1_cs2_premodels_classification

October 2, 2021

[ ]:
```
!nvidia-smi -L
```

```
GPU 0: Tesla K80 (UUID: GPU-1c3b12fd-da17-8ebf-fb47-ea235e676e98)
```

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

```
Mounted at /content/gdrive
```

[ ]:
```
import tensorflow as tf
from tensorflow.keras import models, layers
#from tensorflow.keras.models import Model
#from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import *
from tensorflow.keras.callbacks import LearningRateScheduler,␣
↪ReduceLROnPlateau,ModelCheckpoint, TerminateOnNaN, EarlyStopping,␣
↪TensorBoard,CSVLogger
import datetime
import os
from tensorflow.keras import backend as K
#from tensorflow.keras.models import load_model, save_model
#from tensorflow.keras.preprocessing.image import␣
↪ImageDataGenerator,img_to_array,load_img
import keras
from keras.preprocessing.image import ImageDataGenerator,img_to_array,load_img

import pandas as pd
import numpy as np
import glob
import os
import cv2
from IPython.display import Image
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras.preprocessing import image
```

```
from sklearn.preprocessing import *
#import xgboost as xgb
import warnings
warnings.filterwarnings('ignore')

from scipy import stats
from datetime import *
import datetime
import pickle
```

[ ]:
```
train_data_f = pickle.load(open('/content/gdrive/MyDrive/cs2/data/train_data_f.
↪pkl', 'rb'))
train_data_f.head(3)
```

[ ]:
```
        Image_path Insurance_company  …  Condition  Amount
0  img_4513976.jpg                BQ  …          0     0.0
1  img_7764995.jpg                BQ  …          1  6194.0
2   img_451308.jpg                 A  …          0     0.0

[3 rows x 25 columns]
```

[ ]:
```
x = train_data_f['Image_path']
y = train_data_f['Condition']

train,  cv = train_test_split(train_data_f, test_size = 0.35, random_state = 0)

print(f"train set shape: {train.shape}")
print(f"validation set shape: {cv.shape}")
```

```
train set shape: (909, 25)
validation set shape: (490, 25)
```

[ ]:
```
train['Condition'].value_counts()
```

[ ]:
```
1    847
0     62
Name: Condition, dtype: int64
```

[ ]:
```
cv['Condition'].value_counts()
```

[ ]:
```
1    453
0     37
Name: Condition, dtype: int64
```

[ ]:
```
test_data_f = pickle.load(open('/content/gdrive/MyDrive/cs2/data/test_data_f.
↪pkl', 'rb'))
test_data_f.head(3)
```

[ ]:
```
        Image_path Insurance_company  …  hig_expire  cost_grt_20k
0  img_4538519.jpg                 B  …           0             0
1  img_7766002.jpg                 C  …           1             0
2  img_4637390.jpg                AC  …           0             0

[3 rows x 23 columns]
```

[ ]:
```
train_datagen = ImageDataGenerator(rescale=1./255.,
                                   rotation_range=30,
                                   width_shift_range=(-25,25),
                                   height_shift_range=(-25,25),
                                   shear_range=0.5,
                                   zoom_range=(0.1,0.5),
                                   horizontal_flip=True,
                                   vertical_flip=True,
                                   fill_mode='nearest')

#cv_datagen = ImageDataGenerator(rescale = 1./255.)
test_datagen = ImageDataGenerator(rescale = 1./255.)
```

[ ]:
```
def to_str(data_frame):

  '''takes dataframe,
           return string of conditions'''

  data_frame = data_frame.astype({'Condition' : str})
  return data_frame

train_dat = to_str(train)
cv_dat  = to_str(cv)
```

[ ]:
```
print(train_dat.shape)
print(cv_dat.shape)
```

```
(909, 25)
(490, 25)
```

[ ]:
```
img_train_gen = train_datagen.flow_from_dataframe(
    dataframe = train_dat,
    directory = '/content/gdrive/MyDrive/cs2/data/trainImages/',
    x_col = 'Image_path',
    y_col = 'Condition',
    target_size = (224,224),
    batch_size = 16, #32, #64,
    class_mode = 'binary',
    #subset = 'training',
    shuffle = True)
```

```
img_cv_gen = train_datagen.flow_from_dataframe(
    dataframe = cv_dat,
    directory = '/content/gdrive/MyDrive/cs2/data/trainImages/',
    x_col = 'Image_path',
    y_col = 'Condition',
    target_size = (224,224),
    batch_size = 16, #32, #64,
    class_mode = 'binary',
    #subset = 'validation',
    shuffle = True)

img_test_gen = test_datagen.flow_from_dataframe(
    dataframe = test_data_f,
    directory = '/content/gdrive/MyDrive/cs2/data/testImages/',
    x_col = 'Image_path',
    y_col = None,
    target_size = (224,224),
    batch_size = 16, #32, #64,
    class_mode = None,
    shuffle = True)
```

```
Found 909 validated image filenames belonging to 2 classes.
Found 490 validated image filenames belonging to 2 classes.
Found 600 validated image filenames.
```

#VGG19

##simple

[ ]:
```
base_vgg19 = tf.keras.applications.VGG19(weights='imagenet',include_top=False,␣
↪input_shape=(224,224,3))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80142336/80134624 [==============================] - 1s 0us/step
80150528/80134624 [==============================] - 1s 0us/step
```

[ ]:
```
for layer in base_vgg19.layers[:22]:      ##for sake of checking
  layer.trainable = False
for i,layer in enumerate(base_vgg19.layers):
  print(i, layer.name, layer.trainable)
```

```
0 input_1 False
1 block1_conv1 False
2 block1_conv2 False
3 block1_pool False
4 block2_conv1 False
5 block2_conv2 False
```

```
6 block2_pool False
7 block3_conv1 False
8 block3_conv2 False
9 block3_conv3 False
10 block3_conv4 False
11 block3_pool False
12 block4_conv1 False
13 block4_conv2 False
14 block4_conv3 False
15 block4_conv4 False
16 block4_pool False
17 block5_conv1 False
18 block5_conv2 False
19 block5_conv3 False
20 block5_conv4 False
21 block5_pool False
```

[ ]:
```
##model
def final_model(mod):
  x1 = mod.output
  flat = Flatten()(x1)
  fc1 = Dense(216, activation='relu', kernel_initializer=tf.keras.initializers.
 ↪glorot_normal(seed=0))(flat)
  drop1 = Dropout(0.5)(fc1)
  #fc2 = Dense(512, activation='relu', kernel_initializer=tf.keras.initializers.
 ↪glorot_normal(seed=0))(drop1)
  fc3 = Dense(128, activation='relu', kernel_initializer=tf.keras.initializers.
 ↪glorot_normal(seed=0))(drop1)
  drop2 = Dropout(0.5)(fc3)
  fc4 = Dense(64, activation='relu', kernel_initializer=tf.keras.initializers.
 ↪glorot_normal(seed=0))(drop2)
  out = Dense(1, activation='sigmoid')(fc4)

  mdl_1 = Model(inputs=[mod.input], outputs=[out])
  mdl_1_transfer_lrng = Model(inputs=[mod.input], outputs=[fc4])
  mdl_1.summary()
  return mdl_1, mdl_1_transfer_lrng
```

[ ]:
```
model_1, model_1_transfer_lrng = final_model(base_vgg19)
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
```

```
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv4 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv4 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 216)               5419224
_________________________________________________________________
dropout (Dropout)            (None, 216)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               27776
```

```
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
=================================================================
Total params: 25,479,705
Trainable params: 5,455,321
Non-trainable params: 20,024,384
_________________________________________________________________
```

[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model1/img_only/model1_img_only.
 ↪hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=2)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_1.compile(loss='binary_crossentropy', optimizer = optimizer, metrics=[tf.
 ↪keras.metrics.BinaryAccuracy()])

hstry_1 = model_1.fit(img_train_gen,epochs=10, validation_data=img_cv_gen,␣
 ↪batch_size=8,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 330s 5s/step - loss: 0.3515 -
binary_accuracy: 0.9230 - val_loss: 0.2761 - val_binary_accuracy: 0.9245

Epoch 00001: val_binary_accuracy improved from -inf to 0.92449, saving model to
/content/gdrive/MyDrive/cs2/data/model1/img_only/model1_img_only.hdf5
Epoch 2/10
57/57 [==============================] - 32s 560ms/step - loss: 0.3089 -
binary_accuracy: 0.9252 - val_loss: 0.3120 - val_binary_accuracy: 0.9245

Epoch 00002: val_binary_accuracy did not improve from 0.92449
Epoch 3/10
57/57 [==============================] - 31s 542ms/step - loss: 0.2760 -
binary_accuracy: 0.9318 - val_loss: 0.2604 - val_binary_accuracy: 0.9245

Epoch 00003: val_binary_accuracy did not improve from 0.92449
Epoch 4/10
57/57 [==============================] - 29s 503ms/step - loss: 0.2730 -
```

```
binary_accuracy: 0.9307 - val_loss: 0.2695 - val_binary_accuracy: 0.9245

Epoch 00004: val_binary_accuracy did not improve from 0.92449
Epoch 5/10
57/57 [==============================] - 29s 502ms/step - loss: 0.2831 -
binary_accuracy: 0.9318 - val_loss: 0.2908 - val_binary_accuracy: 0.9245

Epoch 00005: val_binary_accuracy did not improve from 0.92449
Epoch 6/10
57/57 [==============================] - 29s 505ms/step - loss: 0.2779 -
binary_accuracy: 0.9318 - val_loss: 0.2901 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy did not improve from 0.92449
Epoch 7/10
57/57 [==============================] - 29s 507ms/step - loss: 0.2616 -
binary_accuracy: 0.9318 - val_loss: 0.2886 - val_binary_accuracy: 0.9245

Epoch 00007: val_binary_accuracy did not improve from 0.92449
Epoch 8/10
57/57 [==============================] - 29s 508ms/step - loss: 0.2740 -
binary_accuracy: 0.9307 - val_loss: 0.2874 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 29s 508ms/step - loss: 0.2546 -
binary_accuracy: 0.9318 - val_loss: 0.2865 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 29s 507ms/step - loss: 0.2559 -
binary_accuracy: 0.9307 - val_loss: 0.2907 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```
score = model_1.evaluate(img_cv_gen, verbose=1, batch_size=64)
```

```
31/31 [==============================] - 10s 318ms/step - loss: 0.2852 -
binary_accuracy: 0.9245
```

[ ]:
```
from sklearn.metrics import hamming_loss, recall_score, precision_score,␣
↪f1_score

def metrics(y_true, y_pred,):
    print("Recall micro      : ", recall_score(y_true, y_pred, average='micro'))
  print("Precision micro   : ", precision_score(y_true, y_pred,␣
↪average='micro'))
  print("Fl score micro    : ", f1_score(y_true, y_pred, average='micro'))
```

[ ]:
```
y_pred_0 = model_1.predict(img_cv_gen, batch_size=64)
preds_0 = (model_1.predict(img_cv_gen)>0.5).astype("int32")
metrics(cv['Condition'], preds_0)
```

