People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Abdelhamid Mehri – Constantine 2

Faculty of New Technologies of Information and Communication (NTIC)

Department of Softwae Technologies and Information Systems (TLSI)

Mini PROJECT

Module: Deep Learning

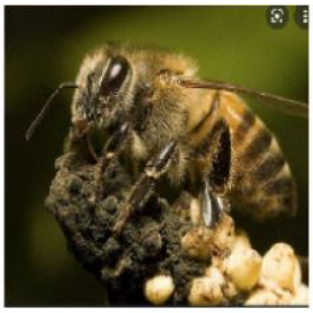
Classification of Harmful Insects Using Convolutional Neural Networks (CNN)

Realized by:	Under supervision of:
Maya Fairouz Menaifi	Dr. Imene BOULEGHLIMAT
Malika Benbouzid	

Introduction

In agriculture, identifying harmful insects is crucial, Manual identification can be time-consuming and error-prone, especially for farmers with limited expertise.

This project aims to develop a deep learning model to automatically classify harmful insects based on images trying to provide a reliable and efficient solution for insect identification.



Objectives

Build a convolutional neural network (CNN) model to classify harmful insects into 15 distinct categories.

Preprocess and augment the dataset to improve the model

Evaluate the model's performance using accuracy, precision, recall, F1-score and confusion matrix

Create a function to classify new images using the trained model.

Problematic

Challenges:

- Class imbalance in the dataset, with some insect classes having fewer images than others.
- Variability in image quality, lighting, and background, and some images shows insects as they hatch which the majority of them look like worms which confuses the model
- Overfitting due to the limited size of the dataset
- Limitation on ressources.

Solutions:

- Manually adding pictures from other datasets and use data augmentation to artificially increase the dataset size and variability.
- Implement dropout layers and regularization techniques to reduce overfitting

- Evaluate the model’s performance on a separate validation images .

Use Cases

Farmers: Quickly identify harmful insects in their fields specially for new farmers and take appropriate action. And use the matching pestasizer for that particular insect

Agricultural Advisors: Provide accurate pest identification and management recommendations.

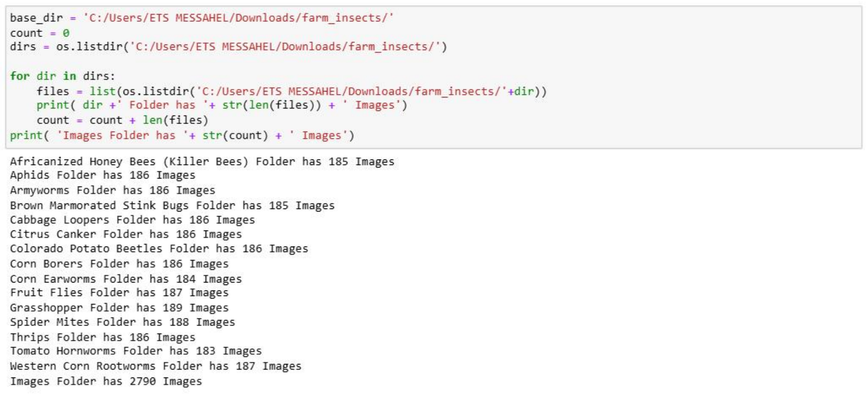
Researchers: Study the distribution and impact of harmful insects in different regions.

Dataset Overview

The dataset consists of 2790 images divided into 15 classes of harmful insects.

Downloaded from Kaggle dataset called “ Dangerous Farm Insects Dataset ” and also from another Kaggle2 dataset called “ Agricultural Pests Image Dataset ” to hellp augment the dataset manually and avoid overfiting

Each class represents a specific type of insect and The images are organized into subfolders with each subfolder corresponding to a class.



```
base_dir = 'C:/Users/ETS MESSAHEL/Downloads/farm_insects/'
count = 0
dirs = os.listdir('C:/Users/ETS MESSAHEL/Downloads/farm_insects/')

for dir in dirs:
    files = list(os.listdir('C:/Users/ETS MESSAHEL/Downloads/farm_insects/'+dir))
    print( dir +' Folder has '+ str(len(files)) + ' Images')
    count = count + len(files)
print( 'Images Folder has '+ str(count) + ' Images')
```

```
Africanized Honey Bees (Killer Bees) Folder has 185 Images
Aphids Folder has 186 Images
Armyworms Folder has 186 Images
Brown Marmorated Stink Bugs Folder has 185 Images
Cabbage Loopers Folder has 186 Images
Citrus Canker Folder has 186 Images
Colorado Potato Beetles Folder has 186 Images
Corn Borers Folder has 186 Images
Corn Earworms Folder has 184 Images
Fruit Flies Folder has 187 Images
Grasshopper Folder has 189 Images
Spider Mites Folder has 188 Images
Thrips Folder has 186 Images
Tomato Hornworms Folder has 183 Images
Western Corn Rootworms Folder has 187 Images
Images Folder has 2790 Images
```

