SR UNIVERSITY

COMPUTER VISION (21CD108)

FLOWER IMAGE CLASSIFICATION USING CNN

A Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

by

VELISHALA ABHIVARUN 2203A52062

Under the guidance of

Dr. Nafis Uddin Khan

Assistant Professor, School of CS&AI.

SR University, Ananthsagar,Warangal,Telagnana-506371

CS & AI

SR University

Ananthasagar, Warangal.

SR UNIVERSITY

CERTIFICATE OF COMPLETION

This is to certify that **Velishala Abhivarun**, bearing Hall Ticket Number **2203A52062**, a student of **CSE-AIML**, **3rd Year – 2nd Semester** at **SR University**, has successfully completed the **Computer Vision** course.

As part of the coursework, the student submitted a project titled **Flower Image Classification**, which demonstrated a solid understanding of the fundamental concepts of computer vision and their practical application in real-world scenarios. The project met the required academic standards and learning objectives of the course.

Dr. Nafis Uddin Khan
Assistant Professor,
SR University
Anathasagar,Warangal

FLOWER IMAGE CLASSIFICATION USING CNN

Description of Project:

This project is a **Computer Vision-based multi-class image classification task** aimed at recognizing different types of flowers using a Convolutional Neural Network (CNN). The notebook is well-structured into logical sections, starting from dataset preparation to model evaluation.

Dataset Overview:

The dataset contains 5 flower classes—Lily, Lotus, Sunflower, Orchid, and Tulip—with 1000 images each, ensuring balance. It is stored in the "FLOWER IMAGES" folder after extraction from a zip file. The directory follows the flow_from_directory format with separate subfolders for each class.

Libraries and Tools Used:

- **Python**: Programming language for writing the code.
- **TensorFlow & Keras**: Libraries for building and training deep learning models.
- **Matplotlib**: Library for plotting graphs and visualizing results (like training accuracy, loss, etc.).
- **NumPy**: Library for numerical operations, like working with arrays.
- **Image Data Generator**: A tool for preprocessing and augmenting images to make the model more robust during training.

Model Architecture

The CNN model likely includes the following layers:

- **Multiple Conv2D layers** with **ReLU activation** to extract features.
- **MaxPooling2D layers** to reduce spatial dimensions and retain important information.
- **Flatten layer** to convert 2D features into a 1D vector.
- **Dense layers**, ending with a **Softmax layer** to classify the images into one of the **5 flower classes**.
- **Loss Function**: categorical_crossentropy
- **Optimizer**: Adam (efficient and widely used)
- **Metric**: Accuracy

TEST ACCURACY:

