INTRODUCTION

Machine learning

Machine learning is a subset of artificial intelligence that focuses on building systems that learn from data and improve with experience. The system identifies patterns and relationships within large datasets, enabling it to handle complex, dynamic problems where manual rule-writing is impractical or impossible.

- **Process:** Input data and corresponding output data are fed into an algorithm during a training phase, which then generates a model (the learned "rules").
- **Adaptability:** The model adapts and improves its performance over time as it is exposed to more data, often without manual intervention in the core logic.
- **Use Cases:** It is ideal for data-rich problems with complex patterns, such as image recognition, natural language processing, recommendation engines, and fraud detection.

Traditional Programming

In traditional programming, the programmer has a clear understanding of the problem and manually defines every step and logic the computer must follow to achieve a predictable, deterministic outcome.

- **Process:** A programmer writes explicit rules (program) to process input data and generate output.
- **Adaptability:** The program's logic is static; any change in requirements or data requires manual code updates and re-deployment.
- **Use Cases:** It is best suited for well-defined problems with clear, stable rules and structured data, such as basic calculations, payroll processing, or inventory management.

OBJECTIVES OF PROJECT

- To develop an image recognition system using Machine Learning techniques.
- To implement deep learning models using TensorFlow for accurate image classification.
- To use Google Colab platform for training and testing the model in an online environment.
- To train the model using a labeled image dataset for automatic object detection and classification.
- To preprocess and upload images for improving detection accuracy.

DATASET DESCRIPTION

In this Image Recognition project, a labeled image dataset is used to train and evaluate the deep learning model. The dataset consists of multiple categories (classes) of images, where each image is assigned a specific label, enabling supervised learning. The images are organized in a directory-based structure, with each folder representing a separate class. The dataset is loaded using TensorFlow and implemented on the Google Colab platform. All images are in RGB format and are resized to a fixed dimension before training to ensure uniformity.

Data preprocessing steps such as normalization (scaling pixel values from 0–255 to 0–1), shuffling, and splitting into training and validation sets are applied to improve model performance and generalization. Typically, the dataset is divided into training data for learning and validation data for performance evaluation. This structured and labeled dataset helps the model learn important visual features and accurately classify new images, enabling the development of an efficient image recognition system.

METHODOLOGY

- Data Collection
  The flower image dataset was obtained from the TensorFlow public dataset containing five classes: roses, tulips, daisies, sunflowers, and dandelions.
- Data Preprocessing
  Images were resized to 180×180 pixels, normalized (pixel values scaled to 0–1), and split into training (80%) and validation (20%) datasets.
- Model Design
  A Convolutional Neural Network (CNN) was built using TensorFlow/Keras with Conv2D, MaxPooling, Flatten, and Dense layers.
- Model Compilation
  The model was compiled using the Adam optimizer, Sparse Categorical Crossentropy loss, and accuracy metric.
- Model Training
  The CNN was trained for 10 epochs using the training dataset while monitoring validation performance.
- Model Evaluation
  Model accuracy and loss were analyzed using the validation dataset to check performance and overfitting.
- Prediction System
  A prediction function processes uploaded images, resizes & normalizes them, and outputs class probabilities.
- Deployment
  A Gradio interface was created to allow users to upload flower images and receive predictions.

CODE

1. !pip install gradio

2. import matplotlib.pyplot as plt

import numpy as np

import os

import PIL

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras.models import Sequential

3. import pathlib

dataset_url="https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz"

data_dir=tf.keras.utils.get_file('flower_photos',origin=dataset_url,untar=True)

data_dir=pathlib.Path(data_dir)

data_dir = data_dir / 'flower_photos' # Correct the path to the nested directory

4. roses=list(data_dir.glob('roses/*'))

print(roses[0])

PIL.Image.open(str(roses[0]))

5. img_height,img_width=180,180

batch_size=32

train_ds=tf.keras.preprocessing.image_dataset_from_directory(

    data_dir,

    validation_split=0.2,

    subset="training",

```
    seed=123,
    image_size=(img_height,img_width),
    batch_size=batch_size
```

6. val_ds=tf.keras.preprocessing.image_dataset_from_directory(

```
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_height,img_width),
    batch_size=batch_size
)
```

7. class_names=train_ds.class_names

```
print(class_names)
```

8. import matplotlib.pyplot as plt

```
plt.figure(figsize=(10,10))
for Images,labels in train_ds.take(1):
  for i in range(9):
    ax=plt.subplot(3,3,i+1)
    plt.imshow(Images[i].numpy().astype("uint8"))
    plt.title(class_names[labels[i]])
    plt.axis("off")
```