Fig :code to count the images for each class

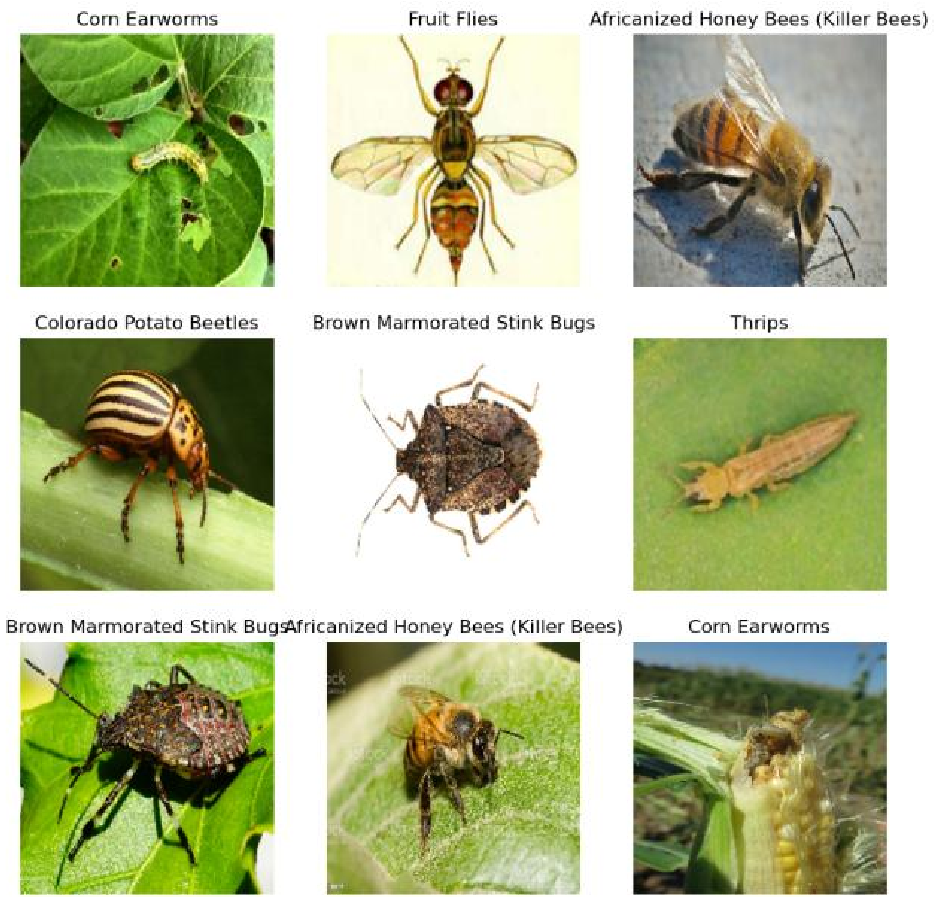

Corn Earworms
Fruit Flies
Africanized Honey Bees (Killer Bees)
Colorado Potato Beetles
Brown Marmorated Stink Bugs
Thrips
Brown Marmorated Stink Bugs
Africanized Honey Bees (Killer Bees)
Corn Earworms

Fig2: images with corresponding classes

Implementation

the necessary libraries for data processing, visualization, and deep learning were imported, including TensorFlow, NumPy, Matplotlib, and Keras components.

After Fetching Image Counts the images are loaded into arrays and split into training and validation datasets using TensorFlow's `image_dataset_from_directory` function.

The `iimage_dataset_from_directory` function is used because it automatically loads images from folders, assigns labels based on folder names, and prepares the data for training by batching, resizing, and normalizing it. This makes it a better choice than manually writing code to handle data loading, as it saves time, reduces complexity, and ensures consistency in preprocessing.

