Bees Species Identification

Maya Dutywa
u23825848@tuks.co.za

Hanno Jacobs
u17000042@tuks.co.za

Abstract

The identification and classification of bee species plays an important role in determining ecological health of ecosystems, ensuring optimal crop pollination, sustainable agriculture, biodiversity studies and many more. The primary objective of our work is to create a computer vision model that is able to identify the species of a bee from an image, with a secondary objective of determining which image augmentation techniques are best for improving the computer vision model when applied to the bee image dataset. Our dataset consists of 1161 images of bees which we labelled for training our computer vision model. Our resultant computer vision model is evaluated using F1, Precision and recall metrics and is able to classify and identify bee species into eight common bee species found in Southern Africa. The resultant model provides high test-set accuracy on two of the species: *Xylocopa* (97% accuracy) and *Thyreus* (85% accuracy) with varying levels of accuracy between 28% and 67% on the six other bee species. The results of our image augmentation test show that the model is improved when using moderate levels of augmentation that include image shearing, height and width shift, image mixup and image copy-paste and found that augmentation techniques that result in significant colour changes result in model degradation.

1. Introduction

Bee identification can play a valuable role in the area of bee conservation efforts. Bees are essential for pollinating plants and ensuring a successful natural ecosystem.

The task of successfully distinguishing bee species from images has a number of challenges because of morphological differences between species, backdrops, and varied image quality. The time-consuming and specialised knowledge needed for existing manual identification methods emphasises the need for automated solutions. This project focuses on three key issues:

1. Obtaining a dataset of different species of bees that can be used to train a computer vision model
2. Improving this dataset using augmentation methods that can adjust to various image characteristics and bee morphological features.
3. Using this dataset to create a computer vision model that can be used to identify and classify different species of bees from images.

(Buschbacher et al., 2020) mentions that the studies on conservation and ecology are severely hampered by the challenging taxonomy and the shortage of specialists; currently there are no commercially available vision-based bee identification models that are able to distinguish between different species of bees. The primary goal of this project is to create a model that can take an image of a bee and identify what species/type it is. According to (Buschbacher et al., 2020) In comparison to conventional descriptor-based methods, Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in object detection and classification in images; in fact, they have even surpassed humans in classification accuracy. The project will make use of a convolutional neural net (CNN) to do object detection on images and distinguish between the species of bees given. The project's secondary goal is to identify the best augmentation methods training dataset for the CNN model.

2. Literature survey

More advanced computer techniques have been used in ecological research to increase the productivity and accuracy of species identification. Particularly, convolutional neural networks (CNNs) are deep learning models that have developed into incredibly powerful instruments for object detection and image classification. The integration of these models into ecological studies has great promise for conservation efforts, particularly in terms of detecting and safeguarding various species of bees, which are vital pollinators in numerous habitats. This review of the literature looks at the application of deep learning to ecological research, focusing on bee species.

Deep learning is a kind of machine learning that uses multi-layered neural networks (thus the term "deep") to identify intricate patterns in huge datasets. In multiple experiments involving pattern recognition, computer vision, and image processing, deep learning systems have outperformed standard ones. In numerous real-world applications and hierarchical systems, transfer learning and deep learning algorithms have been applied to pattern recognition and classification tasks (Gupta et al., 2022; Sohan et al., 2024). CNNs are a kind of deep learning model that work very well on tasks involving images.

Numerous research works have exhibited the effectiveness of CNNs in classifying different kinds of animals. Norouzzadeh et al. (2018), for example, achieved remarkably high accuracy when identifying animal species from camera trap images using deep learning. These uses highlight the possibilities of CNNs to transform ecological monitoring and conservation strategies.

The models in the "You Only Look Once" (YOLO) series showcase the most advanced methods for real-time object identification. YOLOv8 (Ultralytics, 2023), the most recent version, performs better than its predecessors in terms of speed and accuracy (Sohan et al., 2024). Because of its architecture, YOLOv8 is a good option for classifying and detecting items in photos, which is useful for distinguishing different kinds of bees. The image is divided into a grid by the YOLO model, which then forecasts probabilities and bounding boxes for each grid cell. This method gives the model the ability to identify numerous items in an image at once, which is important for ecological studies because photos frequently include multiple individuals or species.

Another important method for diversifying the training dataset without really adding new photos is data augmentation. This entails giving the already-existing images a variety of transformations, including rotations, shearing, brightness tweaks, and width and height changes. According to Bang et al. (2020), data augmentation can aid by producing a more reliable training dataset, which can help resolve the issue of identical objects appearing differently in terms of size and shape depending on the camera's position and altitude.

