BeeML: BAN MachLA
Bee Annotation Machine Learning Algorithm

M Sreerag & C L Srinivas
National Intistute of Science Education and Research
Bhubaneswar

Abstract

The aim of our project was to develop a machine learning algorithm that can classify bee images into distinct species classes. Initial idea of the project involved merging our novel dataset with a pre-existing Kaggle dataset and developing a model on this hybrid dataset. Due to practical reasons, we stuck to working with our own dataset which consists of 3252 hand labelled images. The dataset contains six classes which includes four species of bees and other (non-bee) insects. We ran basic CNN models on this dataset and achieved an accuracy of 92.13% . We plan to run multi-label classification models and hopefully publish this dataset.

1 Insight on related papers

1.1 Image recognition using convolutional neural networks for classification of honey bee subspecies

DOI:- `https://doi.org/10.1007/s13592-022-00918-5`

This paper claims to have achieved a highest accuracy of 0.92, which is the best accuracy achieved for this task. Their dataset had 9887 images They trained their model on the cropped wing images of the various bees, rather than the whole bee image. Hence, their model classified the bees based on wing structure rather than overall morphometry.
They have used various CNN models like ResNet 50, MobileNet V2, Inception Net V3 and Inception ResNet V2 to extract features and have concluded that most of the models yielded same result at the end, even when they produced varying amount of trainable parameters.
This paper discusses methods of feature extraction, bootstrapping, cross validation etc.

1.2 Neural network approach to bee species classification

DOI:- `https://doi.org/10.1016/j.procs.2021.08.067`

This paper claims to have achieved 91% accuracy for classification. Their dataset contained 15,347 images.
They used a similar work flow to classify the images. It provides additional insights on feature extraction, bootstrapping, cross validation etc.

1.3 Assessing the potential for deep learning and computer vision to identify bumble bee species from images

DOI:- `https://doi.org/10.1038/s41598-021-87210-1`

This paper claims to have achieved 91.6% accuracy for classification. Their dataset contained 89,776 images.
It provides additional insights on feature extraction, bootstrapping, cross validation etc.

2 Baselines and Results

1. Basic K-NN model was implemented on the Kaggle dataset containing exclusively bee images. Maximum average accuracy of 75% was achieved using Euclidean distance and experimentally determined "optimal K values".

