day1-04

June 25, 2024

[4]:

```
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
```

[7]:

```
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Art_Dataset',
    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/Art_Dataset',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

```
Found 777 images belonging to 2 classes.
Found 193 images belonging to 2 classes.
```

[8]:

```
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')
])
```

[9]:

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

[10]:

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

```
Epoch 1/5
 4/25 [===>…] - ETA: 1:53 - loss: 2.2691 - accuracy:
0.5391

/usr/local/lib/python3.10/dist-packages/PIL/Image.py:996: UserWarning: Palette
images with Transparency expressed in bytes should be converted to RGBA images
  warnings.warn(

25/25 [==============================] - 181s 7s/step - loss: 0.9500 - accuracy:
0.5405 - val_loss: 0.6746 - val_accuracy: 0.5544
Epoch 2/5
25/25 [==============================] - 130s 5s/step - loss: 0.6747 - accuracy:
0.5676 - val_loss: 0.6347 - val_accuracy: 0.6580
Epoch 3/5
25/25 [==============================] - 127s 5s/step - loss: 0.6565 - accuracy:
0.6049 - val_loss: 0.6251 - val_accuracy: 0.6632
Epoch 4/5
25/25 [==============================] - 130s 5s/step - loss: 0.6028 - accuracy:
0.6731 - val_loss: 0.6216 - val_accuracy: 0.6580
Epoch 5/5
25/25 [==============================] - 144s 6s/step - loss: 0.5158 - accuracy:
0.7477 - val_loss: 0.6959 - val_accuracy: 0.6425
```

[10]:

```
<keras.src.callbacks.History at 0x7d1c5770d330>
```

[11]:

```
model.save("art_classifier.h5","label.text")
```

```
/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(
```

[13]:

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

model = load_model('/content/drive/MyDrive/art_classifier.h5')

test_image = image.load_img('/content/drive/MyDrive/Art_Dataset/AiArtData/
 ↪AI_image.png', target_size=(224, 224))
img = image.img_to_array(test_image)
img = np.expand_dims(img, axis=0)
img = img / 255.0

prediction = model.predict(img)
print(prediction)
```

```
1/1 [==============================] - 0s 161ms/step
[[0.5196866]]
```

[16]:

```
if prediction < 0.5:
    print("prediction: Human_Art (Probablity:)", prediction[0][0])
else:
    print("prediction:AI_Art (Probablity:)", prediction[0][0])
```

```
prediction:AI_Art (Probablity:) 0.5196866
```

[6]:

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

```
Mounted at /content/drive
```