

# Programming Assignment - 1: Building a Machine Learning Application with Streamlit

ITBIN-2110-0159 - Nimantha Colambage

## **About model**

It is a trained machine learning model for classifying images of flowers into categories such as daisy, dandelion, rose, sunflower, and tulip. The model is saved in a file, as are similar images for each category. Indeed, it predicts the type of flower in a photo user upload and can display images of similar flowers from its dataset. This helps provide a visual reference for the predicted category of flowers. Use kaagle free data set by connecting kaagle api

How this model run

the model file cant upload to github because its says too largo to upload so i upload the model to dropbox and save the streamlit app file code inside github and run it

## **Links**

dropbox .h5 model file location

[https://www.dropbox.com/scl/fi/t527snher97bzw4g1tas/flower\\_model.h5?rlkey=d3ltlw10hnso9qlfnefrnckdd&st=zh062pwk&dl=0](https://www.dropbox.com/scl/fi/t527snher97bzw4g1tas/flower_model.h5?rlkey=d3ltlw10hnso9qlfnefrnckdd&st=zh062pwk&dl=0)

github link

[https://github.com/Nimanthan-10/ITBIN\\_2110\\_0159.git](https://github.com/Nimanthan-10/ITBIN_2110_0159.git)

Streamlit app link

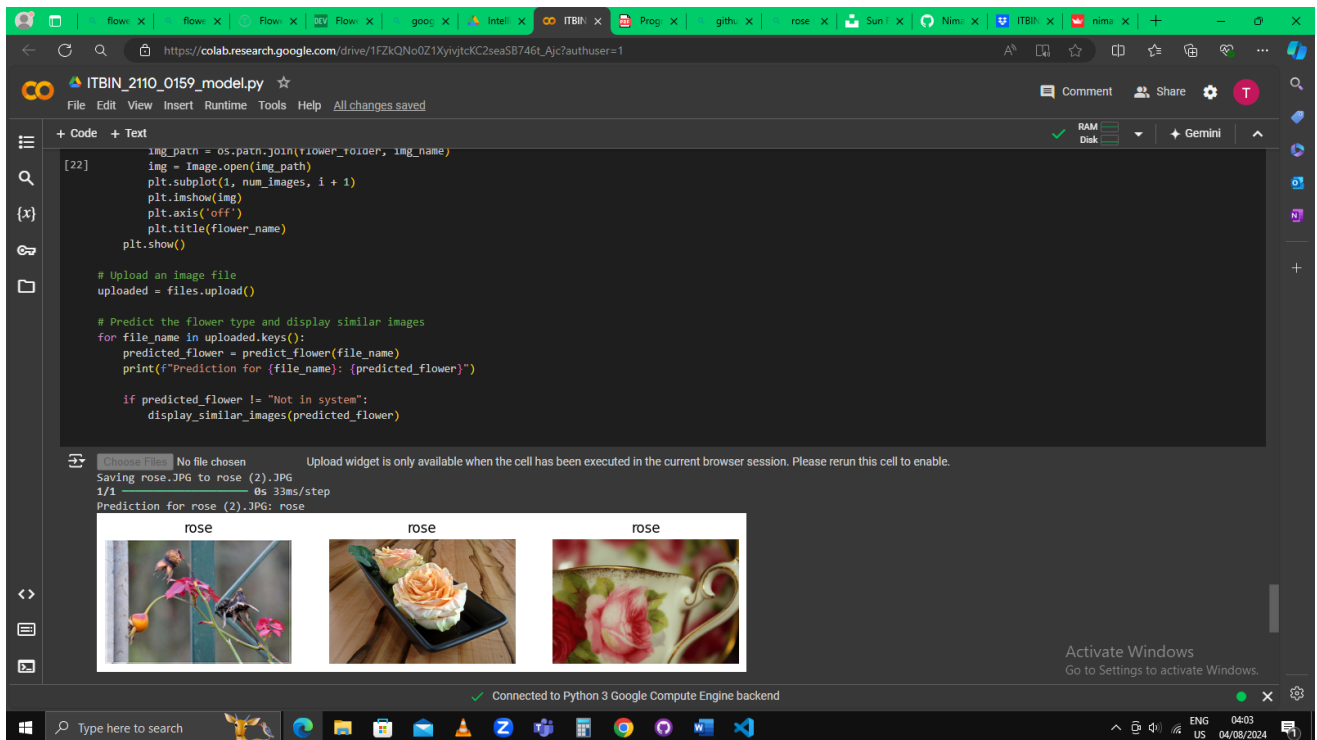
<https://itbin21100159-mpwpkrmrjughp5edypucfy6.streamlit.app>