```
Recall micro      :  0.9244897959183673
Precision micro   :  0.9244897959183673
Fl score micro    :  0.9244897959183674
```

OBSERVATION

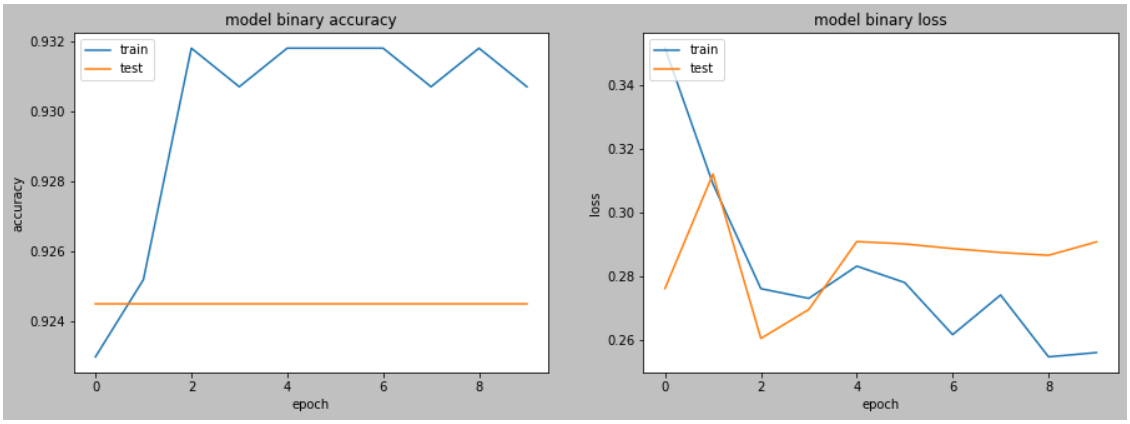
1. having only that class which is damage, as it is only we have predict the details about.

[ ]:
```
def test_pred_cond(model, generator, testframe):

  '''takes model : name of model or number of model,
       generator : generator object for train/cv/test,
       testframe : frame on which we have to predict condition'''

  pred = (model.predict(generator)>0.5).astype('int32')
  test_condition_model = testframe.copy()
  test_condition_model['Condition'] = pd.DataFrame(pred)

  return test_condition_model
```

[ ]:
```
test_condition_model_1 = test_pred_cond(model_1, img_test_gen, test_data_f)
test_condition_model_1.to_csv('/content/gdrive/MyDrive/cs2/data/model1/img_only/
↪test_condition_model_1.csv')
```

[ ]:
```
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_1.history['binary_accuracy'])
  plt.plot(hstry_1.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_1.history['loss'])
  plt.plot(hstry_1.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
```

```
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but after epoch 3, validation loss is marginally higher than training loss indicates model starts overfitting, but epoch 3 is the best balance we are looking for.
2. in binary accuracy plot training accuracy is increasing but test accuracy is constant because whaterever model learns it apllies on test set there is no new to learn from test data perspective, due to test set is limited with similar nature of condition (more damage), and repeatative similar images (i.e less variation in image data), this nature of graph occurs.
3. try to overcome above limitation (point 2) with oversampling of inferior class, or using cars dataset, in version 2.0 of this case study.

[ ]:
```
model_1.save('/content/gdrive/MyDrive/cs2/data/model1/model1_deepl.h5')
```

##transfer learning

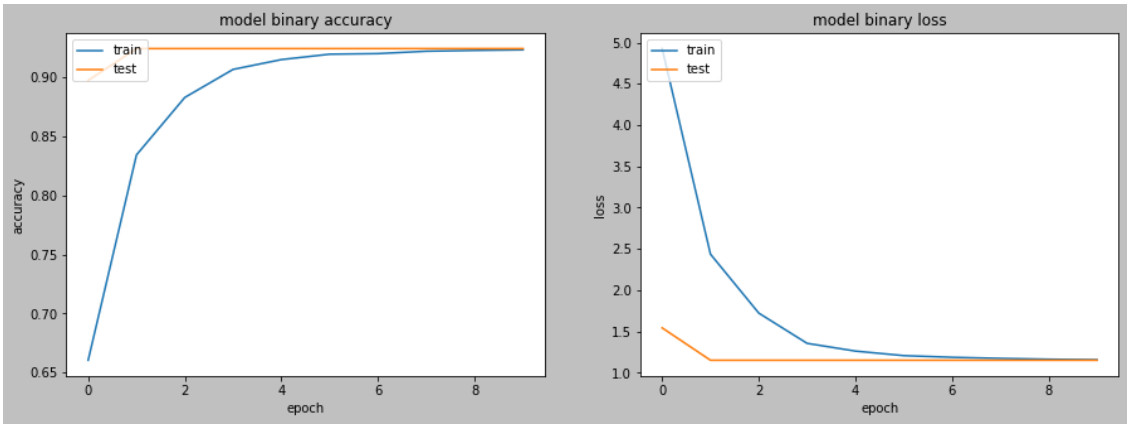
[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model1/transfer_lrng/
 ↪model1_transfer.hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=3,␣
 ↪min_lr=0.000001)
```

```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.
↪999, epsilon=1e-07)
model_1_transfer_lrng.compile(loss='binary_crossentropy', optimizer =␣
↪optimizer, metrics=[tf.keras.metrics.BinaryAccuracy()])

hstry_1_transfer_lrng = model_1_transfer_lrng.fit(img_train_gen,epochs=10,␣
↪validation_data=img_cv_gen, batch_size=64,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 32s 529ms/step - loss: 4.9261 -
binary_accuracy: 0.6604 - val_loss: 1.5434 - val_binary_accuracy: 0.8975

Epoch 00001: val_binary_accuracy improved from -inf to 0.89748, saving model to
/content/gdrive/MyDrive/cs2/data/model1/transfer_lrng/model1_transfer.hdf5
Epoch 2/10
57/57 [==============================] - 30s 532ms/step - loss: 2.4359 -
binary_accuracy: 0.8344 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00002: val_binary_accuracy improved from 0.89748 to 0.92449, saving model
to /content/gdrive/MyDrive/cs2/data/model1/transfer_lrng/model1_transfer.hdf5
Epoch 3/10
57/57 [==============================] - 31s 536ms/step - loss: 1.7221 -
binary_accuracy: 0.8831 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00003: val_binary_accuracy did not improve from 0.92449
Epoch 4/10
57/57 [==============================] - 30s 521ms/step - loss: 1.3560 -
binary_accuracy: 0.9069 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00004: val_binary_accuracy did not improve from 0.92449
Epoch 5/10
57/57 [==============================] - 29s 518ms/step - loss: 1.2622 -
binary_accuracy: 0.9152 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00005: val_binary_accuracy did not improve from 0.92449
Epoch 6/10
57/57 [==============================] - 29s 510ms/step - loss: 1.2073 -
binary_accuracy: 0.9197 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy did not improve from 0.92449
Epoch 7/10
57/57 [==============================] - 29s 509ms/step - loss: 1.1870 -
binary_accuracy: 0.9202 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00007: val_binary_accuracy did not improve from 0.92449
Epoch 8/10
```

```
57/57 [==============================] - 29s 518ms/step - loss: 1.1737 -
binary_accuracy: 0.9223 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 30s 535ms/step - loss: 1.1635 -
binary_accuracy: 0.9230 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 31s 535ms/step - loss: 1.1572 -
binary_accuracy: 0.9234 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```python
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_1_transfer_lrng.history['binary_accuracy'])
  plt.plot(hstry_1_transfer_lrng.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_1_transfer_lrng.history['loss'])
  plt.plot(hstry_1_transfer_lrng.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but after epoch 6, validation loss is more or less equal to training loss indicates model as it is the balance we are looking for.

2. in binary accuracy plot training accuracy is increasing but test accuracy is constant because whaterever model learns it apllies on test set there is no new to learn from test data perspective, due to test set is limited with similar nature of condition (more damage), and repeatative similar images (i.e less variation in image data), this nature of graph occurs.

3. try to overcome above limitation (point 2) with oversampling of inferior class, or using cars dataset, in version 2.0 of this case study.

[ ]:
```
model_1_transfer_lrng.save('/content/gdrive/MyDrive/cs2/data/model1/
↪tranfer_lrng/model1_transfer_learning.h5')
```

[ ]:
```
def get_embedd(model, generator, generator_1, generator_2, batch):

    '''takes input of model : vgg-19/resnet/mobilenet/custom respectively in␣
↪subsequent stages,
      generator, generator_1, generator_2 : image datagen for tain,test,cv
      batch    :  batch size (int)
      returns :  dataframe of train and test having transfer weights'''

    y_pred = model.predict(generator, batch)
    df = pd.DataFrame(y_pred)
    test_condition_model_1_tf = test_data_f.copy()
    test_condition_model_1_tf = pd.concat([test_condition_model_1, df], axis=1,␣
↪join='inner')
    test_condition_model_1_tf.to_csv('/content/gdrive/MyDrive/cs2/data/model1/
↪tranfer_lrng/test_model_1_transfer_learning.csv')
```

```
    y_pred_tr = model.predict(generator_1, batch)
    y_pred_cv = model.predict(generator_2, batch)
    kd0 = pd.concat([train, cv], axis=0)
    kd0.reset_index(inplace=True)
    print(kd0.shape)
    kd1 = pd.concat([pd.DataFrame(y_pred_tr), pd.DataFrame(y_pred_cv)], axis=0)
    kd1.reset_index(inplace=True)
    print(kd1.shape)
    kd_ = pd.concat([kd0,kd1], axis=1)
    kd_.to_csv('/content/gdrive/MyDrive/cs2/data/model1/tranfer_lrng/
 ↪train_model_1_transfer_learning.csv')

    return test_condition_model_1_tf, kd_
```

[ ]:
```
#_,train_transfer_wt = get_embedd(model_1_transfer_lrng, img_train_gen,␣
 ↪img_cv_gen, 16, train=True)
test_transfer_wt,train_transfer_wt = get_embedd(model_1_transfer_lrng,␣
 ↪img_test_gen, img_train_gen, img_cv_gen, 16)#, train=False)
```

```
(1399, 26)
(1399, 65)
```

#RESNET50

0.1 simple

[ ]:
```
base_resnet50 = tf.keras.applications.
 ↪ResNet50(weights='imagenet',include_top=False, input_shape=(224,224,3))
```

