2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Ressource-Efficient Moth Detection for Pest Monitoring with YOLOv5

M. Tallal Farooq, Martin Leipert, Andreas Maier, Vincent Christlein
Pattern Recognition Lab, Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen, Germany
{firstname.lastname}@fau.de

***Abstract*—Moths pose a significant threat to agricultural crops, and identifying them accurately is crucial for effective pest monitoring and crop conservation efforts. However, manually evaluating glue traps is a time-consuming and labor-intensive process, which has led to the development of automated solutions. In this study, we present a deep learning-based automated detection pipeline that can detect moths in images captured by field traps with pheromone-emitting glue pads. To train our model, we collected a comprehensive dataset that includes moths from various environments, such as agricultural plants, homes, and food production facilities. We augmented this dataset and included additional glue pad datasets, enabling the model to detect moths regardless of the species. We base our model on the YOLOv5 algorithm and fine-tune it using transfer learning, which enables us to identify moths in real-time and on embedded hardware. Our evaluation of the algorithm reveals that it achieves an average precision of 98.2 % on a test dataset, which outperforms reference models from previous research. We also assess the model's ability to handle disturbances such as other insects, varying lighting conditions, and foreign objects. Importantly, our solution maintains a tiny memory footprint and low inference time of 2.3 ms, making it a highly efficient and effective tool for moth detection in the field.**

***Index Terms*—Pheromone Trap, Pest Management, Insect Detection, Moths**

I. Introduction

Approximately 40 % of agriculture productivity is being reduced by pests and weed infestations [1], [2]. One of these pests are moths, belonging to the order Lepidoptera, with around 160 000 different species. They are characterized by their distinctive scales-covered wings as visible in Fig. 1. Moths cause damage in their larval stage by feeding different natural materials. Cloth moths, like Tineola bisselliella (common clothes moth, Fig. 1a), are notorious for their infestation of natural fibers such as wool, fur, and silk, causing damage to clothing, carpets, and upholstery. Other moth species, such as the Helicoverpa zea (corn earworm, Fig. 1b), Helicoverpa armigera (cotton bollworm) and Spodoptera exigua (beet armyworm, Fig. 1c), are considered significant crop pests, that cause substantial damage to agricultural crops by feeding on the fruits and leaves. Food moths, like the Plodia interpunctella (indianmeal moth, Fig. 1d) damage stored products, such as grains and nuts, by feeding on them and contaminating them with their larvae and feces [2]. Due to their rapid reproduction rates and recent resistances to pesticides, moths pose significant challenges to crop production and food security.

(a) Tineola bisselliella (b) Helicoverpa zea (c) Spodoptera exigua (d) Plodia interpunctella

Figure 1: Images of different moth species that harm: (a) cloths (b)–(c) agricultural crops, and (d) food stocks. Sources: [3]–[6]

Monitoring systems for pest management in agriculture aid in preventing crop damage from pests including insects, rodents, and crop diseases leading to better crop health and crop yields [7]. Pest management methods include the use of pesticides, biological controls, and physical barriers. Pest management techniques in agriculture can contribute to higher crop yields and better overall crop health by minimizing pest damage. Pest management systems rely on monitoring, in case of moths this is commonly done by pheromone traps with a glue pad. Moth imagos are attracted by pheromones and then stick to the glue. Pheromones are specific to single moth species, allowing precise monitoring of specific pests and damages with glue pads by selecting the right pheromone or pheromone mixture [8]. When monitoring, quick responses to occuring pests are essential. The occurrence of imagoes indicates that reproduction begins, eggs are laid soon, new larvea are due to occur and harm crops. The distinction between images with single insects and trap images is important because trap images may show a greater variety of insect species in different positions, whereas database images show only a single clearly visible bug that completely fills the image. Pest management systems rely on the analysis of trap images by a human expert [9], [10]. This is time-consuming and error prone [11], [12]. Other approaches use modern object detection techniques, but still take pictures from glue pads manually [13]. In other devices, the photos and the detection are done after two or three days, when the moth's features start to deteriorate. In consequence the accuracy of the detection is decreased. With a regular acquisition interval and automated counting, monitoring of the infestation is facilitated.