Test Accuracy: 81.58%

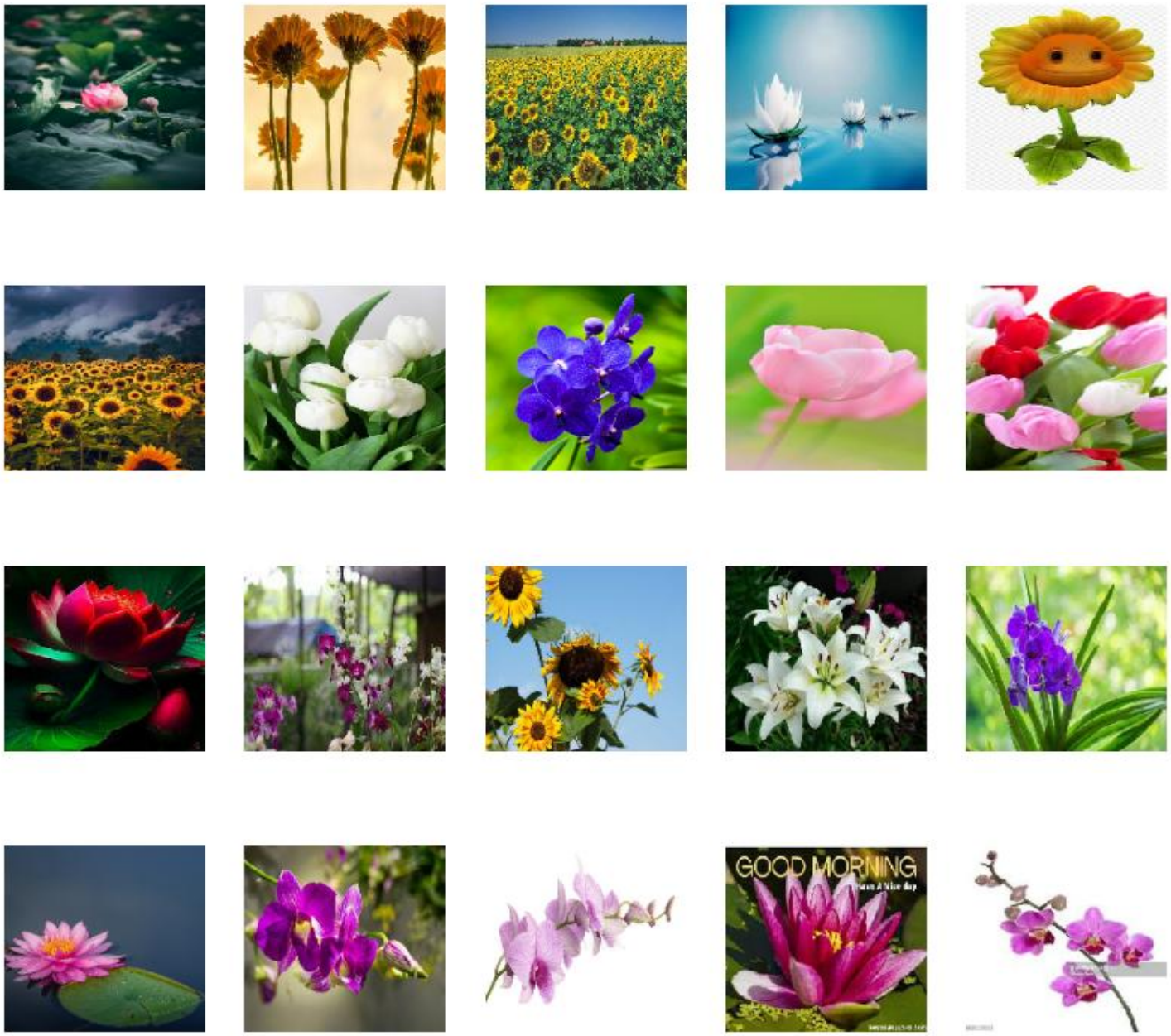
DETAILS OF DATASET:

Total training samples: 5000
Class labels: {'Lilly': 0, 'Lotus': 1, 'Orchid': 2, 'Sunflower': 3, 'Tulip': 4}
Batch size: 32
Sample images shape: (32, 150, 150, 3)
Sample labels shape: (32, 5)

TRAINING SET IMAGES

- Tulip: 1000 images
- Lotus: 1000 images
- Sunflower: 1000 images
- Lilly: 1000 images
- Orchid: 1000 images

DISPLAY SOME IMAGES



DATASET SHAPE:

shape of image: (32, 180, 180, 3)
shape of label: (32,)

FLOWER TYPES: ['Lilly', 'Lotus', 'Orchid', 'Sunflower', 'Tulip']

VISUALISATION OF IMAGE:

