Investigating generative neural-network models for building pest insect detectors in sticky trap images for the Peruvian horticulture

Joel Cabrera
Pontifical Catholic University of Peru
Lima, Peru
jjcabrerarios@gmail.com

Edwin Villanueva
Pontifical Catholic University of Peru
Lima, Peru
ervillanueva@pucp.edu.pe

Abstract

Pest insects are a problem in horticulture, so early detection is key for their control. Sticky traps are an inexpensive way to obtain insect samples in crops, but identifying them manually is a time-consuming task. Building computational models to identify insect species in sticky trap images is therefore highly desirable. However, this is a challenging task due to the difficulty in getting sizeable sets of training images. In this paper, we studied the usefulness of three neural network generative models to synthesize pest insect images (DCGAN, WGAN, and VAE) for augmenting the training set and thus facilitate the induction of insect detector models. Experiments with images of seven species of pest insects of the Peruvian horticulture showed that the WGAN and VAE models are able to learn to generate images of such species. It was also found that the synthesized images can help to induce YOLOv5 detectors with significant gains in detection performance compared to not using synthesized data. A demo app that integrates the detector models can be accessed through the URL `https://bit.ly/3uXW0Ee` . The repository of the project is available at `https://github.com/weirdfish23/pest-insects-GAN`

1 Introduction

One of the main problems in horticulture are pest insects. These can affect the productivity and quality of crops. In Peru, this problem represents a great challenge due to the artisanal nature of the activity and the lack of affordable technological tools to deal with local pests. The usual way that horticulturists approach the problem is through the scheduled and intensive application of pesticides, which generates negative impacts on the environment and people [Cañedo et al., 2011].

A way to improve the manner of dealing with insect pests is by collecting timely information on the distribution of insect populations and species in crops. This can help implement on-necessity pesticide application strategies, which can reduce production costs and environmental impacts. Sticky traps are simple devices for collecting insect samples in crops [Li et al., 2021]. These are pieces of colored sticky paper that attract and immobilize insects when land on them. Despite its low cost, the process of identifying and classifying insect species in sticky traps is a time-consuming task that requires specialized personnel. It has been suggested that computer vision and machine learning techniques could help automate this process [Wang et al., 2021, Li et al., 2021, Zhong et al., 2018].

Indeed, some deep learning models have recently been proposed to identify insect species in sticky trap images [Espinoza et al., 2016, Huang et al., 2019, Lu et al., 2019, Rustia et al., 2019, Xia et al., 2015, Zhong et al., 2018, Zhou et al., 2019, Wang et al., 2021, Li et al., 2021]. However, most of these works assume that a large set of insect images is available for training or have invested effort in

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

getting it. Obtaining a sizeable set of pest insects images could be one of the most expensive parts in building insect detectors. For the case of Peru, at the best of our knowledge there is not a public collection of images of the most relevant insect species for the Peruvian horticulture.

In this paper we investigated the application of generative neural network models in order to build a sizeable and realistic set of training images that serve to induce effective models for the identification and classification of insect species in sticky traps for Peruvian horticulture. The investigated models are three recent neural network architectures: Conditional Deep Convolutional Generative Adversarial Network (DCGAN) [Radford et al., 2016, Goodfellow et al., 2014], Conditional Wasserstein GAN (WGAN) [Arjovsky et al., 2017] and Conditional Variational Autoencoder (VAE) [vae]. These techniques are evaluated in their usefulness to induce effective insect classifiers and compared against the classical data augmentation technique based on image perturbations (rotations, mirroring, modifications of brightness, contrast, etc.). To the best of our knowledge, this is the first study to investigate and compare different generative neural models for artificially creating images of pest insects for inducing detectors.

The rest of the paper is organized as follow. Section 2 describe the data collection and pre-processing, the generative models and the detection model considered in the study. Section 3 presents the experiments and results obtained. Finally, Section 4 concludes the work and suggests future works.

2 Materials And Methods

2.1 Data collection and pre-processing

For training the generative models we collected an initial set of images of the following insect species relevant in the Peruvian horticulture: *Prodiplosis longifila*, *Liriomyza huidobrensis*, *Brevicoryne brassicae*, *Trips tabaci*, *Bemisia tabaci*, *Macrolophus pygmaeus*, *Nesidiocoris tenuis*. For the last three species we use images of the "Yellow Sticky Trap Dataset" [Nieuwenhuizen et al., 2019]. For the other species we scraped web pages with information of pest insects. All the images were pre-processed (cut, filled and scaled) to obtain patches of 64x64 pixels containing each one a single insect. The number of collected images for each specie is showed in table 1.