In summary, deep learning models—in particular, CNNs and YOLO architectures—have a lot of potential to improve bee species identification and conservation. These models can reach great accuracy and efficiency by utilising methods like data augmentation and transfer learning, which can aid in ecological research and conservation activities. Technology will probably become more and more important in protecting biodiversity and sustaining healthy ecosystems as it develops.

3. Methodology

To train the bees species identification model images of the bee species in question (*Allodapula, Apis-mellifera-scutellata, Braunsapis, Lasioglossum, Meliponula, Seladonia, Thyreus, Xylocopa)* 1161 images are scraped from the internet that comprise images of the eight classes of bees. This data is then labelled using Roboflow's image labelling tool (Dywer et al., 2024).

Initial preliminary testing includes testing a zero-shot model that has already been trained on a huge number of common objects. The hope for this approach is to see if the zero-shot model is able to identify and classify the different bee species without any training on any extra data. The tested zero-shot model is a model called YOLO-world (Cheng et al., 2024).

The main testing will include using the dataset obtained to train a YOLOv8-nano (Ultralytics, 2023) CNN model. These images are then separated into a 70%/20%/10% train-validation-test split for cross-validation.

The models generated is then evaluated to get the following metrics:

- Training metrics
    - Train loss
    - Validation loss
    - Classification loss
    - Distributional Focal loss
    - Precision
    - Recall
    - mean Average Precision 50% confidence (mAP50)
    - mean Average Precision 50%-95% confidence (mAP50-95)
- Evaluation metrics
    - F1-confidence curve
    - Precision-confidence curve
    - Precision-recall curve
    - Recall-confidence curve

These metrics give a good understanding of the performance of the model, that indicate the model's ability to generalise to never before seen data, such as the data in the test dataset.

The next step is to apply augmentation techniques to the training dataset and see if the training and evaluation metrics improve. The augmentation techniques used are:

- Image shearing (Khalifa et al., 2022) where the image is sheared along different axes (x, y, etc) to allow for the underlying properties of the image to be learnt even when the shape is distorted.
- Image mixup (Bao et al., 2023) where the bounding boxes of one image are overlaid over the bounding boxes of another. This helps with learning to distinguish between different species of classes. This is very helpful for classification tasks like distinguishing between different species of bees.
- Image copy-paste (Ghiasi et al., 2021), this augmentation technique copied bounding boxes from one image to another image to allow for the model to learn different backgrounds and learn occlusion of objects.
- Image cropping (Takahashi et al., 2019) allows for the model to learn the central features of the image and prevents background distractions from reducing model performance.

Applying these augmentation techniques to our dataset should provide us with a dataset that is increased in size and variability which should also improve model generalisation (Shorten et al., 2019).

4. Exploratory data analysis

Our bee identification and classification task requires obtaining open-source images of the eight classes of bees in question in a natural scene and then subsequently using an open source labelling tool such as Roboflow (Dwyer et al., 2024) to draw labelled bounding boxes around the bees. The resulting dataset will be a realistic representation of the types of bee images that researchers would expect to use our bee identification and classification model on (for ecological purposes). This data is then finally used to train the selected YOLOv8 model in question.

We were unable to find any open-source datasets that contain the bee species we were interested in identifying. Therefore, in order to obtain enough images of the different species of bees a web-scraping application is built in Python to search for the name of the bee species and then download an arbitrary number of images from the search engine. This application is used to download the dataset of images which are then manually labelled in Roboflow. Figure 1 shows a summary of the number of instances of each class after labelling the images in the dataset. The total number of instances of each of the classes in the dataset is shown along with the average normalised size of the bounding boxes in the dataset in figure 2. These show that the classes are somewhat unbalanced and more images of the Seladonia and Allodapula class should be added to create a dataset that is more equal in the number of instances of each class (we had difficulties in finding unique images of these classes using the web scraping tool).

