Software Technology 1 Capstone Project

Ethan Morrison // u3258480

<https://www.kaggle.com/datasets/utkarshsaxenadn/flower-classification-5-classes-roselilyetc>

<https://github.com/ethanl-m/Software-Technology-Capstone>

# Introduction

The objective of this project is to develop a flower classification system using Convolutional Neural Networks (CNNs). The dataset utilized for this project, named "Flower Classification V2," comprises images of various flower species, including Aster, Daisy, Iris, Lavender, Lily, Marigold, Orchid, Poppy, Rose, and Sunflower. The project encompasses two primary files: Train.py for model training and GUI.py for the graphical user interface integration with the trained model.

# Exploratory Data Analysis (EDA)

## Questions for exploration:

1. What is the distribution of flower species/classes in the dataset?
2. How do various features (e.g., petal length, petal width, sepal length, sepal width) vary across different species?
3. Are there any correlations between different features?
4. Is there any outlier detection needed?
5. Can we visualize clusters within the dataset?

# Model Building and Training

The CNN model was constructed using TensorFlow and Keras within Train.py. It consisted of multiple convolutional and pooling layers followed by fully connected layers. Key steps included:

* Preprocessing images, resizing them to a standard size, and normalizing pixel values.
* Splitting the dataset into training, validation, and testing sets.
* Training the model using the training set, validating it on the validation set, and evaluating its performance on the testing set.
* Achieved an accuracy of 79% on the testing set.

The trained CNN model was saved as 'flower\_classification\_model.h5' in the working directory using Keras' model.save() method.

# GUI.py - Graphical User Interface

The GUI.py file integrates the trained model into a user-friendly interface using Tkinter. It offers functionality for users to upload an image and obtain predictions for the flower class. Upon image upload, the system preprocesses the image, feeds it into the trained model, and displays the predicted flower class.

Classification of Different Classes

* Aster: Description and classification details.
* Daisy: Description and classification details.
* Iris: Description and classification details.
* Lavender: Description and classification details.
* Lily: Description and classification details.
* Marigold: Description and classification details.
* Orchid: Description and classification details.
* Poppy: Description and classification details.
* Rose: Description and classification details.
* Sunflower: Description and classification details.

# Conclusion

The project successfully implements a flower classification system using CNNs. The GUI facilitates user interaction by enabling them to upload images for classification. The trained model demonstrates accurate predictions for various flower species, providing a practical solution for automated flower classification.

# Future Improvements

* Implement advanced CNN architectures
* Enhance the GUI with more features, such as batch prediction, image editing tools, and result visualization.