## Model Run

Model create using colab and when user upload image it predicts flower name also output similar flowers photos that inside that data set if that flower not in that data set output not in system.

Only get 5 flowers for training

## Example



The screenshot displays a Google Colab notebook interface. The notebook title is "ITBIN\_2110\_0159\_model.py". The code in the cell is as follows:

```
[22] img_path = os.path.join(flower_folder, img_name)
img = image.open(img_path)
plt.subplot(1, num_images, i + 1)
plt.imshow(img)
plt.axis('off')
plt.title(flower_name)
plt.show()

# Upload an image file
uploaded = files.upload()

# Predict the flower type and display similar images
for file_name in uploaded.keys():
    predicted_flower = predict_flower(file_name)
    print(f"Prediction for {file_name}: {predicted_flower}")

    if predicted_flower != "Not in system":
        display_similar_images(predicted_flower)
```

The output of the code execution is shown below the code cell:

```
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving rose.JPG to rose (2).JPG
1/1 0s 33ms/step
Prediction for rose (2).JPG: rose
```

Below the output text, three images of roses are displayed, each with the caption "rose" above it. The first image shows a rose on a branch, the second shows a rose in a vase, and the third shows a close-up of a rose.

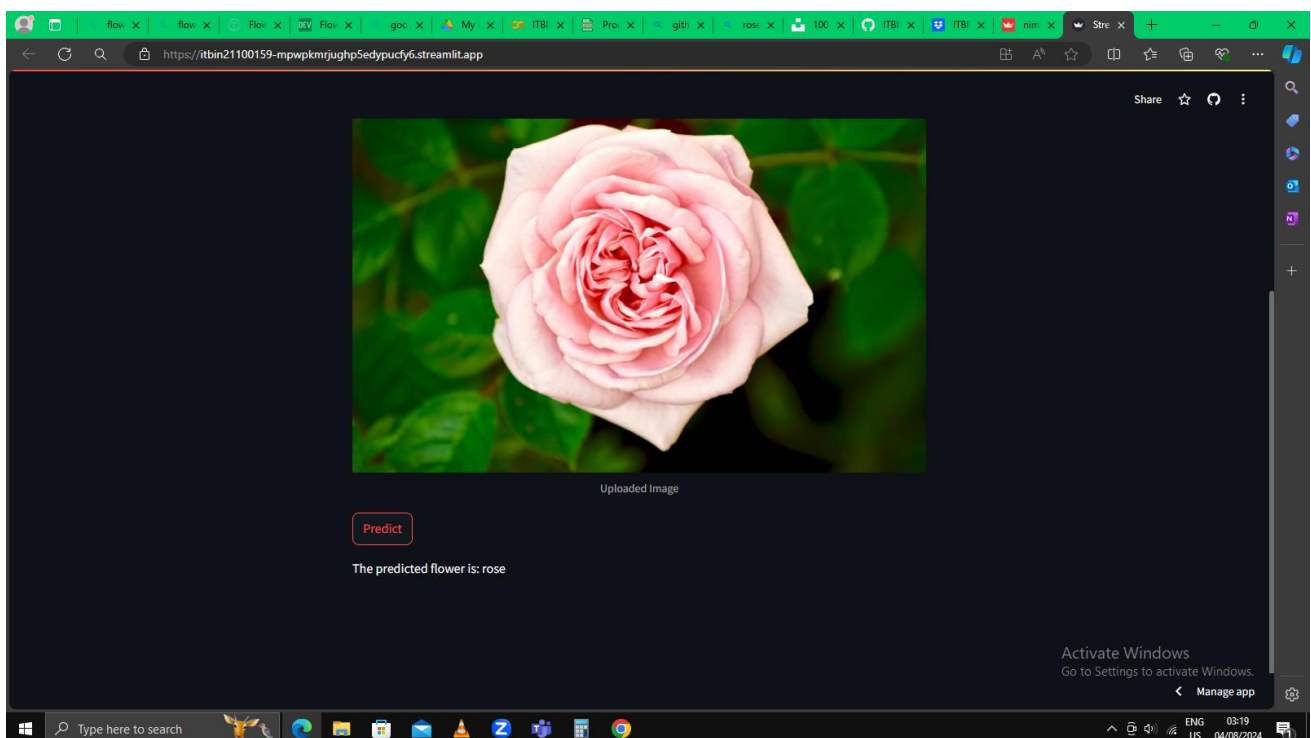
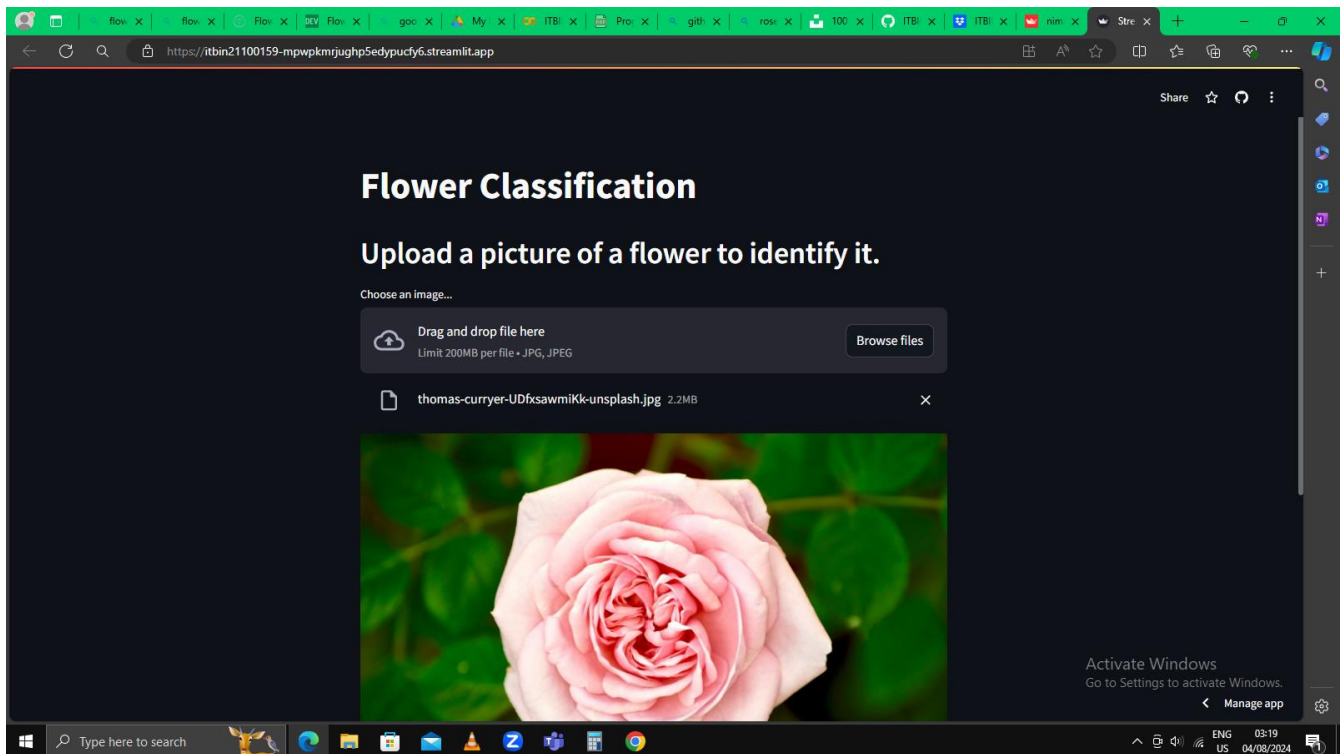
At the bottom of the notebook interface, a status bar indicates "Connected to Python 3 Google Compute Engine backend".

