Team - 4

# Pest Detection

An ML Model, identifies the pest and suggest pesticides

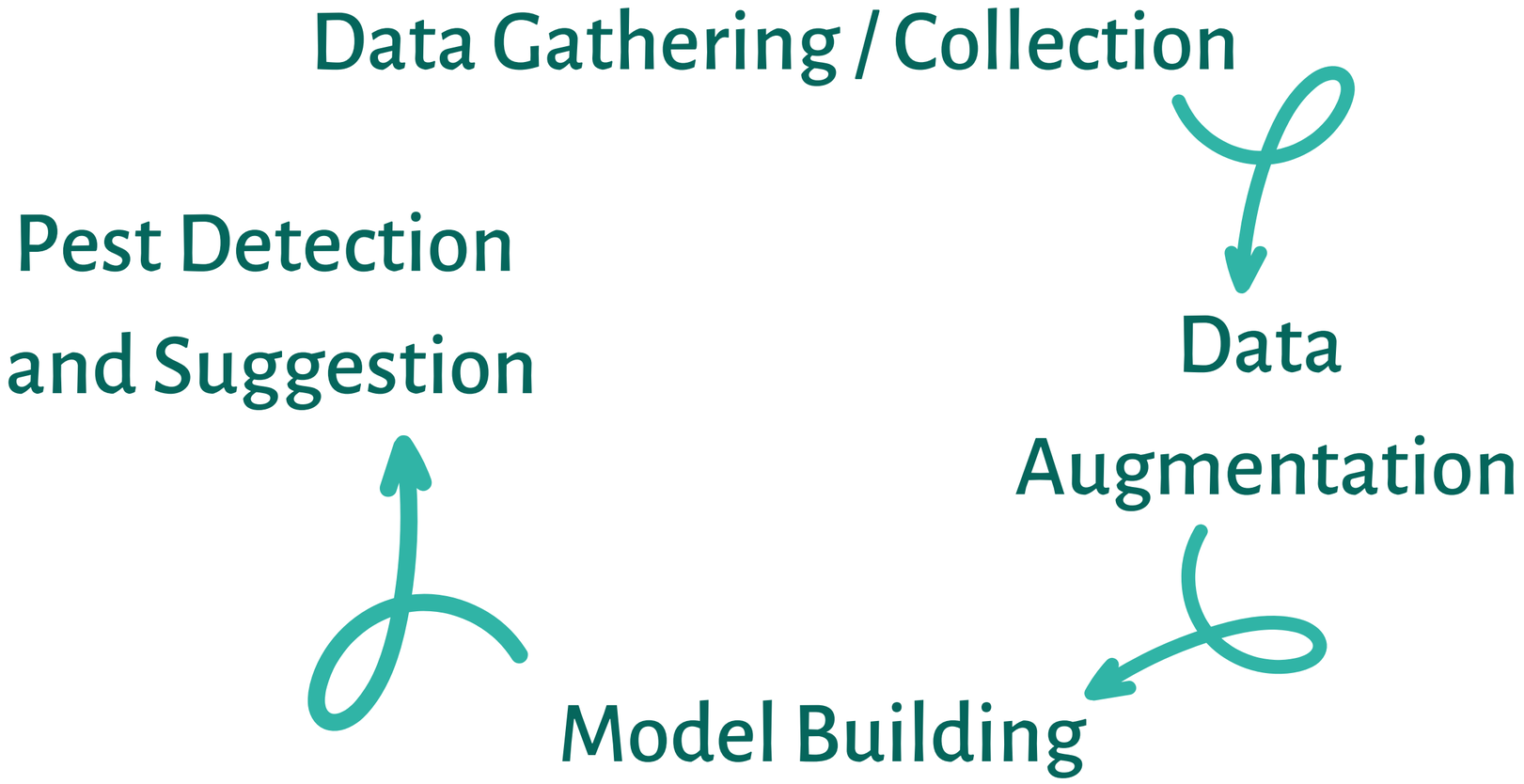
# Table of Contents



# Problem Statement

**Farmers now have a dilemma with *pest detection*. Here, we attempt to apply *machine learning* to identify pests in diverse crops and also recommend a suitable pesticide.**

# Workflow Explanation




# **Data Gathering /** Collection

For Simplicity, we divided the crops into three groups:

1. Field Crops
2. Vegetable Crops
3. Fruit Crops

Then on average, we chose 10 crops of each type and 5 pests for each crop. Rather than just importing a premade dataset, we tried to collect all the data manually. In all, our dataset contains 13,682 items. ( As dataset is huge, we are still working on it)

Dataset Link :
https://drive.google.com/drive/folders/1yp3z1cXpFDyM4A1xdMnFopsInecw7i6R


# Data Gathering / Collection

We used listing of pests for each crop in the reference below.