Our work applies state-of-the-art deep learning algorithms to automate moth detection and counting. Our target is to detect

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

moths of different species in a resource- and memory-efficient way in images obtained from automated field traps under varying conditions and environments. The general algorithm is shown in Fig. 2. Therefore, we evaluate different model sizes of the YOLOv5 model [14] and compare them to other models used in literature. We evaluate the trained models regarding mAP, inference time, and recall. We have implemented species-agnostic detection in our model since we train our model on various species of moths. We can determine the caught species using different pheromones, making the automated trap widely applicable. To cope with our small dataset, we use transfer learning from ImageNet [15], data augmentation, and we include other available datasets. Detecting pests from a distance under various conditions allows for efficient, continuous and seamless monitoring of insects in different environments.

Figure 2: Description of the general moth detection and glue pad monitoring procedure.

```
SetupTrap()
InitializeSystem()
while True do
    image = CaptureImage()
    image = PreProcessImage(image)
    detectedObjects = DetectMothsWithNeuralNetwork(image)
    detectedMothsCount = ProcessDetections(detectedObjects)
    if detectedMothsCount >= UserDefinedThreshold then
        NotifyUser("Glue pad is full. Change glue pad.")
    end if
    WaitUserDefinedTimeInterval()
end while
```

Figure 3: The Traplinked Tom trap used for data acquisition

II. Related Work

Moths and insect detection on glue pads lies in the fields of image classification and object detection. Before the breakthrough of neural networks, approaches with scale invariant feature transform (SIFT) and bag of words were applied for automation in insect detection. Those reached accuracy rates of 85.4 % on single images of insects [16]. For detection of moths from trap images Bakkay et al. [17] combined contour-based and region-based segmentation. With adaptive k-means clustering, using the contours convex hull and region merging algorithm, they reached a detection rate of 70 % for insects.

The breakthrough of Convolutional Neural Networks (CNNs) also overcame the limitations of machine learning in agricultural research [18], [19]. In 2016, Ding and Taylor achieved an AP of 93.1 % on 40 images for moths in trap images with a CNN for image area classification and a sliding window for image segmentation [19]. With the advances of deep learning, different pipelines and algorithms for moth monitoring evolved. Bjerge et al. [20] developed a monitoring system to monitor eight different moth species with a light trap based on a canvas and a light ring. The detection branch is based on a custom made CNN and achieves an F1-score of 0.93.

For different prototypical glue pad traps, Sütő developed a detection and classification system for the codling moth [21]. With MobileNetV2 on an OpenMV Cam H7 microcontroller board he reached an accuracy of 82 %. Similarly, Hoye et al. [9] developed a full pipeline for online moth detection using a custom CNN and different prototype traps. They detected eight different species with an automated trap. With a custom-made lightweight network, they claim an F1 performance of 93 %. Hong et al. [13] investigated a variety of standard detector algorithms on a dataset containing three different moth species. They achieved the best mAP of 90.25 % using a Faster R-CNN detector with a ResNet-101 backbone. In a 2022 review by Sütő [18], different hardware and algorithms were studied for moth detection with automated traps, with special focus on practical application. He found that the problem of automated insect counting is not yet solved due to the lack of sufficient data and the small insect size in the images. In summary, there are various approaches that achieve good scores. However, these were not validated in actual application contexts, such as using actual trap photos for pest monitoring.

III. Material and Methods

There are two main challenges in detecting pests from trap images. The first challenge is the main constraint of the setup: The power consumption limits the hardware and the field hardware is also price constrained. The second factor is inconsistency and variability of conditions inside the trap: non-pest insects, objects and illumination conditions.

