Bees Detection Using YOLOv5

Hady Khaled Mohamed
Faculty of Engineering
University of Ottawa
Ottawa, Canada
hhamm043@uottawa.ca

Mohamed Sayed Hussein
Faculty of Engineering
University of Ottawa
Ottawa, Canada
mhuss073@uottawa.ca

Sarah Hussien Mohamed
Faculty of Engineering
University of Ottawa
Ottawa, Canada
smoha279@uottawa.ca

Abdallah Medhat Rashed
Faculty of Engineering
University of Ottawa
Ottawa, Canada
arash015@uottawa.ca

***Abstract*—Machine Learning is now an important approach used in many aspects of our life. Particularly now a days it is heading to computer vision applications. Computer Vision is one of the important fields in Artificial intelligence. That allows the computer-based systems to interpret and understand the images. It plays a significant role in solving challenges related to object detection problems. The main purpose of object detection is to locate and identify the needed targets. In this paper we will display the object detection techniques and experiments applied to help in detecting bees.**

Keywords—Computer Vision, Object Detection, Object Localization, Bees Detection, YOLO V5.

I. Introduction

"Over the past 50 years, the number of crops that depend on pollinators has tripled." [1]. Bees have a vital role in keeping our life, we need much more food every day as the world's population is increasing massively every day. Everyone considers the bee as a source for honey. But it is wrong. Bees' products can be helpful in many sectors like Food, Healthcare, cosmetics ...etc. Bees are extremely helpful to keep the biodiversity, so they are considered as a natural biosensor either in labs or in fields. We can measure the functioning wellness of an ecosystem by the number of bees and the quality of products they produce, as mentioned by José Graziano da Silva, Director-General of the Food and Agriculture Organization. Bees' detection problems can help monitor their natural habitat which can be helpful in behavioral studies. Which gives the scientific community more insights on how to help bees to increase their productivity. And know the places where the bees' huddles and groups. Supplying a way to detect the number of bees leaving and returning to the colony might help in bees migrations control.

Implementing, using, and exploiting computer vision techniques to help the scientific community, in their studies towards helping nature and helping the Competent authorities like the FAO reaching sustainable food production, is a duty and an obligation towards humanity.

II. Related Work

In [2], they presented a systembased on a deep convolutional neural network classifier used to detect bees on 48x48 pixel images, supplied a method for getting data from images having bees for the aggregation of classification results, they compared colour models and suggested a tournament-based technique.
In [3], they took the advantage of the Deep Learning models and their ability to work with images without features manually extracted. They implemented multiple models. MLP was not an excellent choice for this problem. While CNN got a better performance and recognition capabilities. They trained the model over 50k images, and the model had the ability to stabilize just after 50 epochs.
In [4], they made use of deep learning, specifically Convolutional Neural Networks (CNN). The YOLOv3 model is used here. they achieved reliable results (Precision 86.6% Recall 88.9% F1 87.8%).
In [5], they used Convolutional Neural networks. ConvNet is an enormously powerful choice in their case. They were able to localize multiple bees in an overly complex backgorunds. To train their VGG-based network, they assigned a class label tensor for input images of size 42x42x4.
In [6], three CNN models employed to find bumble bees at the species level all had equal accuracy rates. Wide-ResNet achieved the greatest test accuracy (91.7%), followed by InceptionV3 (91.6%) and ResNet100(91.6%).

III. Data

Using a suitable data set is the key to solving this problem. Bees' detection is very sensitive problem, as there are many challenges, as the bees' sizes and the places we can find them that may be overlapped in color with the bees. So, we use a huge amount of data, so we collect our dataset from the inaturalist.org [7]. This dataset has 700 images of bees that have been uploaded by Inaturalist users for the purposes of recording the observation and identification. We only used images that their users have licensed under CC0 license. We can see some samples in figure (1). The data is composed of multiple images with single and multiple bees. We have a lot of diversity of bees with different sides, rotations, and sizes. Which is an important aspect to consider. As if the model is trained only on a big bee's sizes in the image it will not be robust enough to solve the main problem.