The dataset is resized to a uniform size (180x180) and split into 80% training and 20% validation sets we started exploring the dataset the class names were extracted, and a bar plot was created to visualize the distribution of images across classes. So we can confirme the dataset's organization and see if there is class imbalances.

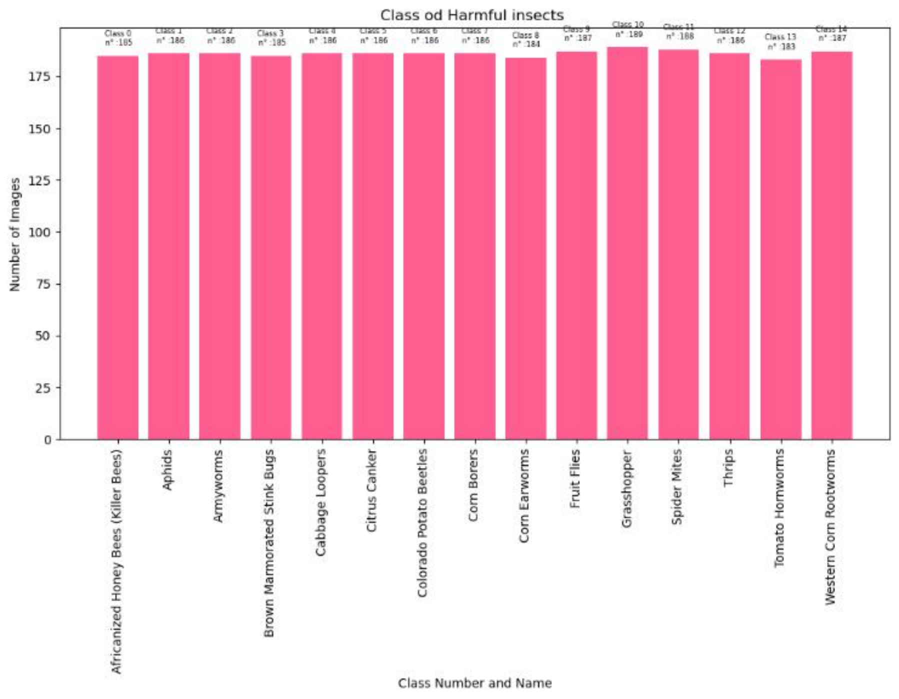


Fig2: class distrubution of insects

Data Preprocessing and Augmentation

The dataset was optimized using TensorFlow's AUTOTUNE feature

THis feature is utilized to optimize resource allocation dynamically during data loading and preparation since we are limited ressourse wise .

The dataset is cached, shuffled, and prefetched to streamline data processing so we can insure smooth and efficient model training and evaluation. Where :

- **caching** saves the dataset in memory to avoid repeated loading,
- **shuffling** mixes the data to improve model performance,
- and **prefetching** prepares the next batch in the background.

And the same thing dor the validation dataset.

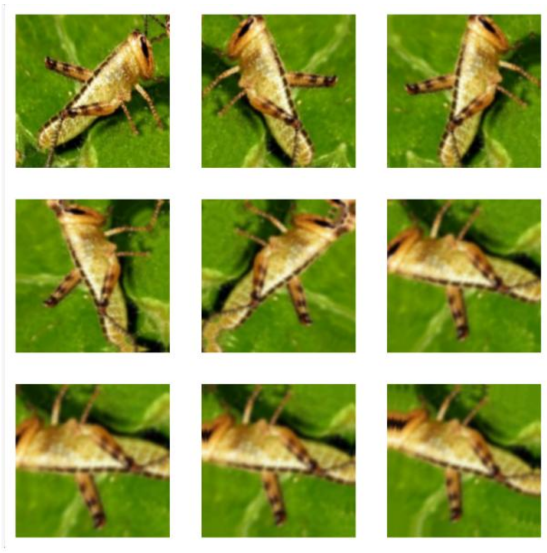
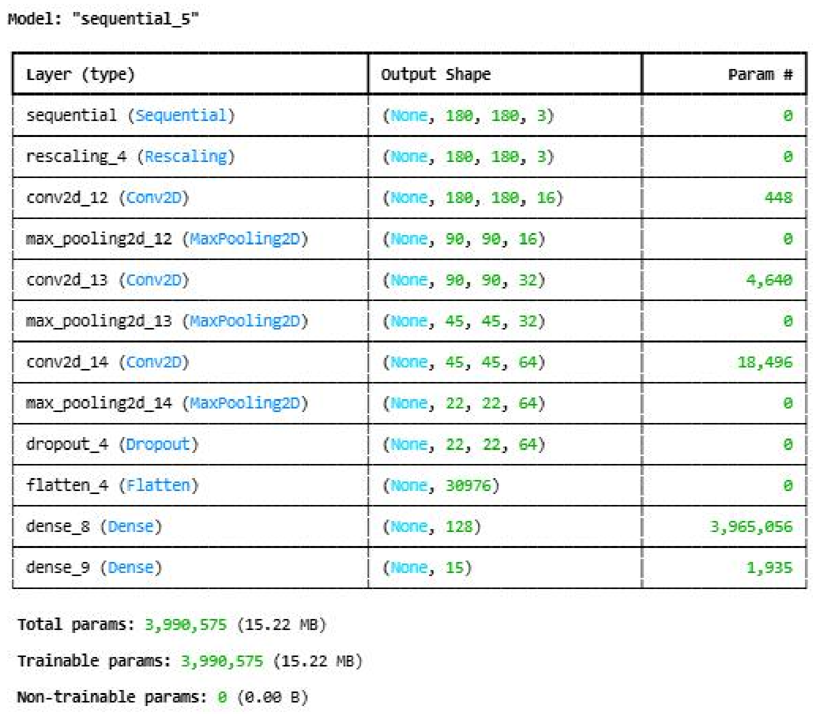