[ ]:
```
for layer in base_resnet50.layers[:175]:      ##for sake of checking
  layer.trainable = False
for i,layer in enumerate(base_resnet50.layers):
  print(i, layer.name, layer.trainable)
```

```
0 input_3 False
1 conv1_pad False
2 conv1_conv False
3 conv1_bn False
4 conv1_relu False
5 pool1_pad False
6 pool1_pool False
7 conv2_block1_1_conv False
8 conv2_block1_1_bn False
9 conv2_block1_1_relu False
10 conv2_block1_2_conv False
11 conv2_block1_2_bn False
12 conv2_block1_2_relu False
13 conv2_block1_0_conv False
```

```
14 conv2_block1_3_conv False
15 conv2_block1_0_bn False
16 conv2_block1_3_bn False
17 conv2_block1_add False
18 conv2_block1_out False
19 conv2_block2_1_conv False
20 conv2_block2_1_bn False
21 conv2_block2_1_relu False
22 conv2_block2_2_conv False
23 conv2_block2_2_bn False
24 conv2_block2_2_relu False
25 conv2_block2_3_conv False
26 conv2_block2_3_bn False
27 conv2_block2_add False
28 conv2_block2_out False
29 conv2_block3_1_conv False
30 conv2_block3_1_bn False
31 conv2_block3_1_relu False
32 conv2_block3_2_conv False
33 conv2_block3_2_bn False
34 conv2_block3_2_relu False
35 conv2_block3_3_conv False
36 conv2_block3_3_bn False
37 conv2_block3_add False
38 conv2_block3_out False
39 conv3_block1_1_conv False
40 conv3_block1_1_bn False
41 conv3_block1_1_relu False
42 conv3_block1_2_conv False
43 conv3_block1_2_bn False
44 conv3_block1_2_relu False
45 conv3_block1_0_conv False
46 conv3_block1_3_conv False
47 conv3_block1_0_bn False
48 conv3_block1_3_bn False
49 conv3_block1_add False
50 conv3_block1_out False
51 conv3_block2_1_conv False
52 conv3_block2_1_bn False
53 conv3_block2_1_relu False
54 conv3_block2_2_conv False
55 conv3_block2_2_bn False
56 conv3_block2_2_relu False
57 conv3_block2_3_conv False
58 conv3_block2_3_bn False
59 conv3_block2_add False
60 conv3_block2_out False
61 conv3_block3_1_conv False
```

```
62 conv3_block3_1_bn False
63 conv3_block3_1_relu False
64 conv3_block3_2_conv False
65 conv3_block3_2_bn False
66 conv3_block3_2_relu False
67 conv3_block3_3_conv False
68 conv3_block3_3_bn False
69 conv3_block3_add False
70 conv3_block3_out False
71 conv3_block4_1_conv False
72 conv3_block4_1_bn False
73 conv3_block4_1_relu False
74 conv3_block4_2_conv False
75 conv3_block4_2_bn False
76 conv3_block4_2_relu False
77 conv3_block4_3_conv False
78 conv3_block4_3_bn False
79 conv3_block4_add False
80 conv3_block4_out False
81 conv4_block1_1_conv False
82 conv4_block1_1_bn False
83 conv4_block1_1_relu False
84 conv4_block1_2_conv False
85 conv4_block1_2_bn False
86 conv4_block1_2_relu False
87 conv4_block1_0_conv False
88 conv4_block1_3_conv False
89 conv4_block1_0_bn False
90 conv4_block1_3_bn False
91 conv4_block1_add False
92 conv4_block1_out False
93 conv4_block2_1_conv False
94 conv4_block2_1_bn False
95 conv4_block2_1_relu False
96 conv4_block2_2_conv False
97 conv4_block2_2_bn False
98 conv4_block2_2_relu False
99 conv4_block2_3_conv False
100 conv4_block2_3_bn False
101 conv4_block2_add False
102 conv4_block2_out False
103 conv4_block3_1_conv False
104 conv4_block3_1_bn False
105 conv4_block3_1_relu False
106 conv4_block3_2_conv False
107 conv4_block3_2_bn False
108 conv4_block3_2_relu False
109 conv4_block3_3_conv False
```

```
110 conv4_block3_3_bn False
111 conv4_block3_add False
112 conv4_block3_out False
113 conv4_block4_1_conv False
114 conv4_block4_1_bn False
115 conv4_block4_1_relu False
116 conv4_block4_2_conv False
117 conv4_block4_2_bn False
118 conv4_block4_2_relu False
119 conv4_block4_3_conv False
120 conv4_block4_3_bn False
121 conv4_block4_add False
122 conv4_block4_out False
123 conv4_block5_1_conv False
124 conv4_block5_1_bn False
125 conv4_block5_1_relu False
126 conv4_block5_2_conv False
127 conv4_block5_2_bn False
128 conv4_block5_2_relu False
129 conv4_block5_3_conv False
130 conv4_block5_3_bn False
131 conv4_block5_add False
132 conv4_block5_out False
133 conv4_block6_1_conv False
134 conv4_block6_1_bn False
135 conv4_block6_1_relu False
136 conv4_block6_2_conv False
137 conv4_block6_2_bn False
138 conv4_block6_2_relu False
139 conv4_block6_3_conv False
140 conv4_block6_3_bn False
141 conv4_block6_add False
142 conv4_block6_out False
143 conv5_block1_1_conv False
144 conv5_block1_1_bn False
145 conv5_block1_1_relu False
146 conv5_block1_2_conv False
147 conv5_block1_2_bn False
148 conv5_block1_2_relu False
149 conv5_block1_0_conv False
150 conv5_block1_3_conv False
151 conv5_block1_0_bn False
152 conv5_block1_3_bn False
153 conv5_block1_add False
154 conv5_block1_out False
155 conv5_block2_1_conv False
156 conv5_block2_1_bn False
157 conv5_block2_1_relu False
```

```
158 conv5_block2_2_conv False
159 conv5_block2_2_bn False
160 conv5_block2_2_relu False
161 conv5_block2_3_conv False
162 conv5_block2_3_bn False
163 conv5_block2_add False
164 conv5_block2_out False
165 conv5_block3_1_conv False
166 conv5_block3_1_bn False
167 conv5_block3_1_relu False
168 conv5_block3_2_conv False
169 conv5_block3_2_bn False
170 conv5_block3_2_relu False
171 conv5_block3_3_conv False
172 conv5_block3_3_bn False
173 conv5_block3_add False
174 conv5_block3_out False
```

[ ]:
```
model_2, model_2_transfer_lrng = final_model(base_resnet50)
```

```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 224, 224, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 230, 230, 3)  0           input_3[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 112, 112, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 112, 112, 64) 256
conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 112, 112, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 114, 114, 64) 0
conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 56, 56, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
```

```
------------------
conv2_block1_1_conv (Conv2D)    (None, 56, 56, 64)   4160
pool1_pool[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_1_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block1_1_conv[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_1_relu (Activation (None, 56, 56, 64)   0
conv2_block1_1_bn[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block1_1_relu[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block1_2_conv[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_2_relu (Activation (None, 56, 56, 64)   0
conv2_block1_2_bn[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_0_conv (Conv2D)    (None, 56, 56, 256)  16640
pool1_pool[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block1_2_relu[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_0_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block1_0_conv[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block1_3_conv[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_add (Add)          (None, 56, 56, 256)  0
conv2_block1_0_bn[0][0]
conv2_block1_3_bn[0][0]
------------------------------------------------------------------------------
------------------
conv2_block1_out (Activation)   (None, 56, 56, 256)  0
conv2_block1_add[0][0]
```

```
__________________________________________________________________________________
__________________
conv2_block2_1_conv (Conv2D)    (None, 56, 56, 64)   16448
conv2_block1_out[0][0]
__________________________________________________________________________________
__________________
conv2_block2_1_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block2_1_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block2_1_relu (Activation (None, 56, 56, 64)   0
conv2_block2_1_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block2_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block2_1_relu[0][0]
__________________________________________________________________________________
__________________
conv2_block2_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block2_2_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block2_2_relu (Activation (None, 56, 56, 64)   0
conv2_block2_2_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block2_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block2_2_relu[0][0]
__________________________________________________________________________________
__________________
conv2_block2_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block2_3_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block2_add (Add)          (None, 56, 56, 256)  0
conv2_block1_out[0][0]
conv2_block2_3_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block2_out (Activation)   (None, 56, 56, 256)  0
conv2_block2_add[0][0]
__________________________________________________________________________________
__________________
conv2_block3_1_conv (Conv2D)    (None, 56, 56, 64)   16448
conv2_block2_out[0][0]
__________________________________________________________________________________
__________________
conv2_block3_1_bn (BatchNormali (None, 56, 56, 64)   256
```

```
conv2_block3_1_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block3_1_relu (Activation (None, 56, 56, 64)   0
conv2_block3_1_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block3_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block3_1_relu[0][0]
__________________________________________________________________________________
__________________
conv2_block3_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block3_2_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block3_2_relu (Activation (None, 56, 56, 64)   0
conv2_block3_2_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block3_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block3_2_relu[0][0]
__________________________________________________________________________________
__________________
conv2_block3_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block3_3_conv[0][0]
__________________________________________________________________________________
__________________
conv2_block3_add (Add)          (None, 56, 56, 256)  0
conv2_block2_out[0][0]
conv2_block3_3_bn[0][0]
__________________________________________________________________________________
__________________
conv2_block3_out (Activation)   (None, 56, 56, 256)  0
conv2_block3_add[0][0]
__________________________________________________________________________________
__________________
conv3_block1_1_conv (Conv2D)    (None, 28, 28, 128)  32896
conv2_block3_out[0][0]
__________________________________________________________________________________
__________________
conv3_block1_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block1_1_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block1_1_relu (Activation (None, 28, 28, 128)  0
conv3_block1_1_bn[0][0]
__________________________________________________________________________________
__________________
```

```
conv3_block1_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block1_1_relu[0][0]
__________________________________________________________________________________
__________________
conv3_block1_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block1_2_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block1_2_relu (Activation (None, 28, 28, 128)  0
conv3_block1_2_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block1_0_conv (Conv2D)    (None, 28, 28, 512)  131584
conv2_block3_out[0][0]
__________________________________________________________________________________
__________________
conv3_block1_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block1_2_relu[0][0]
__________________________________________________________________________________
__________________
conv3_block1_0_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block1_0_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block1_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block1_3_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block1_add (Add)          (None, 28, 28, 512)  0
conv3_block1_0_bn[0][0]
conv3_block1_3_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block1_out (Activation)   (None, 28, 28, 512)  0
conv3_block1_add[0][0]
__________________________________________________________________________________
__________________
conv3_block2_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block1_out[0][0]
__________________________________________________________________________________
__________________
conv3_block2_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block2_1_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block2_1_relu (Activation (None, 28, 28, 128)  0
conv3_block2_1_bn[0][0]
__________________________________________________________________________________
```

```
------------------
conv3_block2_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block2_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block2_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_2_relu (Activation (None, 28, 28, 128)  0
conv3_block2_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block2_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block2_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_add (Add)          (None, 28, 28, 512)  0
conv3_block1_out[0][0]
conv3_block2_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_out (Activation)   (None, 28, 28, 512)  0
conv3_block2_add[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block2_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block3_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_relu (Activation (None, 28, 28, 128)  0
conv3_block3_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block3_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block3_2_conv[0][0]
```

```
__________________________________________________________________________________
__________________
conv3_block3_2_relu (Activation (None, 28, 28, 128)  0
conv3_block3_2_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block3_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block3_2_relu[0][0]
__________________________________________________________________________________
__________________
conv3_block3_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block3_3_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block3_add (Add)          (None, 28, 28, 512)  0
conv3_block2_out[0][0]
conv3_block3_3_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block3_out (Activation)   (None, 28, 28, 512)  0
conv3_block3_add[0][0]
__________________________________________________________________________________
__________________
conv3_block4_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block3_out[0][0]
__________________________________________________________________________________
__________________
conv3_block4_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block4_1_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block4_1_relu (Activation (None, 28, 28, 128)  0
conv3_block4_1_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block4_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block4_1_relu[0][0]
__________________________________________________________________________________
__________________
conv3_block4_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block4_2_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block4_2_relu (Activation (None, 28, 28, 128)  0
conv3_block4_2_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block4_3_conv (Conv2D)    (None, 28, 28, 512)  66048
```

```
conv3_block4_2_relu[0][0]
__________________________________________________________________________________
__________________
conv3_block4_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block4_3_conv[0][0]
__________________________________________________________________________________
__________________
conv3_block4_add (Add)          (None, 28, 28, 512)  0
conv3_block3_out[0][0]
conv3_block4_3_bn[0][0]
__________________________________________________________________________________
__________________
conv3_block4_out (Activation)   (None, 28, 28, 512)  0
conv3_block4_add[0][0]
__________________________________________________________________________________
__________________
conv4_block1_1_conv (Conv2D)    (None, 14, 14, 256)  131328
conv3_block4_out[0][0]
__________________________________________________________________________________
__________________
conv4_block1_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block1_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block1_1_relu (Activation (None, 14, 14, 256)  0
conv4_block1_1_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block1_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block1_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block1_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block1_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block1_2_relu (Activation (None, 14, 14, 256)  0
conv4_block1_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block1_0_conv (Conv2D)    (None, 14, 14, 1024) 525312
conv3_block4_out[0][0]
__________________________________________________________________________________
__________________
conv4_block1_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block1_2_relu[0][0]
__________________________________________________________________________________
__________________
```

```
conv4_block1_0_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block1_0_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block1_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block1_3_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block1_add (Add)          (None, 14, 14, 1024) 0
conv4_block1_0_bn[0][0]
conv4_block1_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block1_out (Activation)   (None, 14, 14, 1024) 0
conv4_block1_add[0][0]
__________________________________________________________________________________
__________________
conv4_block2_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block1_out[0][0]
__________________________________________________________________________________
__________________
conv4_block2_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block2_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block2_1_relu (Activation (None, 14, 14, 256)  0
conv4_block2_1_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block2_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block2_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block2_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block2_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block2_2_relu (Activation (None, 14, 14, 256)  0
conv4_block2_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block2_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block2_2_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block2_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block2_3_conv[0][0]
__________________________________________________________________________________
```

```
__________________
conv4_block2_add (Add)          (None, 14, 14, 1024) 0
conv4_block1_out[0][0]
conv4_block2_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block2_out (Activation)   (None, 14, 14, 1024) 0
conv4_block2_add[0][0]
__________________________________________________________________________________
__________________
conv4_block3_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block2_out[0][0]
__________________________________________________________________________________
__________________
conv4_block3_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block3_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block3_1_relu (Activation (None, 14, 14, 256)  0
conv4_block3_1_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block3_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block3_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block3_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block3_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block3_2_relu (Activation (None, 14, 14, 256)  0
conv4_block3_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block3_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block3_2_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block3_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block3_3_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block3_add (Add)          (None, 14, 14, 1024) 0
conv4_block2_out[0][0]
conv4_block3_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block3_out (Activation)   (None, 14, 14, 1024) 0
```

```
conv4_block3_add[0][0]
__________________________________________________________________________________
__________________
conv4_block4_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block3_out[0][0]
__________________________________________________________________________________
__________________
conv4_block4_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block4_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block4_1_relu (Activation (None, 14, 14, 256)  0
conv4_block4_1_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block4_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block4_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block4_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block4_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block4_2_relu (Activation (None, 14, 14, 256)  0
conv4_block4_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block4_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block4_2_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block4_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block4_3_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block4_add (Add)          (None, 14, 14, 1024) 0
conv4_block3_out[0][0]
conv4_block4_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block4_out (Activation)   (None, 14, 14, 1024) 0
conv4_block4_add[0][0]
__________________________________________________________________________________
__________________
conv4_block5_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block4_out[0][0]
__________________________________________________________________________________
__________________
```

```
conv4_block5_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block5_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block5_1_relu (Activation (None, 14, 14, 256)  0
conv4_block5_1_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block5_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block5_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block5_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block5_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block5_2_relu (Activation (None, 14, 14, 256)  0
conv4_block5_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block5_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block5_2_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block5_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block5_3_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block5_add (Add)          (None, 14, 14, 1024) 0
conv4_block4_out[0][0]
conv4_block5_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block5_out (Activation)   (None, 14, 14, 1024) 0
conv4_block5_add[0][0]
__________________________________________________________________________________
__________________
conv4_block6_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block5_out[0][0]
__________________________________________________________________________________
__________________
conv4_block6_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block6_1_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block6_1_relu (Activation (None, 14, 14, 256)  0
conv4_block6_1_bn[0][0]
__________________________________________________________________________________
```

```
__________________
conv4_block6_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block6_1_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block6_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block6_2_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block6_2_relu (Activation (None, 14, 14, 256)  0
conv4_block6_2_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block6_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block6_2_relu[0][0]
__________________________________________________________________________________
__________________
conv4_block6_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block6_3_conv[0][0]
__________________________________________________________________________________
__________________
conv4_block6_add (Add)          (None, 14, 14, 1024) 0
conv4_block5_out[0][0]
conv4_block6_3_bn[0][0]
__________________________________________________________________________________
__________________
conv4_block6_out (Activation)   (None, 14, 14, 1024) 0
conv4_block6_add[0][0]
__________________________________________________________________________________
__________________
conv5_block1_1_conv (Conv2D)    (None, 7, 7, 512)    524800
conv4_block6_out[0][0]
__________________________________________________________________________________
__________________
conv5_block1_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block1_1_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block1_1_relu (Activation (None, 7, 7, 512)    0
conv5_block1_1_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block1_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block1_1_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block1_2_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block1_2_conv[0][0]
```

```
__________________________________________________________________________________
__________________
conv5_block1_2_relu (Activation (None, 7, 7, 512)    0
conv5_block1_2_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block1_0_conv (Conv2D)    (None, 7, 7, 2048)   2099200
conv4_block6_out[0][0]
__________________________________________________________________________________
__________________
conv5_block1_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block1_2_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block1_0_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block1_0_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block1_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block1_3_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block1_add (Add)          (None, 7, 7, 2048)   0
conv5_block1_0_bn[0][0]
conv5_block1_3_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block1_out (Activation)   (None, 7, 7, 2048)   0
conv5_block1_add[0][0]
__________________________________________________________________________________
__________________
conv5_block2_1_conv (Conv2D)    (None, 7, 7, 512)    1049088
conv5_block1_out[0][0]
__________________________________________________________________________________
__________________
conv5_block2_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block2_1_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block2_1_relu (Activation (None, 7, 7, 512)    0
conv5_block2_1_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block2_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block2_1_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block2_2_bn (BatchNormali (None, 7, 7, 512)    2048
```

```
conv5_block2_2_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block2_2_relu (Activation (None, 7, 7, 512)    0
conv5_block2_2_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block2_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block2_2_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block2_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block2_3_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block2_add (Add)          (None, 7, 7, 2048)   0
conv5_block1_out[0][0]
conv5_block2_3_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block2_out (Activation)   (None, 7, 7, 2048)   0
conv5_block2_add[0][0]
__________________________________________________________________________________
__________________
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 512)    1049088
conv5_block2_out[0][0]
__________________________________________________________________________________
__________________
conv5_block3_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block3_1_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block3_1_relu (Activation (None, 7, 7, 512)    0
conv5_block3_1_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block3_1_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block3_2_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block3_2_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block3_2_relu (Activation (None, 7, 7, 512)    0
conv5_block3_2_bn[0][0]
__________________________________________________________________________________
__________________
```

```
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block3_2_relu[0][0]
__________________________________________________________________________________
__________________
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block3_3_conv[0][0]
__________________________________________________________________________________
__________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0
conv5_block2_out[0][0]
conv5_block3_3_bn[0][0]
__________________________________________________________________________________
__________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0
conv5_block3_add[0][0]
__________________________________________________________________________________
__________________
flatten_1 (Flatten)             (None, 100352)       0
conv5_block3_out[0][0]
__________________________________________________________________________________
__________________
dense_4 (Dense)                 (None, 216)          21676248    flatten_1[0][0]
__________________________________________________________________________________
__________________
dropout_2 (Dropout)             (None, 216)          0           dense_4[0][0]
__________________________________________________________________________________
__________________
dense_5 (Dense)                 (None, 128)          27776       dropout_2[0][0]
__________________________________________________________________________________
__________________
dropout_3 (Dropout)             (None, 128)          0           dense_5[0][0]
__________________________________________________________________________________
__________________
dense_6 (Dense)                 (None, 64)           8256        dropout_3[0][0]
__________________________________________________________________________________
__________________
dense_7 (Dense)                 (None, 1)            65          dense_6[0][0]
==================================================================================
==================
Total params: 45,300,057
Trainable params: 21,712,345
Non-trainable params: 23,587,712
__________________________________________________________________________________
__________________
```