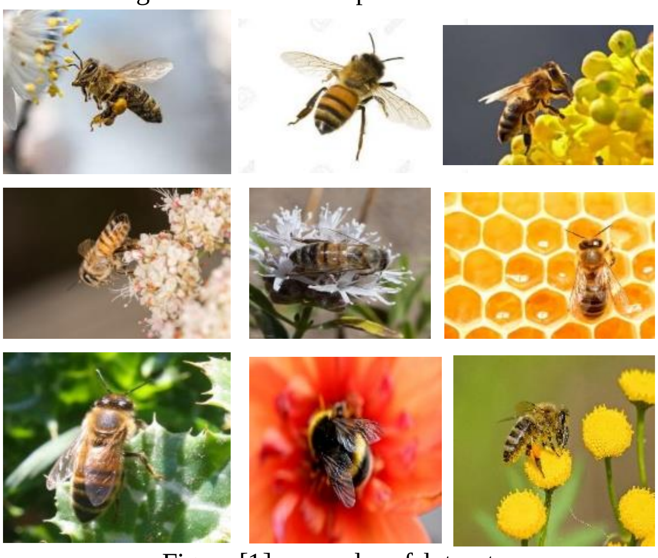

Figure [1] examples of dataset

Then we used IBM cloud annotation [8] to annotate images by adding a bounding box around the target object in the image to use it in our model so each image has a ".txt" file with the border dimensions. Then split our dataset into 600 images as a training set and 100 images as a validation set before sending our dataset to our model. After the first usage of the data, we saw that the data is not enough and missed a lot of details of the images. Using dataset of 700 image considered to be small dataset in such detailed problems. So, we had to implement data augmentation approach of Vertical flips, Horizontal flips, Vertical-Horizontal flips and 90º Rotation. The augmented samples are presented in figure 2. So, we managed to increase the data to 2500 annotated images. Ready to be used in the retraining process.

Figure [2] examples of data augmentation

Now we adjusted the 2500 images to be of size 256x256. Then using another splitting for new dataset after data augmentation so split our dataset into 2000 images as training set and 500 images as validation set. By increasing the amount of training data, we believe that we can get over the challenges of our dataset. The dataset contains multiple images with different sized bees, it is important multiple different sized bees, to make sure that the model is robust enough to be implemented in a complex environment.

Figure [3]. Small Sized bee

In figure 3 we can see a sample of small sized bees, while in figure 4 and figure 5 we can see medium and large bee sized.

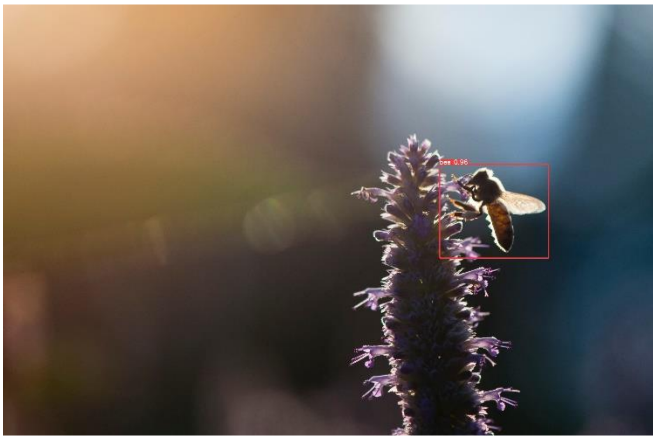

Figure [4]. Medium sized bee

Figure [5]. Large sized bees

Obviously, the image sizes are different, but later during the experiments they will all be of the same size 256x256.

IV. Methods

We applied data augmentation as it's a set of techniques for producing additional data from current data to artificially increase the amount of data available. Making small changes to data or utilizing models to produce additional data are examples of this. So, we used the following data augmentation techniques (Vertical Flips, Horizontal Flips, Vertical-horizontal flips, 90 Rotation) to:

- Improving the accuracy of model prediction.
- Increasing the amount of training data in the models.
- Avoiding data scarcity to improve models.
- Creating diversity in data and decreasing data overfitting (a statistical error in which a function correlates too closely to a small number of data points).
- Enhancing the models' ability to generalize.
- Lowering the cost of data collection and labelling.
- Allows for the prediction of unusual events.
- Prevents issues with data privacy.

After developing data augmentation, we noticed that the model performs better and more accurately.

YOLO is an abbreviation meaning You Only Look Once. We're using Ultralytics' Version 5, which was released in June 2020 and is now the most powerful object recognition algorithm available. For our problem we decided to use the pretrained model YOLOV5m instead of using YOLOV5s that will be more suitable for a precise problem which depends on many detailed features. It's a manufacturer convolutional neural network (CNN) that accurately recognizes objects in real time. This method processes the entire image with a single neural network, then divides it into pieces and predicts bounding boxes and probabilities for each part. The predicted probability is used to weigh these bounding boxes. In the sense that it produces predictions after only one forward propagation through the neural network, the method "just looks once" at the image.