A. Data

The images were obtained using a Traplinked glue pad trap with image acquisition device named Tom shown in Fig. 3. This device is equipped with an RGB camera featuring a resolution of 1080 × 1920 pixels and a field of view spanning 220 degrees. The device takes pictures at regular six-hour intervals and transmits them to a server for further analysis. The images were captured during day- and nighttime, to provide a comprehensive range of real-world scenarios. The trap is set up at various locations indoor and outdoor to capture images of different moth species, mainly Tineola bisselliella, Plodia interpunctella, and Ephestia kuehniella. The trapped species is determined by the used pheromones. The obtained data are then annotated with the makesense platform [22]. An annotated image is shown in Fig. 4. The trap images have a simple background consisting of the glue pad without complex objects. During data annotation, all moths are labeled with a bounding box and assigned to the

Table I: Statistics of the created dataset.

Dataset	Images with Moth	Images without Moth	Average Moths per image
Training	994	91	11.45
Validation	212	22	10.62
Test	216	20	11.05

class 'moths'. Non-moth objects, such as flies and other insects, are not annotated and are considered as negative examples, typically referred to as the 'background' class. As a result of data collection and labelling, we get a total of 474 annotated images. Our data exhibits significant variations in terms of lighting conditions because data are obtained at different times of day, as well as indoors and outdoors. The influence of daytimes is shown in Fig. 7. Additionally, the moth species vary.

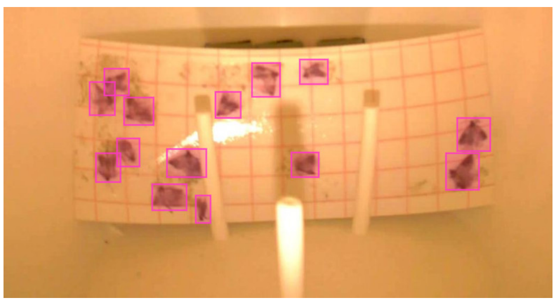

Figure 4: Image from the trap annotated with makesense. The lighting conditions lead to a yellowish tint.

Since our own dataset is relatively small, we incorporate several open-source images from the Lobesia botrana [23] and Cydia pomonella [24] datasets. Both datasets provide glue pad images and differ mainly in illumination conditions and the glue pads colour.

To create an even more extensive dataset, we additionally apply augmentation techniques on our data. To account for variations that occur in the field, we employed data augmentation techniques using the Keras Image Data Generator. These techniques include changes in brightness, perspective, and rotation of the images. The latter accounts for differently rotated moths. By combining the open source dataset with our augmented dataset, we were able to obtain a total of 1422 labeled images of moths. The collected and augmented dataset is divided into three subsets: the training set, the validation set, and the test set, in the proportion of 70 %, 15 %, and 15 %, respectively as shown in Table I.

B. Detection Network Models

In this work, we aim to use a light-weight and fast model for object detection in real-time, thus we decided for the YOLOv5 [14] architecture, especially in the small and nano versions. Additionally, we evaluate also the bigger YOLOv5 models as well as models from the literature, i.e., the best performing model from previous works (FasterRCNN

Table II: Model sizes, computation costs and number of parameters for different YOLOv5 and reference models.

Models	Parameters (Million)	Model Size (MB)	Flops (Billion)
YOLOv5n	1.9	3.7	4.5
YOLOv5s	7.2	13.9	16.5
YOLOv5m	21.2	41.1	49
YOLOv5l	46.5	90.5	109.1
YOLOv5x	86.7	168.9	205.7
MobileNetV1 FPN	36.2	22.4	123.3
MobileNetV2 FPN Lite	3.4	16.8	1.5
FasterRCNN ResNet-101	44	181.1	7.8

w. ResNet-101 backbone) and another lightweight network (MobileNet).