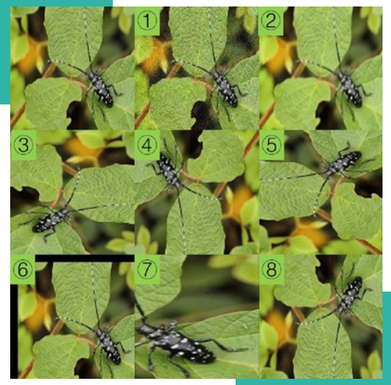
We selected 5 pests for  each crop, and collected images using google search. Some pests are common to several crops.

Collection of pest images off google image search is challenging because there are limited number of images of the wide spectrum of pests. We were not to find more that 10 images for several pests.

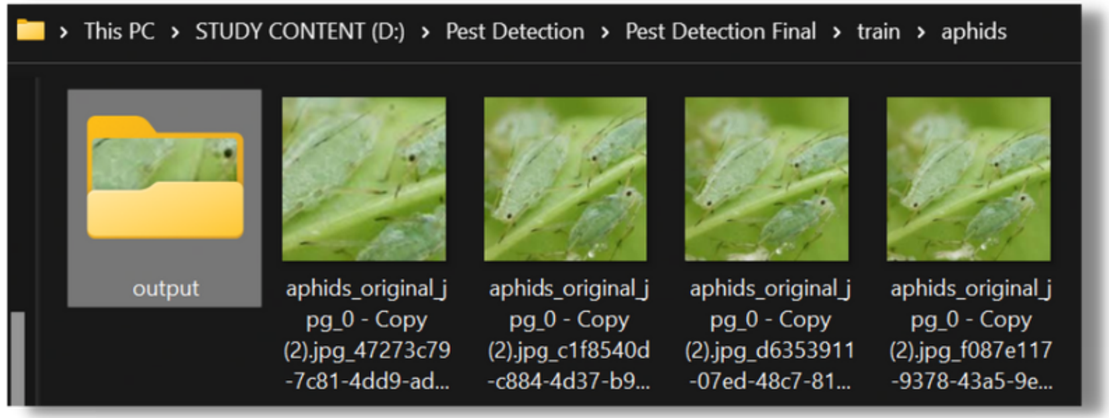
Therefore we need to regroup pests.

Reference Link :
https://databases.nbair.res.in/insectpests/pestsearch.php?cropname=Rice

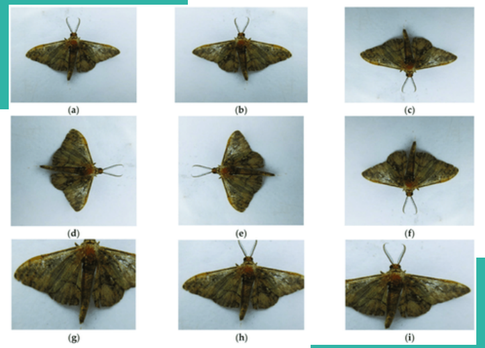
# Data Augmentation




```python
# this function fix the number of augmented images and handle RGBA images
def aug_imgg():

    count = 0
    dir_path = path_of_foldd              # that dir path
    dir_list = (os.listdir(dir_path))     #file name of particular dir

    for mm in range(len(dir_list)):

        if dir_list[mm]=='output':
            pass
        else:
            im = cv2.imread(dir_path+dir_list[mm])

            h,w,c = im.shape
            if c==4:
                os.remove(dir_path+dir_list[mm])
            else:
                count=count+1

    augmented_img = 300-count
    if augmented_img>0:
        augmented_img = augmented_img
    else:
        augmented_img = 0
    return augmented_img
```