2. We ran basic CNN model on our novel dataset and achieved an accuracy of 92.13%.

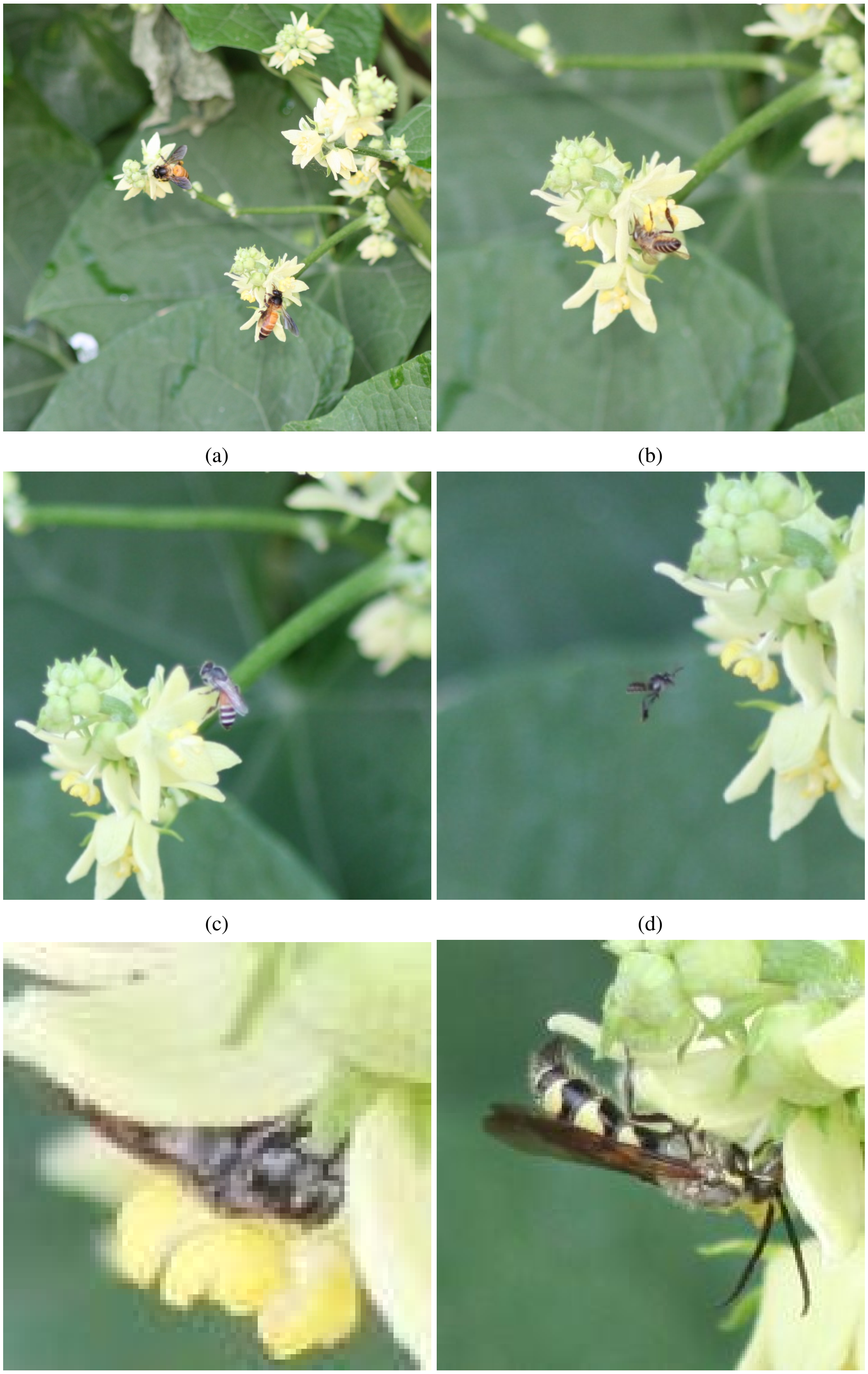
3 Dataset Curation

1. The original data consists of 3,846 images from camera traps setup on site at (insert so and so place). The photographs are captured by the camera (Canon EOS 400D) automatically at intervals. Images were collected in two sessions at two different sites (from 0820hrs to 1550hrs on 16^{th} January 2014 at site 1 and from 0700hr to 1240hrs on 11^{th} April 2014 at site 2). Each image is 24 bit and 1936 x 1288 pixels.

2. Images were annotated by hand using morphological cues and a visual key. Makesense.ai was used to mark bees in the images using bounding boxes. We have used six different labels to mark the boxes; bee, notbee, dorsata, cerana, florea and trigona. The last four correspond to the species names of the four different bee species present in our dataset. Objects that were positively identified to be not bee (eg:- ants, spiders, wasps etc.) were marked as "notbee". Objects that were identified as bee but not as belonging to any of the four species classes were marked as "bee". Within each labelled directory, there exists a sub-directory "notclear". The images which could not be classified into a label without using contextual information available to us, were deemed as unclear, in the context of ML training data and hence segregated into a separate directory within each class. The contextual information here refers to previous frame/next frame information which allows us to label a rather "unclear" image with it's true label, although there might not be enough visual cues present in the image under consideration in order to accurately and confidently label it.

3. The annotations were exported, saved and complied into a single csv file containing file name, label, coordinates of the top left corner of the bounding box, height of the bounding box and width of the bounding box. Using this csv file, the area within the bounding boxes containing the object of interest was cropped out and saved to a directory with the respective label name. This data, containing 3252 images was then used to train the ML models.