The YOLO family, originally developed by Redmon et al. [25], refers to a single-shot, fast real-time object detector. The term single shot refers to the simultaneous object detection and classification in the same feature branch. The general focus of YOLO lies in a light-weighted design and fast inference speed, which makes these able for real time applications on embedded systems. Additionally, YOLO is well known for being adaptable for different objects, memory, and dataset sizes [26]. The YOLO version used here is YOLOv5 [14].

YOLOv5 consists of 3 main parts: Backbone, Neck, and Head (for the architecture see Fig. 5). CSPDarknet53 is used as a backbone to extract features from the desired image [14]. Cross Stage Partial DenseNet (CSP) [27] is itself based on the DenseNet [28] architecture and improves the feature propagation and gradient flow of DenseNet. YOLOv5 introduces a focus layer and replaces the original three input layers from YOLO v3 and reduces the number of convolutions by slicing the input image to 4 matrices with half resolution and then performing a convolution on this vector. The focusing layer reduces memory requirements and improves gradient backward propagation. However, this mechanism is suspected to diminish the performance on tiny objects [29].

As model neck, YOLOv5 uses the Path Aggregation Network [30]. Due to data linkage, this network passes features from one level to all other levels and therefore preserves spatial information especially well. This is important for the model head (YOLO layer), that extracts the position of the box and the classification result [26].

YOLO architectures are popular in pest detection. In its small version (YOLOv5s), it is applied by Teixeira et al. [31] in two different datasets. They demonstrate the effectiveness of transfer learning with small YOLO models by achieving a detection AP of 96.5 % and a counting error of 63.3 % for bedbugs. For grapevine moth, the AP is 90.9 % and the counting error 6.7 %. Onler [32] evaluates different YOLOv5 sizes in his study in 2021 for harmful caterpillar detection in agriculture. For these relatively simple tasks, small YOLO versions show no drop in performance while maintaining superior speed. Regarding complexity, YOLO v5 consists of two-dimensional convolutions. Assume N to be the image side length, and then the time complexity is $O(N^2)$.

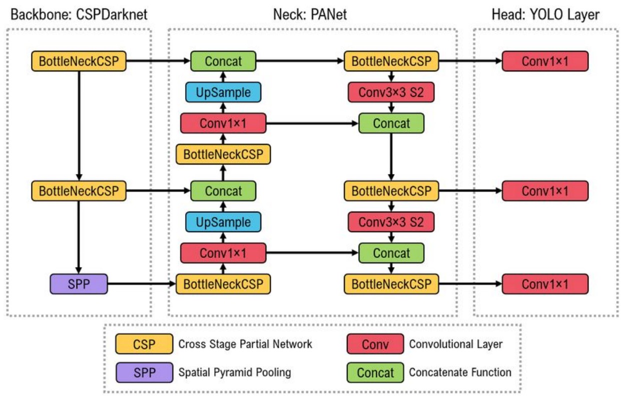

Figure 5: YOLOv5 architecture [14]

C. Evaluation Protocol

For a field application, the relevant factors are not only prediction performance, but also memory, and computational efficiency. As resource consumption has a cost and environmental impact. To demonstrate the effectiveness and performance of YOLOv5 regarding memory and computational efficiency, we compare the following networks: all five variants of YOLOv5 (from nano to extra large). The best performing network of the comparison by Hong et al. [13], which is a Faster RCNN with ResNet-101 backend. As well as SSD MobileNet that is regarded as memory efficient and which has been suggested by Sütő [21]. All models were trained from scratch as well as on pretrained weights from ImageNet. Pretrained models usually show faster convergence. The YOLOv5 models were trained for 100 epochs using a learning rate of 0.01 and a batch size of 4. The chosen optimizer for training was stochastic gradient descent (SGD). The reference models, including MobileNetV1 FPN, MobileNetV2 FPN Lite, and FasterRCNN ResNet-101, the parameters used were identical, except the initial learning rate of 0.04.