Lotus

Lotus

Orchid

Orchid

Sunflower

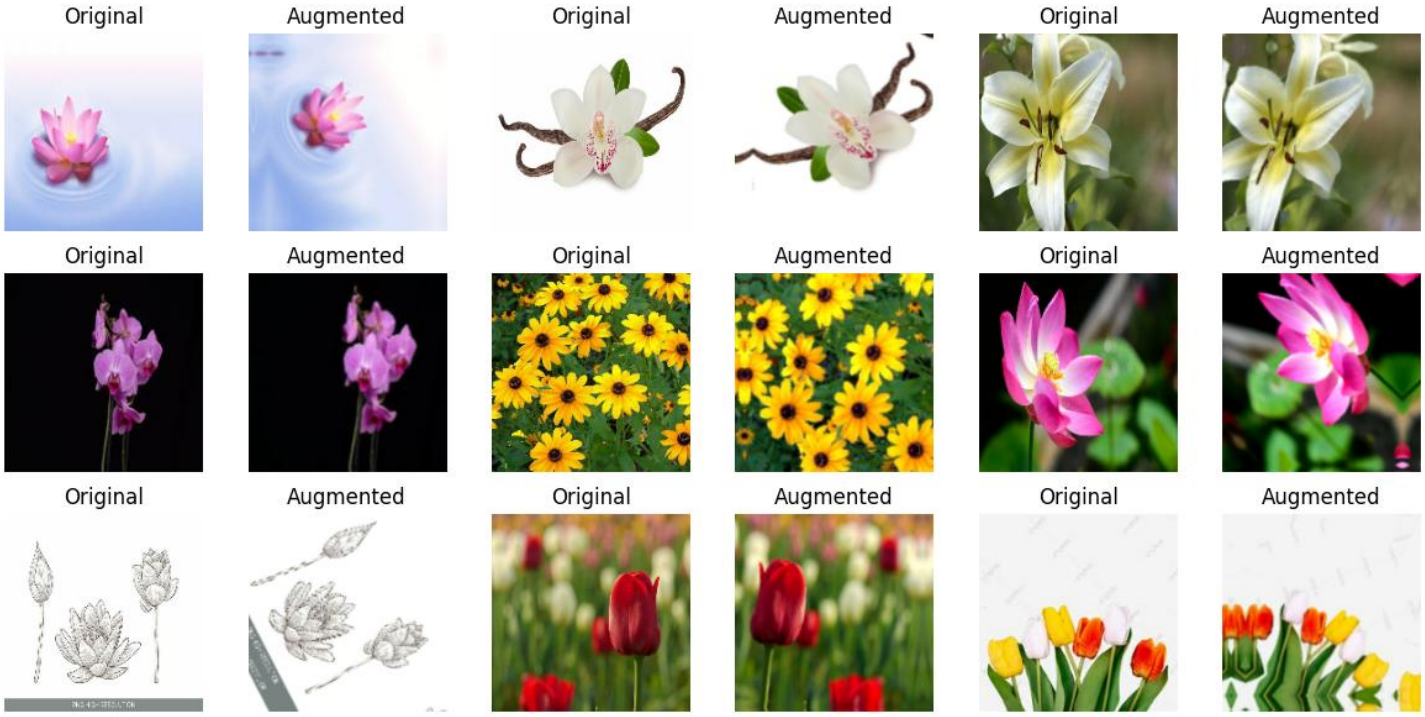
Sunflower

Lilly

Orchid

Sunflower

IMAGE DATA AUGMENTATION:



COUNT OF FLOWER SAMPLES:

Class Number	Flower Name	Number of Samples
0	Lily	821
1	Lotus	803
2	Orchid	800
3	Sunflower	791
4	Tulip	785

CLASSIFICATION REPORT:

	precision	recall	f1-score	support
Lilly	0.59	0.84	0.69	178
Lotus	0.85	0.83	0.84	197
Orchid	0.82	0.74	0.78	200
Sunflower	0.96	0.92	0.94	209
Tulip	0.93	0.76	0.83	215
accuracy			0.82	999
macro avg	0.83	0.82	0.82	999
weighted avg	0.84	0.82	0.82	999

CNN MODEL BUILDING:

Layer (type)	Output Shape	Param #
sequential (Sequential)	(1, 180, 180, 3)	0
rescaling (Rescaling)	(1, 180, 180, 3)	0
conv2d (Conv2D)	(1, 180, 180, 16)	448
max_pooling2d (MaxPooling2D)	(1, 90, 90, 16)	0
conv2d_1 (Conv2D)	(1, 90, 90, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(1, 45, 45, 32)	0
conv2d_2 (Conv2D)	(1, 45, 45, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(1, 22, 22, 64)	0
dropout (Dropout)	(1, 22, 22, 64)	0
flatten (Flatten)	(1, 30976)	0
dense (Dense)	(1, 128)	3,965,056
dense_1 (Dense)	(1, 5)	645

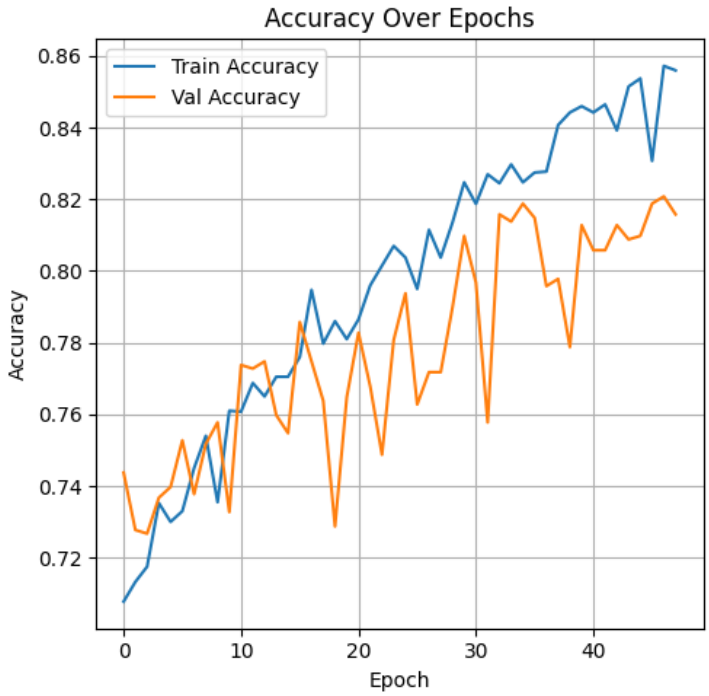
Total params: 3,989,285 (15.22 MB)
Trainable params: 3,989,285 (15.22 MB)
Non-trainable params: 0 (0.00 B)

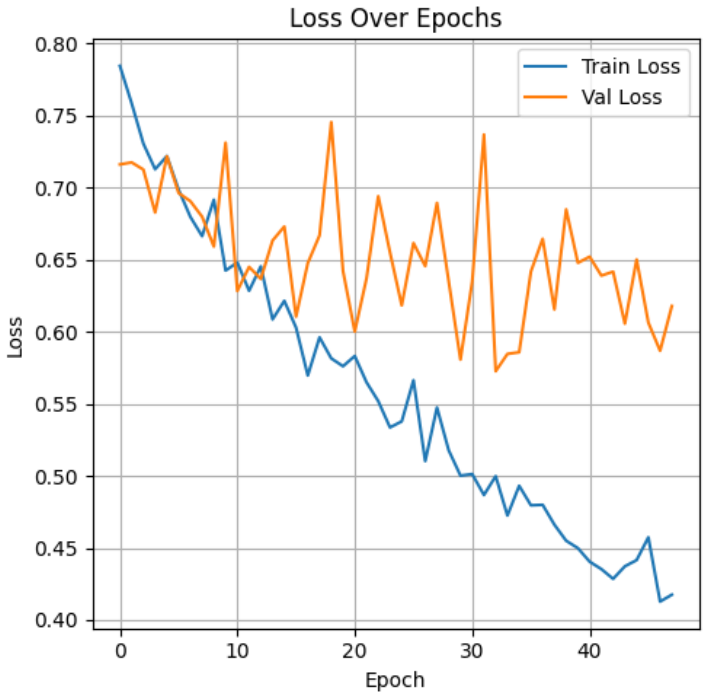
EPOCH: Epoch 48/65
125/125 ━━━━━━━━━━━━━━━━━━━━ **9s** 75ms/step - accuracy: 0.8623 - loss: 0.4082 - val_accuracy: 0.8158 - val_loss: 0.6178