Class	Number of Data Points	Percentage Abundance
Dorsata	111	3.40%
Cerana	66	2.03%
Florea	821	25.25%
Trigona	1135	34.90%
Bee	569	17.50%
Notbee	550	16.91%
TOTAL	3252	100%



(a) (b)

(c) (d)

(e) (f)

Figure 1: *From (a) to (d)* A.dorsata, A.cerana, A.florea, A.trigona. *(e) object identified as "bee", (f) object identified as "notbee", which in this case is a wasp.*

4 Models and Results

4.1 Model

We ran basic CNN model on our dataset since relevant literature shows CNN models to be optimal for image classification task.

1. Augmentation
   - Augmentation was performed by flipping (horizontal and vertical), rotation, color jitter, afine transformation, perspective transformation and grayscale.
2. Class Equalization
   - Class equalization was achieved by augmenting and sampling classes according to their abundances, i.e. minority classes were selectively augmented and sampled more than majority classes.
   - Augmentation factor used was 2, i.e. the target class size for each class was 2 x the size of the largest class. If the size of a particular class is less than the target size, an image is selected from the class at random, transformed and added to the dataset. This process is continued until the required target size is achieved for that class. Same operation was carried out with all other classes.
3. Other model details
   - CNN model with five 2d convolution layers and one max pool layer.
   - Learning rate - 0.0001
   - Activation function - Relu
   - Optimizer - Adam
   - Loss - Cross entropy loss

4.2 Results

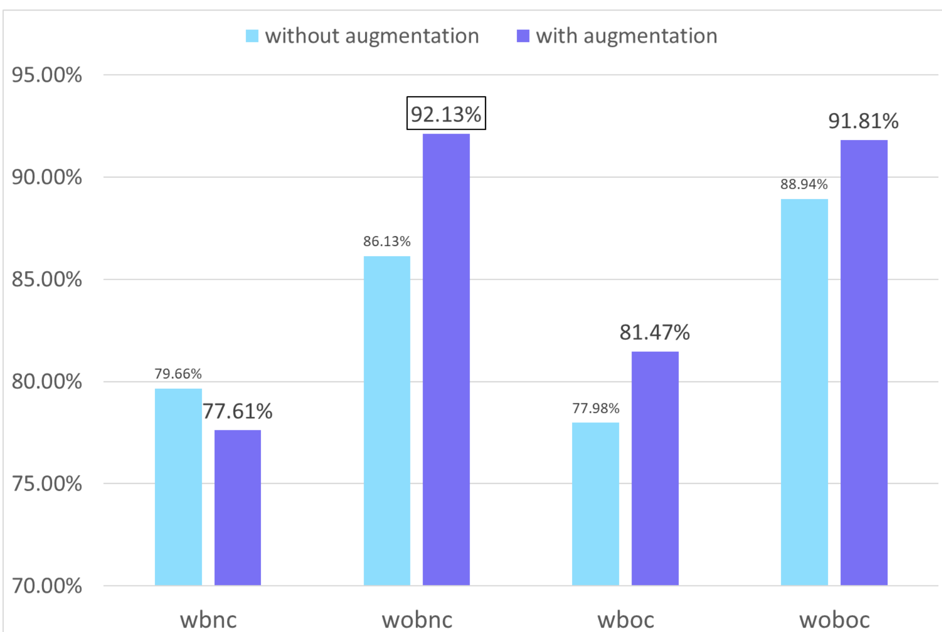


Figure 2: *Accuracy scores of our CNN model on validation. The model was run under two scenarios; with augmentation and class equalization of the data and without. Additionally, four different datasets were made from our complete set; woboc - which contained data from all the classes except the "bee" class and did not include images under the "not clear" sub-directory within each class, wobnc - same as woboc but contains the images under "not clear sub-directory within each class, wboc - contains the datapoints from all classes but does not contain the images under the sub-directory "not clear" within each class, wbnc - complete dataset.*

5 Future Direction

1. We plan to train multi label classification models on this dataset which will enable us to utilize all the six classes in training. Additionally, we believe this will produce a much more meaningful and useful classification system.
2. Standardize the dataset and hopefully publish it.