9. num_classes=5

```
model=Sequential([
    layers.Rescaling(1./255,input_shape=(img_height,img_width,3)),
    layers.Conv2D(16,3,padding='same',activation='relu'),
```

```
  layers.MaxPooling2D(),
  layers.Conv2D(32,3,padding='same',activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64,3,padding='same',activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128,activation='relu'),
  layers.Dense(num_classes,activation='softmax')
])
```

10. model.compile(optimizer='adam',

```
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=['accuracy'])
```

11. from IPython.core import history

```
epochs=10
history=model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs
)
```

12. def predict_image(img):

```
 # img is a numpy array from Gradio
 # Convert to TensorFlow tensor
 img_tensor = tf.convert_to_tensor(img, dtype=tf.float32)
 # Resize the image to the target size (img_height, img_width)
 img_resized = tf.image.resize(img_tensor, (img_height, img_width))
 # Add a batch dimension: (1, img_height, img_width, 3)
```

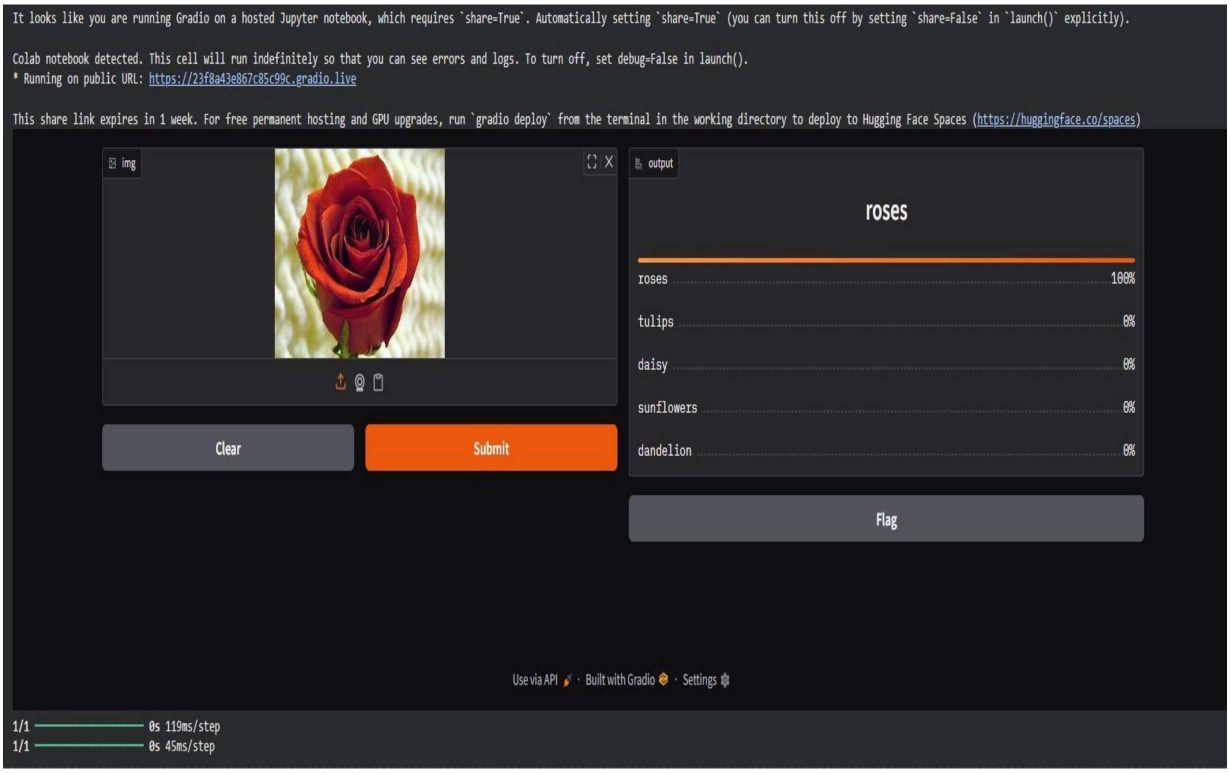
```
  img_batch = tf.expand_dims(img_resized, 0)
  prediction = model.predict(img_batch)[0]
  return {class_names[i]:float(prediction[i]) for i in range(len(class_names))}
```