[ ]:
```
from tensorflow.keras.callbacks import *
```

```
filepath = '/content/gdrive/MyDrive/cs2/data/model2/img_only/model2_img_only.
 ↪hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=2)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_2.compile(loss='binary_crossentropy', optimizer = optimizer, metrics=[tf.
 ↪keras.metrics.BinaryAccuracy()])

hstry_2 = model_2.fit(img_train_gen,epochs=10, validation_data=img_cv_gen,␣
 ↪batch_size=8,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 289s 4s/step - loss: 0.5679 -
binary_accuracy: 0.8295 - val_loss: 0.3984 - val_binary_accuracy: 0.9245

Epoch 00001: val_binary_accuracy improved from -inf to 0.92449, saving model to
/content/gdrive/MyDrive/cs2/data/model2/img_only/model2_img_only.hdf5
Epoch 2/10
57/57 [==============================] - 30s 533ms/step - loss: 0.3878 -
binary_accuracy: 0.8746 - val_loss: 0.2665 - val_binary_accuracy: 0.9245

Epoch 00002: val_binary_accuracy did not improve from 0.92449
Epoch 3/10
57/57 [==============================] - 28s 496ms/step - loss: 0.3586 -
binary_accuracy: 0.8922 - val_loss: 0.2716 - val_binary_accuracy: 0.9245

Epoch 00003: val_binary_accuracy did not improve from 0.92449
Epoch 4/10
57/57 [==============================] - 28s 500ms/step - loss: 0.3143 -
binary_accuracy: 0.9131 - val_loss: 0.3165 - val_binary_accuracy: 0.9245

Epoch 00004: val_binary_accuracy did not improve from 0.92449
Epoch 5/10
57/57 [==============================] - 28s 500ms/step - loss: 0.3329 -
binary_accuracy: 0.9175 - val_loss: 0.2775 - val_binary_accuracy: 0.9245

Epoch 00005: val_binary_accuracy did not improve from 0.92449
Epoch 6/10
57/57 [==============================] - 29s 511ms/step - loss: 0.2867 -
binary_accuracy: 0.9219 - val_loss: 0.2937 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy did not improve from 0.92449
Epoch 7/10
```

```
57/57 [==============================] - 29s 505ms/step - loss: 0.3274 -
binary_accuracy: 0.9230 - val_loss: 0.2768 - val_binary_accuracy: 0.9245

Epoch 00007: val_binary_accuracy did not improve from 0.92449
Epoch 8/10
57/57 [==============================] - 29s 517ms/step - loss: 0.3098 -
binary_accuracy: 0.9252 - val_loss: 0.2745 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 29s 504ms/step - loss: 0.3150 -
binary_accuracy: 0.9252 - val_loss: 0.2775 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 29s 505ms/step - loss: 0.2948 -
binary_accuracy: 0.9307 - val_loss: 0.2752 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```
score = model_2.evaluate(img_cv_gen, verbose=1, batch_size=64)
```

```
31/31 [==============================] - 10s 315ms/step - loss: 0.2709 -
binary_accuracy: 0.9245
```

[ ]:
```
y_pred_0 = model_2.predict(img_cv_gen, batch_size=64)
preds_0 = (model_2.predict(img_cv_gen)>0.5).astype("int32")
metrics(cv['Condition'], preds_0)
```

```
Recall micro      :  0.9244897959183673
Precision micro   :  0.9244897959183673
Fl score micro    :  0.9244897959183674
```

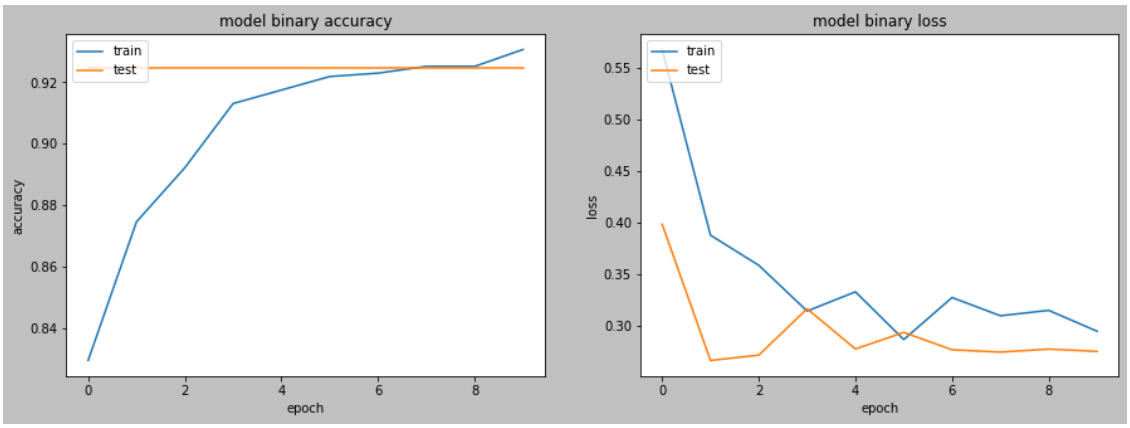
OBSERVATION

1. having only that class which is damage, as it is only we have predict the details about.