Table 1: Total of pest insects images per species and source.

Species	Total of images	Source
Prodiplosis longifila	35	Web
Liriomyza huidobrensis	112	Web
Brevicoryne brassicae	58	Web
Trips tabaci	53	Web
Bemisia tabaci	5807	Yellow Sticky Trap Dataset
Macrolophus pygmaeus	1619	Yellow Sticky Trap Dataset
Nesidiocoris tenuis	688	Yellow Sticky Trap Dataset
Total	**8372**	

The collected dataset was further increased using a series of image transformation procedures. We applied: random horizontal and vertical flip, random rotation and color jitter. Fig. 1 shows an example of the result of applying these transformation procedures several times to a single base image. Original and pre-processed images are available in our repository.

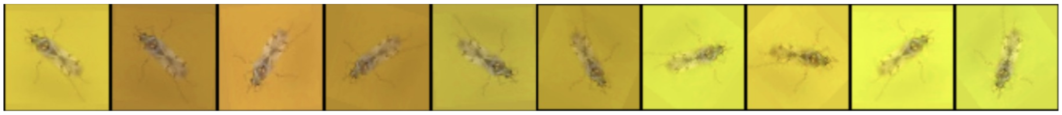

Figure 1: Images resulting of applying traditional image data augmentation procedures to a single insect image (leftmost image).

2.2 Generative models

We adjusted three generative models (DCGAN, WGAN and VAE) to acquire the capacity to synthesize pest insect images to be used as data augmentation procedures for subsequent classifier induction. Details of the model architectures and their training process can be found in Appendix A. The models are conditioned on the species to be generated. All generative models were adjusted with the data generated as described in the previous section. The trained models can be found in our repository.

2.3 Generation of sticky trap images

The trained generative models, WGAN and VAE, were used to synthesize sticky trap images to train and validate insect species detectors. We constructed yellow sticky trap images by placing model-generated insect images randomly in an area of dimensions compatible with what is found in real settings. Original images obtained in the pre-processing step (Section 2.1) are also considered to be placed in the resulting images using a mixing ratio. In the experimental phase we tested different mixing ratios. To simulate realistic conditions we also considered the addition of different objects in the sticky traps images, like sticks, rocks and random noise. All the information about the class of the images added, the location of the pest insects in the image as bounding boxes and the proportion of generated and real images is stored in separated files. These files were required to train the detection model.

2.4 Insect detection and classification with YOLOv5

In order to detect the pest insects in the yellow sticky traps images we use the pretrained model YOLOv5m [Jocher et al., 2021] and fine tuned it in the custom datasets. In this version of the model, Cross Stage Partial Network (CSPNet) [Wang et al., 2020] is used as the model back- bone and Path Aggregation Network (PANet) [Liu et al., 2018] as the neck for feature aggregation. The head is retrained to make the prediction of the desired classes and bounding boxes of the species of insects. In appendix B we show the high level architecture of this model.

3 Results And Discussion

In this section we describe the experiments performed and results obtained. First, we evaluate the three types of generative models described above in their ability to reproduce the diversity of the insect images of each target specie in the original dataset. Next, we present results about the usefulness of the synthesized datasets to induce good models for detection and classification of pest insects in sticky trap images.

All models were implemented in Python language using the Pytorch deep learning library. Experiments were carried on a Lambda Deep Learning Workstation (lambdalabs.com) with two GeForce RTX 2080 Ti GPU cards installed in a system running Ubuntu 18.04.5 LTS operating system.

3.1 Performance evaluation of pest insect image synthesis

Each trained generative model was qualitatively and quantitatively evaluated on its ability to generate realistic insect images and their variabilities of the target species.

For the qualitative evaluation, we generated 500 images of each insect specie with each generative model. Then, we applied the t-SNE technique [van der Maaten and Hinton, 2008] to find a low-dimension representation of the images. This procedure was also performed in the original image set obtained in Section 2.1. Fig. 2 a) shows a scatter plot of the 2D representation of the original images mapped with t-SNE. In figure 2 b), c) and d) show equivalent representations for the images generated with each generative model. We can visually appreciate that the models WGAN and VAE seems to have learned closely the distribution of the original image set. All separated clusters and overlapped clusters in the original set are also reproduced in the synthetic sets generated with that models. The DCGAN model seems to have had difficulty learning the variety of the images in most species since it tends to produce very similar images for each specie (very narrow clusters in the t-SNE representation in Fig. 2 b)), problem known as mode collapse. This can happen because the discriminator learns faster than the generator during the training procedure and the generator doesn't have to opportunity to learn better representations.