EVALUATION:
32/32 ━━━━━━━━━━━━━━━━━━━━ **2s** 51ms/step - accuracy: 0.8352 - loss: 0.5239
Validation Loss: 0.572620689868927
Validation Accuracy: 0.815815806388855

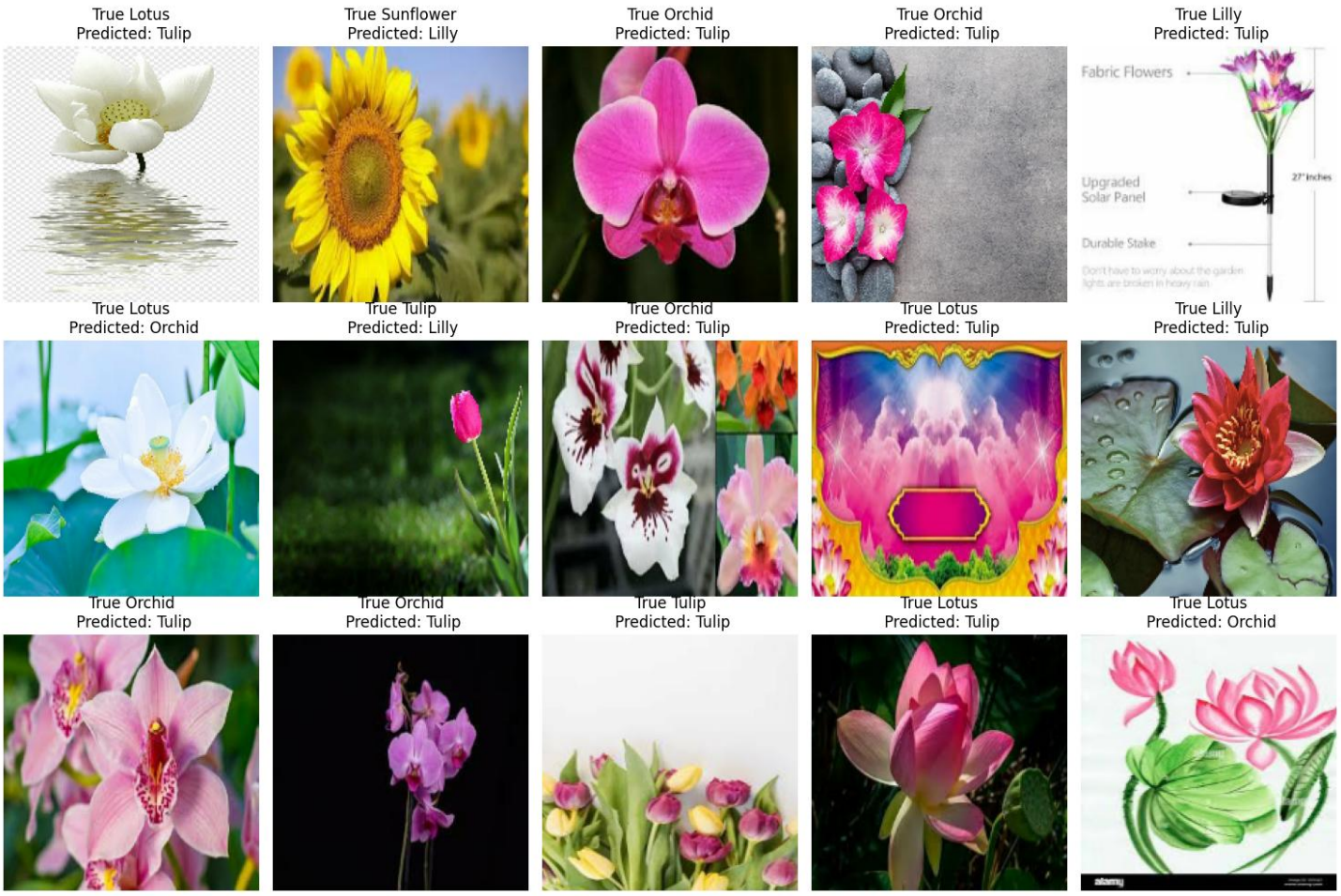
Model Prediction
Prediction vector: [[7.1037477e-01 5.9976440e-02 2.2765869e-01 1.4574826e-12 1.9901409e-03]]
Sum of probabilities: 1.0000001
Predicted class index: 0
Prediction: Lilly (71.04%)