[ ]:
```
test_condition_model_2 = test_pred_cond(model_2, img_test_gen, test_data_f)
test_condition_model_2.to_csv('/content/gdrive/MyDrive/cs2/data/model2/img_only/
↪test_condition_model_2.csv')
```

[ ]:
```
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_2.history['binary_accuracy'])
  plt.plot(hstry_2.history['val_binary_accuracy'])
```

```
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_2.history['loss'])
  plt.plot(hstry_2.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but at epoch 3,5, validation loss (here test loss) is more or less equal to training loss indicates model as it is the balance we are looking for.
2. in binary accuracy plot training accuracy is increasing but test accuracy is constant because whaterever model learns it apllies on test set there is no new to learn from test data perspective, due to test set is limited with similar nature of condition (more damage), and repeatative similar images (i.e less variation in image data), this nature of graph occurs.
3. try to overcome above limitation (point 2) with oversampling of inferior class, or using cars dataset, in version 2.0 of this case study.

[ ]:
```
model_2.save('/content/gdrive/MyDrive/cs2/data/model2/model2_deepl.h5')
```

##transfer learning

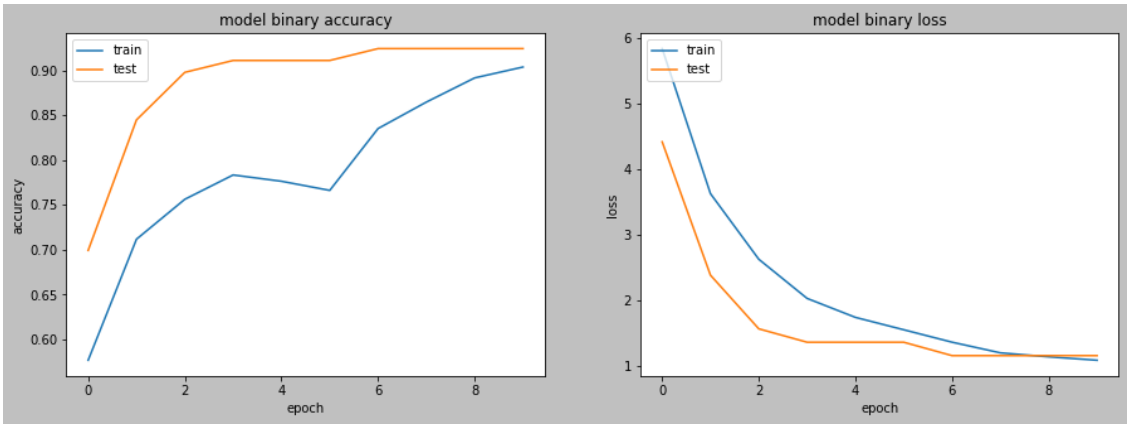
[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/
 ↪model2_transfer.hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=3,␣
 ↪min_lr=0.000001)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_2_transfer_lrng.compile(loss='binary_crossentropy', optimizer =␣
 ↪optimizer, metrics=[tf.keras.metrics.BinaryAccuracy()])

hstry_2_transfer_lrng = model_2_transfer_lrng.fit(img_train_gen,epochs=10,␣
 ↪validation_data=img_cv_gen, batch_size=16,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 35s 548ms/step - loss: 5.8424 -
binary_accuracy: 0.5766 - val_loss: 4.4181 - val_binary_accuracy: 0.6992

Epoch 00001: val_binary_accuracy improved from -inf to 0.69920, saving model to
/content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/model2_transfer.hdf5
Epoch 2/10
57/57 [==============================] - 31s 536ms/step - loss: 3.6260 -
binary_accuracy: 0.7117 - val_loss: 2.3804 - val_binary_accuracy: 0.8449

Epoch 00002: val_binary_accuracy improved from 0.69920 to 0.84490, saving model
to /content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/model2_transfer.hdf5
Epoch 3/10
57/57 [==============================] - 31s 536ms/step - loss: 2.6238 -
binary_accuracy: 0.7563 - val_loss: 1.5611 - val_binary_accuracy: 0.8980

Epoch 00003: val_binary_accuracy improved from 0.84490 to 0.89796, saving model
to /content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/model2_transfer.hdf5
Epoch 4/10
57/57 [==============================] - 30s 535ms/step - loss: 2.0257 -
binary_accuracy: 0.7834 - val_loss: 1.3563 - val_binary_accuracy: 0.9112

Epoch 00004: val_binary_accuracy improved from 0.89796 to 0.91122, saving model
to /content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/model2_transfer.hdf5
Epoch 5/10
57/57 [==============================] - 30s 535ms/step - loss: 1.7360 -
binary_accuracy: 0.7764 - val_loss: 1.3563 - val_binary_accuracy: 0.9112
```

```
Epoch 00005: val_binary_accuracy did not improve from 0.91122
Epoch 6/10
57/57 [==============================] - 29s 503ms/step - loss: 1.5467 -
binary_accuracy: 0.7662 - val_loss: 1.3563 - val_binary_accuracy: 0.9112

Epoch 00006: val_binary_accuracy did not improve from 0.91122
Epoch 7/10
57/57 [==============================] - 29s 505ms/step - loss: 1.3569 -
binary_accuracy: 0.8352 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00007: val_binary_accuracy improved from 0.91122 to 0.92449, saving model
to /content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/model2_transfer.hdf5
Epoch 8/10
57/57 [==============================] - 30s 533ms/step - loss: 1.1941 -
binary_accuracy: 0.8648 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 29s 508ms/step - loss: 1.1333 -
binary_accuracy: 0.8918 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 29s 508ms/step - loss: 1.0812 -
binary_accuracy: 0.9039 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```python
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_2_transfer_lrng.history['binary_accuracy'])
  plt.plot(hstry_2_transfer_lrng.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_2_transfer_lrng.history['loss'])
  plt.plot(hstry_2_transfer_lrng.history['val_loss'])

  plt.title('model binary loss')
```

```
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but after epoch 7, validation loss is slightly more than training loss indicates model will overfit after this, at epoch 8 as it is the balance we are looking for.
2. in binary accuracy plot training accuracy is increasing, test accuracy is also increasing whatever model is learning able to apply well on test data, it can be more iterated for more convergence.

[ ]:
```
model_2_transfer_lrng.save('/content/gdrive/MyDrive/cs2/data/model2/
↪transfer_lrng/model2_transfer_learning.h5')
```

[ ]:
```
def get_embedd(model, generator, generator_1, generator_2, batch):

    '''takes input of model : vgg-19/resnet/mobilenet/custom respectively in␣
↪subsequent stages,
      generator, generator_1, generator_2 : image datagen for tain,test,cv
      batch   :  batch size (int)
      returns :  dataframe of train and test having transfer weights'''

    y_pred = model.predict(generator, batch)
    df = pd.DataFrame(y_pred)
    test_condition_model_1_tf = test_data_f.copy()
```

```
    test_condition_model_1_tf = pd.concat([test_condition_model_2, df], axis=1,␣
↪join='inner')
    test_condition_model_1_tf.to_csv('/content/gdrive/MyDrive/cs2/data/model2/
↪transfer_lrng/test_model_2_transfer_learning.csv')

    y_pred_tr = model.predict(generator_1, batch)
    y_pred_cv = model.predict(generator_2, batch)
    kd0 = pd.concat([train, cv], axis=0)
    kd0.reset_index(inplace=True)
    print(kd0.shape)
    kd1 = pd.concat([pd.DataFrame(y_pred_tr), pd.DataFrame(y_pred_cv)], axis=0)
    kd1.reset_index(inplace=True)
    print(kd1.shape)
    kd_ = pd.concat([kd0,kd1], axis=1)
    kd_.to_csv('/content/gdrive/MyDrive/cs2/data/model2/transfer_lrng/
↪train_model_2_transfer_learning.csv')

    return test_condition_model_1_tf, kd_
```

[ ]:
```
#_,train_transfer_wt = get_embedd(model_1_transfer_lrng, img_train_gen,␣
↪img_cv_gen, 16, train=True)
test_transfer_wt, train_transfer_wt = get_embedd(model_2_transfer_lrng,␣
↪img_test_gen, img_train_gen, img_cv_gen, 16)#, train=False)
```

```
(1399, 26)
(1399, 65)
```

1 Mobile Net

[ ]:
```
base_mobilenet = tf.keras.applications.
↪MobileNet(weights='imagenet',include_top=False, input_shape=(224,224,3))
```

[ ]:
```
for layer in base_mobilenet.layers[:86]:      ##for sake of checking
  layer.trainable = False
for i,layer in enumerate(base_mobilenet.layers):
  print(i, layer.name, layer.trainable)
```

```
0 input_6 False
1 conv1 False
2 conv1_bn False
3 conv1_relu False
4 conv_dw_1 False
5 conv_dw_1_bn False
6 conv_dw_1_relu False
7 conv_pw_1 False
8 conv_pw_1_bn False
9 conv_pw_1_relu False
```

```
10 conv_pad_2 False
11 conv_dw_2 False
12 conv_dw_2_bn False
13 conv_dw_2_relu False
14 conv_pw_2 False
15 conv_pw_2_bn False
16 conv_pw_2_relu False
17 conv_dw_3 False
18 conv_dw_3_bn False
19 conv_dw_3_relu False
20 conv_pw_3 False
21 conv_pw_3_bn False
22 conv_pw_3_relu False
23 conv_pad_4 False
24 conv_dw_4 False
25 conv_dw_4_bn False
26 conv_dw_4_relu False
27 conv_pw_4 False
28 conv_pw_4_bn False
29 conv_pw_4_relu False
30 conv_dw_5 False
31 conv_dw_5_bn False
32 conv_dw_5_relu False
33 conv_pw_5 False
34 conv_pw_5_bn False
35 conv_pw_5_relu False
36 conv_pad_6 False
37 conv_dw_6 False
38 conv_dw_6_bn False
39 conv_dw_6_relu False
40 conv_pw_6 False
41 conv_pw_6_bn False
42 conv_pw_6_relu False
43 conv_dw_7 False
44 conv_dw_7_bn False
45 conv_dw_7_relu False
46 conv_pw_7 False
47 conv_pw_7_bn False
48 conv_pw_7_relu False
49 conv_dw_8 False
50 conv_dw_8_bn False
51 conv_dw_8_relu False
52 conv_pw_8 False
53 conv_pw_8_bn False
54 conv_pw_8_relu False
55 conv_dw_9 False
56 conv_dw_9_bn False
57 conv_dw_9_relu False
```

```
58 conv_pw_9 False
59 conv_pw_9_bn False
60 conv_pw_9_relu False
61 conv_dw_10 False
62 conv_dw_10_bn False
63 conv_dw_10_relu False
64 conv_pw_10 False
65 conv_pw_10_bn False
66 conv_pw_10_relu False
67 conv_dw_11 False
68 conv_dw_11_bn False
69 conv_dw_11_relu False
70 conv_pw_11 False
71 conv_pw_11_bn False
72 conv_pw_11_relu False
73 conv_pad_12 False
74 conv_dw_12 False
75 conv_dw_12_bn False
76 conv_dw_12_relu False
77 conv_pw_12 False
78 conv_pw_12_bn False
79 conv_pw_12_relu False
80 conv_dw_13 False
81 conv_dw_13_bn False
82 conv_dw_13_relu False
83 conv_pw_13 False
84 conv_pw_13_bn False
85 conv_pw_13_relu False
```

[ ]:
```
model_3, model_3_transfer_lrng = final_model(base_mobilenet)
```

```
Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
conv1 (Conv2D)               (None, 112, 112, 32)      864
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 32)      128
_________________________________________________________________
conv1_relu (ReLU)            (None, 112, 112, 32)      0
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0
```