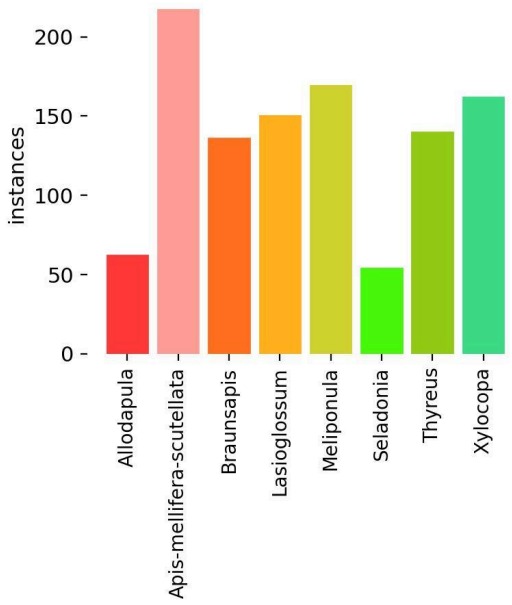

Figure 1: Label instances

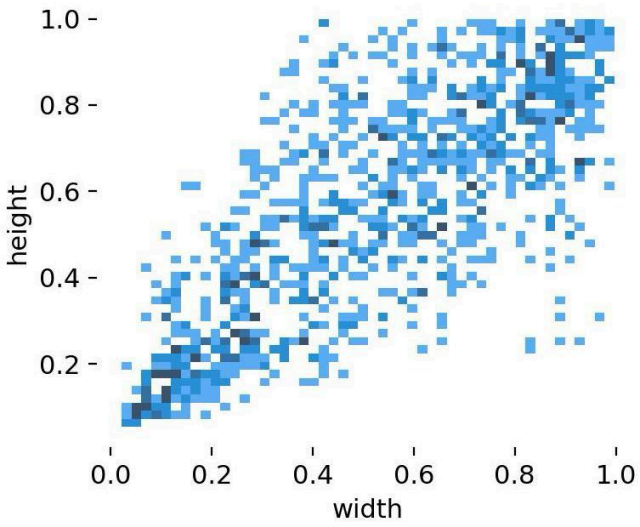

Figure 2: Normalised label sizes

The final dataset used for training consists of 818 images in the train set, 229 in the validation and 114 in the test set. This dataset that consists of the 1161 total images and labels is 52 MB in size which is stored in local storage on the PC that is used to train the model. This dataset is relatively small and results in quick training of the YOLOv8 model.

Initial preliminary testing on the dataset using the YOLO-World model, as shown in figure 3, results in most bees in our test set mosaic being identified as *Lasioglossum* (except for one which is *Xylocopa* and *Allodapula*) which is not correct. Therefore, this zero-shot model shows no promise for practically identifying different species of bees.

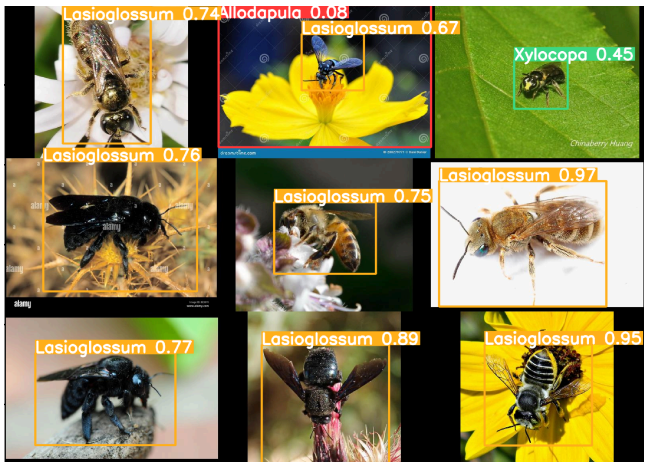

Figure 3: YOLO-World inference results

Data Augmentation

The dataset is also augmented using the following augmentation steps, as discussed in the methodology section: image shearing, image mixup, image copy-paste, and image cropping which are used to increase the model's ability to generalise and prevent overfitting. The augmented data visualisation used is built with Streamlit. The augmentation visualisation tool is shown below in figure 4.