GRAPH:

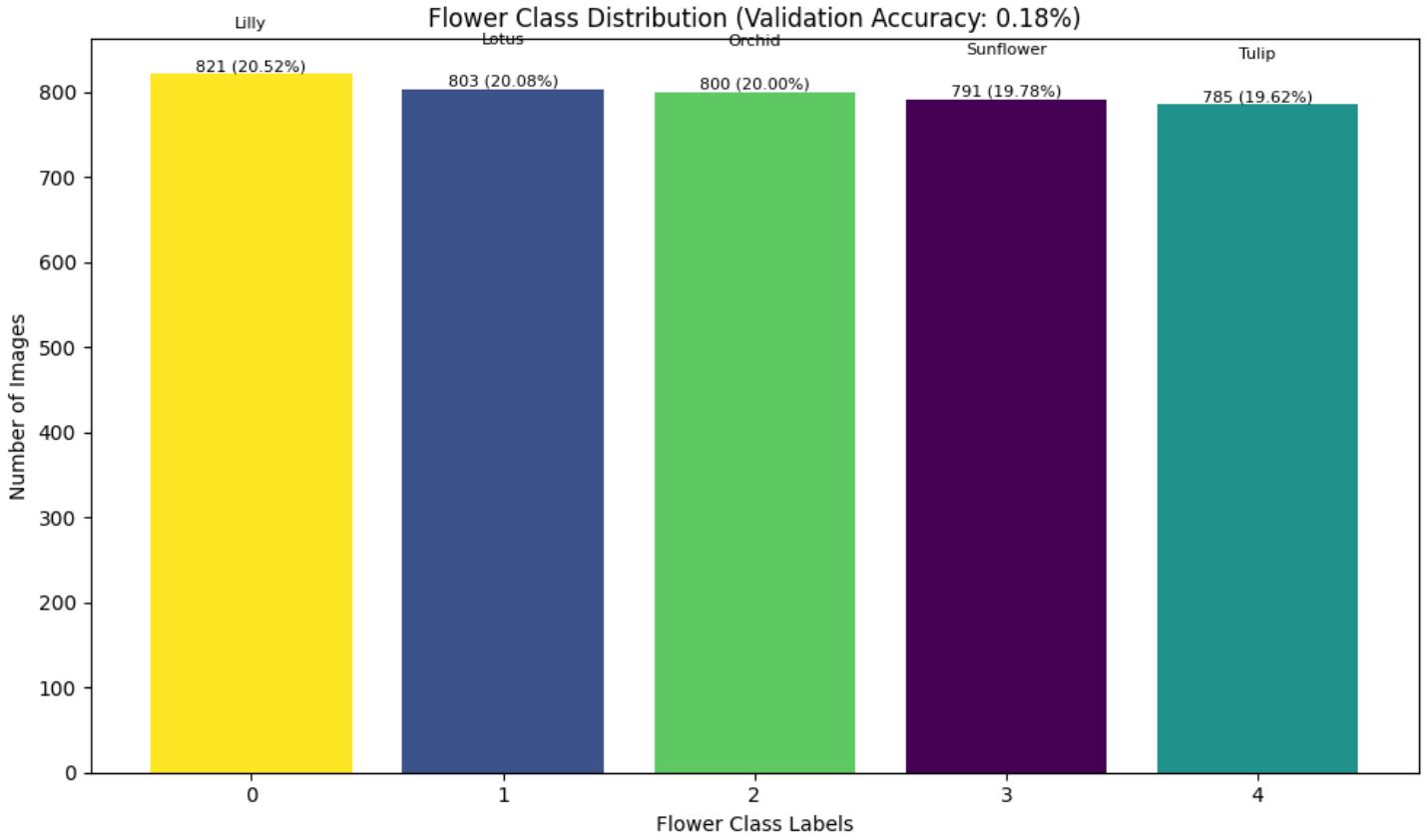