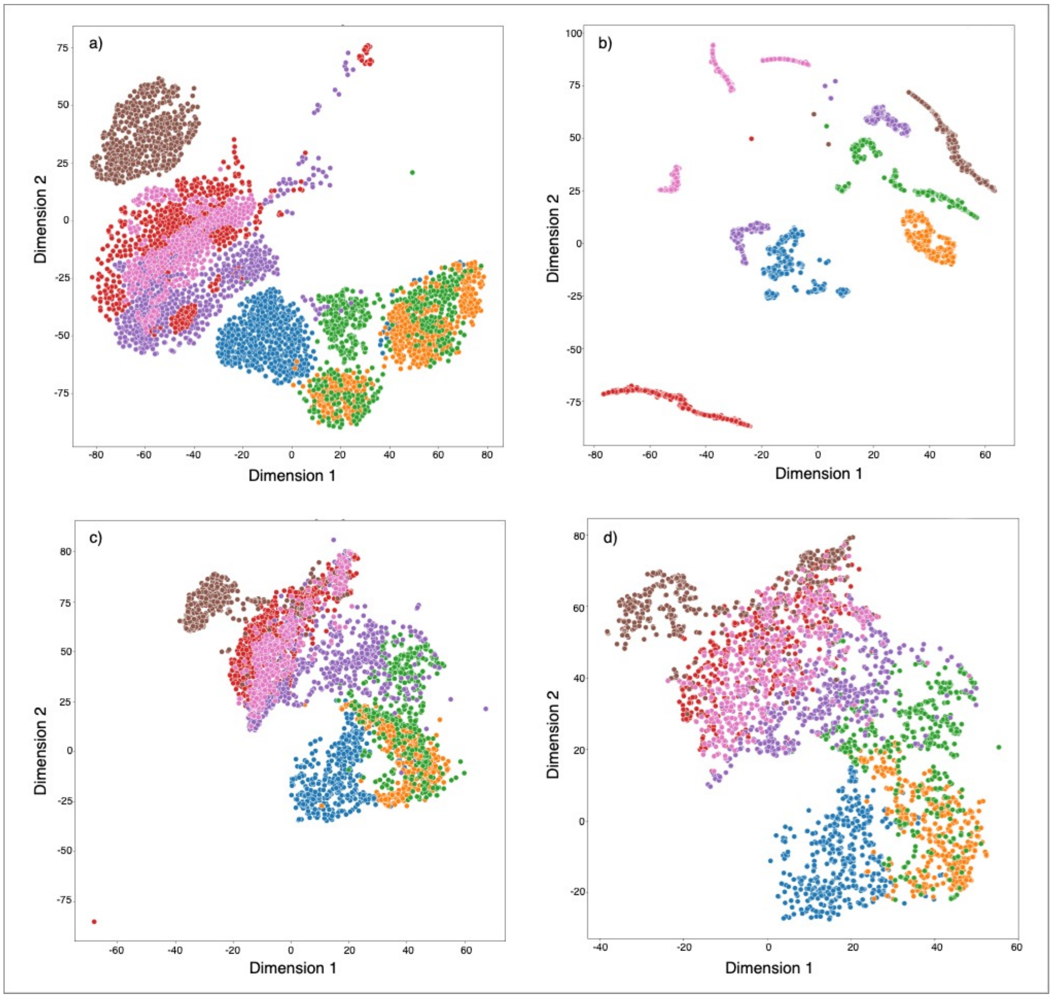

Figure 2: Two-dimensional t-SNE representations of 3500 pest insects images each one. a)Real images, the rest are from generated images as follow: b)DCGAN, c)WGAN and d)VAE. Colors identify the insect species (Blue: *Bemisia tabaci*, Orange: *Macrolophus pygmaeus*, Green: *Nesidiocoris tenuis*, Red: *Brevicoryne brassicae*, Purple: *Liriomyza huidobrensis*, Brown: *Prodiplosis longifila* and Pink: *Trips tabaci*).

