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

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

```
!pip3 install autokeras
```

```
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: tensorboard~=2.5 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: keras-nightly~=2.5.0.dev in /usr/local/lib/python3.
Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: gast==0.4.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: h5py~=3.1.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python
Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages
Collecting terminaltables
  Downloading https://files.pythonhosted.org/packages/9b/c4/4a21174f32f8a7e1104798
Collecting colorama
  Downloading https://files.pythonhosted.org/packages/44/98/5b86278fbbf250d239ae0e
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/pyt
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.7/d
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/
Requirement already satisfied: cached-property; python_version < "3.8" in /usr/loc
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /usr/
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/d
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in /usr/loca
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/di
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-pa
Building wheels for collected packages: keras-tuner, terminaltables
  Building wheel for keras-tuner (setup.py) ... done
  Created wheel for keras-tuner: filename=keras_tuner-1.0.2-cp37-none-any.whl size
  Stored in directory: /root/.cache/pip/wheels/bb/a1/8a/7c3de0efb3707a1701b36ebbfd
```

```
  Building wheel for terminaltables (setup.py) ... done
  Created wheel for terminaltables: filename=terminaltables-3.1.0-cp37-none-any.wh
  Stored in directory: /root/.cache/pip/wheels/30/6b/50/6c75775b681fb36cdfac7f1979
Successfully built keras-tuner terminaltables
Installing collected packages: terminaltables, colorama, keras-tuner, autokeras
Successfully installed autokeras-1.0.13 colorama-0.4.4 keras-tuner-1.0.2 terminalt
```

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'  # or any {'0', '1', '2'}

from numpy import mean
from numpy import std
import numpy as np
from matplotlib import pyplot
from sklearn.model_selection import KFold
from keras.datasets import mnist
#from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dense
from keras.layers import Flatten
from keras.optimizers import SGD
from keras.layers import Dropout
from keras.layers import BatchNormalization
import keras
import tensorflow as tf
from keras.models import Model
from keras import backend as K
import matplotlib.pyplot as plt
import sklearn
from sklearn.metrics import confusion_matrix
from scipy.io import loadmat
import numpy as np
import PIL
import cv2
from sklearn.model_selection import train_test_split
import autokeras as ak
import tensorflow
import warnings


'''
x = list()
data = list()
z = 0
y=[]
##Class-1 images##
folder_path_class1 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_co
folder_path_class2 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_co
folder_path_class3 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_co
folder_path_class4 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_co
folder_path_class5 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_co
folder_path_class1b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_c
```

```
folder_path_class2b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_c
folder_path_class3b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_c
folder_path_class4b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_c
folder_path_class5b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_c

paths = [folder_path_class1,folder_path_class1b,folder_path_class2,folder_path_class2b,fol
         folder_path_class5,folder_path_class5b]
for i in range(len(paths)):
  print(i)
  if i%2==0:
    y_l = 0
  elif i%2!=0:
    y_l=1
  for image in os.walk(paths[i]):
    data.append(image[2])
#    print(image[2])

  for j in range(len(data[0])):
    str_complete = paths[i] + data[0][j]
    #print(str_complete)
    img = cv2.imread(str_complete)
    img = cv2.resize(img, (128, 128))
    x.append(img)#Ensure all images are loaded
    y.append(y_l)
  data = []

data_x = np.asarray(x)
np.save('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_complete/scheme1

y = np.asarray(y)
np.save('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_complete/scheme1
'''



x = np.load('/content/drive/MyDrive/PCG_signal_time_frequency_image/github_database_5Class
y = np.load('/content/drive/MyDrive/PCG_signal_time_frequency_image/github_database_5Class

indices = np.arange(x.shape[0])
np.random.shuffle(indices)
x = x[indices]
y = y[indices]


#loaded_model = tensorflow.keras.models.load_model("/content/drive/MyDrive/PCG_signal_time
#print(loaded_model.summary())
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 128, 3)]     0
_________________________________________________________________
cast_to_float32 (CastToFloat (None, 128, 128, 3)       0
_________________________________________________________________
normalization (Normalization (None, 128, 128, 3)       7
_________________________________________________________________
```

```
conv2d (Conv2D)              (None, 126, 126, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 124, 124, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 62, 62, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 62, 62, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 246016)            0
_________________________________________________________________
dropout_1 (Dropout)          (None, 246016)            0
_________________________________________________________________
dense (Dense)                (None, 1)                 246017
_________________________________________________________________
classification_head_1 (Activ (None, 1)                 0
=================================================================
Total params: 265,416
Trainable params: 265,409
Non-trainable params: 7
_________________________________________________________________
None
```

```
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
warnings.filterwarnings("ignore")
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)