Fig4: example image after augmentation

Model Creation

The model uses Conv2D layers to find patterns like edges and textures in images,
and MaxPooling2D layers to focus on the most important parts.
A Dropout layer helps prevent the model from memorizing the data too much.
The Flatten layer turns the 2D data into a 1D list, which is then used by Dense layers to classify the images into 15 types with 3.9 million parameters,



```
Model: "sequential_5"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ sequential (Sequential)              │ (None, 180, 180, 3)         │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ rescaling_4 (Rescaling)              │ (None, 180, 180, 3)         │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_12 (Conv2D)                   │ (None, 180, 180, 16)        │             448 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_12 (MaxPooling2D)      │ (None, 90, 90, 16)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_13 (Conv2D)                   │ (None, 90, 90, 32)          │           4,640 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_13 (MaxPooling2D)      │ (None, 45, 45, 32)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_14 (Conv2D)                   │ (None, 45, 45, 64)          │          18,496 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_14 (MaxPooling2D)      │ (None, 22, 22, 64)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout_4 (Dropout)                  │ (None, 22, 22, 64)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten_4 (Flatten)                  │ (None, 30976)               │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_8 (Dense)                      │ (None, 128)                 │       3,965,056 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_9 (Dense)                      │ (None, 15)                  │           1,935 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘

 Total params: 3,990,575 (15.22 MB)

 Trainable params: 3,990,575 (15.22 MB)

 Non-trainable params: 0 (0.00 B)
```

Fig5: summary about the layers of the model

Model Compilation

The model was compiled using the Adam optimizer and SparseCategoricalCrossentropy loss function and Accuracy .

The Adam optimizer was chosen because it combines the benefits of two popular optimizers, AdaGrad and RMSProp , it adjusts the learning rate for each parameter automatically and speeds up training . it is useful in cases with complex datasets like ours ,where the model needs to learn features from a variety of images.

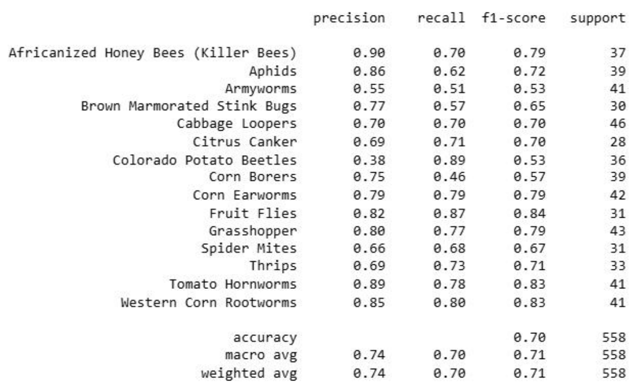
The SparseCategoricalCrossentropy loss function was used because the dataset has multiple classes represented as integer labels, and this loss function is specifically designed for such multi-class classification problems.

Accuracy was selected as the evaluation metric since it provides an easy-to-interpret measure of how well the model correctly classifies images into the right categories

The model was trained for 50 epochs, with validation performed on a separate dataset.