To have a more objective analysis of the capabilities of the models in learning the varieties of each species, we calculated the Universal Divergence (UD) [Wang et al., 2009]. The UD metric assess the divergence between two distributions: the distribution of the t-SNE embedding of the original images of a given insect species against the distribution of the t-SNE embedding of the model-generated images of the same insect species. Lower UD values mean that the distributions are closer. Table 2 shows the resulting UD metrics for each insect species and generative model. These results suggest that VAE and WGAN have similar capabilities to capture the distribution of the original images, being the VAE model slightly better, as evidenced in the average UD. We also verify that DCGAN model present the most divergent distributions compared to the original images, which was also observed in the qualitative evaluation (Fig. 2).

3.2 Evaluation of pest insect detection and classification

To further assess the utility of the synthetic data, we induced YOLOv5m models with such data and evaluated their performance in identifying and classifying insect species in test sticky trap images.

Following the procedure described in Section 2.3 we generated six sets of sticky trap images, each set containing 300 yellow sticky trap images, each image with around 100 insects of the seven target

Table 2: Divergence between real images and genereted images per model and species.

	DCGAN	**WGAN**	**VAE**
Bemisia tabaci	7.207	7.431	7.058
Macrolophus pygmaeus	6.953	7.227	7.264
Nesidiocoris tenuis	7.247	6.604	6.878
Brevicoryne brassicae	6.759	6.022	6.070
Liriomyza huidobrensis	5.721	5.880	5.742
Prodiplosis longifila	7.707	6.278	5.997
Trips tabaci	7.475	6.507	6.494
Average	7.010	6.564	**6.500**

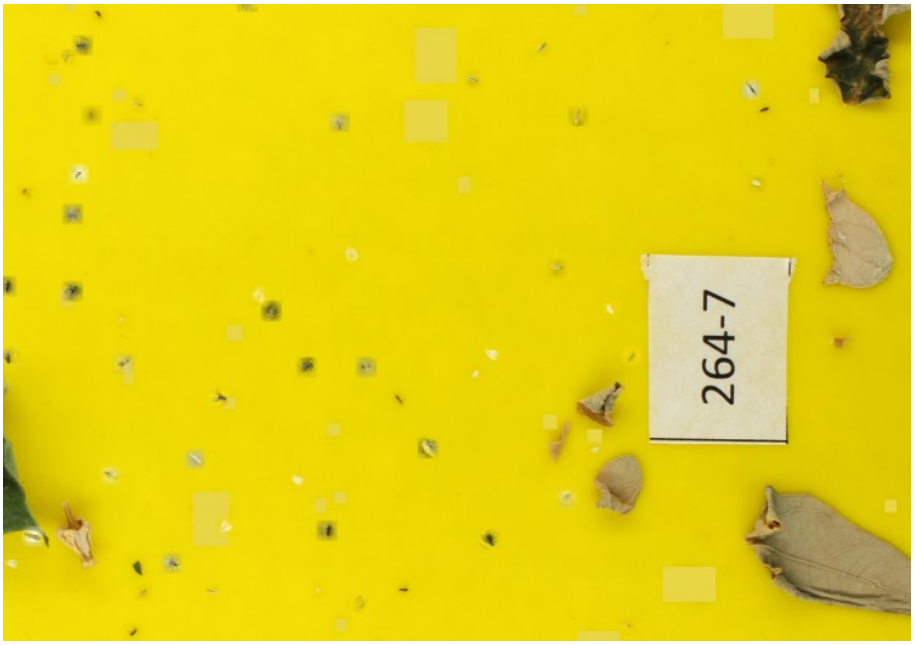

Figure 3: Example of sticky trap image generated with 20% of synthetic pest insects images and other elements found in the field.

species. Fig. 3 shows an example of a generated sticky trap. The difference between the sets is the proportion of the synthetic insects added to the sticky trap. We tested the following proportions: 0% (only real insects), 20%, 40%, 60%, 80% and 100% (only synthetic insects). Fig. 4 shows an example of the output of a YOLOv5m model (bounding boxes and species labels) in some sticky trap image.

To evaluate the detection performance of the YOLOv5m models we use the area under the curve precision-recall (AUC) in testing data. We evaluated this metric per species. Table 3 shows the AUC metrics of the models trained with the 6 different datasets and for each species. We can observe that in five of the seven species the AUC metric improves when we use a fraction of synthetic data in the training set. In some cases the increase in performance has been more than 14% (Liriomyza huidobrensis). In the species Prodiplosis longifila and Trips tabaci no performance improvement was observed with any proportion of synthetic data. For the case of Prodiplosis longifila, this species is quite different from the rest, as observed in the t-SNE plot (Fig. 6), so its identification is not problematic and the generation of synthetic data is not very helpful. For the species trips tabaci, this has high overlapping with Brevicoryne brassicae. Probably the base images of these species with which the generative models were constructed have not had enough diversity to capture discriminating features and it would be necessary to have more real data to learn more details about these species.