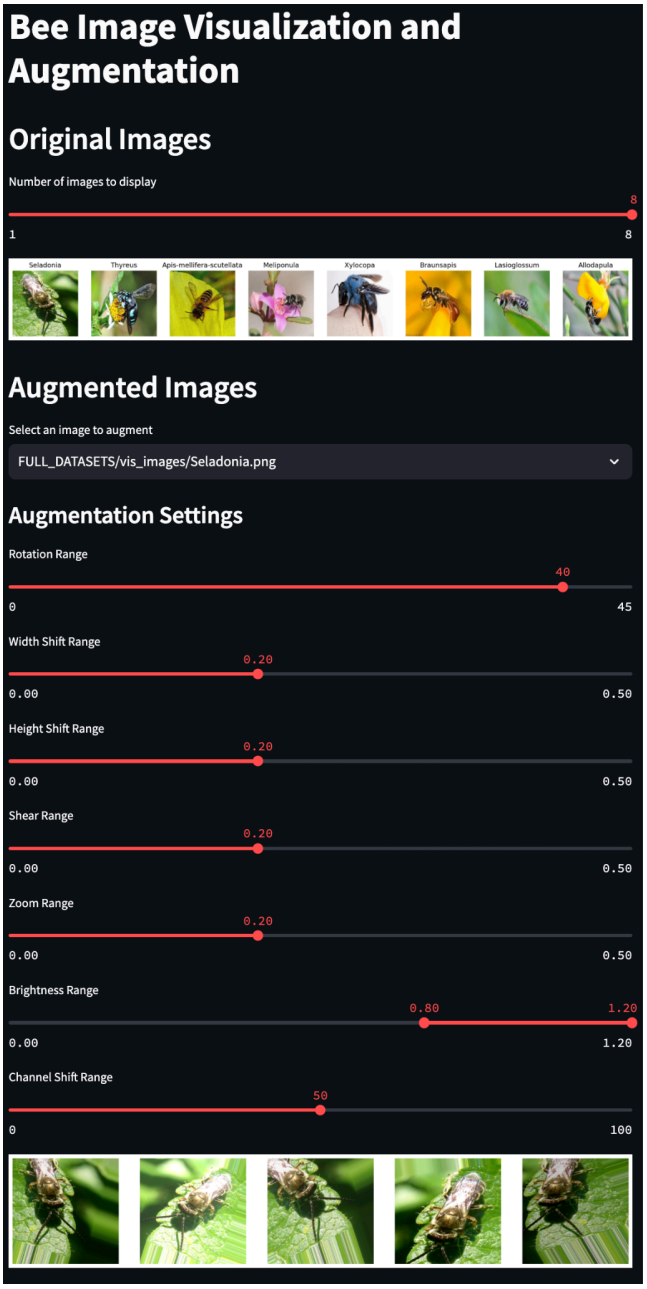

Figure 4: Streamlit augmentation visualisation tool

The data augmentation visualisation shows how several augmentation methods are applied to a chosen picture. Users can pick an image from the dataset, and the program will create augmented versions of the image by applying rotation, zooming, and horizontal flipping, to get an idea of the augmentation that takes place on the dataset in the training process. Figure 5 shows another example of the augmentation done on another image in the dataset.

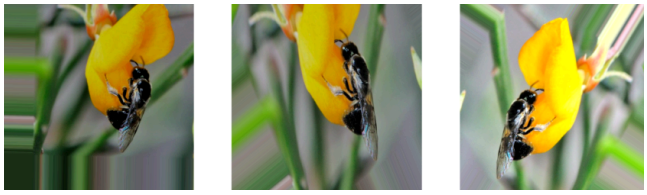

Figure 5: Augmentation example

5. Modelling and Visualisation

The training and modelling of our CNN model on the augmented dataset of 818 training images, 229 validation images and 114 test images allows us to cross-validate the model's performance and its ability to generalise to unseen data (in the case of the test-dataset).

The training results below show the progression of the model's training metrics during the 700 epochs of training. The model's batch size is 32 and the images in the dataset are resized to 640 x 640 resolution for the training process. The learning rate is 0.01 and the momentum for the training is 0.937 with the Adam optimizer.

The ideal training for a computer vision object detection model would result in the model's train and validation loss decreasing to monotonously at similar rates, with the validation loss and the train loss curves staying close to each other throughout the training process.

The training metrics in figure 6 show decreases in the loss parameters. This indicates that the model could still be "improving" during the training process however, the increases in the precision and recall metrics stagnate between 100-200 epochs which is contrary to what is expected since the loss is still decreasing, this shows that the decreasing loss metrics may only be a result of overfitting that doesn't actually contribute to model performance and generalisation. Overall the model could still actually benefit from more training epochs (this will be done after the training dataset is further improved).