# Data Augmentation

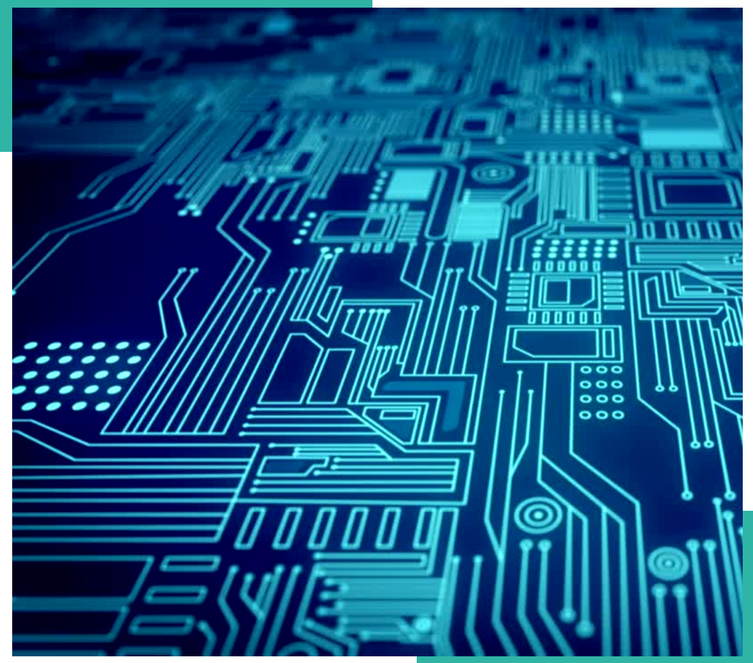


```
from pathlib import Path
import shutil
import os

path_to_data = "D:/Pest Detection/Pest Detection Final/train/"

for XD in range(len(total_class_in_dataset)):
    path_of_new_img = path_to_data+total_class_in_dataset[XD]+'/'+'output'+'/'
    path_of_fold = path_to_data+total_class_in_dataset[XD]+'/'
    files=os.listdir(path_of_new_img)

     # iterating over all the files in the source directory
    for file_name in files:
       shutil.copy(path_of_new_img+file_name, path_of_fold+file_name)
    print("Files are copied successfully")
```

```
import shutil

path_to_data = "D:/Pest Detection/Pest Detection Final/train/"

for lol in range(len(total_class_in_dataset)):
    folder_path = path_to_data+total_class_in_dataset[lol]+'/'+'output'+'/'
    shutil.rmtree(folder_path)

# removing the new dir that is created during data augmentation
print("Successfully removed created folder")
```

```
Successfully removed created folder
```

# MODEL BUILDING



```
input_shape = (BATCH_SIZE,IMAGE_SIZE,IMAGE_SIZE,CHANNELS)
n_classes = 13

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32,(4,4),activation='relu',input_shape = input_shape),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(4,4),activation='relu',input_shape = input_shape),
    layers.MaxPooling2D((2,2)),

    layers.Conv2D(64,(4,4),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(4,4),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(4,4),activation='relu'),
    layers.MaxPooling2D((2,2)),
    # the value in Convo2D and MaxPolling is fully based on hit and trail approach

    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(n_classes, activation='softmax'),

])

model.build(input_shape=input_shape)
```

# **Results** And Suggestions

- We proposed a model that would help farmers in detecting pests using Convolutional Neural Network (CNN) and image processing.
- It not only investigates the different pests that attack the fields but also gives information on how they can be controlled and managed.
- The searching and comparison of captured images to a stack of pest images were implemented using a model based on CNN.


Dataset Link : https://docs.google.com/document/d/1AXLbtG_55ecK4ppI422Kq6Lab-Jmd_ESnrNhODzLbOo/edit

# **Results** And Suggestions

- Collected images were pre-processed and used in training the model. The model was able to achieve a final training accuracy of 82% at epoch value 4.

- The trained model can perform prediction or can classify images with a low percentage of error.

```
Epoch 1/5
265/265 [==============================] - ETA: 0s - loss: 2.2619 - accuracy: 0.2758
Epoch 1: val_loss improved from inf to 2.38917, saving model to .\model_weights.h5
265/265 [==============================] - 1333s 5s/step - loss: 2.2619 - accuracy: 0.2758 - val_loss: 2.3892 - val_accuracy:
0.2118
Epoch 2/5
265/265 [==============================] - ETA: 0s - loss: 1.8355 - accuracy: 0.4095
Epoch 2: val_loss improved from 2.38917 to 2.16843, saving model to .\model_weights.h5
265/265 [==============================] - 1204s 5s/step - loss: 1.8355 - accuracy: 0.4095 - val_loss: 2.1684 - val_accuracy:
0.3299
Epoch 3/5
265/265 [==============================] - ETA: 0s - loss: 1.2921 - accuracy: 0.5914
Epoch 3: val_loss improved from 2.16843 to 1.32333, saving model to .\model_weights.h5
265/265 [==============================] - 1169s 4s/step - loss: 1.2921 - accuracy: 0.5914 - val_loss: 1.3233 - val_accuracy:
0.6111
Epoch 4/5
265/265 [==============================] - ETA: 0s - loss: 0.8347 - accuracy: 0.7341
Epoch 4: val_loss improved from 1.32333 to 0.94210, saving model to .\model_weights.h5
265/265 [==============================] - 1041s 4s/step - loss: 0.8347 - accuracy: 0.7341 - val_loss: 0.9421 - val_accuracy:
0.7569
Epoch 5/5
265/265 [==============================] - ETA: 0s - loss: 0.5562 - accuracy: 0.8263
Epoch 5: val_loss did not improve from 0.94210
265/265 [==============================] - 979s 4s/step - loss: 0.5562 - accuracy: 0.8263 - val_loss: 1.0176 - val_accuracy: 0.
7396
```

# **Teams And** Individual Contribution

- **Data Collection**
  - Yash Doshi, Safia Labdi, Hala Jadallah
- **Data Augmentation**
  - Bhushan Mahajan
- **Model Building**
  - Vineet Sharma
- **Results and Suggestions**
  - Arya Dubey


ThankYou