A demo web app was deployed integrating the detector models and a simple user interface. The user can submit its sticky trap images and obtain the resulting detection bounding boxes as in Fig. 11. This tool can be accessed through the url `https://bit.ly/3uXWOEe`.

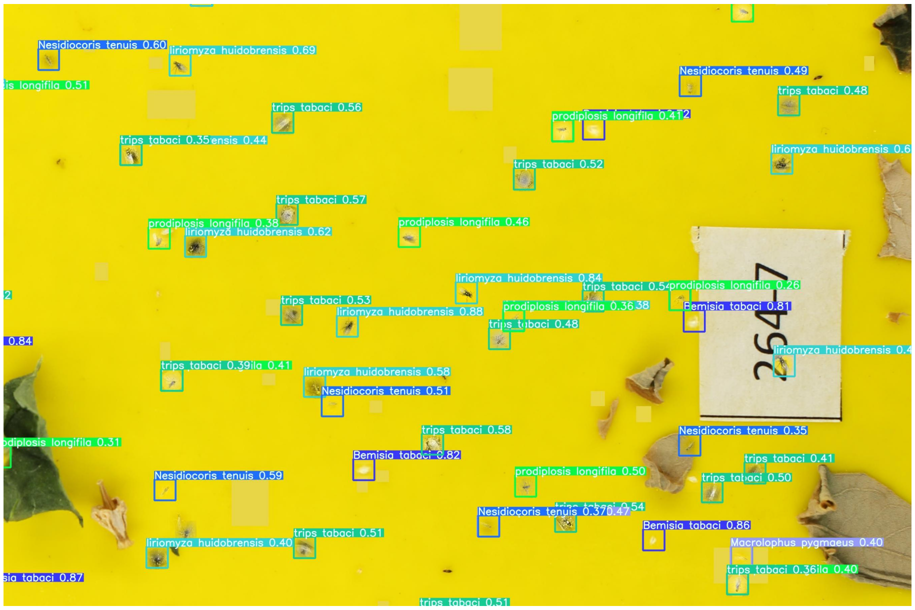

Figure 4: Example of an output of the YOLOv5m model after detecting and classifying the insects in a sticky trap image.

Table 3: AUC metrics of different YOLOv5m models (columns) trained with different proportions of synthetic data and differentiated by insect species (rows). Last column indicates the percentage difference in AUC of the best model using synthetic data in relation to the model trained without synthetic data

		Use of generated imgs.						
Species	**# Imgs.**	**0%**	**20%**	**40%**	**60%**	**80%**	**100%**	**Max. Diff.**
Bemisia tabaci	5807	0,91	0,91	0,92	0,87	**0,93**	0,90	**2,10%**
Macrolophus pygmaeus	1619	0,70	0,68	0,72	0,72	0,75	**0,81**	**10,60%**
Nesidiocoris tenuis	688	0,61	0,52	**0,65**	0,36	0,44	0,49	**4,00%**
Brevicoryne brassicae	58	0,47	**0,51**	0,37	0,26	0,40	0,47	**4,50%**
Liriomyza huidobrensis	112	0,81	0,81	0,87	0,86	0,89	**0,95**	**14,60%**
Prodiplosis longifila	35	**0,77**	0,75	0,61	0,44	0,44	0,51	0,00%
Trips tabaci	53	**0,52**	0,49	0,45	0,44	0,42	0,42	0,00%

4 Conclusion

Automating the identification of pest insects in sticky trap images is highly desirable. However, this is a challenging task due to the difficulty in obtaining sizeble sets of training images. In this article we studied the usefulness of three generative models in synthesizing pest insect images (DCGAN, WGAN and VAE) in order to increase the training set and thus facilitate the induction of identification and detection models. In a series of experiments with images of seven insect species of interest for the Peruvian horticulture we demonstrated that the WGAN and VAE models are able to capture the variability of the images of such species. Additionally, it was found the synthetic data generated by such models can help to induce YOLOv5 detectors with significant gains in identification performance compared to not using synthesized data.

The present work can be extended with the use of novel generative models. One example of this is the model SAGAN [Zhang et al., 2019] that use self–attention layers in order to learn better spatial and structural information from the images. This model could be better able to learn features such as the shape of the insects. In some cases these features could be more relevant for distinguishing insect species than the texture or color features that are well learned by the generative models based on convolution layers, like the experimented in the present study.