##Define model
test_acc_list = []
K_cappa_list = []
precision_list = []
recall_list = []
F1_list = []

for iter in range(5):
  print("Trial Number : ", (iter+1))
  loaded_model = tensorflow.keras.models.load_model("/content/drive/MyDrive/PCG_signal_tim
  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=1)
  x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=1/8, rando


  y_tr_one_hot  = np.zeros((np.array(y_train).shape[0],5))
  for i in range(np.array(y_train).shape[0]):
    label = y_train[i]
    y_tr_one_hot[i][int(label)] = 1

  K.clear_session()
  #modelf = model_define()

  #print(modelf.summary())
  optimizer = keras.optimizers.Adam(lr=0.001)
 # loaded_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accur

  y_val_one_hot  = np.zeros((np.array(y_val).shape[0],5))
  for i in range(np.array(y_val).shape[0]):
    label = y_val[i]
```

```
      y_val_one_hot[i][int(label)] = 1

  #  hist = loaded_model.fit(np.array(x_train), y_tr_one_hot, validation_data=(np.array(x_va

    y_te_one_hot  = np.zeros((np.array(y_test).shape[0],5))
    for i in range(np.array(y_test).shape[0]):
      label = y_test[i]
      y_te_one_hot[i][int(label)] = 1

    test_loss, test_acc = loaded_model.evaluate(np.array(x_test), np.array(y_te_one_hot), ve
    print('Accuracy:',test_acc)
    test_acc_list.append(test_acc)

    ##Evaluating Sensitivity, Accuracy and Kappa scores
    y_prob = loaded_model.predict(x_test)
    Y_pred = y_prob.argmax(axis=-1)

    K_cappa = sklearn.metrics.cohen_kappa_score(y_test,Y_pred)
    print("cohen kappa scores:" ,K_cappa)
    K_cappa_list.append(K_cappa)

    cm1 = confusion_matrix(y_test,Y_pred)
    print("confusion matrix \n",cm1)

    precision = sklearn.metrics.precision_score(y_test,Y_pred,average='micro')
    print('precision : ', precision)
    precision_list.append(precision)

    recall = sklearn.metrics.recall_score(y_test,Y_pred,average='micro')
    print('recall : ', recall)
    recall_list.append(recall)

    F1 = sklearn.metrics.f1_score(y_test,Y_pred,average="micro")
    print("F1 : ", F1)
    F1_list.append(F1)


print('5-Trial Accuracy:',sum(test_acc_list)/len(test_acc_list))
print("5-Trial cohen kappa scores:" ,sum(K_cappa_list)/len(K_cappa_list))
print('5-Trial precision : ', sum(precision_list)/len(precision_list))
print('5-Trial recall : ', sum(recall_list)/len(recall_list))
print("5-Trial F1 : ", sum(F1_list)/len(F1_list))

print("Accuracy list: ",test_acc_list)
print("Cohen Kappa scores: ", K_cappa_list)
print("Precision scores: ", precision_list)
print("Recall list", recall_list)
print("F1 scores", F1_list)
```

```
      WARNING:absl:Importing a function (__inference_block5d_activation_layer_call_and_r
      WARNING:absl:Importing a function (__inference_block5g_expand_activation_layer_cal
      WARNING:absl:Importing a function (__inference_block1c_activation_layer_call_and_r
      WARNING:absl:Importing a function (__inference_block5h_activation_layer_call_and_r
      WARNING:absl:Importing a function (__inference_block6e_expand_activation_layer_cal
      WARNING:absl:Importing a function (__inference_block5h_expand_activation_layer_cal
```

```
WARNING:absl:Importing a function (__inference_block2a_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block5c_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4b_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block2g_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block3d_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4a_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block1b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4j_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_efficientnetb7_layer_call_and_retur
WARNING:absl:Importing a function (__inference_block5i_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block5i_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2f_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6l_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5h_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block5j_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block1d_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6c_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6i_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block5e_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block5f_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2g_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4i_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block1d_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block3b_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6a_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block1a_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5d_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2a_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4d_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4i_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_stem_activation_layer_call_and_retu
WARNING:absl:Importing a function (__inference_block5a_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6h_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block3f_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block2b_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block7b_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5i_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block3e_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block7b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5g_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2a_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6m_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block3f_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6k_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block5g_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block3e_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6j_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block2g_activation_layer_call_and_r
```

```
'''
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=1)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=1/8, random_

y_tr_one_hot  = np.zeros((np.array(y_train).shape[0],5))
for i in range(np.array(y_train).shape[0]):
```

```
  label = y_train[i]
  y_tr_one_hot[i][int(label)] = 1

clf = ak.ImageClassifier(overwrite=True, max_trials=4)
clf.fit(x_train, y_tr_one_hot,epochs=40)
results = clf.predict(x_test)

#model = clf.export_model()
#model.save("/content/drive/MyDrive/PCG_signal_time_frequency_image/model_physionet_databa
'''


