Flower Classification Model using VGG16

Auhona Chakraborty

Project Introduction

In this project, I aimed to develop a flower classification model using deep learning techniques. The goal was to accurately classify images of flowers into one of five categories: dandelion, daisy, tulip, sunflower, and rose. The model was trained on a dataset consisting of 4242 images collected from various sources such as Flickr, Google Images, and Yandex Images. Each class contained approximately 800 photos, with varying resolutions and proportions, around 320x240 pixels.

Libraries used

Numpy

Utilized for efficient handling of image data

Keras

A high-level deep learning API that facilitated the creation and training of neural networks.

Pandas

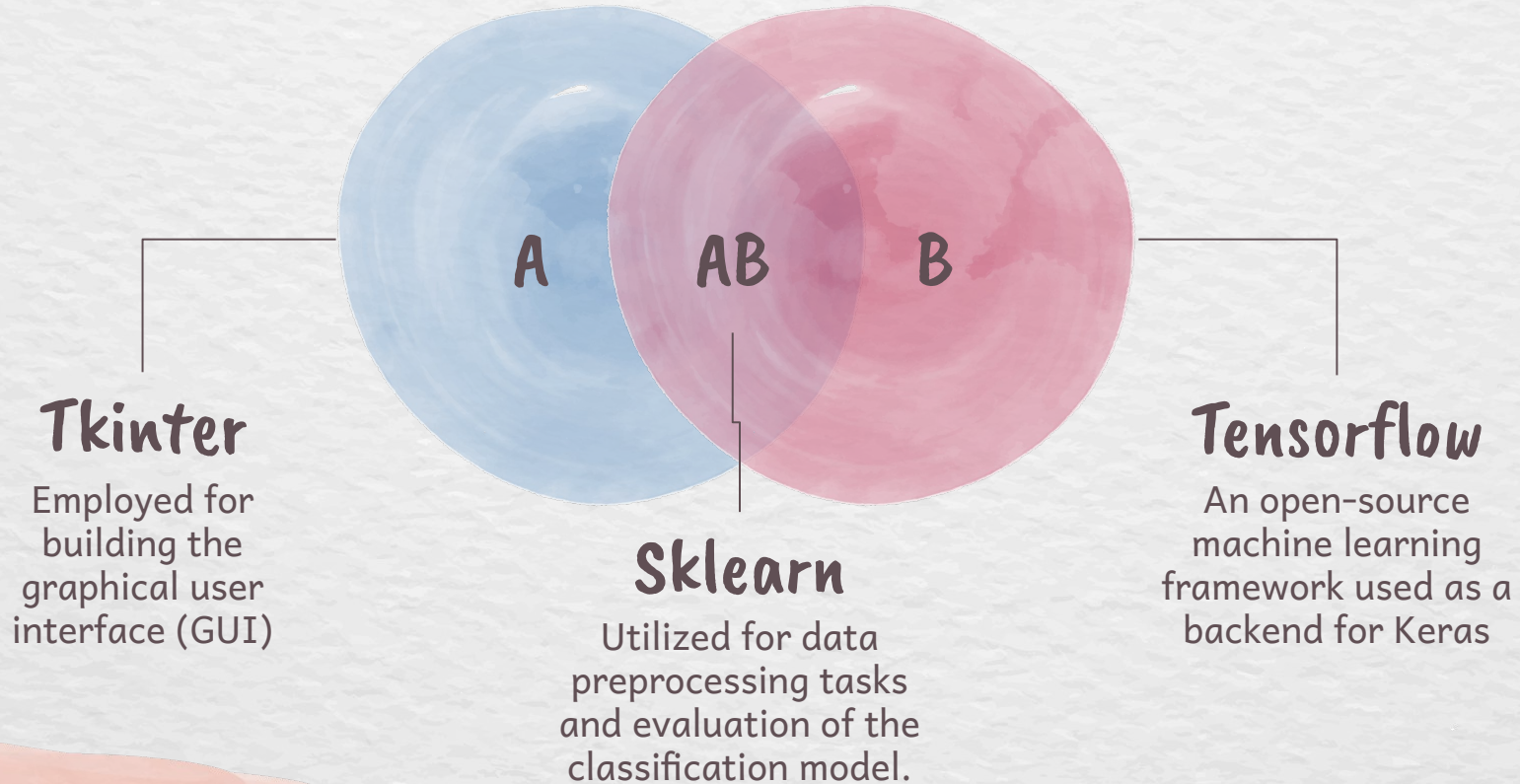Used for data manipulation and analysis.