Acknowledgments and Disclosure of Funding

The authors gratefully acknowledge Artificial Intelligence Group of Pontifical Catholic University of Peru (IA–PUCP) for the support with the computational infrastructure for the experimental part of the present study.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

Veronica Cañedo, Armando Alfaro-Tapia, and Jürgen Kroschel. *Manejo Integrado de plagas de insectos en hortalizas Principios y referencias técnicas para la Sierra Central de Perú*. Centro Internacional de la Papa, 2011.

Karlos Espinoza, Diego L. Valera, José A. Torres, Alejandro López, and Francisco D. Molina-Aiz. Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture. *Computers and Electronics in Agriculture*, 127:495–505, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

Jiayi Huang, Mengxi Zeng, Weixia Li, and Xiangbao Meng. Application of Data Augmentation and Migration Learning in Identification of Diseases and Pests in Tea Trees. In *2019 Boston, Massachusetts July 7- July 10, 2019*. American Society of Agricultural and Biological Engineers, 2019.

Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, 2021.

Wenyong Li, Dujin Wang, Ming Li, Yulin Gao, Jianwei Wu, and Xinting Yang. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Computers and Electronics in Agriculture*, 183, 2021.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018.

Chen-Yi Lu, Dan Jeric Arcega Rustia, and Ta-Te Lin. Generative adversarial network based image augmentation for insect pest classification enhancement. *IFAC-PapersOnLine*, 52(30):1–5, 2019.

A.T. (Ard) Nieuwenhuizen, J. (Jochen) Hemming, D. (Dirk) Janssen, H.K. (Hyun) Suh, L. (Lien) Bosmans, V. (Vincent) Sluydts, N. (Nathalie) Brenard, E. (Estefanía) Rodríguez, and M.D.M. (Maria del Mar) Tellez. Raw data from yellow sticky traps with insects for training of deep learning convolutional neural network for object detection, 2019. URL `https://data.4tu.nl/articles/dataset/Raw_data_from_Yellow_Sticky_Traps_with_insects_for_training_of_deep_learning_Convolutional_Neural_Network_for_object_detection/12707066/1`.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

Dan Jeric Rustia, Jun-Jee Chao, Jui-Yung Chung, and Ta-Te Lin. An online unsupervised deep learning approach for an automated pest insect monitoring system. In *2019 ASABE Annual International Meeting*, 2019.

Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008.

Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1571–1580, 2020.

Dujin Wang, Yizhong Wang, Ming Li, Xinting Yang, Jianwei Wu, and Wenyong Li. Using an improved Yolov4 deep learning network for accurate detection of whitefly and thrips on sticky trap images. *Transactions of the ASABE*, 64(3):919–927, 2021.

Qing Wang, Sanjeev R. Kulkarni, and Sergio Verdu. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5): 2392–2405, 2009.

Chunlei Xia, Tae-Soo Chon, Zongming Ren, and Jang-Myung Lee. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. *Ecological Informatics*, 29:139–146, 2015.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019. https://arxiv.org/abs/1805.08318.

Yuanhong Zhong, Junyuan Gao, Qilun Lei, and Yao Zhou. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors*, 18:1489, 2018.

Huiling Zhou, Haiwei Miao, Jiangtao Li, Fuji Jian, and Digvir S. Jayas. A low-resolution image restoration classifier network to identify stored-grain insects from images of sticky boards. *Computers and Electronics in Agriculture*, 162:593–601, 2019.

A Generative model

A.1 Conditional Deep Convolutional Generative Adversarial Network (DCGAN)

This model, initially proposed in Radford et al. [2016] and Goodfellow et al. [2014], is composed by two modules: a generator and a discriminator, which are trained in an adversarial game. The goal of the generator is to synthesize images that are similar to the original ones and the discriminator tries to discriminate the original images from the synthetic ones. The output of the discriminator is used as feedback to the generator to improve the quality of the images and fool the discriminator so this can not discriminate if an image is real or fake. Both modules are trained jointly in order to facilitate the task to the generator at the beginning of the training. If the discriminator is pre-trained to discriminate the images, the generator would not have the opportunity to learn to produce good quality images. Fig. 5 shows the high level architecture of the DCGAN with its integrating modules.

The generator is composed of five transposed convolutional layers. Each layer, except the last one, is followed by a batch normalization layer and ReLu as activation function. In the last it is used the tanh activation function. The input of this module is a vector of length 64 of random noise concatenated with a one-hot label vector of the image class (insect specie) to be generated.

