day02

June 26, 2024

[ ]:
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 244
BATCH_SIZE=32
```

[ ]:
```

```

[ ]:
```
train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/1SV21CS035/data',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator =
    train_datagen.flow_from_directory('/content/driv
    e/MyDrive/1SV21CS035/data',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
```

```
Found 143 images belonging to 2 classes.
Found 35 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') #output layer
])
```

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

[ ]:
```
model.fit(train_generator, validation_data=val_generator,epochs=5) # Fixed typo:
↪ train_genertor -> train_generator
```

```
Epoch 1/5
5/5 [==============================] - 63s 13s/step - loss: 1.3043 - accuracy:
0.5000 - val_loss: 0.6984 - val_accuracy: 0.5000
Epoch 2/5
5/5 [==============================] - 25s 5s/step - loss: 0.6941 - accuracy:
0.5000 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 3/5
5/5 [==============================] - 27s 5s/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 4/5
5/5 [==============================] - 26s 5s/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 5/5
5/5 [==============================] - 28s 6s/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
```

[ ]:
```
<keras.src.callbacks.History at 0x7877f2acf6d0>
```

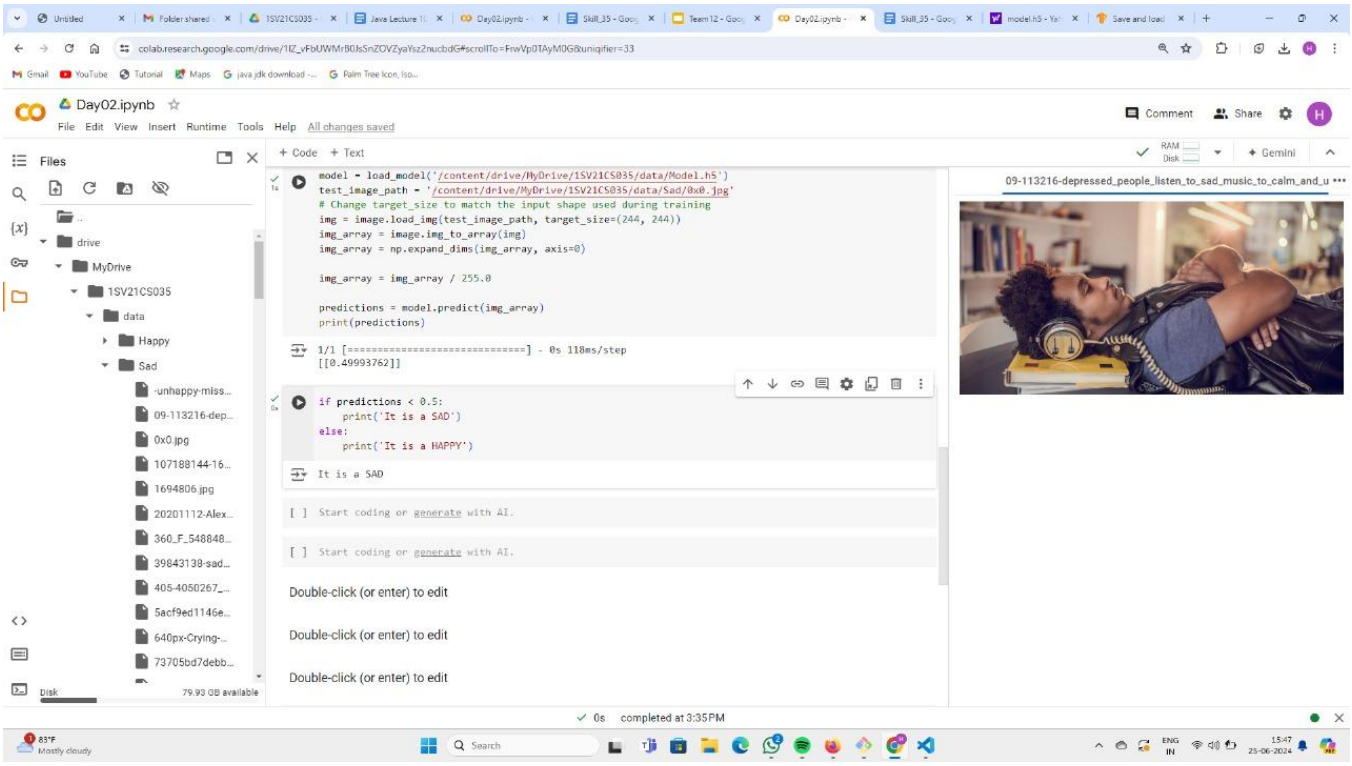
[ ]:
```
model.save("Model.h5","label.txt")
```

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

model = load_model('/content/drive/MyDrive/1SV21CS035/data/Model.h5')
test_image_path = '/content/drive/MyDrive/1SV21CS035/data/Sad/0x0.jpg'
# Change target_size to match the input shape used during training
img = image.load_img(test_image_path, target_size=(244, 244))
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 118ms/step
[[0.49993762]]
```

[ ]:
```python
if predictions < 0.5:
    print('It is a SAD')
else:
    print('It is a HAPPY')
```

```
It is a SAD
```


Day02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
Gemini
model = load_model('/content/drive/MyDrive/1SV21CS035/data/Model.h5')
test_image_path = '/content/drive/MyDrive/1SV21CS035/data/Sad/0x0.jpg'
# Change target_size to match the input shape used during training
img = image.load_img(test_image_path, target_size=(244, 244))
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 118ms/step
[[0.49993762]]
if predictions < 0.5:
print('It is a SAD')
else:
print('It is a HAPPY')
It is a SAD
[ ] Start coding or generate with AI.
[ ] Start coding or generate with AI.
Double-click (or enter) to edit
Double-click (or enter) to edit
Double-click (or enter) to edit
drive
MyDrive
1SV21CS035
data
Happy
Sad
09-113216-depressed_people_listen_to_sad_music_to_calm_and_u
0s completed at 3:35PM