OpenCV

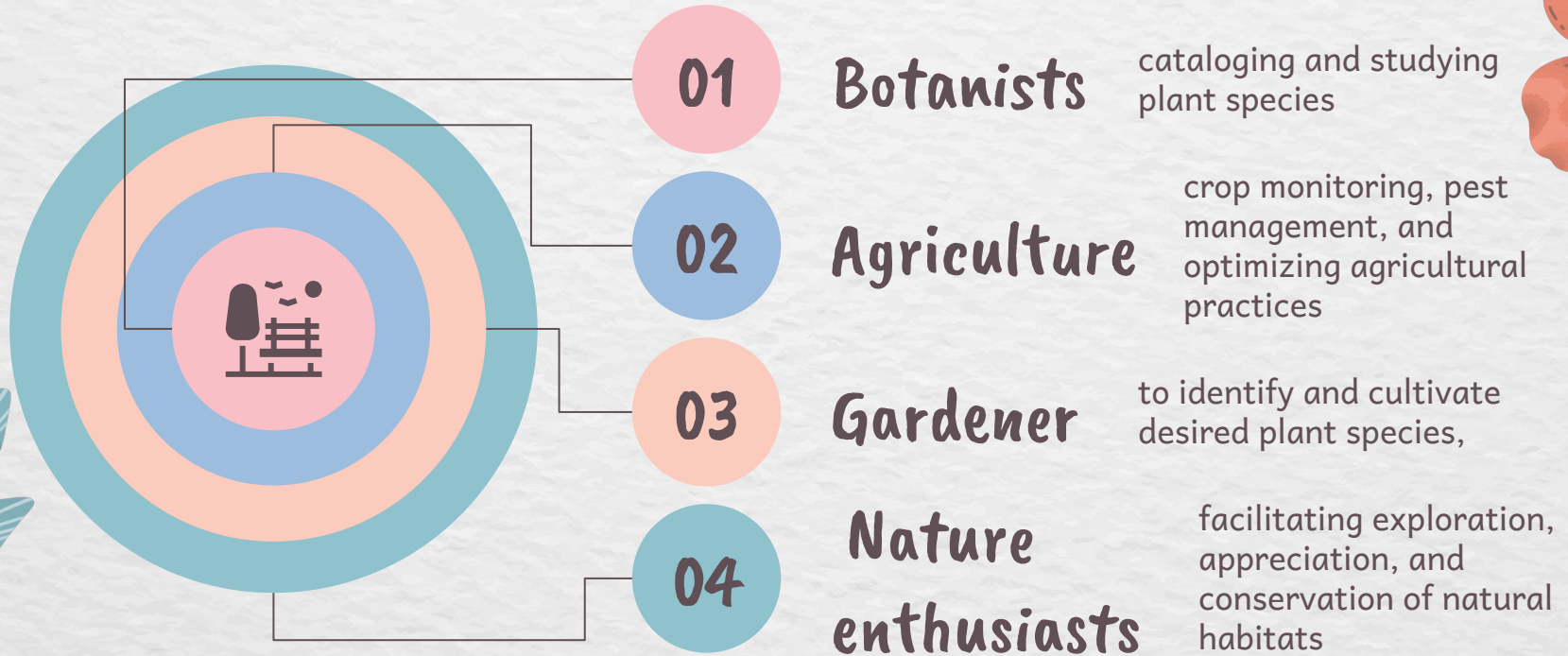Used for image preprocessing tasks such as reading and resizing images.

Libraries Used Continued...

A   AB   B

Tkinter

Employed for building the graphical user interface (GUI)

Sklearn

Utilized for data preprocessing tasks and evaluation of the classification model.

Tensorflow

An open-source machine learning framework used as a backend for Keras

Purpose of the Project

01 Botanists — cataloging and studying plant species

02 Agriculture — crop monitoring, pest management, and optimizing agricultural practices

03 Gardener — to identify and cultivate desired plant species,

04 Nature enthusiasts — facilitating exploration, appreciation, and conservation of natural habitats

About VGG16

Intro — VGG16, or Visual Geometry Group 16, is a deep convolutional neural network architecture
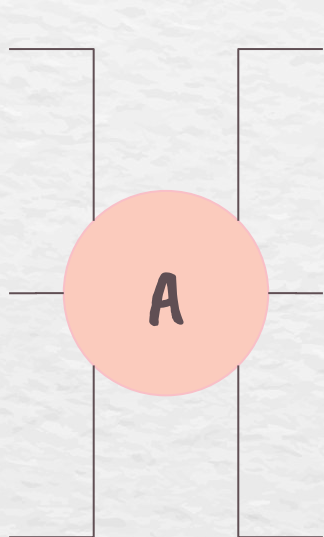
Depth — It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers.

Conv — The network utilizes small 3x3 convolutional filters followed by ReLU activation and max-pooling.

A

FC layer — Three fully connected layers process the high-dimensional feature vector for classification

pretrained — VGG16 is often used as a pre-trained model, fine-tuned for specific tasks.

Apps — Widely applied in image classification, detection, and feature extraction tasks

Validation Accuracy : 86.57%

Validation Loss : 61.20%

THANK YOU!