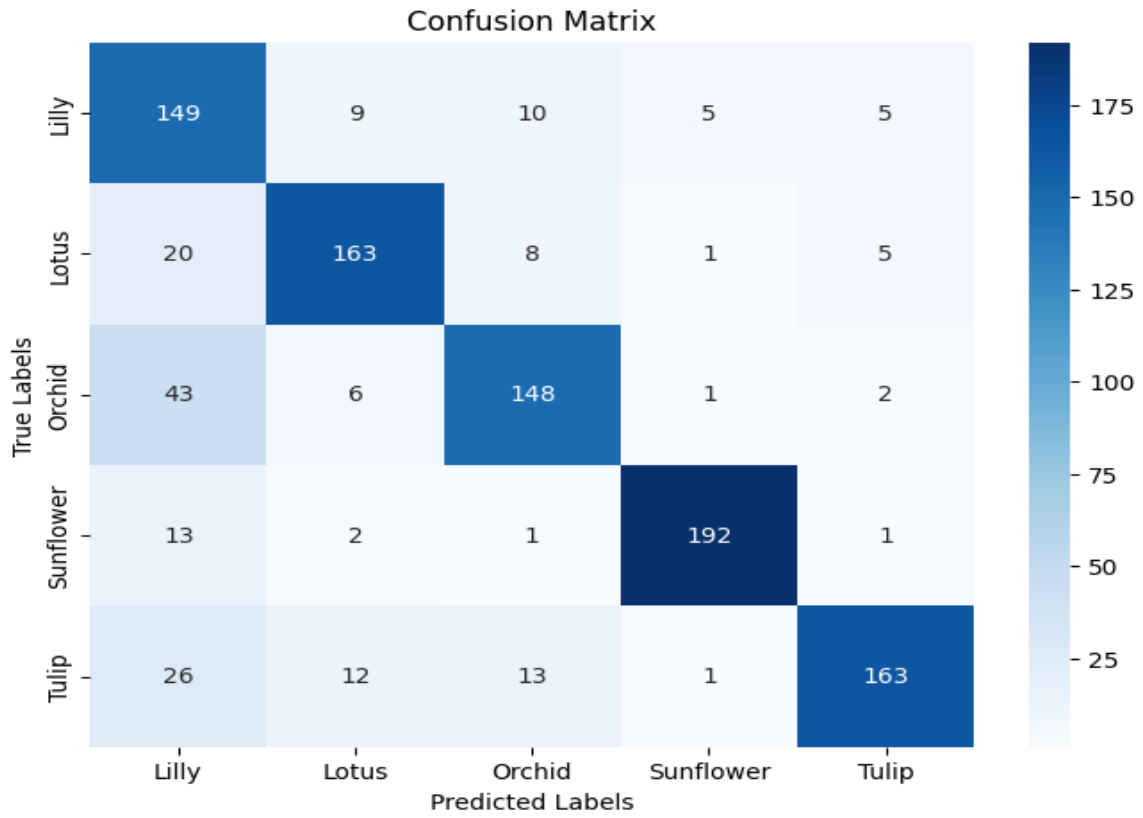
FUNCTION FOR PREDICTIONS:



