, Yuan M, Ren L, Wang J, Chen Z (2018) Automatic in-trap pest detection using deep learning for pheromone-based Dendroctonus valens monitoring. Biosyst Eng 176:140–150. https://doi.org/10.1016/j.biosystemseng.2018.10.012
5. Muppala C, Guruviah V (2019) Machine vision detection of pests, diseases and weeds: a review. J Phytol 12:9–19
6. He Y, Zhou Z, Tian L, Liu Y, Luo X (2020) Brown rice planthopper (Nilaparvata lugens stal) detection based on deep learning. Precision Agric 21:1385–1402. https://doi.org/10.1007/s11119-020-09726-2
7. Rustia DJA, Lu CY, Chao JJ, Wu YF, Chung JY, Hsu JC, Lin TT (2021) Online semi-supervised learning applied to an automated insect pest monitoring system. Biosys Eng 208:28–44. https://doi.org/10.1016/j.biosystemseng.2021.05.006
8. Preti M, Moretti C, Scarton G, Giannotta G, Angeli S (2021) Developing a smart trap prototype equipped with camera for tortricid pests remote monitoring. Bull Insectol 74:147–160
9. Lima MCF, Leandro MEDA, Valero C, Coronel LCP, Bazzo COG (2020) Automatic detection and monitoring of insect pests - A review. Agriculture 10:161. https://doi.org/10.3390/agriculture10050161
10. Suto J (2022) Condling moth monitoring with camera-equipped automated traps: a review. Agriculture 12:1721. https://doi.org/10.3390/agriculture12101721
11. Mamdouh N, Khattab A (2021) YOLO-based deep learning framework for olive fruit fly detection and counting. IEEE Access 9:84255–84262
12. Roosjen PPJ, Kellenberger B, Kooistra L, Green DR, Fahrentrapp J (2020) Deep learning for automated detection of Drosophila suzukii: potential for UAV-based monitoring. Pest Manag Sci 76:2994–3002. https://hdl.handle.net/10863/17644. Accessed 01.03.2023
13. Hong SJ, Kim SY, Kim E, Lee CH, Lee JS, Lee DS, Bang J, Kim G (2020) Moth detection from pheromone trap images using deep learning object detectors. Agriculture 10:170. https://doi.org/10.3390/agriculture10050170
14. Li W, Zhu T, Li X, Dong J, Liu J (2022) Recommending advanced deep learning models for efficient insect pest detection. Agriculture 12:1065. https://doi.org/10.3390/agriculture12071065
15. Barbedo JGA (2020) Detecting and classifying pests in crops using proximal images and machine learning: a review. AI 1:312–328. https://doi.org/10.3390/ai1020021
16. Cardoso B, Silva C, Costa J, Ribeiro B (2022) Internet of Things meets computer vision to make an intelligent pest monitoring network. Appl Sci 12:9397. https://doi.org/10.3390/app12189397
17. Diller Y, Shamsian A, Shaked B, Altman Y, Danziger BC, Manrakhan A, Serfontein R, Bali E, Wernicke M, Egartner A, Colacci M, Sciarretta A, Chechik G, Alchanatis V, Papadopoulos NT, Nestel D (2023) A real-time remote surveillance system for fruit flies of economic importance: sensitivity and image analysis. J Pest Sci 96:611–622. https://doi.org/10.1007/s10340-022-01528-x
18. Júnior TDC, Rieder R (2020) Automatic identification of insects from digital images: A survey. Comput Electron Agric 178:105784. https://doi.org/10.1016/j.compag.2020.105784
19. Shi Z, Dang H, Liu Z, Zhou X (2020) Detection and identification of stored-grain insects using deep learning: a more efficient neural network. IEEE Access 8:163703–163714

20. Du L, Sun Y, Chen S, Feng J, Zhao Y, Yan Z, Zhang X, Bian Y (2022) A novel object detection model based on faster R-CNN for Spodoptera frugiperda according to feeding trace of corn leaves. Agriculture 12:248. https://doi.org/10.3390/agriculture12020248
21. Li W, Zheng T, Yang Z, Li M, Sun C, Yang X (2021) Classification and detection of insects from field images using deep learning for smart trap management: a systematic review. Ecol Inform 66:101460. https://doi.org/10.1016/j.ecoinf.2021.101460
22. Kaur P, Khehra BS, Mavi B (2021) Data augmentation for object detection: a review. In: IEEE international midwest symposium on circuits and systems. Lansing, pp 537–543
23. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. Kauai, pp 511–518
24. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV). Santiago, pp 1440–1448. https://doi.org/10.48550/arXiv.1504.08083
25. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39:1137–1149
26. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Columbus, pp 580–587. https://doi.org/10.48550/arXiv.1311.2524
27. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hayes J et al (2014) Microsoft COCO: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference. Zurich, pp 740–755. https://doi.org/10.48550/arXiv.1405.0312
28. Domingues T, Brandao T, Ribeiro R, Ferreira JC (2022) Insect detection in stick trap images of tomato crops using machine learning. Agriculture 12:1967. https://doi.org/10.3390/agriculture12111967
29. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Sixth international conference on computer vision. Bombay, pp 839–846
30. Ding W, Taylor G (2016) Automatic moth detection from trap images for pest management. Comput Electron Agric 123:17–28. https://doi.org/10.1016/j.compag.2016.02.003
31. Suto J (2021) Embedded system-based sticky paper trap with deep learning-based insect counting algorithm. Electronics 10:1754. https://doi.org/10.3390/electronics10151754
32. Suto J (2022) A novel plug-in board for remote insect monitoring. Agriculture 12:1897. https://doi.org/10.3390/agriculture12111897

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.