For training, we used 1085 annotated images to train each YOLOv5 model for object recognition. Table II shows the number of parameters and the model size for all used models. Training and evaluation was done in PyTorch within the Google Colab Pro environment with a Tesla T4 graphics card. The image patches used have a resolution of 640×640 as per our image dimensions.

In our experiments, we use the following evaluation metrics. Mean average precision mAP with an intersection over union (IoU) threshold of 0.5, denoted as mAP@IoU=0.5. Furthermore, we report the precision, i.e., measuring the number of correctly predicted samples amongst the predictions. The recall, measuring the number of positive samples correctly found by the detection algorithm. As we want to use the algorithm in field, the inference time is also used as a metric of computational performance (although this will be computationally slower on real hardware). For further explanation of the metrics we refer to [29].

IV. Results and Discussion

In Table III, the results for the models used with and without transfer learning are shown. The best performing model regarding mAP is YOLOv5x with an mAP of 98.4 %. In terms of mAP, all YOLOv5 models (even the nano and small) outperform the reference models SSD MobileNet and FasterRCNN ResNet-101. The latter model has the best Recall with 98.2 %. However, even YOLOv5n is only 1.9 % worse in Recall, while both SSD MobileNet models are 31.2 % and 39.0 % behind in terms of Recall. In summary, the YOLO models perform significantly better than the 'lightweight' reference models and comparable to the much larger Faster RCNN ResNet-101 model.

A. Impact of Transfer Learning

In Table III the scores are shown both for the networks with and without transfer learning. For the training from scratch, YOLOv5l shows the best mAP with 98.3 % and Recall with 96.5 % rates. For the approach with transfer learning, YOLOv5x has the best mAP with 98.4 %. While the Faster RCNN - ResNet-101 has the best Recall rate with 97.4 %. So large models perform comparable with and without transfer learning. In contrast, the smaller models perform significantly worse than with transfer learning. With transfer learning, YOLOv5n reaches a mAP of 98.2 %, which is an improvement of +3.8 % compared to the model trained from scratch. Also the recall is 4.6 % better than without transfer learning. Especially small models with few parameters benefit from using transfer learning. Additionally, for the YOLOv5 models with transfer learning, the difference in performance for the nano and small versions to the extralarge versions is only -1.1 % in recall and -0.2 in mAP. During training, the transfer learning approach converges faster than training from scratch, and also shows fewer fluctuations. The training saturates rather quickly, after 10 to 15 epochs, in contrast to training from scratch, where this takes about 10 epochs longer.

B. Performance

In Table III, also the computation time on our hardware is shown. YOLOv5n predicts in 2.3 ms and is the fastest. While FasterRCNN with ResNet-101 is the slowest with a computation time of 197 ms. Moreover, all YOLO variants are faster than the fastest reference network (SSD MobileNet v2 - FPN Lite). As shown in Table II, YOLOv5n is smaller, has less parameters, and fewer Multiply-Adds than all the reference models. FasterRCNN with the ResNet-101 Backend is about 30 times larger in number of parameters than YOLOv5n while MobileNet v2 with FPN-Lite is about 70 % bigger. As the performance is not significantly worse for YOLOv5n than for large and complex models, we assume, that the complexity is sufficient for our task and that YOLOv5n offers the best trade-off between prediction performance and computational complexity.

Table III: Evaluation of individual performance metrics for Moth detection.