```
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512
_________________________________________________________________
conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768
_________________________________________________________________
conv_pw_4_bn (BatchNormaliza (None, 28, 28, 256)       1024
_________________________________________________________________
conv_pw_4_relu (ReLU)        (None, 28, 28, 256)       0
_________________________________________________________________
conv_dw_5 (DepthwiseConv2D)  (None, 28, 28, 256)       2304
```

```
_________________________________________________________________
conv_dw_5_bn (BatchNormaliza (None, 28, 28, 256)       1024
_________________________________________________________________
conv_dw_5_relu (ReLU)        (None, 28, 28, 256)       0
_________________________________________________________________
conv_pw_5 (Conv2D)           (None, 28, 28, 256)       65536
_________________________________________________________________
conv_pw_5_bn (BatchNormaliza (None, 28, 28, 256)       1024
_________________________________________________________________
conv_pw_5_relu (ReLU)        (None, 28, 28, 256)       0
_________________________________________________________________
conv_pad_6 (ZeroPadding2D)   (None, 29, 29, 256)       0
_________________________________________________________________
conv_dw_6 (DepthwiseConv2D)  (None, 14, 14, 256)       2304
_________________________________________________________________
conv_dw_6_bn (BatchNormaliza (None, 14, 14, 256)       1024
_________________________________________________________________
conv_dw_6_relu (ReLU)        (None, 14, 14, 256)       0
_________________________________________________________________
conv_pw_6 (Conv2D)           (None, 14, 14, 512)       131072
_________________________________________________________________
conv_pw_6_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_6_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_7 (DepthwiseConv2D)  (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_7_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_7_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_7 (Conv2D)           (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_7_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_7_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_8 (DepthwiseConv2D)  (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_8_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_8_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_8 (Conv2D)           (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_8_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_8_relu (ReLU)        (None, 14, 14, 512)       0
```

```
_________________________________________________________________
conv_dw_9 (DepthwiseConv2D)  (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_9_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_9_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_9 (Conv2D)           (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_9_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_9_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_10 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_10 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_11 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_11 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pad_12 (ZeroPadding2D)  (None, 15, 15, 512)       0
_________________________________________________________________
conv_dw_12 (DepthwiseConv2D) (None, 7, 7, 512)         4608
_________________________________________________________________
conv_dw_12_bn (BatchNormaliz (None, 7, 7, 512)         2048
_________________________________________________________________
conv_dw_12_relu (ReLU)       (None, 7, 7, 512)         0
_________________________________________________________________
conv_pw_12 (Conv2D)          (None, 7, 7, 1024)        524288
_________________________________________________________________
conv_pw_12_bn (BatchNormaliz (None, 7, 7, 1024)        4096
```

```
_________________________________________________________________
conv_pw_12_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_dw_13 (DepthwiseConv2D) (None, 7, 7, 1024)        9216
_________________________________________________________________
conv_dw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 50176)             0
_________________________________________________________________
dense_8 (Dense)              (None, 216)               10838232
_________________________________________________________________
dropout_4 (Dropout)          (None, 216)               0
_________________________________________________________________
dense_9 (Dense)              (None, 128)               27776
_________________________________________________________________
dropout_5 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_10 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 65
=================================================================
Total params: 14,103,193
Trainable params: 10,874,329
Non-trainable params: 3,228,864
_________________________________________________________________
```

[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model3/img_only/model3_img_only.
 ↪hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=2)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_3.compile(loss='binary_crossentropy', optimizer = optimizer, metrics=[tf.
 ↪keras.metrics.BinaryAccuracy()])
```

```
hstry_3 = model_3.fit(img_train_gen,epochs=10, validation_data=img_cv_gen,␣
↪batch_size=8,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 30s 490ms/step - loss: 0.6635 -
binary_accuracy: 0.9131 - val_loss: 0.2824 - val_binary_accuracy: 0.9245

Epoch 00001: val_binary_accuracy improved from -inf to 0.92449, saving model to
/content/gdrive/MyDrive/cs2/data/model3/img_only/model3_img_only.hdf5
Epoch 2/10
57/57 [==============================] - 27s 478ms/step - loss: 0.4529 -
binary_accuracy: 0.8966 - val_loss: 0.2970 - val_binary_accuracy: 0.9245

Epoch 00002: val_binary_accuracy did not improve from 0.92449
Epoch 3/10
57/57 [==============================] - 27s 473ms/step - loss: 0.4053 -
binary_accuracy: 0.9065 - val_loss: 0.3060 - val_binary_accuracy: 0.9245

Epoch 00003: val_binary_accuracy did not improve from 0.92449
Epoch 4/10
57/57 [==============================] - 27s 469ms/step - loss: 0.3247 -
binary_accuracy: 0.9164 - val_loss: 0.2778 - val_binary_accuracy: 0.9245

Epoch 00004: val_binary_accuracy did not improve from 0.92449
Epoch 5/10
57/57 [==============================] - 27s 475ms/step - loss: 0.2926 -
binary_accuracy: 0.9208 - val_loss: 0.2752 - val_binary_accuracy: 0.9224

Epoch 00005: val_binary_accuracy did not improve from 0.92449
Epoch 6/10
57/57 [==============================] - 26s 465ms/step - loss: 0.2913 -
binary_accuracy: 0.9208 - val_loss: 0.2472 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy did not improve from 0.92449
Epoch 7/10
57/57 [==============================] - 26s 465ms/step - loss: 0.2812 -
binary_accuracy: 0.9164 - val_loss: 0.2535 - val_binary_accuracy: 0.9265

Epoch 00007: val_binary_accuracy improved from 0.92449 to 0.92653, saving model
to /content/gdrive/MyDrive/cs2/data/model3/img_only/model3_img_only.hdf5
Epoch 8/10
57/57 [==============================] - 28s 496ms/step - loss: 0.2629 -
binary_accuracy: 0.9274 - val_loss: 0.2480 - val_binary_accuracy: 0.9224

Epoch 00008: val_binary_accuracy did not improve from 0.92653
Epoch 9/10
```

```
57/57 [==============================] - 27s 478ms/step - loss: 0.2673 -
binary_accuracy: 0.9252 - val_loss: 0.2491 - val_binary_accuracy: 0.9204

Epoch 00009: val_binary_accuracy did not improve from 0.92653
Epoch 10/10
57/57 [==============================] - 26s 466ms/step - loss: 0.2490 -
binary_accuracy: 0.9241 - val_loss: 0.2549 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92653
```

[ ]:
```
score = model_3.evaluate(img_cv_gen, verbose=1, batch_size=64)
```

```
31/31 [==============================] - 9s 289ms/step - loss: 0.2599 -
binary_accuracy: 0.9245
```

[ ]:
```
y_pred_0 = model_3.predict(img_cv_gen, batch_size=64)
preds_0 = (model_3.predict(img_cv_gen)>0.5).astype("int32")
metrics(cv['Condition'], preds_0)
```

```
Recall micro      :  0.9183673469387755
Precision micro   :  0.9183673469387755
Fl score micro    :  0.9183673469387755
```

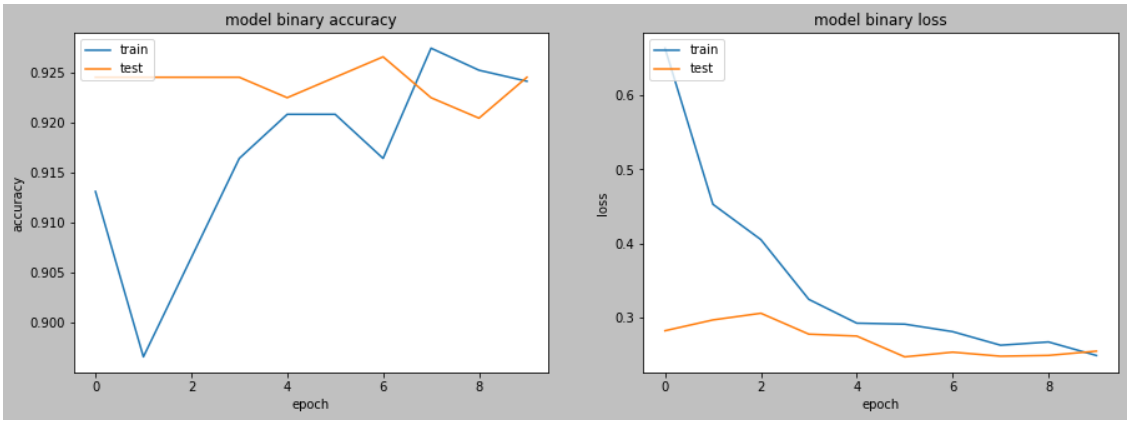
OBSERVATION

1. having only that class which is damage, as it is only we have predict the details about.

[ ]:
```
test_condition_model_3 = test_pred_cond(model_3, img_test_gen, test_data_f)
test_condition_model_3.to_csv('/content/gdrive/MyDrive/cs2/data/model3/img_only/
↪test_condition_model_3.csv')
```

