Mohamed Alaa 202200980
Mohamed Mahmoud Emam 202202236

Drone Detection Project Report

A. *Introduction and Problem Understanding:*

Our goal in this project is to work with a set of images. These images have drones in them, and the model needs to identify the drones and their locations.

B. *Data Explanation:*

- **Data or Input:** 1359 images have drones in different places and difficult positions.
- **Label or Target or Output:** Drone Location Consist of Four numbers:
  - First number represents the **`x_center`**.
  - Second number represents the **`y_center`**.
  - Third number represents the **`width of the bounding box`**.
  - Fourth number represents the **`height of the bounding box`**.

C. Preprocessing:

1) Download the Dataset
2) Read the Data and put the images in list named **train_images** and the targets in list named **train_annotation**
3) Change the variable type from **List** into **Numpy** array to deal with it
4) Normalize the data to make all pixels be from 0 to 1
5) Print Images shape (1359, 256, 256, 3) → we have 1359 image each image has Dimension (256 , 256 , 3) **RGB**
6) Print Targets shape (1359, 4) → we have 4 numbers in the label (`x_center,y_center,width,height`)
7) Split the Data into **Train** and **Test** (1087 image for Train , 272 image for test)
8) Next comes the **Data Augmentation** step. We took each image and created three versions:
   - First version is the original image.
   - Second version is the image flipped.

- Third version is the image rotated.

After the **Data Augmentation** step the number of images = 3261

D. *Model Architecture:*

- We chose the Inception model because it stores **more information** and reduces the **number of parameters**, which will shorten the **training time**. Also, the Inception model prevents the **vanishing gradient** problem and avoids **overfitting**, resulting in higher accuracy. First, we tried using the VGG16 model. It was fast, but the accuracy was not high because the model is simple. Then we tested the ResNet152 model and found that the training accuracy was very high, but the validation accuracy was very low, which indicates overfitting. As a result, the training performance was poor. So, we decided to try ResNet with fewer parameters and used ResNet50, but we still noticed overfitting.

- Steps:
    1) Downloaded InceptionV3 model that trained on imagenet dataset without the top or without the fully connected layers
    2) Apply Fine tuning on last 10 layers to improve Accuracy and change the weights
    3) Adding the fully connected layers with relu activation function
    4) Adding the output layer with 4 neurons because the output should be 4 numbers, and without activation function because the model should predict numbers from 0 to 1 (Regression not Classification)
    5) Chose Adam optimizer (common practice) , and the Mean Squared Error loss function because we are working on Regression problem
    6) Train the model with 150 epochs and 32 batch size
    7) Train Accuracy = 0.9241 , Validation Accuracy = 0.8407
    8) Apply the model on the Test dataset to predict Drone's location

E. *Visualization of Results:*

- We visualized the results by taking the predictions and drawing a green rectangle on the test images to check if the green rectangle is correctly placed on the drone or not.
- Next, we plotted a graph showing the accuracy over the epochs for both training and validation, and we found a small difference between them.
- Also, we plotted a graph showing the loss over the epochs for both training and validation.