13. import gradio as gr

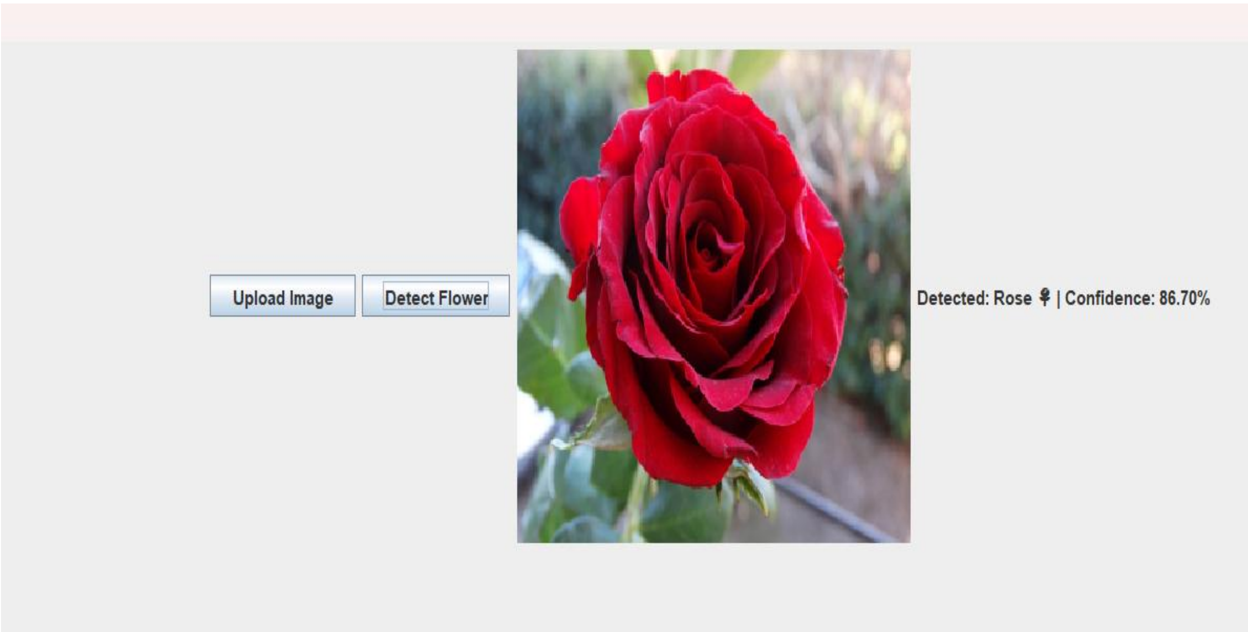
```
image=gr.Image()
label=gr.Label(num_top_classes=5)
gr.Interface(fn=predict_image,inputs=image,outputs=label).launch(debug=True)
```

OUTPUT

Machine learning output:



Traditional programming output:



APPLICATIONS

1.Agriculture & Farming

- Identify flower species for crop monitoring
- Detect plant health and diseases
- Help farmers in flower cultivation planning

2.Mobile Applications

- Apps that recognize flowers using camera (AI-based plant apps)
- Learning apps for students and gardeners

3.Botanical Research

- Classify and document plant species
- Support scientific research and biodiversity studies

4.Gardening & Horticulture

- Help gardeners identify unknown flowers
- Provide care instructions (water, soil, sunlight needs)

5.**Medical & Herbal Industry**

- Identify medicinal plants and flowers
- Support herbal medicine research

CONCLUSION

A flower recognition system is an advanced and intelligent application of artificial intelligence, machine learning, and image processing technologies that enables accurate identification and classification of different flower species from digital images. This system minimizes human effort, reduces the chances of error, and significantly speeds up the process of flower identification. By using deep learning models and computer vision techniques, it can analyze various features of flowers such as color, shape, size, texture, and petal structure to provide precise results. As a result, it becomes a reliable tool for users who need quick and correct identification.

The flower recognition system has wide-ranging applications in agriculture, horticulture, botanical research, education, environmental monitoring, healthcare, and mobile applications. In agriculture, it helps farmers in crop monitoring, disease detection, and improving flower production. In education, it supports interactive learning by enabling students to easily identify flowers and understand plant biodiversity.

REFERENCES

1. https://in.search.yahoo.com/search?fr=mcafee&type=E211IN826G0&p=tensorflow
2. https://www.gradio.app/
3. https://colab.research.google.com/drive/1Mk4R_kveq4qH7xCewaydu4XNyW5F9ntG

4.https://chatgpt.com/c/699e9919-c2b8-83a3-986d-188783041748