The discriminator is composed of four convolutional layers, each one followed by a batch normalization layer and LeakyRelu as activation function. In this case the inputs of the module are the original images or the fake ones concatenated with the one-hot label vector as channels.

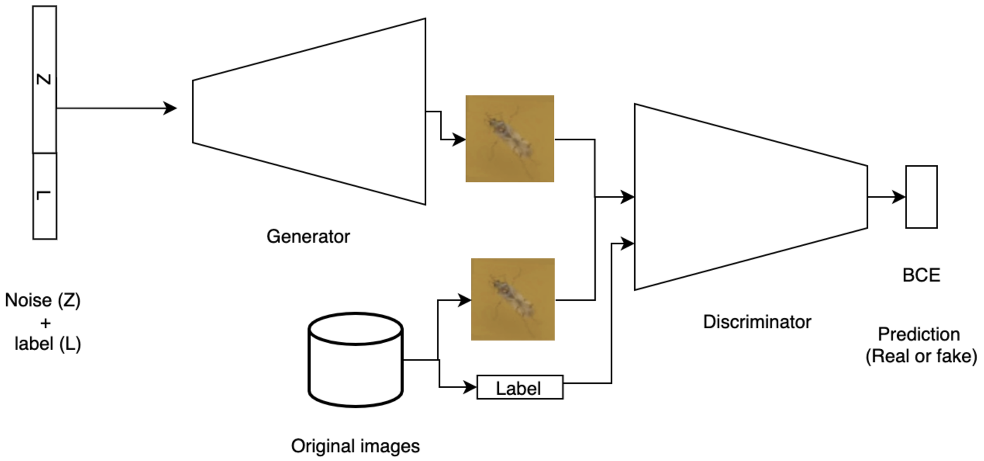

Figure 5: High level architecture of the generative DCGAN model used to synthesize images of pest insects.

A.2 Conditional Wasserstein GAN (WGAN)

This model, originally proposed in Arjovsky et al. [2017], replaces the second module of the traditional GAN, (the *discriminator*) with a module named *critic*. Instead of classifying the images into real or fake labels, the critic outputs a numerical score that indicates how realistic an image is. This score brings more information to the generator to improve the image generation. In order to ensure that the output score is valid, the critic needs to be 1-Lipschitz continuous (1-L). In other words, the norm of the gradient should be at most 1. A penalty factor is added to the loss function to enforce 1-L continuity, this factor is calculated with the interpolation between real and generated images. Fig. 6 shows the high level architecture of the WGAN model.

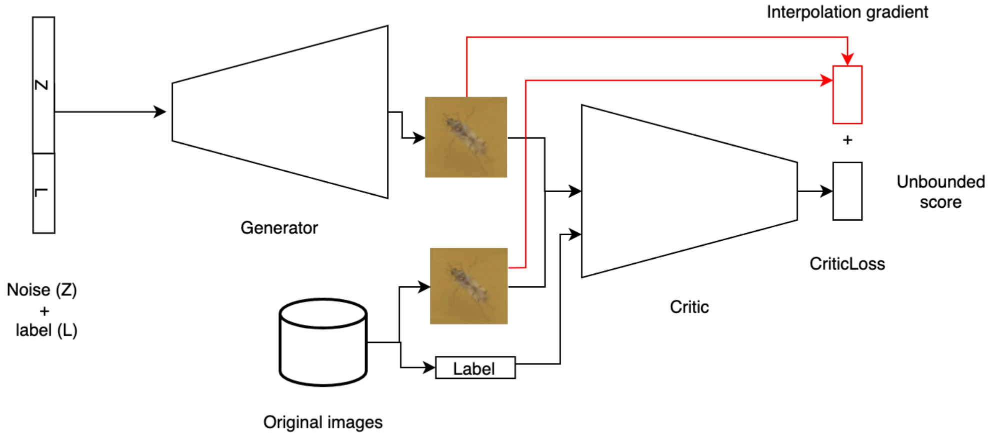

Figure 6: High level architecture of the WGAN [Arjovsky et al., 2017] model used to synthesize images of pest insects.

A.3 Conditional Variational Autoencoder (VAE)

This model, initially proposed in vae, is composed of an encoder and a decoder. The encoder has as input the original image concatenated with the one-hot label vector of the image class. It encodes the image to a set of normal distributions. Then, a vector is sampled from these distributions and concatenated again with the one-hot label vector in order to be passed to the decoder. The decoder reconstructs the image from this vector. The decoder is used to generate images from a random noise vector concatenated with the one-hot label vector of the target class. Fig. 7 shows the high level architecture of the VAE model.