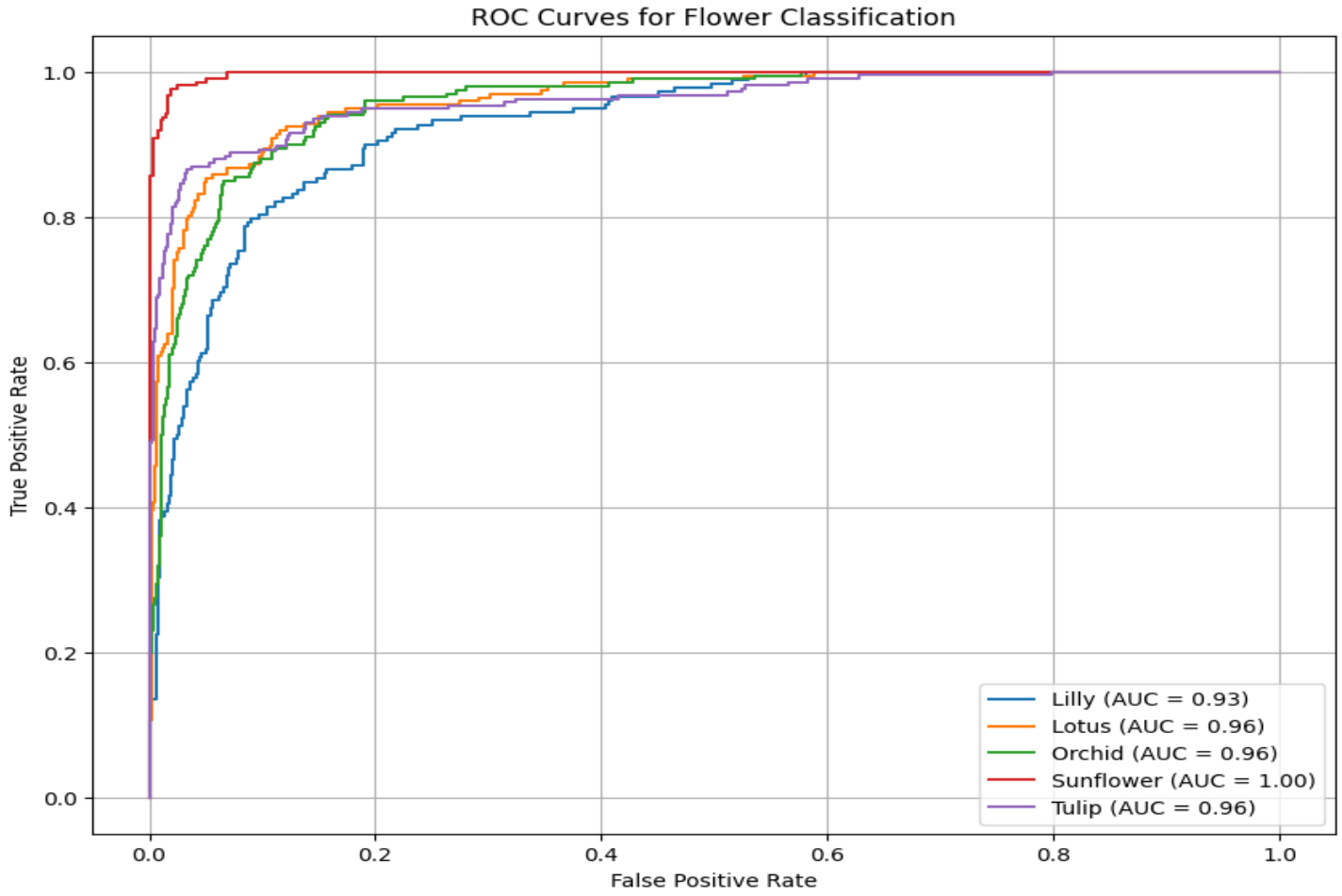
HISTOGRAM:



CONFUSION MATRIX:



ROC CURVE:



DISPLAY IMAGE INTO RGB, GRAYSCALE, WATERSHED TRANSFORM, EDGE DETECTION