Models	Transfer Learning Precision (%)	Recall (%)	mAP @IoU:0.5 (%)	Inference time (ms)	Without Transfer learning Precision (%)	Recall (%)	mAP @IoU:0.5 (%)	Inference time (ms)
YOLOv5n	96.0	96.3	98.2	**2.3**	92.6	91.7	94.4	**2.3**
YOLOv5s	96.3	97.1	98.2	5.4	95.8	94.7	97.5	4.7
YOLOv5m	**96.9**	96.2	97.3	12.3	95.8	94.4	96.6	11.3
YOLOv5l	96.6	97.3	97.6	19.4	**96.3**	**96.5**	**98.1**	20.3
YOLOv5x	96.6	97.4	**98.4**	36.4	95.3	94.2	96.5	35.2
Faster RCNN - Resnet-101	–	**98.2**	96.1	197	–	–	–	–
SSD MobileNet v1 - FPN	–	66.2	93.2	100	–	–	–	–
SSD MobileNet V2 - FPNLite	–	58.4	89.0	55.2	–	–	–	–

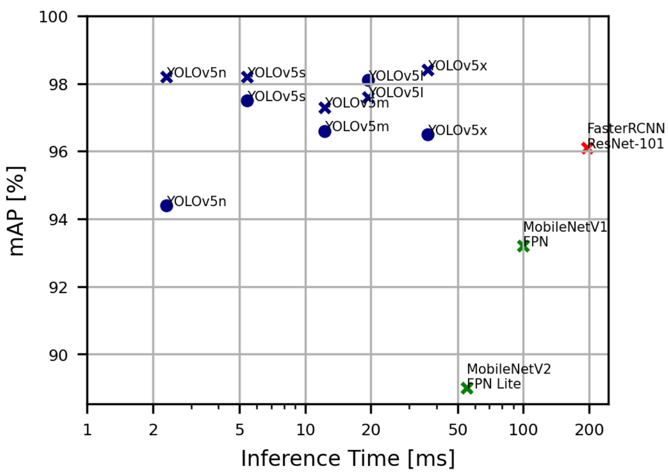

Figure 6: Inference time and mAP plotted for all examined models. Results with transfer learning are marked with an x, results from scratch with an o.

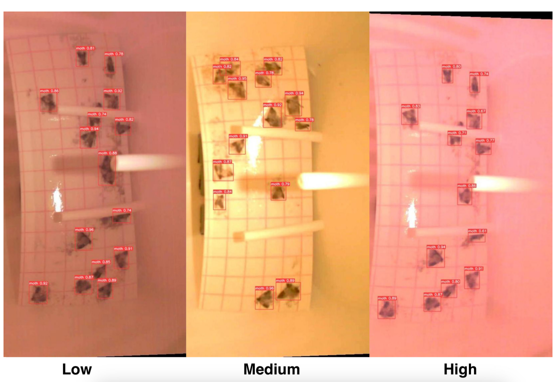

Figure 7: Detection results with YOLOv5n (trained with transfer learning) at various times of the day under different levels of brightness.

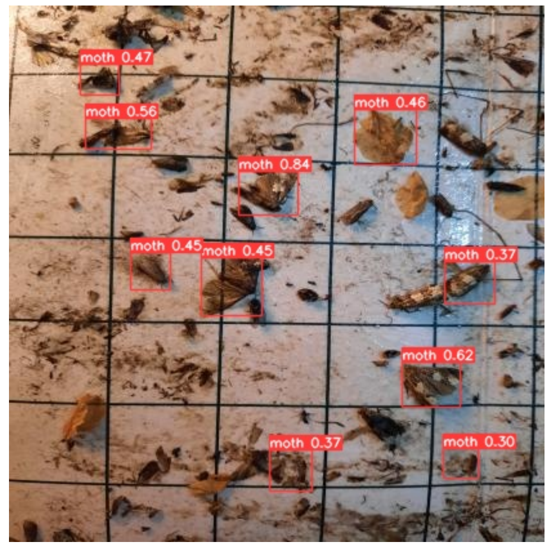

Figure 8: Different disturbances (other insects, other objects). The algorithm (YOLOv5n) still detects the majority of moths correctly.