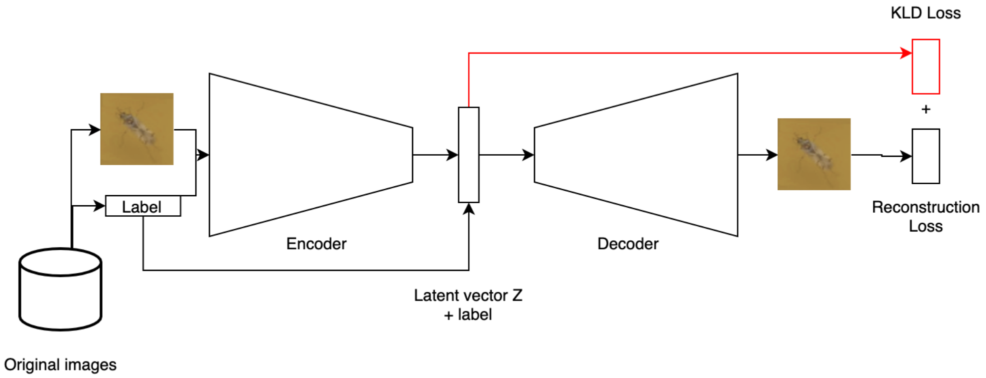

Figure 7: High level architecture of the model VAE [vae] model used to synthesize images of pest insects.

B Detection model (YOLOv5)

In the Figure 8 we show the high level architecture of the model YOLOv5 with its main modules, back–bone with Cross Stage Partial Network (CSPNet) [Wang et al., 2020], Path Aggregation Network (PANet) [Liu et al., 2018] as the neck for feature aggregation and the head.

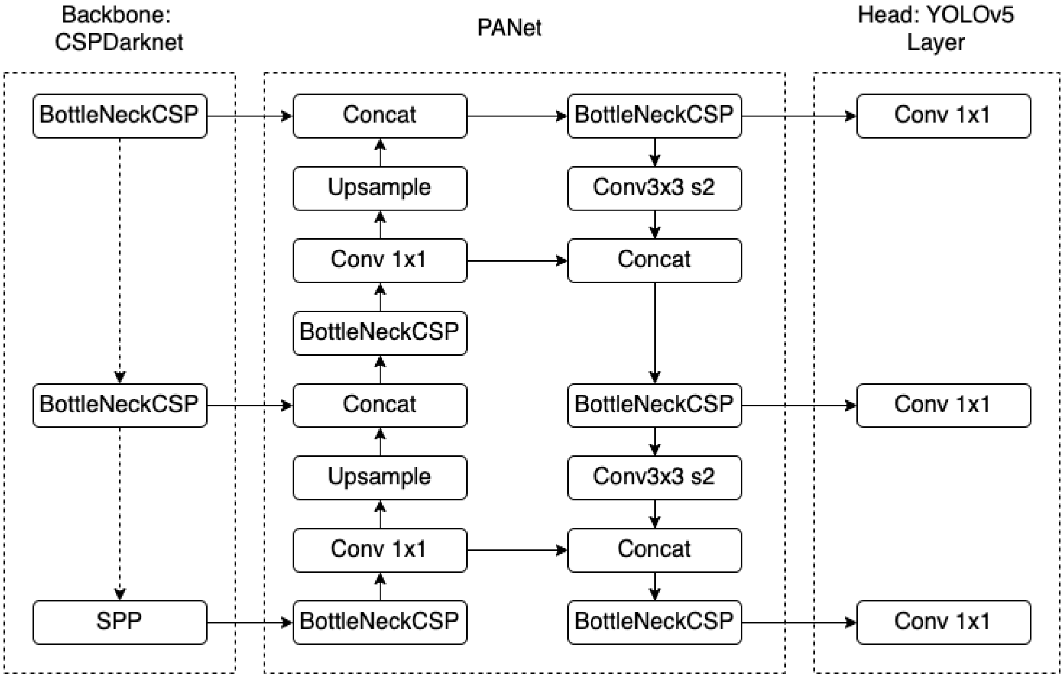

Figure 8: YOLOv5 Architecture [Jocher et al., 2021] (SPP: Spatial Pyramid Partial Network, CSP: Cross Stage Partial Network, Conv: Convolutional layer)