Results

1. Precision, recall, F1-score :



```
                                     precision    recall  f1-score   support

Africanized Honey Bees (Killer Bees)      0.90      0.70      0.79        37
                              Aphids      0.86      0.62      0.72        39
                           Armyworms      0.55      0.51      0.53        41
          Brown Marmorated Stink Bugs     0.77      0.57      0.65        30
                      Cabbage Loopers     0.70      0.70      0.70        46
                        Citrus Canker     0.69      0.71      0.70        28
             Colorado Potato Beetles      0.38      0.89      0.53        36
                         Corn Borers      0.75      0.46      0.57        39
                       Corn Earworms      0.79      0.79      0.79        42
                         Fruit Flies      0.82      0.87      0.84        31
                         Grasshopper      0.80      0.77      0.79        43
                        Spider Mites      0.66      0.68      0.67        31
                              Thrips      0.69      0.73      0.71        33
                    Tomato Hornworms      0.89      0.78      0.83        41
              Western Corn Rootworms      0.85      0.80      0.83        41

                            accuracy                          0.70       558
                           macro avg      0.74      0.70      0.71       558
                        weighted avg      0.74      0.70      0.71       558
```

Fig7 : Classification repport

The model achieves a solid 70% accuracy with strong performance for classes like Fruit Flies and Tomato Hormworms with high precision and recall.

While there’s room to improve for classes like Colorado Potato Beetles and Armyworms, the whole consistency (precision and recall averages around 0.70-0.74) is promising.

2. Training and validation accuracy:

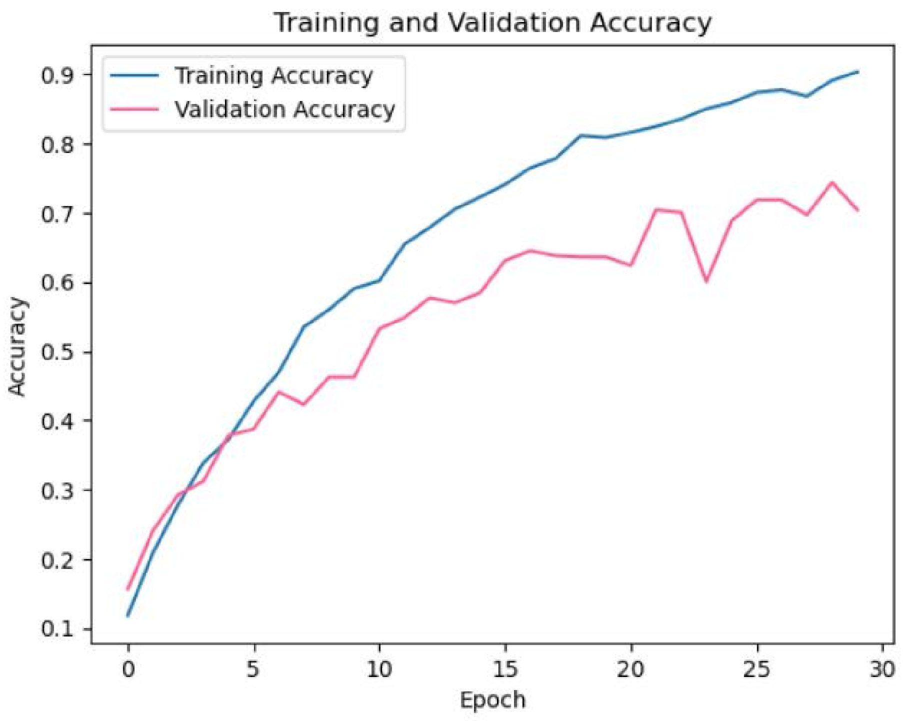


Fig 8:Training and validation accuracy

The graph demonstrates steady improvement in both training and validation accuracy over 30 epochs, highlighting the model's ability to learn patterns effectively.

The validation accuracy shows consistent growth overall and the model is performing well on unseen data.