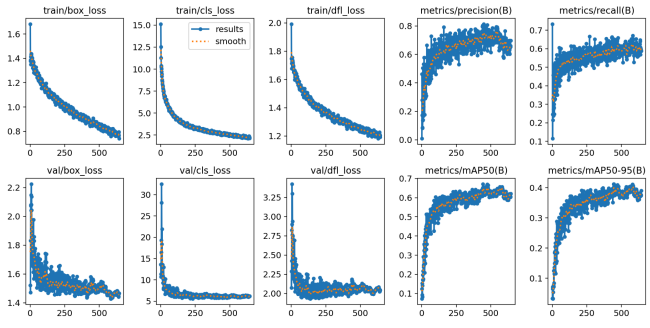

Figure 6: Training metrics

Precision curve

The precision of a model on a class indicates the model's ability to avoid false-positive (FP) detections. The precision-confidence curve in figure 7 shows the model's ability to avoid making a detection when there is nothing there. The ideal precision-confidence curve would give high precision, even at low

confidence levels. This would result in a curve that is high and dropping off on the left at very-low confidence levels. The ideal model should have a 1.00 precision value at as low a confidence level as possible, indicating that the model doesn't have FP detections, even when it is not very sure of the detection. Our model has good precision even at low confidence levels on some of the classes such as *Seladonia* and *Thyreus*. This means that the model is able to avoid making incorrect detections for those classes, even at low confidence levels.

The performance of the model drops off dramatically for several of the classes at an approximately 0.9 confidence level. This is indicative of a model that needs more well-labelled training examples. The curves for all the classes should also be smooth and monotonous, which indicates that the model steadily improves at higher confidence levels.

$$Precision = TP/(TP + FP)$$

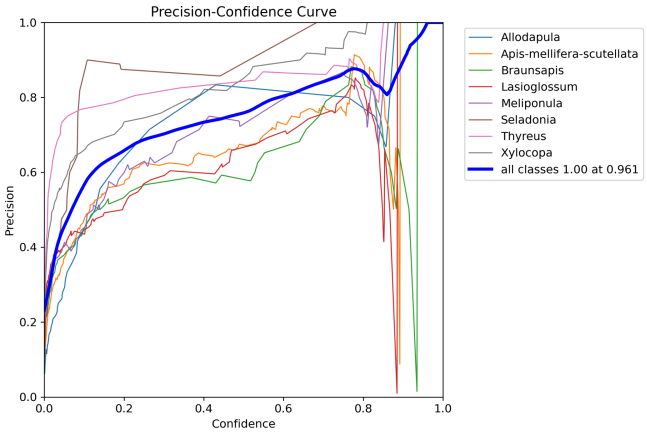

Figure 7: Precision-Confidence curve

Recall curve

The recall of a model on a class indicates the model's ability to avoid false-negative (FN) detections. This means that the model is able to avoid missing true detections. The ideal recall-curve would result in high recall even for high confidence levels, which would give a curve that is high and only drops off at very high confidence values. The ideal recall curve should also be smooth and monotonous.

Our model has good recall performance for *Thyreus*, *Xylocopa* and *Allodapula* classes, as shown in figure 8, which indicates that the model rarely misses a detection where a true case of those bees are present, whilst struggling for other classes, especially the *Meliponula* class. These classes' recall performance can be increased by adding more well-labelled examples of those bees to the dataset.

$$Recall = TP/(TP + FN)$$

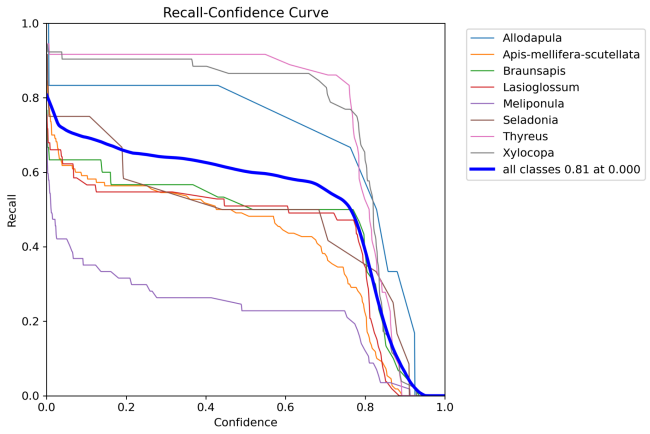

Figure 8: Recall-Confidence curve

Precision-recall curve

The ideal computer vision model for our application would have high-precision and high-recall for all of the classes, which is indicative of low false-positives and low false-negatives. This would mean that the ideal curves we would expect would be high and to the top-right corner of the plot. The classes that perform well, from this metric as demonstrated in figure 9, are the *Xylocopa* and *Thyreus* classes with decent precision-recall performance from the *Allodapula* and *Seladonia* classes, with especially poor performance in the *Meliponula* class. These classes' precision-recall performance can be increased by adding more well-labelled examples of those bees to the dataset.

