flower-detect-cnn

June 28, 2024

[ ]:
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
IMG_SIZE = 224
BATCH_SIZE = 32
```

[ ]:
```
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/flower dataset/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/flower dataset/val',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

```
Found 445 images belonging to 2 classes.
Found 2 images belonging to 2 classes.
```

[ ]:
```
model = keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_SIZE,␣
 ↪IMG_SIZE, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])
```

[ ]:
```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32,␣
 ↪(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2)
])
```

[ ]:
```
model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
```

[ ]:
```
model.fit(train_generator, epochs=5, validation_data=val_generator)
```

```
Epoch 1/5
14/14 [==============================] - 59s 4s/step - loss: 0.6075 - accuracy:
0.8292 - val_loss: 0.6957 - val_accuracy: 0.5000
Epoch 2/5
14/14 [==============================] - 55s 4s/step - loss: 0.2816 - accuracy:
0.8787 - val_loss: 0.3226 - val_accuracy: 1.0000
Epoch 3/5
14/14 [==============================] - 56s 4s/step - loss: 0.2205 - accuracy:
0.9191 - val_loss: 0.2425 - val_accuracy: 1.0000
Epoch 4/5
14/14 [==============================] - 54s 4s/step - loss: 0.1698 - accuracy:
0.9438 - val_loss: 1.2198 - val_accuracy: 0.5000
Epoch 5/5
14/14 [==============================] - 54s 4s/step - loss: 0.1567 - accuracy:
0.9483 - val_loss: 0.8982 - val_accuracy: 0.5000
```

[ ]:
```
<keras.src.callbacks.History at 0x7b2dceecb2e0>
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[ ]:
```
model.save("flower_dataset.h5","label.txt")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

[ ]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('/content/flower_dataset.h5')
test_image_path = '/content/drive/MyDrive/flower dataset/train/rose/
 ↪3446770760_1c8fee5f7c_c.jpg'
img = image.load_img(test_image_path, target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)
```

```
1/1 [==============================] - 0s 152ms/step
[[0.00316723]]
```

[ ]:
```
if predictions < 0.5:
    print('It is a Rose')
else:
    print('It is a Waterlily')
```

```
It is a Rose
```