[ ]:
```
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_3.history['binary_accuracy'])
  plt.plot(hstry_3.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_3.history['loss'])
  plt.plot(hstry_3.history['val_loss'])

  plt.title('model binary loss')
```

```
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but at epoch 9, validation loss (here test loss) is more or less equal to training loss indicates model as it is the balance we are looking for, after epoch 9 it may start overfitting.
2. in binary accuracy plot training accuracy is increasing, test loss have hapazard movement, may be because whatever model learing not able to apply in that epoch, more epoch require to have definate say, moreover at epoch 9, we see balance train and test accuracy.

[ ]:
```
model_3.save('/content/gdrive/MyDrive/cs2/data/model3/model3_deepl.h5')
```

##transfer learning

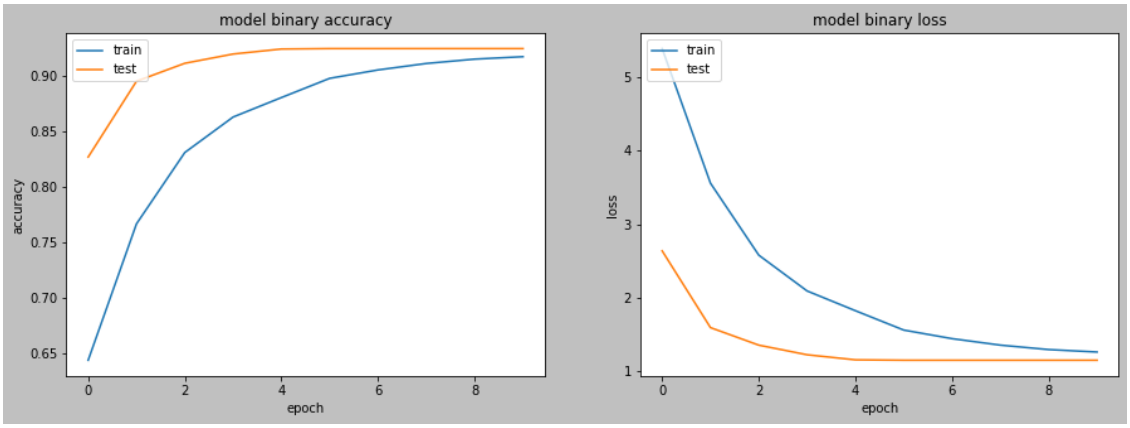
[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/
 ↪model3_transfer.hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=3,␣
 ↪min_lr=0.000001)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
```

```
model_3_transfer_lrng.compile(loss='binary_crossentropy', optimizer =␣
 ↪optimizer, metrics=[tf.keras.metrics.BinaryAccuracy()])

hstry_3_transfer_lrng = model_3_transfer_lrng.fit(img_train_gen,epochs=10,␣
 ↪validation_data=img_cv_gen, batch_size=16,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 29s 477ms/step - loss: 5.3835 -
binary_accuracy: 0.6433 - val_loss: 2.6366 - val_binary_accuracy: 0.8266

Epoch 00001: val_binary_accuracy improved from -inf to 0.82659, saving model to
/content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 2/10
57/57 [==============================] - 27s 476ms/step - loss: 3.5558 -
binary_accuracy: 0.7663 - val_loss: 1.5945 - val_binary_accuracy: 0.8950

Epoch 00002: val_binary_accuracy improved from 0.82659 to 0.89503, saving model
to /content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 3/10
57/57 [==============================] - 27s 478ms/step - loss: 2.5781 -
binary_accuracy: 0.8307 - val_loss: 1.3563 - val_binary_accuracy: 0.9112

Epoch 00003: val_binary_accuracy improved from 0.89503 to 0.91122, saving model
to /content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 4/10
57/57 [==============================] - 27s 480ms/step - loss: 2.0922 -
binary_accuracy: 0.8627 - val_loss: 1.2257 - val_binary_accuracy: 0.9196

Epoch 00004: val_binary_accuracy improved from 0.91122 to 0.91958, saving model
to /content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 5/10
57/57 [==============================] - 27s 478ms/step - loss: 1.8239 -
binary_accuracy: 0.8803 - val_loss: 1.1569 - val_binary_accuracy: 0.9241

Epoch 00005: val_binary_accuracy improved from 0.91958 to 0.92411, saving model
to /content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 6/10
57/57 [==============================] - 27s 481ms/step - loss: 1.5610 -
binary_accuracy: 0.8977 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy improved from 0.92411 to 0.92449, saving model
to /content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/model3_transfer.hdf5
Epoch 7/10
57/57 [==============================] - 27s 470ms/step - loss: 1.4445 -
binary_accuracy: 0.9052 - val_loss: 1.1515 - val_binary_accuracy: 0.9245
```

```
Epoch 00007: val_binary_accuracy did not improve from 0.92449
Epoch 8/10
57/57 [==============================] - 26s 457ms/step - loss: 1.3565 -
binary_accuracy: 0.9111 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 26s 456ms/step - loss: 1.2961 -
binary_accuracy: 0.9150 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 27s 475ms/step - loss: 1.2631 -
binary_accuracy: 0.9172 - val_loss: 1.1515 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_3_transfer_lrng.history['binary_accuracy'])
  plt.plot(hstry_3_transfer_lrng.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_3_transfer_lrng.history['loss'])
  plt.plot(hstry_3_transfer_lrng.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but after epoch 9, validation loss will be converging with training loss indicates at epoch 9 or after as it is the balance we are looking for.
2. in binary accuracy plot training accuracy is increasing, test accuracy is also increasing whatever model is learning able to apply well on test data, it can be more iterated for more convergence.

[ ]:
```
model_3_transfer_lrng.save('/content/gdrive/MyDrive/cs2/data/model3/
↪transfer_lrng/model3_transfer_learning.h5')
```

[ ]:
```
def get_embedd(model, generator, generator_1, generator_2, batch):

    '''takes input of model : vgg-19/resnet/mobilenet/custom respectively in␣
↪subsequent stages,
      generator, generator_1, generator_2 : image datagen for tain,test,cv
      batch   :  batch size (int)
      returns :  dataframe of train and test having transfer weights'''


    y_pred = model.predict(generator, batch)
    df = pd.DataFrame(y_pred)
    test_condition_model_1_tf = test_data_f.copy()
    test_condition_model_1_tf = pd.concat([test_condition_model_3, df], axis=1,␣
↪join='inner')
    test_condition_model_1_tf.to_csv('/content/gdrive/MyDrive/cs2/data/model3/
↪transfer_lrng/test_model_3_transfer_learning.csv')

    y_pred_tr = model.predict(generator_1, batch)
    y_pred_cv = model.predict(generator_2, batch)
```

```
    kd0 = pd.concat([train, cv], axis=0)
    kd0.reset_index(inplace=True)
    print(kd0.shape)
    kd1 = pd.concat([pd.DataFrame(y_pred_tr), pd.DataFrame(y_pred_cv)], axis=0)
    kd1.reset_index(inplace=True)
    print(kd1.shape)
    kd_ = pd.concat([kd0,kd1], axis=1)
    kd_.to_csv('/content/gdrive/MyDrive/cs2/data/model3/transfer_lrng/
 ↪train_model_3_transfer_learning.csv')

    return test_condition_model_1_tf, kd_
```

[ ]:
```
#_,train_transfer_wt = get_embedd(model_1_transfer_lrng, img_train_gen,␣
 ↪img_cv_gen, 16, train=True)
test_transfer_wt, train_transfer_wt = get_embedd(model_3_transfer_lrng,␣
 ↪img_test_gen, img_train_gen, img_cv_gen, 16)#, train=False)
```

```
(1399, 26)
(1399, 65)
```

#custom model

[ ]:
```
#structure taken from https://www.kaggle.com/ankitachoudhury01/
 ↪image-classification-and-xgboost-for-beginners

def custom_model():
  input = Input(shape=(224,224,3))
  x = Conv2D(filters=16, kernel_size=(3,3),activation='relu')(input)
  x = Conv2D(filters=16, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=16, kernel_size=(3,3),activation='relu')(x)
  x = MaxPooling2D(pool_size=(2,2))(x)

  x = Conv2D(filters=64, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=64, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=64, kernel_size=(3,3),activation='relu')(x)
  x = MaxPooling2D(pool_size=(2,2))(x)

  x = Conv2D(filters=128, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=128, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=128, kernel_size=(3,3),activation='relu')(x)
  x = MaxPooling2D(pool_size=(2,2))(x)

  x = Conv2D(filters=256, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=256, kernel_size=(3,3),activation='relu')(x)
  x = Conv2D(filters=256, kernel_size=(3,3),activation='relu')(x)
  x = MaxPooling2D(pool_size=(2,2))(x)
```

```
 x = Conv2D(filters=512, kernel_size=(3,3),activation='relu')(x)
 x = Conv2D(filters=512, kernel_size=(3,3),activation='relu')(x)
 x = Conv2D(filters=512, kernel_size=(3,3),activation='relu')(x)
 x = MaxPooling2D(pool_size=(2,2))(x)


 x = Conv2D(filters=1024, kernel_size=(3,3),activation='relu',padding =␣
↪'same')(x)
 x = MaxPooling2D(pool_size=(2,2), padding = "same")(x)


 flat = Flatten()(x)    ### input layer
 fc1 = Dense(216, activation='relu', kernel_initializer=tf.keras.initializers.
↪glorot_normal(seed=0))(flat)
 drop1 = Dropout(0.5)(fc1)
 #fc2 = Dense(512, activation='relu', kernel_initializer=tf.keras.initializers.
↪glorot_normal(seed=0))(drop1)
 fc3 = Dense(128, activation='relu', kernel_initializer=tf.keras.initializers.
↪glorot_normal(seed=0))(drop1)
 drop2 = Dropout(0.5)(fc3)
 fc4 = Dense(64, activation='relu', kernel_initializer=tf.keras.initializers.
↪glorot_normal(seed=0))(drop2)
 out = Dense(1, activation='sigmoid')(fc4)

 mdl_1 = Model(inputs=[input], outputs=[out])
 mdl_1_transfer_lrng = Model(inputs=[input], outputs=[fc4])
 mdl_1.summary()
 return mdl_1, mdl_1_transfer_lrng
```

[ ]:
```
model_4, model_4_transfer_lrng = custom_model()
```

```
Model: "model_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_8 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 222, 222, 16)      448
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 220, 220, 16)      2320
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 218, 218, 16)      2320
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 109, 109, 16)      0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 107, 107, 64)      9280
```

```
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 105, 105, 64)      36928
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 103, 103, 64)      36928
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 51, 51, 64)        0
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 49, 49, 128)       73856
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 47, 47, 128)       147584
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 45, 45, 128)       147584
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 22, 22, 128)       0
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 20, 20, 256)       295168
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 18, 18, 256)       590080
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 16, 16, 256)       590080
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 8, 8, 256)         0
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 6, 6, 512)         1180160
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 4, 4, 512)         2359808
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 2, 2, 512)         2359808
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 1, 1, 512)         0
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 1, 1, 1024)        4719616
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 1, 1, 1024)        0
_________________________________________________________________
flatten_4 (Flatten)          (None, 1024)              0
_________________________________________________________________
dense_16 (Dense)             (None, 216)               221400
_________________________________________________________________
dropout_8 (Dropout)          (None, 216)               0
_________________________________________________________________
dense_17 (Dense)             (None, 128)               27776
_________________________________________________________________
dropout_9 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_18 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_19 (Dense)             (None, 1)                 65
```

```
=================================================================
Total params: 12,809,465
Trainable params: 12,809,465
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model4/img_only/model4_img_only.
 ↪hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=2)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_4.compile(loss='binary_crossentropy', optimizer = optimizer, metrics=[tf.
 ↪keras.metrics.BinaryAccuracy()])

hstry_4 = model_4.fit(img_train_gen,epochs=10, validation_data=img_cv_gen,␣
 ↪batch_size=8,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 40s 592ms/step - loss: 0.4379 -
binary_accuracy: 0.9219 - val_loss: 0.3279 - val_binary_accuracy: 0.9245

Epoch 00001: val_binary_accuracy improved from -inf to 0.92449, saving model to
/content/gdrive/MyDrive/cs2/data/model4/img_only/model4_img_only.hdf5
Epoch 2/10
57/57 [==============================] - 30s 517ms/step - loss: 0.2932 -
binary_accuracy: 0.9318 - val_loss: 0.2799 - val_binary_accuracy: 0.9245

Epoch 00002: val_binary_accuracy did not improve from 0.92449
Epoch 3/10
57/57 [==============================] - 29s 510ms/step - loss: 0.3283 -
binary_accuracy: 0.9318 - val_loss: 0.3705 - val_binary_accuracy: 0.9245

Epoch 00003: val_binary_accuracy did not improve from 0.92449
Epoch 4/10
57/57 [==============================] - 29s 510ms/step - loss: 0.2830 -
binary_accuracy: 0.9318 - val_loss: 0.2813 - val_binary_accuracy: 0.9245

Epoch 00004: val_binary_accuracy did not improve from 0.92449
Epoch 5/10
57/57 [==============================] - 29s 512ms/step - loss: 0.2801 -
binary_accuracy: 0.9318 - val_loss: 0.2709 - val_binary_accuracy: 0.9245
```

```
Epoch 00005: val_binary_accuracy did not improve from 0.92449
Epoch 6/10
57/57 [==============================] - 29s 503ms/step - loss: 0.2703 -
binary_accuracy: 0.9318 - val_loss: 0.2683 - val_binary_accuracy: 0.9245

Epoch 00006: val_binary_accuracy did not improve from 0.92449
Epoch 7/10
57/57 [==============================] - 29s 507ms/step - loss: 0.2763 -
binary_accuracy: 0.9318 - val_loss: 0.2710 - val_binary_accuracy: 0.9245

Epoch 00007: val_binary_accuracy did not improve from 0.92449
Epoch 8/10
57/57 [==============================] - 29s 500ms/step - loss: 0.2707 -
binary_accuracy: 0.9318 - val_loss: 0.2672 - val_binary_accuracy: 0.9245

Epoch 00008: val_binary_accuracy did not improve from 0.92449
Epoch 9/10
57/57 [==============================] - 29s 505ms/step - loss: 0.2667 -
binary_accuracy: 0.9318 - val_loss: 0.2706 - val_binary_accuracy: 0.9245

Epoch 00009: val_binary_accuracy did not improve from 0.92449
Epoch 10/10
57/57 [==============================] - 29s 502ms/step - loss: 0.2705 -
binary_accuracy: 0.9318 - val_loss: 0.2746 - val_binary_accuracy: 0.9245