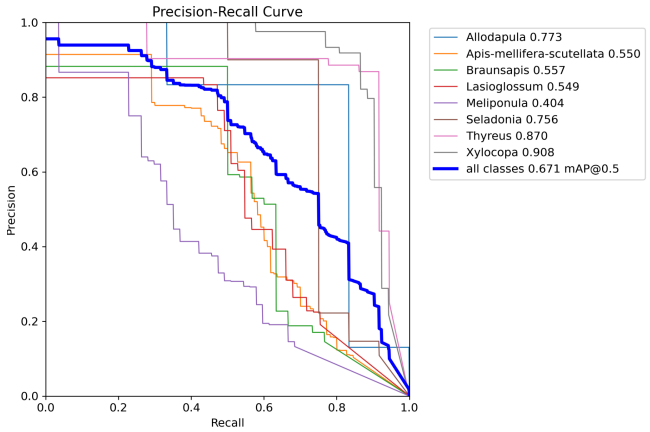

Figure 9: Precision-Recall curve

F1-confidence curve

The F1 curve in figure 10 shows the confidence levels for each class to obtain a given F1 score, which is a combination of the model's precision and recall which shows the model's performance on each class. Ideally the curve should be as high as possible, with the flat plateau-like shape that it has for the "all classes" curve. The jagged shape of the curves for most classes is indicative of a low number of training examples for each bee type; incorrect labelling; or class imbalances (unbalanced number of examples for each class). These can all be improved to a great extent by adding more well-labelled examples to the dataset especially for the classes with poor performance such as the *Meliponula* and other classes that fall below the average line. The

curves for the *Xylocopa* and *Thyreus* classes are relatively high, which indicates that the model is good at identifying and classifying those bee species in the dataset.

$$F1 = (2 \times Precision \times Recall)/(Precision \times Recall)$$

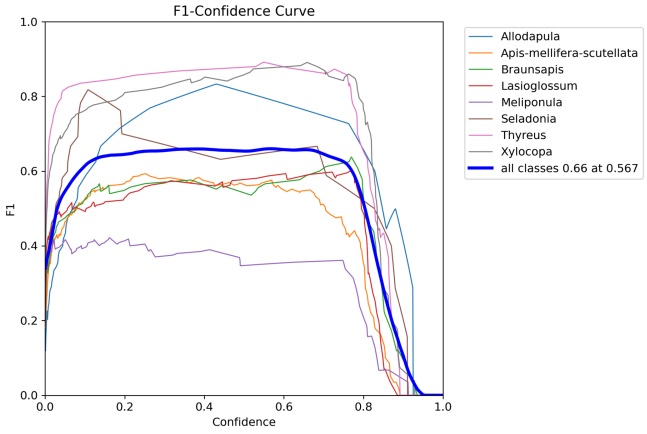

Figure 10: F1-Confidence curve

Confusion matrix

This confusion matrix in figure 11 shows that there is poor precision for many of the classes that is indicative of either low data quantity for certain classes, or poor labelling of images (both of these are likely the case seeing as the bee species look so similar that labelling is very difficult, and can lead to incorrect ground truths). The diagonal of the matrix should consist of numbers that are very close to 1 (this indicates recall), however the diagonal only has high values for two of the species: *Xylocopa* (97% accuracy) and *Thyreus* (85% accuracy) with varying levels of accuracy between 28% and 67% on the six other bee species. This is primarily due to the uniqueness of the *Xylocopa* and *Thyreus* bees, with their blue colour (for male *Xylocopa* bees) and the blue-spotted with unique wings for the *Thyreus* bees. The secondary reason for the good performance of the model for those classes is the relative simplicity of labelling those bees since they are very distinct from the other bee species in question and therefore there is expected to be a very low percentage of bees that are incorrectly labelled in the dataset. Incorrect labelling for other bees in the dataset is suspected since our project made use of lay persons in the field of bee identification.

Figure 11: Confusion matrix

The metrics of the model provide confidence that the model is in fact classifying the different species of bees well for the given data.

The model gives its best F1-confidence value of 0.66 at 0.567 confidence level. This could be improved given that many other YOLOv8 models can have this number in the +-0.8 range, however the relative similarity of the visual characteristics of the different species of bees mean that we are satisfied with this number, given that we have already employed all the augmentation techniques that are commonly used in computer vision. An example of the model's predictions on all of the classes is shown in figure 12.