The training process successfully captures the essential features of the dataset,

3. Training and validation Loss:

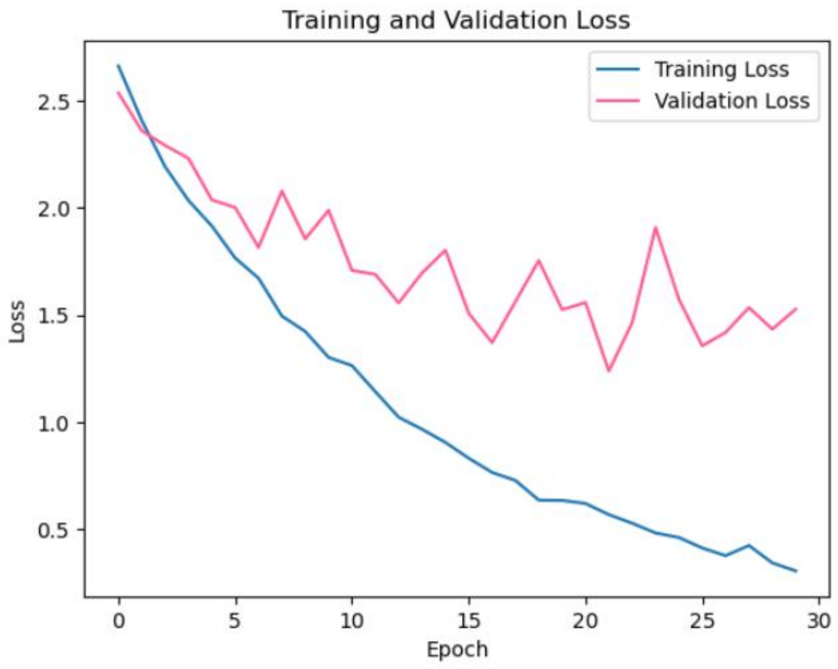
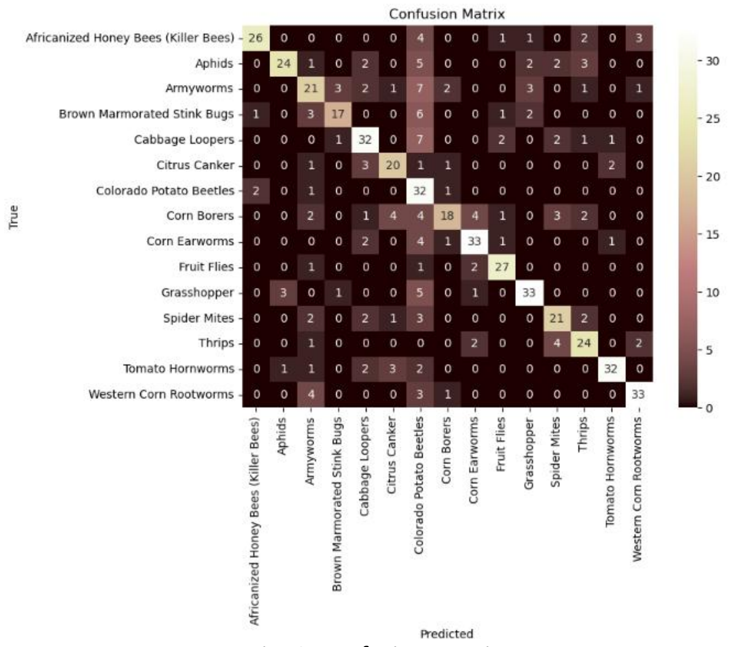


Fig9: Training and validation loss

The graph reflects a consistent decrease in training loss over 30 epochs, demonstrating the model's ability to learn and fit the training data effectively.

The validation loss shows some fluctuations, it remains relatively stable, indicating that the model is performing reasonably well on unseen data.

4. Confusion matrice



True \ Predicted	Africanized Honey Bees (Killer Bees)	Aphids	Armyworms	Brown Marmorated Stink Bugs	Cabbage Loopers	Citrus Canker	Colorado Potato Beetles	Corn Borers	Corn Earworms	Fruit Flies	Grasshopper	Spider Mites	Thrips	Tomato Hornworms	Western Corn Rootworms
Africanized Honey Bees (Killer Bees)	26	0	0	0	0	0	4	0	0	1	1	0	2	0	3
Aphids	0	24	1	0	2	0	5	0	0	0	2	2	3	0	0
Armyworms	0	0	21	3	2	1	7	2	0	0	3	0	1	0	1
Brown Marmorated Stink Bugs	1	0	3	17	0	0	6	0	0	1	2	0	0	0	0
Cabbage Loopers	0	0	0	1	32	0	7	0	0	2	0	2	1	1	0
Citrus Canker	0	0	1	0	3	20	1	1	0	0	0	0	0	2	0
Colorado Potato Beetles	2	0	1	0	0	0	32	1	0	0	0	0	0	0	0
Corn Borers	0	0	2	0	1	4	4	18	4	1	0	3	2	0	0
Corn Earworms	0	0	0	0	2	0	4	1	33	1	0	0	0	1	0
Fruit Flies	0	0	1	0	0	0	1	0	2	27	0	0	0	0	0
Grasshopper	0	3	0	1	0	0	5	0	1	0	33	0	0	0	0
Spider Mites	0	0	2	0	2	1	3	0	0	0	0	21	2	0	0
Thrips	0	0	1	0	0	0	0	0	2	0	0	4	24	0	2
Tomato Hornworms	0	1	1	0	2	3	2	0	0	0	0	0	0	32	0
Western Corn Rootworms	0	0	4	0	0	0	3	1	0	0	0	0	0	0	33