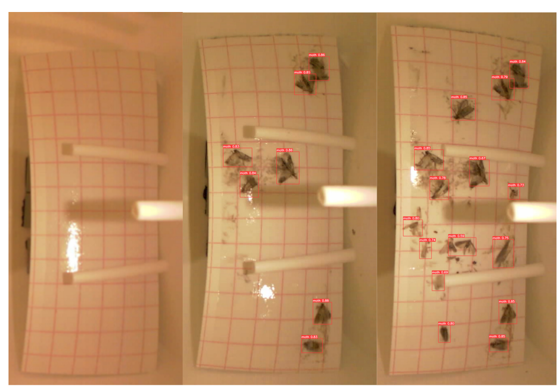

Figure 9: Time-lapse of a trap: Empty trap (left), few moths arrive (center) and start to deteriorate (right).

C. Qualitative results

Our research focused on developing a method for detecting moths in images from a trap in field. Thus, we use selected images to demonstrate the effectiveness of our method under the varying conditions in the trap. In Fig. 7, the algorithm precisely predicts moths in the trap at different times of the day (morning to evening), under all these conditions, the YOLOv5n recognizes all moths and draws correct bounding boxes.

In Fig. 8, the influence of other insects and foreign objects on trap images is shown. With a lot of disturbances like petals, the detection quality deteriorates. Nevertheless, most moths are detected correctly, so pest monitoring still works.

In Fig. 9, we demonstrate that moths are detected correctly even if they already show signs of deterioration.

V. Conclusion

In this study, we evaluated different algorithms for moth monitoring on a real field glue pad trap. With YOLOv5 nano we found a lightweight algorithm, that may also run on embedded hardware and in real-time. When trained with transfer learning, this algorithm performs comparable to more complex reference models in our one-class moth detection problem. While in multi-class problems YOLOv5n drops in performance, e.g., in case of a 21 class problem [33]. We can demonstrate that our solution is robust to varying lighting conditions, foreign objects, and other (non-moth) insects in the trap and to different moth species in the dataset. Even when trained on a comparably small dataset. Additionally, the detection works universally for moths of different species in the trap.

These findings are limited by the size of the training and testing set and that we cannot classify individual species. As shown in Fig. 1, moth species have similar features and thus classification, especially of moths on a glue pad, is tricky. There are numerous datasets available, especially for the codling moth, which are often different in the acquisition technique and the aspect of the glue pad, resulting in color, lighting and contrast variations. For future work, one universal dataset for different traps and species could potentially be created from different available moth datasets with the help of style transformations using GANs [34].

Acknowledgment

We would like to thank Traplinked GmbH from Nuremberg for providing the trap image data.