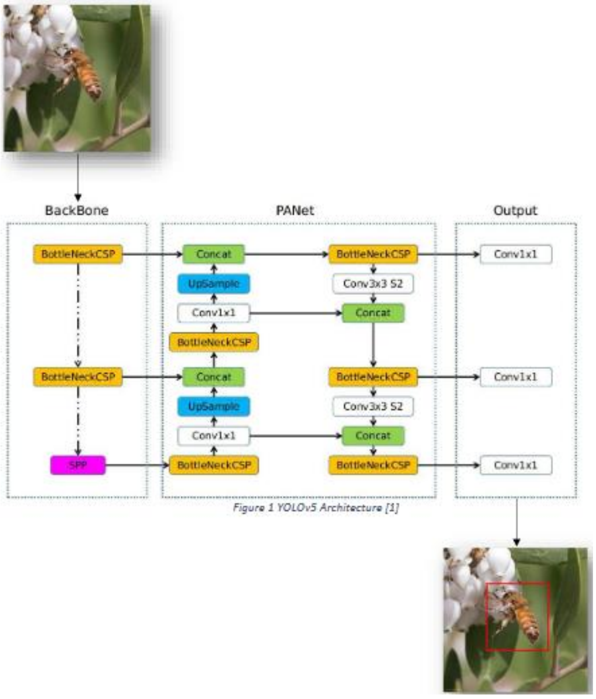

Figure [6]. YOLOv5 Architecture

The model's backbone it is the element dedicated to taking the input image and extracting feature maps from. This is an important step in any object detector because it is the principal structure in charge of gathering contextual information from the input image and abstracting that information into patterns. In YOLOv5, we experimented with replacing the previous backbone with two different alternatives. ResNet [10] is a well-known structure that utilizes residual connections to lessen the effects of diminishing returns in deeper neural networks. DenseNet [11] makes use of similar connections to preserve as much information as possible safe as it flows through the network. Implementing these structures in place necessitates breaking them down into their fundamental blokes and ensuring that the layers communicate appropriately.

The structure between the head and the backbone (see Figure 6) is referred to as the "neck," and its purpose is to aggregate as much information gathered by the backbone as possible before it is fed to the head. This structure is critical for transmitting small-object information to higher levels of abstraction by avoiding it from being lost. It accomplishes this by up sampling the feature maps' resolution once again, allowing distinct layers from the backbone to be aggregated and impact the detection step once more. The model's head oversees taking feature maps and inferring bounding boxes and classes from numerous aggregated feature maps from the neck. Other than the parameters it receives, this structure can be left alone because it is a fundamental component of the model that does not have the same impact on small object detection as the other elements. Other factors, on the other hand, can influence the performance of small object detection. Aside from the input image size, the depth and width of the model can be changed to modify which part of the network receives much of the processing. The way layers are joined in the neck and head can also be altered manually.

Let's compare between YOLOv5 small and YOLOv5 medium architectures which we have used in our experiments, the size of yolov5s is only 14 Megabytes, while the size of yolov5m 41 megabytes. the inference speed of yolov5s 2.2-mile seconds and the inference speed of yolov5m is 2.9-mile seconds. To make the model well detected with good performance and accuracy So, we used these training batches as we trained many types of all bees whether large or small, far, or close in the training data as shown in the following image.