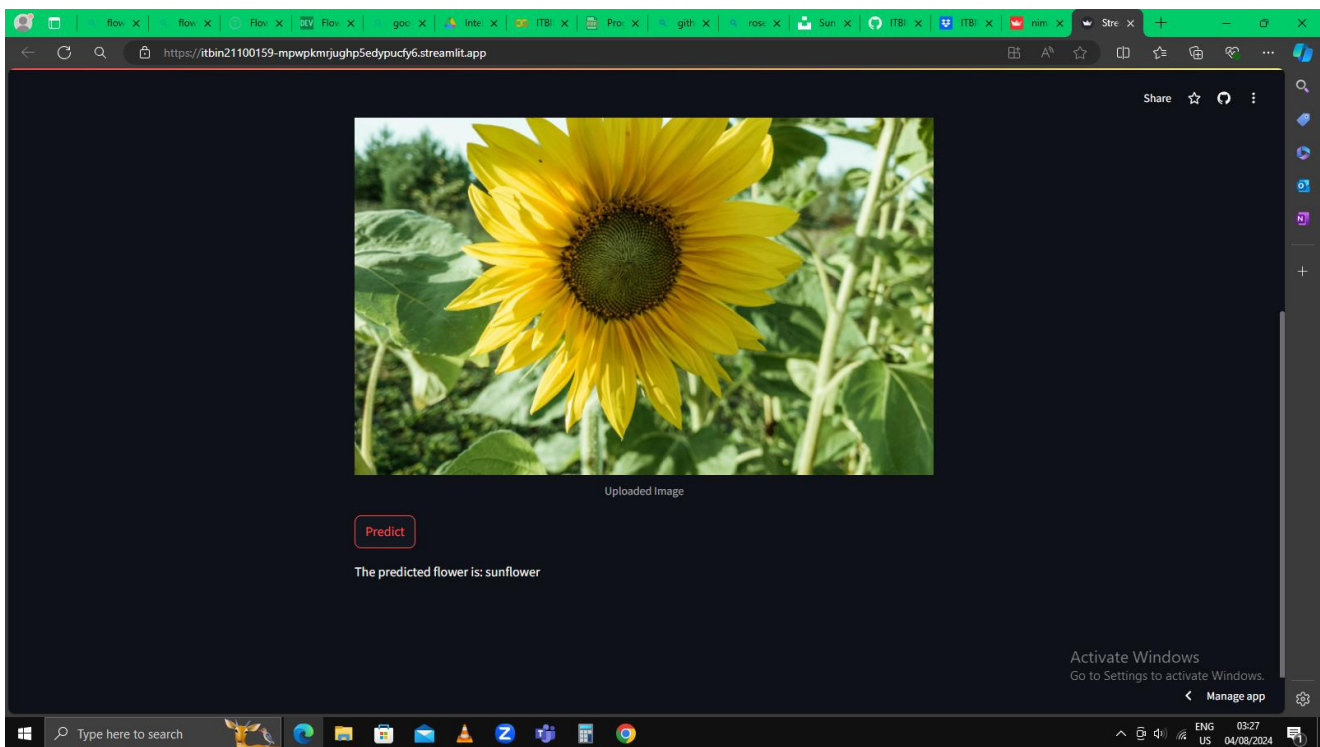
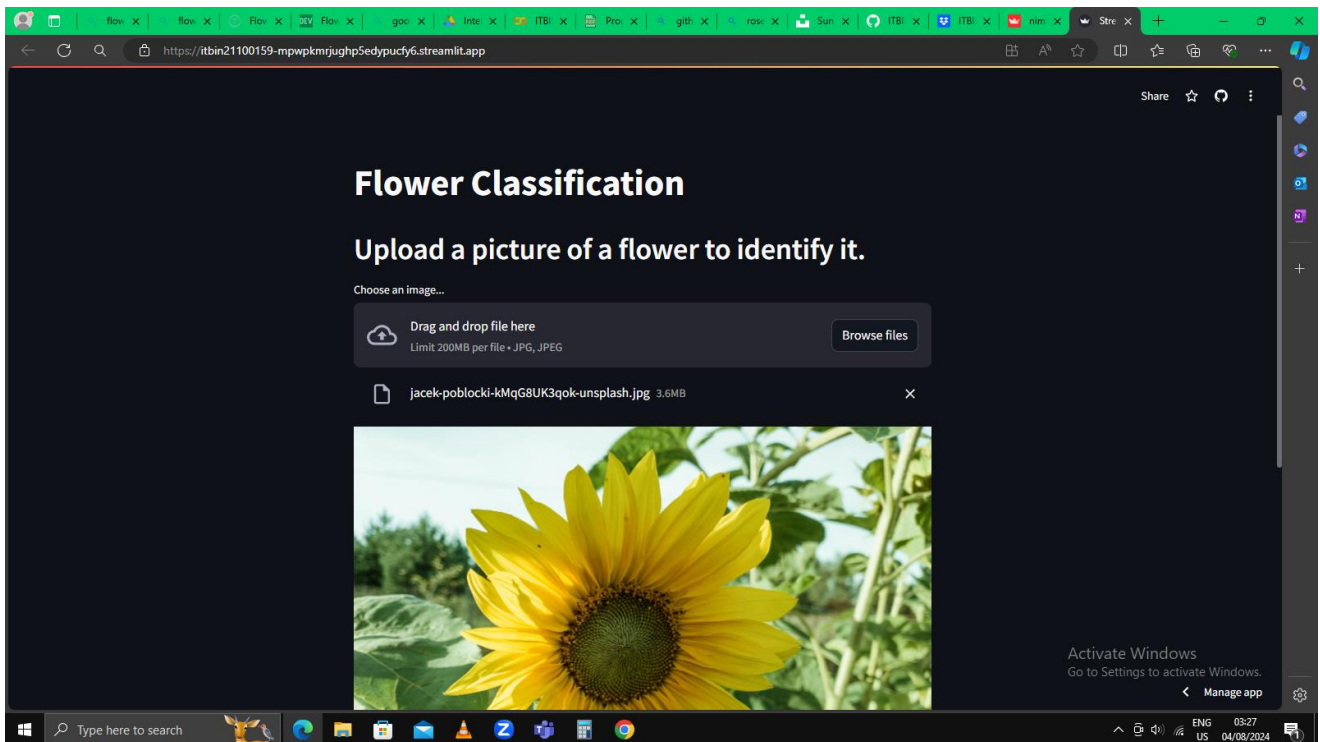
## Stramlit app

Flower\_model inside dropbox and connect to App.py file inside github run the stremlit application and display the prediction

**When user upload flower image it predict that flower name**

## Examples





## **Links**

### **Github Link**

[https://github.com/Nimanthan-10/ITBIN\\_2110\\_0159.git](https://github.com/Nimanthan-10/ITBIN_2110_0159.git)

### **dropbox .h5 model file location**

[https://www.dropbox.com/scl/fi/t527snher97bzw4g1tas/flower\\_model.h5?rlkey=d3ltlw10hnso9qlfnefrnckdd&st=zh062pwk&dl=0](https://www.dropbox.com/scl/fi/t527snher97bzw4g1tas/flower_model.h5?rlkey=d3ltlw10hnso9qlfnefrnckdd&st=zh062pwk&dl=0)

### **Streamlit app**

<https://itbin21100159-mpwpkmrjughp5edypucfy6.streamlit.app>