References

[1] T. Reiche, "We have to handle our food better – protect you better," Online, 2022, accessed: April 26, 2023. [Online]. Available: https://www.traplinked.com/en/pest-control-portal/ 1
[2] D. Jiang, S. Chen, M. Hao, J. Fu, and F. Ding, "Mapping the potential global codling moth (cydia pomonella l.) distribution based on a machine learning method," *Sci. Rep.*, vol. 8, no. 1, pp. 1–8, 2018. 1
[3] F. Welter-Schultes, CC 0, via Wikimedia Commons. 1
[4] C. M. Andy Reago, CC BY 2.0, via Wikimedia Commons. 1
[5] D. Hobern, CC BY 2.0, via Wikimedia Commons. 1
[6] Kaldari, CC 0, via Wikimedia Commons. 1
[7] M. F. Potter. (2016) Clothes moths. [Online]. Available: https://entomology.ca.uky.edu/ef609 1
[8] H. D. Astrid Groot, Teun Dekker, "The genetic basis of pheromone evolution in moths," *Annu Rev Entomol*, vol. 61, pp. 99–117, 2016. 1
[9] T. T. Høye, J. Ärje, K. Bjerge, O. L. Hansen, A. Iosifidis, F. Leese, H. M. Mann, K. Meissner, C. Melvad, and J. Raitoharju, "Deep learning and computer vision will transform entomology," *Proc. Natl. Acad. Sci. USA*, vol. 118, no. 2, p. e2002545117, 2021. 1, 2
[10] J. Wang, C. Lin, L. Ji, and A. Liang, "A new automatic identification system of insect images at the order level," *Knowl Based Syst*, vol. 33, pp. 102–110, 2012. 1
[11] Z. Yuan, W. Fang, Y. Zhao, and V. S. Sheng, "Research of insect recognition based on improved yolov5," *J. Artif. Intell.*, vol. 3, no. 4, p. 145, 2021. 1
[12] Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu, "Application of deep learning in integrated pest management: A real-time system for detection and diagnosis of oilseed rape pests," *Mob. Inf. Syst.*, vol. 2019, 2019. 1
[13] S.-J. Hong, S.-Y. Kim, E. Kim, C.-H. Lee, J.-S. Lee, D.-S. Lee, J. Bang, and G. Kim, "Moth detection from pheromone trap images using deep learning object detectors," *Agriculture*, vol. 10, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2077-0472/10/5/170 1, 2, 4
[14] Ultralytics, "YOLOv5: YOLOv5: Real-time object detection with PyTorch," https://github.com/ultralytics/yolov5, 2020, accessed April 24, 2023. 2, 3, 4
[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255. 2
[16] C. Wen, D. E. Guyer, and W. Li, "Local feature-based identification and classification for orchard insects," *Biosyst. Eng.*, vol. 104, no. 3, pp. 299–307, 2009. 2
[17] M. C. Bakkay, S. Chambon, H. A. Rashwan, C. Lubat, and S. Barsotti, "Automatic detection of individual and touching moths from trap images by combining contour-based and region-based segmentation," *IET Comput. Vis.*, vol. 12, no. 2, pp. 138–145, 2018. 2
[18] J. Suto, "Codling moth monitoring with camera-equipped automated traps: A review," *Agriculture*, vol. 12, no. 10, p. 1721, 2022. 2
[19] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Comput Electron Agric*, vol. 123, pp. 17–28, 2016. 2
[20] K. Bjerge, J. B. Nielsen, M. V. Sepstrup, F. Helsing-Nielsen, and T. T. Høye, "An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning," *Sensors*, vol. 21, no. 2, p. 343, 2021. 2
[21] J. Sütő, "Embedded system-based sticky paper trap with deep learning-based insect-counting algorithm," *Electronics*, vol. 10, no. 15, p. 1754, 2021. 2, 4
[22] P. Skalski, "Make Sense," https://github.com/SkalskiP/make-sense/, 2019. 2
[23] GORSENTAM, "cydia_pomonella dataset," mar 2023, visited on 2023-05-11. 3
[24] C. Teixeira, "grape_moth_1 dataset," may 2022. 3
[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788. 3
[26] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 029–13 038. 3
[27] "Cspnet: A new backbone that can enhance learning capability of cnn." 3
[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269. 3
[29] D. Padilla Carrasco, H. A. Rashwan, M. Á. García, and D. Puig, "T-yolo: Tiny vehicle detection based on yolo and multi-scale convolutional neural networks," *IEEE Access*, vol. 11, pp. 22 430–22 440, 2023. 3, 4
[30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018. 3
[31] A. C. Teixeira, R. Morais, J. J. Sousa, E. Peres, and A. Cunha, "A deep learning approach for automatic counting of bedbugs and grape moth," *Procedia Comput. Sci.*, vol. 219, pp. 145–152, 2023. 3
[32] c. E. Önler, "Real time pest detection using yolov5," *Int J Agric Sci*, vol. 14, no. 3, pp. 232–246, 2021. 3
[33] I. Ahmad, Y. Yang, Y. Yue, C. Ye, M. Hassan, X. Cheng, Y. Wu, and Y. Zhang, "Deep learning based detector yolov5 for identifying insect pests," *Appl. Sci.*, vol. 12, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/10167 6
[34] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2019. 6