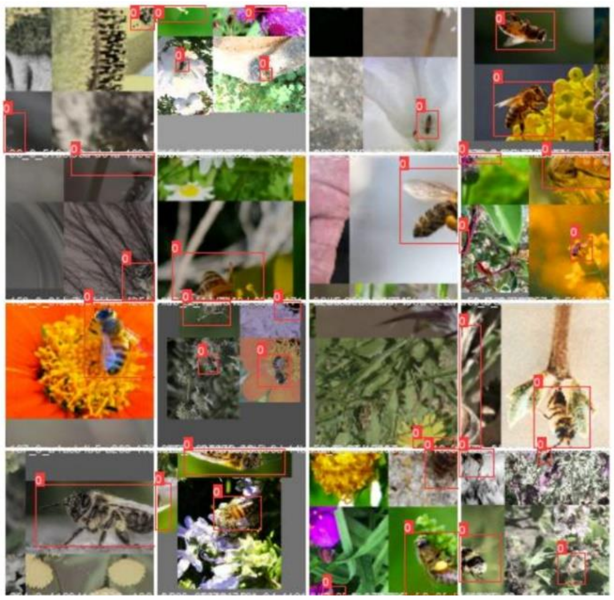

Figure [7]. Training batches example

V. EXPERIMENTS

Our project based on multiple experiments. First experiment considered to be the baseline for our later work and to figure out the weaknesses of our system so that we can start to improve our "Try and Error" approach. We used 700 images with no augmentation techniques and a small YOLOv5 Architecture, therefore our results were not satisfactory. We only got 50% mAP, after investigation and error analysis, we got the main challenges that we have to deal with, like:

- Bees are small size creatures.
- The background and the bees color may be overlapped.
- Flowers and bees are the same hue.
- Some of the training images were not of satisfactory quality.

So, our model can't detect the bees and got confused, as shown in figure 8.

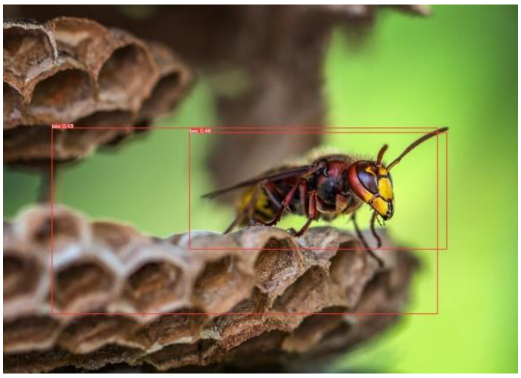

Figure [8]. First Experiment (Error analysis)

In the second experiment we tried to address all these challenges by:

- Increasing the number of clear images in the training set.
- Include variety of bee sizes.
- Apply data augmentation techniques.
- Change from small to medium YOLOv5 Architecture.

Also, we adapted the hyper-parameters of the model as follows:

- Image size = 256
- Batch size = 128
- Number of Epochs = 200
- Weights = yolov5m.pt

The model achieves a high mAP of 99% using the techniques discussed above. When compared to the first experiment, which had mAP of 50%, this is a significant improvement. It proves the model's significant improvement and robustness. It also achieves a precision of 98%, showing that approximately all the results are accurately recognized, and a recall of 99 %, showing that the most relevant results are found.

VI. Results

Table 1 Result of Experiments

	mAP	Precision	Recall
Experiment 1	50%	47%	49%
Experiment 2	99%	98%	99%

Our experiments ended up with a high mAP of 99 %. This is a substantial number in comparison to the early trials. It shows how much the model has improved and how stable it is. It also achieves a precision of 98%, showing that almost all the results are correctly recognized, and a recall of 99 %, showing that the most important findings are achieved as shown in table (1).

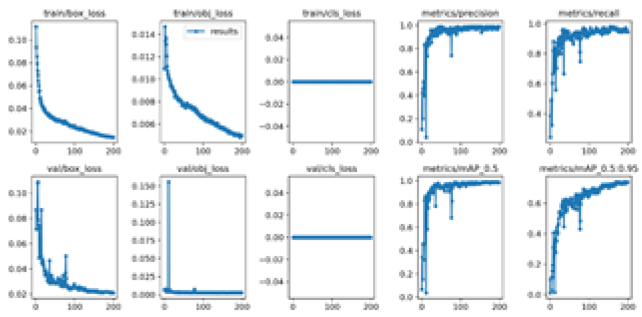

Figure 9. Metrics of Experiment 2

We used mean average precision as a metric to measure our model's performance in detecting bees. The mAP and loss curve shown in figure 9 show that the mAP increases as the number of epochs increases, despite the loss curve decreasing as the number of epochs increases. Finally, we got a mAP of 98.6% on the test set.