Fig10: confusion matrice new

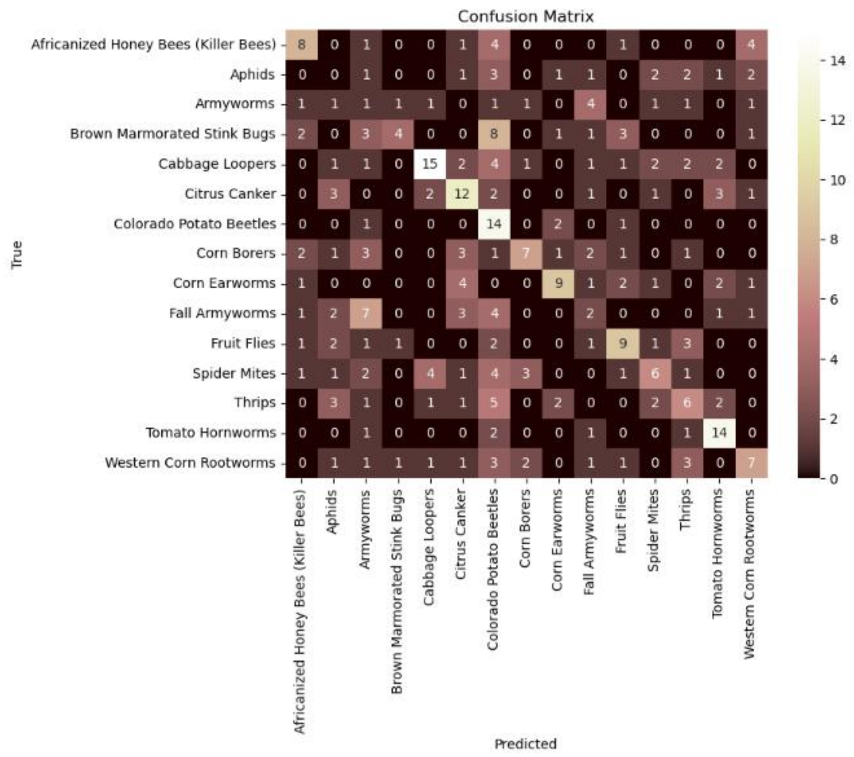


Fig11: confusion matrice old

the confusion matrix (Fig. 10) shows how well the model performed across 15 insect classes.

Most predictions match the true labels, with higher values along the diagonal, indicating good accuracy for many classes.
There are a few misclassifications, but they are small which mean the model has learned the differences between the insects well.

When comparing this to the previous confusion matrix (Fig. 11) we can see a big improvement. The presence of more zeros shows the model can better tell the classes apart, and the higher diagonal scores mean it is classifying the insects more accurately.