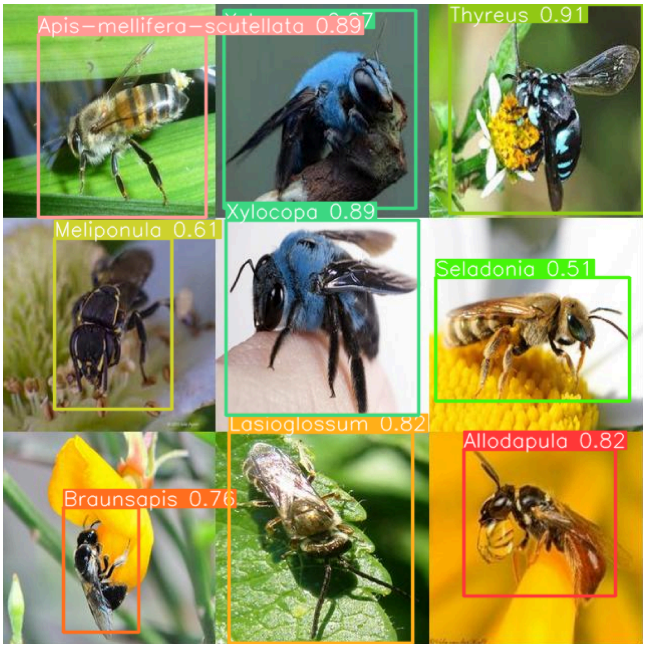

Figure 12: YOLOv8 model inference results

Bee Species detector - Streamlit deployment

The bee detection and classification model is deployed with Streamlit, as shown in figure 13, so that any image of a bee can be given to the detector and a confidence threshold can be selected for inference and the bee will then be classified into its species. This model can in future be deployed with the use of a smartphone app that will allow for many bees to be identified by volunteers and thereby capture valuable ecological information.

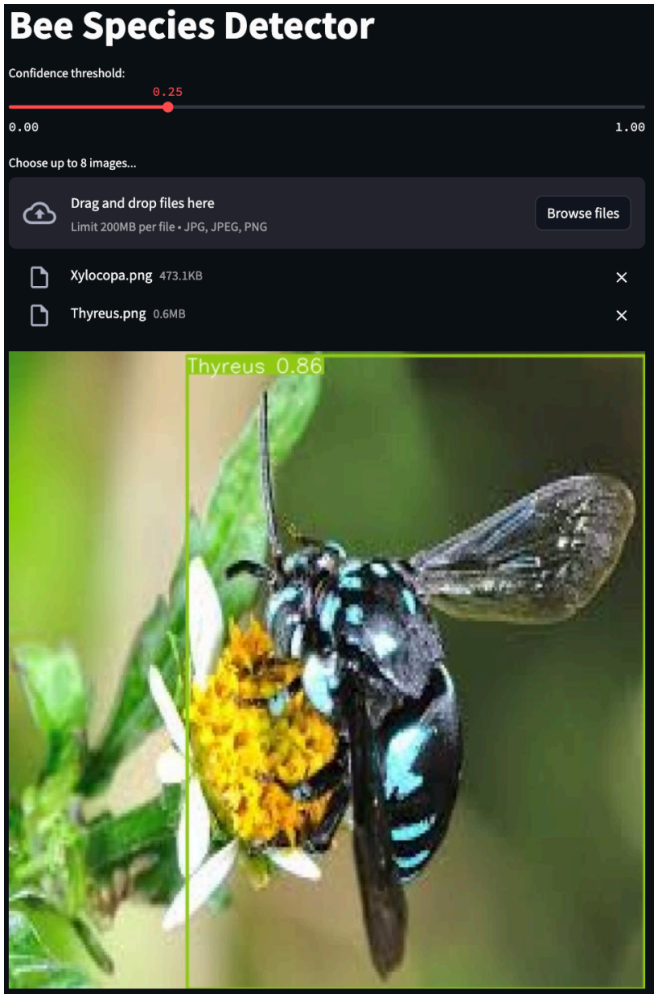

Figure 13: Streamlit bee species detector

6. PCS Discussion

Predictability

The model's predictability is measured and validated by the metrics of the training given at the last section of the report. The model is completely deterministic therefore the model will always give the same detections/performance on a particular image. There is no temporal component to the data that the model is trained and inferred on.

Computability

The computability of this model is able to be done on any device that has a cpu and can run Python code. The model is able to be gpu accelerated for inference if the edge device supports gpu compatibility. The model is only 6.3MB in size and can inference on most images and devices in under 1 second (and in 0.25 seconds when running on the CPU of a 14" M1-pro Macbook pro) This inference speed is good enough considering that we expect the model to inference on one image at a time and don't require real-time processing for most of the use-cases we expect our model to be used on. In future we expect this model to be able to be deployed on smartphones for simple usage in the field.