Epoch 00010: val_binary_accuracy did not improve from 0.92449
```

[ ]:
```
score = model_4.evaluate(img_cv_gen, verbose=1, batch_size=64)
```

```
31/31 [==============================] - 9s 293ms/step - loss: 0.2753 -
binary_accuracy: 0.9245
```

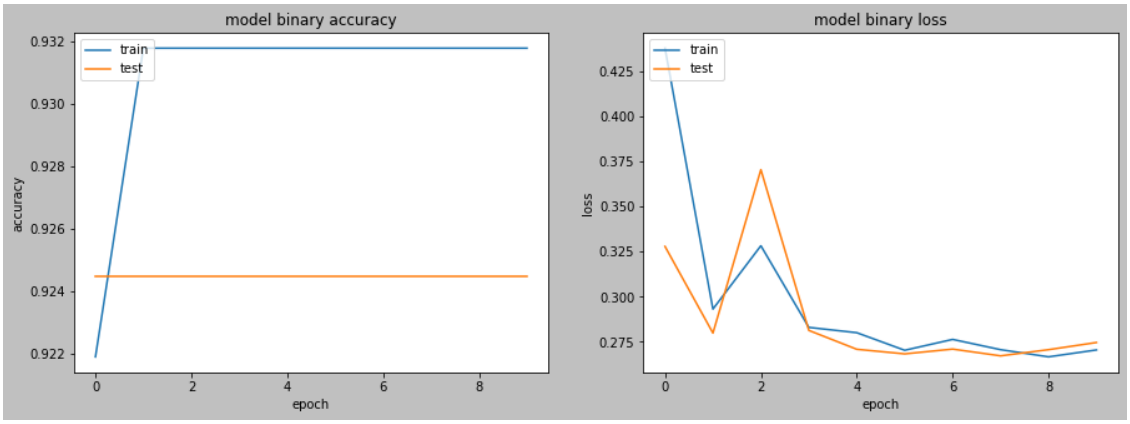
[ ]:
```
y_pred_0 = model_4.predict(img_cv_gen, batch_size=64)
preds_0 = (model_4.predict(img_cv_gen)>0.5).astype("int32")
metrics(cv['Condition'], preds_0)
```

```
Recall micro      :  0.9244897959183673
Precision micro   :  0.9244897959183673
Fl score micro    :  0.9244897959183674
```

[ ]:
```
test_condition_model_4 = test_pred_cond(model_4, img_test_gen, test_data_f)
test_condition_model_4.to_csv('/content/gdrive/MyDrive/cs2/data/model4/img_only/
↪test_condition_model_4.csv')
```

[ ]:
```
#import matplotlib.pyplot as plt

def plot_los():
```

```
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_4.history['binary_accuracy'])
  plt.plot(hstry_4.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_4.history['loss'])
  plt.plot(hstry_4.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, but at epoch 1-3 and after 7, validation loss (here test loss) is more than training loss indicates model is overfitting, at epoch 3 or 7, strikes the balance we are looking for.

2. in binary accuracy plot training accuracy is increasing but test accuracy is constant because whaterever model learns it apllies on test set there is no new to learn from test data perspective, due to test set is limited with similar nature of condition (more damage), and repeatative similar images (i.e less variation in image data), this nature of graph occurs.

3. try to overcome above limitation (point 2) with oversampling of inferior class, or using cars dataset, in version 2.0 of this case study.

[ ]:
```
model_4.save('/content/gdrive/MyDrive/cs2/data/model4/model4_deepl.h5')
```

##transfer learning

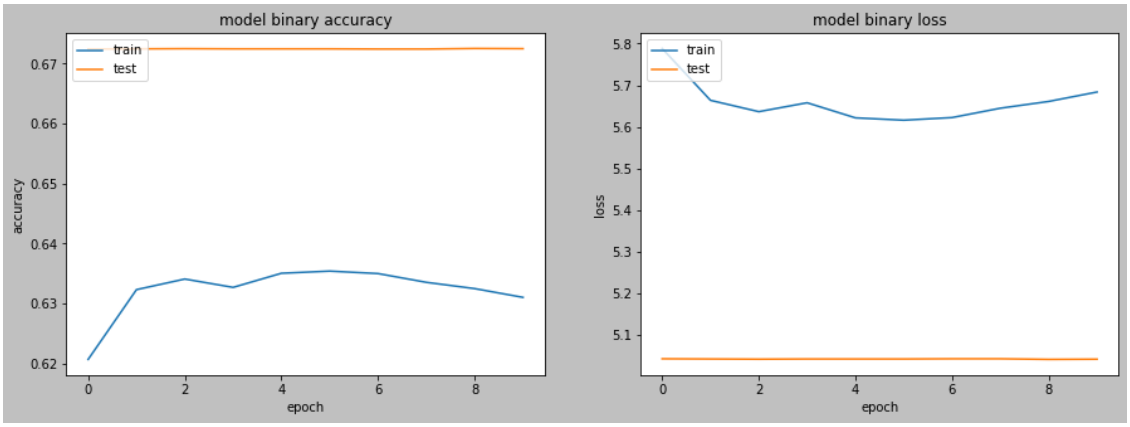
[ ]:
```
from tensorflow.keras.callbacks import *

filepath = '/content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/
 ↪model4_transfer.hdf5'
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_binary_accuracy',␣
 ↪verbose=1,save_best_only=True, mode='auto', save_freq='epoch')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=3,␣
 ↪min_lr=0.000001)

optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.
 ↪999, epsilon=1e-07)
model_4_transfer_lrng.compile(loss='binary_crossentropy', optimizer =␣
 ↪optimizer, metrics=[tf.keras.metrics.BinaryAccuracy()])

hstry_4_transfer_lrng = model_4_transfer_lrng.fit(img_train_gen,epochs=10,␣
 ↪validation_data=img_cv_gen, batch_size=16,
          callbacks=[checkpoint, reduce_lr])
```

```
Epoch 1/10
57/57 [==============================] - 32s 537ms/step - loss: 5.7883 -
binary_accuracy: 0.6207 - val_loss: 5.0431 - val_binary_accuracy: 0.6724

Epoch 00001: val_binary_accuracy improved from -inf to 0.67245, saving model to
/content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/model4_transfer.hdf5
Epoch 2/10
57/57 [==============================] - 30s 530ms/step - loss: 5.6638 -
binary_accuracy: 0.6323 - val_loss: 5.0426 - val_binary_accuracy: 0.6725

Epoch 00002: val_binary_accuracy improved from 0.67245 to 0.67248, saving model
to /content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/model4_transfer.hdf5
Epoch 3/10
57/57 [==============================] - 30s 534ms/step - loss: 5.6365 -
binary_accuracy: 0.6341 - val_loss: 5.0421 - val_binary_accuracy: 0.6725

Epoch 00003: val_binary_accuracy improved from 0.67248 to 0.67251, saving model
to /content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/model4_transfer.hdf5
Epoch 4/10
57/57 [==============================] - 30s 530ms/step - loss: 5.6579 -
binary_accuracy: 0.6327 - val_loss: 5.0426 - val_binary_accuracy: 0.6725

Epoch 00004: val_binary_accuracy did not improve from 0.67251
```

```
Epoch 5/10
57/57 [==============================] - 30s 521ms/step - loss: 5.6217 -
binary_accuracy: 0.6350 - val_loss: 5.0426 - val_binary_accuracy: 0.6725

Epoch 00005: val_binary_accuracy did not improve from 0.67251
Epoch 6/10
57/57 [==============================] - 29s 518ms/step - loss: 5.6161 -
binary_accuracy: 0.6354 - val_loss: 5.0426 - val_binary_accuracy: 0.6725

Epoch 00006: val_binary_accuracy did not improve from 0.67251
Epoch 7/10
57/57 [==============================] - 30s 523ms/step - loss: 5.6224 -
binary_accuracy: 0.6350 - val_loss: 5.0431 - val_binary_accuracy: 0.6724

Epoch 00007: val_binary_accuracy did not improve from 0.67251
Epoch 8/10
57/57 [==============================] - 30s 520ms/step - loss: 5.6449 -
binary_accuracy: 0.6335 - val_loss: 5.0431 - val_binary_accuracy: 0.6724

Epoch 00008: val_binary_accuracy did not improve from 0.67251
Epoch 9/10
57/57 [==============================] - 30s 522ms/step - loss: 5.6611 -
binary_accuracy: 0.6325 - val_loss: 5.0416 - val_binary_accuracy: 0.6725

Epoch 00009: val_binary_accuracy improved from 0.67251 to 0.67254, saving model
to /content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/model4_transfer.hdf5
Epoch 10/10
57/57 [==============================] - 30s 523ms/step - loss: 5.6837 -
binary_accuracy: 0.6310 - val_loss: 5.0421 - val_binary_accuracy: 0.6725

Epoch 00010: val_binary_accuracy did not improve from 0.67254
```

[ ]:
```python
#import matplotlib.pyplot as plt

def plot_los():
  fig = plt.figure(figsize=(15, 5)).patch.set_facecolor('silver')
  plt.subplot(121)

  plt.plot(hstry_4_transfer_lrng.history['binary_accuracy'])
  plt.plot(hstry_4_transfer_lrng.history['val_binary_accuracy'])
  plt.title('model binary accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')

  plt.subplot(122)
  plt.plot(hstry_4_transfer_lrng.history['loss'])
```

```
  plt.plot(hstry_4_transfer_lrng.history['val_loss'])

  plt.title('model binary loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train','test'], loc='upper left')
  plt.show()

plot_los()
```



OBSERVATION

1. from the loss curve, initially in epoch loss curve training loss is higher than validation loss that means underftting, which is quite evident as it is starting stage, that to is too large in quantam, no sign of convergence, underfit because less layers in model (less depth) in custom architecture as compare to other models (transfer learning models) therefore not able to learn the appropriate features , it has to be ignored while considering.

[ ]:
```
model_4_transfer_lrng.save('/content/gdrive/MyDrive/cs2/data/model4/
↪transfer_lrng/model4_transfer_learning.h5')
```

[ ]:
```
def get_embedd(model, generator, generator_1, generator_2, batch):

    '''takes input of model : vgg-19/resnet/mobilenet/custom respectively in␣
↪subsequent stages,
      generator, generator_1, generator_2 : image datagen for tain,test,cv
      batch   :  batch size (int)
      returns :  dataframe of train and test having transfer weights'''

    y_pred = model.predict(generator, batch)
    df = pd.DataFrame(y_pred)
    test_condition_model_1_tf = test_data_f.copy()
```

```
    test_condition_model_1_tf = pd.concat([test_condition_model_3, df], axis=1,␣
↪join='inner')
    test_condition_model_1_tf.to_csv('/content/gdrive/MyDrive/cs2/data/model4/
↪transfer_lrng/test_model_4_transfer_learning.csv')

    y_pred_tr = model.predict(generator_1, batch)
    y_pred_cv = model.predict(generator_2, batch)
    kd0 = pd.concat([train, cv], axis=0)
    kd0.reset_index(inplace=True)
    print(kd0.shape)
    kd1 = pd.concat([pd.DataFrame(y_pred_tr), pd.DataFrame(y_pred_cv)], axis=0)
    kd1.reset_index(inplace=True)
    print(kd1.shape)
    kd_ = pd.concat([kd0,kd1], axis=1)
    kd_.to_csv('/content/gdrive/MyDrive/cs2/data/model4/transfer_lrng/
↪train_model_4_transfer_learning.csv')

    return test_condition_model_1_tf, kd_
```

[ ]:
```
#_,train_transfer_wt = get_embedd(model_1_transfer_lrng, img_train_gen,␣
↪img_cv_gen, 16, train=True)
test_transfer_wt, train_transfer_wt = get_embedd(model_4_transfer_lrng,␣
↪img_test_gen, img_train_gen, img_cv_gen, 16)
```

```
(1399, 26)
(1399, 65)
```