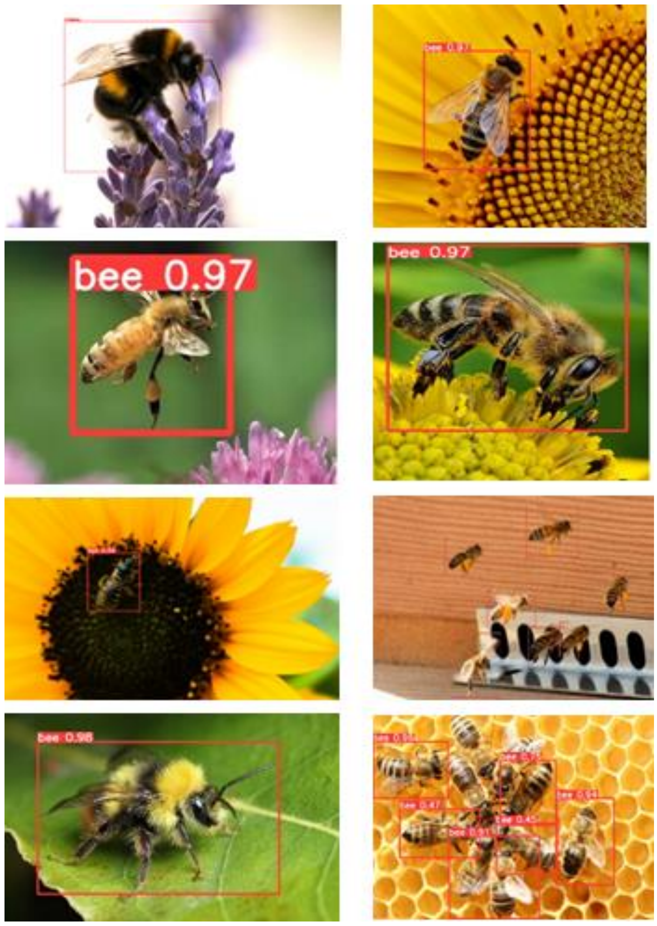

Figure [10] Model Output

As shown in figure 10, the model can now accurately detect bees; as we can see, it could differentiate the bee on the disc floret of a sunflower with a high confidence score, and it also did well with images with hues extremely like a bee's color, and it can detect multiple bees in the same image.

VII. Conclusion

Bees' detection problems can help monitor their natural habitat which can be helpful in behavioral studies. Which gives the scientific community more insights on how to help bees to increase their productivity. Since it might be hard to detect small objects in images. So, our approach has to be more accurate, we have to increase the data, we have to provide more suitable images with a lot of different sized bees. The model needed to be enhanced and get enough data to make the model well learnt. We increased our dataset as we are gathering more images and developing data augmentation techniques. We developed a bees detection model using pretrained model YOLOV5 m instead of using YOLOV5 s that it will act well on a precise problem which depends on many detailed features. mAP (mean average precision), loss curve are our metrics, we increased the number of epochs to show a well learnt model.

EFERENCES

[1] Marko, S. (2022). The importance of bees. Worldbeeday.org. Retrieved 9 February 2022, from https://www.worldbeeday.org/en/about/the-importance-of-bees.html.

[2] Dembski, J., & Szymański, J. (2018). Bees Detection on Images: Study of Different Color Models for Neural Networks. Distributed Computing And Internet Technology, 295-308. https://doi.org/10.1007/978-3-030-05366-6_25

[3] Tiwari. (2018). A Deep Learning Approach to Recognizing Bees in Video Analysis of Bee Traffic. ProQuest Dissertations Publishing.

[4] Tresson, P., Tixier, P., Puech, W., & Carval, D. (2019). Insect interaction analysis based on object detection and CNN. 2019 IEEE 21St International Workshop On Multimedia Signal Processing (MMSP). https://doi.org/10.1109/mmsp.2019.8901798

[5] Duan, L., Shen, M., Gao, W., Cui, S., & Deussen, O. (2017). Bee pose estimation from single images with convolutional neural network. 2017 IEEE International Conference On Image Processing (ICIP). https://doi.org/10.1109/icip.2017.8296800

[6] Spiesman, B., Gratton, C., Hatfield, R., Hsu, W., Jepsen, S., & McCornack, B. et al. (2021). Assessing the potential for deep learning and computer vision to identify bumble bee species from images. Scientific Reports, 11(1). https://doi.org/10.1038/s41598-021-87210-1

[7] iNaturalist. iNaturalist. (2022). Retrieved 13 April 2022, from https://www.inaturalist.org/.

[8] Cloud Annotations. Cloud.annotations.ai. (2022). Retrieved 13 April 2022, from https://cloud.annotations.ai/.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8691 LNCS(PART 3):346–361, jun 2014,from https://bibbase.org/network/publication/he-zhang-ren-sun-spatialpyramidpoolingindeepconvolutionalnetworksforvisualrecognition-2014 .

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December:770–778, dec 2015 ,from https://arxiv.org/abs/1512.03385 .

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. Technical report,from https://arxiv.org/abs/1608.06993 .