Stability

The model stability is completely determined by the quality and veracity of the data that is used to train the model and the initial weights that the model is initialised to at the start of the training process (this effect is negligible). The training of the model automatically selects batches to train on randomly. This ensures that the model's training is not negatively influenced by sequences of mislabelled images in the way that it would be if the batches are selected sequentially. This is essential if the images that are used to train the models are labelled by multiple labelers that might not label the images with the exact same labelling methodology.

7. Conclusion

Our work shows that the usage of a CNN-based object detection model such as YOLOv8 provides promising results in the task of identification and classification of bee species. Our model shows promising results from a limited dataset of just 1161 images and our work demonstrates that the model is able to be improved using augmentation techniques such as image shearing, height and width shift, image mixup and image copy-paste and found that augmentation techniques that result in significant colour changes result in model degradation. The model we provide has good predictability since the model is completely deterministic and always provides the same results for a given image as long as the same model is used. The model can infer an image in a short period (0.25 seconds per image on CPU) which should translate to sufficient speeds for non-real-time applications on low-powered devices.

Future work

Future work for the model includes increasing the size of the dataset and making use of trained labellers that are able to effectively label bees in the dataset into their correct classes. Future work could also include making use of a larger CNN model (that uses more processing resources) such as YOLOv8-x (Ultralytics, 2023) which is 10x larger in size or YOLOv10 which is the recently released state-of-the-art CNN-based model (Wang et al., 2024) or using a Vision Transformer model (ViT) such as CrossViT (Chen et al., 2021) instead of a CNN-based model.

8. References

Bang, S., Baek, F., Park, S., Kim, W., & Kim, H. (2020). Image augmentation to improve construction resource detection using generative adversarial networks, cut-and-paste, and image transformation techniques. *Automation in Construction*, *115*(September 2019), 103198. https://doi.org/10.1016/j.autcon.2020.103198

Bao, W., Pittaluga, F., Kumar B G, V., & Bindschaedler, V. (2023). DP-Mix: Mixup-based Data Augmentation for Differentially Private Learning. arXiv preprint arXiv:2311.01295.

Buschbacher, K., Ahrens, D., Espeland, M., & Steinhage, V. (2020). Image-based species identification of wild bees using convolutional neural networks. *Ecological Informatics*, *55*(September 2019), 101017. https://doi.org/10.1016/j.ecoinf.2019.101017

Chen, C.-F., Fan, Q., & Panda, R. (2021). CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. Cornell University, Computer Science > Computer Vision and Pattern Recognition. arXiv:2103.14899. https://doi.org/10.48550/arXiv.2103.14899.

Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., & Shan, Y. (2024). YOLO-World: Real-Time Open-Vocabulary Object Detection. arXiv preprint arXiv:2401.17270.

Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [Software]. Available from https://roboflow.com. computer vision.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., & Zoph, B. (2021). Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. arXiv preprint arXiv:2012.07177.

Gupta, J., Pathak, S., & Kumar, G. (2022). Deep Learning (CNN) and Transfer Learning: A Review. *Journal of Physics: Conference Series*, *2273*(1). https://doi.org/10.1088/1742-6596/2273/1/012029

Khalifa, Nour Eldeen & Loey, Mohamed & Mirjalili, Seyedali. (2022). A comprehensive survey of recent trends in deep learning for digital images augmentation. Artificial Intelligence Review. 55. 10.1007/s10462-021-10066-4.

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(25), E5716–E5725. https://doi.org/10.1073/pnas.1719367115

Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

Sohan, M., Sai Ram, T., & Rami Reddy, C. V. (2024). *A Review on YOLOv8 and Its Advancements*. *January*, 529–545. https://doi.org/10.1007/978-981-99-7962-2_39

Takahashi, R., Matsubara, T., & Uehara, K. (2020). Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. IEEE Transactions on Circuits and Systems for Video Technology, 30(9), 2917–2931. http://dx.doi.org/10.1109/TCSVT.2019.2935128

Ultralytics, J., et al. (2023). ultralytics/yolov8: Yolov8 - new state-of-the-art yolo model. GitHub. Retrieved from https://github.com/ultralytics/ultralytics, accessed: 2024-05-31.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOv10: Real-Time End-to-End Object Detection. arXiv. https://arxiv.org/abs/2405.14458