Comment   Share

+ Code + Text   RAM Disk

```
[1]
!pip install rarfile
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting rarfile
  Downloading rarfile-4.0-py3-none-any.whl (28 kB)
Installing collected packages: rarfile
Successfully installed rarfile-4.0
```

```
[2]
import rarfile

with rarfile.RarFile('/content/drive/MyDrive/confident-unconfident.rar') as rf:
    rf.extractall()
```

```python
# Import necessary libraries
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.applications import VGG19
import os
import numpy as np
import cv2
from sklearn.metrics import confusion_matrix, f1_score, accuracy_score, precision_score, roc_curve, roc_auc_score
import matplotlib.pyplot as plt

# Declared paths for train, val and test sets
train_data_dir = '/content/confident-unconfident/train'
validation_data_dir = '/content/confident-unconfident/val'
test_data_dir = '/content/confident-unconfident/test'

# Declared image dimensions and batch size
img_height = 128
img_width = 128
batch_size = 32
num_classes=2

# Loading train, validation and test sets

# Creating an ImageDataGenerator for train,test and validation data
train_datagen = ImageDataGenerator(
    rescale=1./255)
validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Loading the train,test and validation data from directory using flow_from_directory method
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    color_mode='grayscale',
    class_mode='binary')

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    color_mode='grayscale',
    class_mode='binary')

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    color_mode='grayscale',
    class_mode='binary')

# Defining the model architecture
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu' ,padding="same", input_shape=(img_height, img_width, 1)),
```

```python
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu' ,padding="same"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',padding="same"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

# printing model summary
print(model.summary())

...
# Define the model architecture
base_model = VGG19(weights='imagenet', include_top=False, input_shape=(img_width, img_height, 3))

for layer in base_model.layers:
    layer.trainable = False

model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))  #my inference remove dropout layer because may be my model required more feature
model.add(Dense(num_classes, activation='softmax'))
...
# Defining the optimizer with a specific learning rate
Optimizer = tf.keras.optimizers.Adam(learning_rate=0.0002)

# Compiling the model
model.compile(optimizer=Optimizer,
              loss=tf.losses.BinaryCrossentropy(),
              metrics=['accuracy'])


# Defining callbacks
logdir = 'logs'
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)
early_stopping_callback = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)

# Training the model
history = model.fit(train_generator,
                    epochs=8,
                    validation_data=validation_generator,
                    callbacks=[tensorboard_callback, early_stopping_callback])


# Saving the model
model.save('/content/final_model.h5')
```

```
Found 22354 images belonging to 2 classes.
Found 6014 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
Model: "sequential_5"
```

Layer (type)	Output Shape	Param #
conv2d_15 (Conv2D)	(None, 128, 128, 32)	320
max_pooling2d_15 (MaxPoolin g2D)	(None, 64, 64, 32)	0
conv2d_16 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_16 (MaxPoolin g2D)	(None, 32, 32, 64)	0
conv2d_17 (Conv2D)	(None, 32, 32, 32)	18464
max_pooling2d_17 (MaxPoolin g2D)	(None, 16, 16, 32)	0
flatten_5 (Flatten)	(None, 8192)	0
dense_10 (Dense)	(None, 128)	1048704

```
dense_11 (Dense)              (None, 1)                  129

=================================================================
Total params: 1,086,113
Trainable params: 1,086,113
Non-trainable params: 0
_____
None
Epoch 1/8
699/699 [==============================] - 21s 28ms/step - loss: 0.6375 - accuracy: 0.6282 - val_loss: 0.6006 - val_accuracy: 0.6643
Epoch 2/8
699/699 [==============================] - 20s 28ms/step - loss: 0.5877 - accuracy: 0.6810 - val_loss: 0.5663 - val_accuracy: 0.6947
                                                            0.5543 - accuracy: 0.7078 - val_loss: 0.5571 - val_accuracy: 0.7075
Epoch 4/8
699/699 [==============================] - 17s 24ms/step - loss: 0.5212 - accuracy: 0.7323 - val_loss: 0.5301 - val_accuracy: 0.7225
Epoch 5/8
699/699 [==============================] - 20s 28ms/step - loss: 0.4917 - accuracy: 0.7559 - val_loss: 0.5343 - val_accuracy: 0.7305
Epoch 6/8
699/699 [==============================] - 17s 24ms/step - loss: 0.4635 - accuracy: 0.7765 - val_loss: 0.5238 - val_accuracy: 0.7368
Epoch 7/8
699/699 [==============================] - 17s 24ms/step - loss: 0.4324 - accuracy: 0.7963 - val_loss: 0.5150 - val_accuracy: 0.7399
Epoch 8/8
699/699 [==============================] - 17s 24ms/step - loss: 0.4002 - accuracy: 0.8126 - val_loss: 0.5320 - val_accuracy: 0.7288
```

```
[ ]  !pip install tensorboard
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorboard in /usr/local/lib/python3.9/dist-packages (2.11.2)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (2.27.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (1.8.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (0.40.0)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (1.4.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (2.16.2)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (3.4.1)
                                                        5.0 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (0.6.1)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (67.6.0)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (2.2.3)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (3.19.6)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (0.4.6)
Requirement already satisfied: numpy>=1.12.0 in /usr/local/lib/python3.9/dist-packages (from tensorboard) (1.22.4)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from google-auth<3,>=1.6.3->tensorboard)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.9/dist-packages (from google-auth<3,>=1.6.3->tensorboard)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages (from google-auth<3,>=1.6.3->tensorboard) (1.16.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.9/dist-packages (from google-auth<3,>=1.6.3->tensorboard) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.9/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.9/dist-packages (from markdown>=2.6.8->tensorboard) (6.1.
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests<3,>=2.21.0->tensorboard) (1.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3,>=2.21.0->tensorboard) (2022.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests<3,>=2.21.0->tensorboard) (3.4)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests<3,>=2.21.0->tensorboard)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.9/dist-packages (from werkzeug>=1.0.1->tensorboard) (2.1.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorb
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.9/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.9/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauth
```

```
[ ]  %load_ext tensorboard
```

```
%tensorboard --logdir=/content/logs --port 6001
```

TensorBoard   TIME SERIES   SCALARS   GRAPHS   INACTIVE

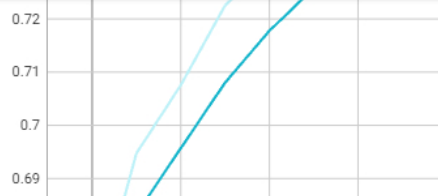Filter runs (regex)   Filter tags (regex)   All   Scalars   Image   Histogram   Settings

Run

train

validation

epoch_accuracy

```
0.72


0.71


0.7


0.69
```

epoch_loss

epoch_loss
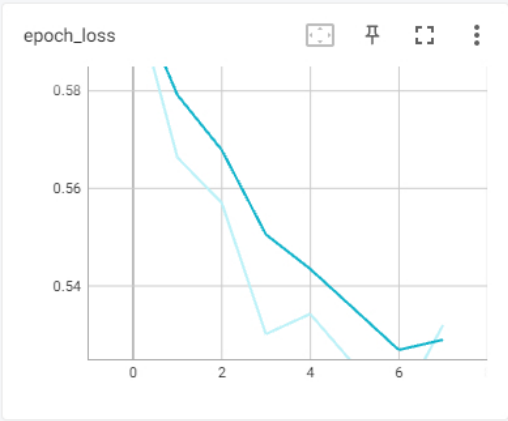
Cancel contracts here

2s   completed at 1:47 PM