Testing

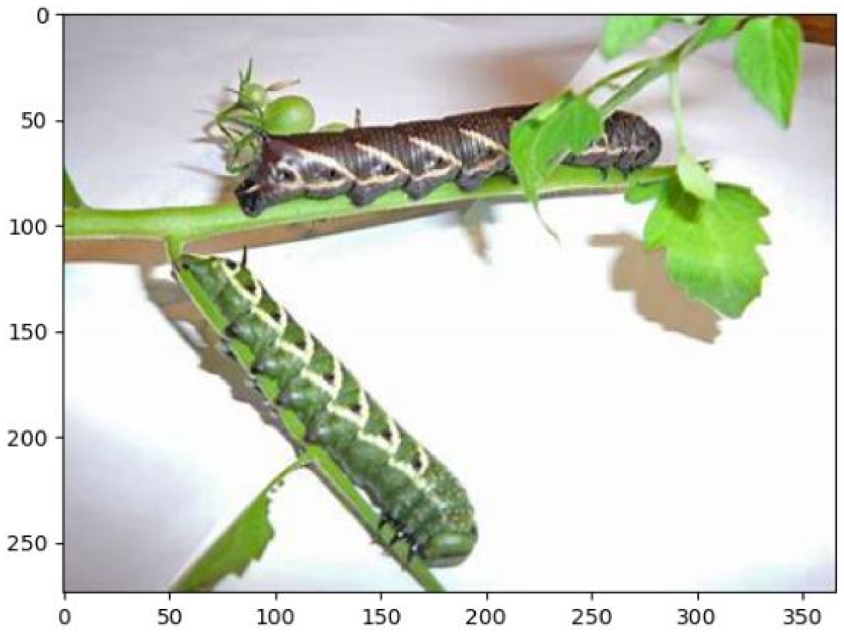
1. Colorado Potato Beetles:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
**************************************
tf.Tensor(
[ 5.981478  5.981478  5.981478  5.981478  5.981478  5.981478 16.259296
  5.981478  5.981478  5.981494  5.981478  5.981478  5.981478  5.981478
  5.981478], shape=(15,), dtype=float32)
Out[55]: 'The Image belongs to Colorado Potato Beetles with a score of 16.26 %'
```

2. Corn Earworms

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
****************************************
tf.Tensor(
[ 6.1166525  6.120806   7.0908628  6.2290463  6.657831   6.116724
  6.186966   6.1166925 12.553731   6.1199293  6.1610904  6.11693
  6.117655   6.1771502  6.117936 ], shape=(15,), dtype=float32)
Out[57]: 'The Image belongs to Corn Earworms with a score of 12.55 %'
```

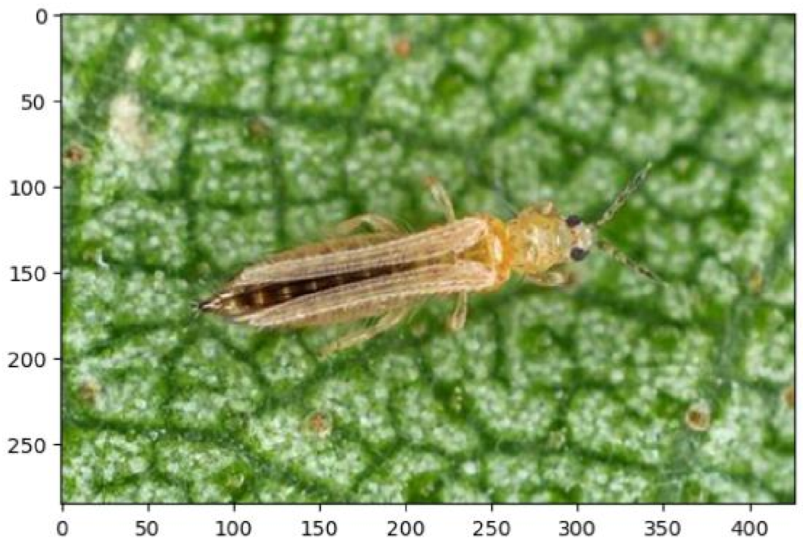
3. Tomato Hornworms



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
****************************************
tf.Tensor(
[ 5.9832478  5.983245   5.9837813  5.9832444  5.983259   5.9834933
  5.9984603  5.983466   5.9835153  5.983664   5.983242   5.983243
  5.9834647 16.217432   5.98324  ], shape=(15,), dtype=float32)
```

```
Out[59]: 'The Image belongs to Tomato Hornworms with a score of 16.22 %'
```

4. Thrips :



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
****************************************
tf.Tensor(
[ 5.987265   5.9873247  6.0000973  5.987703   5.987265   5.987282
  5.9999204  5.987864   5.987265   5.987265   5.987513   6.0053964
 16.121443   5.987265   5.9991355], shape=(15,), dtype=float32)
```

```
Out[72]: 'The Image belongs to Thrips with a score of 16.12 %'
```

Conclusion

This project demonstrated the use of deep learning for harmful insect classification. The CNN model provided some what an efficient and reliable solution for pest identification, addressing challenges such as class imbalance and overfitting through careful preprocessing and augmentation.

And we aim fo the future to use transfer learnning techniques to improve preformance