#model = clf.export_model()
#model.save("/content/drive/MyDrive/PCG_signal_time_frequency_image/github_database_5Class
```

### 5-FOLD CV

```
x = np.load('/content/drive/MyDrive/PCG_signal_time_frequency_image/github_database_5Class
y = np.load('/content/drive/MyDrive/PCG_signal_time_frequency_image/github_database_5Class

indices = np.arange(x.shape[0])
np.random.shuffle(indices)
x_cv = x[indices]
y_cv = y[indices]

y_one_hot  = np.zeros((np.array(y_cv).shape[0],5))

for i in range(np.array(y_cv).shape[0]):
  label = y_cv[i]
  y_one_hot[i][int(label)] = 1


from sklearn.model_selection import StratifiedKFold
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
test_acc_list = []
K_cappa_list = []
precision_list = []
recall_list = []
F1_list = []
iter = 0

for train,test in kfold.split(x_cv,y_cv):
  print("Fold Number : ", (iter+1))
  iter = iter+1
  loaded_model = tf.keras.models.load_model("/content/drive/MyDrive/PCG_signal_time_freque

  K.clear_session()
  #modelf = model_define()

  #print(modelf.summary())
  optimizer = keras.optimizers.Adam(lr=0.001)
  #loaded_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accur

  #hist = loaded_model.fit(x_cv[train], y_one_hot[train],validation_data=(x_cv[test],y_one
```

```
    test_loss, test_acc = loaded_model.evaluate(x_cv[test],y_one_hot[test],batch_size=8, ver
    print('Accuracy:',test_acc)
    test_acc_list.append(test_acc)

    ##Evaluating Sensitivity, Accuracy and Kappa scores
    y_prob = loaded_model.predict(x_cv[test])
    Y_pred = y_prob.argmax(axis=-1)

    K_cappa = sklearn.metrics.cohen_kappa_score(y_cv[test],Y_pred)
    print("cohen kappa scores:" ,K_cappa)
    K_cappa_list.append(K_cappa)

    cm1 = confusion_matrix(y_cv[test],Y_pred)
    print("confusion matrix \n",cm1)

    precision = sklearn.metrics.precision_score(y_cv[test],Y_pred,average='micro')
    print('precision : ', precision)
    precision_list.append(precision)

    recall = sklearn.metrics.recall_score(y_cv[test],Y_pred,average='micro')
    print('recall : ', recall)
    recall_list.append(recall)

    F1 = sklearn.metrics.f1_score(y_cv[test],Y_pred,average="micro")
    print("F1 : ", F1)
    F1_list.append(F1)


print('5-FOLD CV Accuracy:',sum(test_acc_list)/len(test_acc_list))
print("5-FOLD CV cohen kappa scores:" ,sum(K_cappa_list)/len(K_cappa_list))
print('5-FOLD CV precision : ', sum(precision_list)/len(precision_list))
print('5-FOLD CV recall : ', sum(recall_list)/len(recall_list))
print("5-FOLD CV F1 : ", sum(F1_list)/len(F1_list))

print("Accuracy list: ",test_acc_list)
print("Cohen Kappa scores: ", K_cappa_list)
print("Precision scores: ", precision_list)
print("Recall list", recall_list)
print("F1 scores", F1_list)
```

```
    Fold Number :  1
    WARNING:absl:Importing a function (__inference_block2b_activation_layer_call_and_r
    WARNING:absl:Importing a function (__inference_block2a_activation_layer_call_and_r
    WARNING:absl:Importing a function (__inference_block7d_se_reduce_layer_call_and_re
    WARNING:absl:Importing a function (__inference_block6a_activation_layer_call_and_r
    WARNING:absl:Importing a function (__inference_model_layer_call_and_return_conditi
    WARNING:absl:Importing a function (__inference_block6e_se_reduce_layer_call_and_re
    WARNING:absl:Importing a function (__inference_block6m_expand_activation_layer_cal
    WARNING:absl:Importing a function (__inference_block6d_se_reduce_layer_call_and_re
    WARNING:absl:Importing a function (__inference_block6m_expand_activation_layer_cal
    WARNING:absl:Importing a function (__inference_block6m_activation_layer_call_and_r
    WARNING:absl:Importing a function (__inference_block7a_expand_activation_layer_cal
    WARNING:absl:Importing a function (__inference_block2b_se_reduce_layer_call_and_re
    WARNING:absl:Importing a function (__inference_block4c_expand_activation_layer_cal
    WARNING:absl:Importing a function (__inference_block5j_activation_layer_call_and_r
```

```
WARNING:absl:Importing a function (__inference_block6h_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4e_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6k_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5g_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4d_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block1a_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block4a_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2d_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5f_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block5j_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block3c_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6i_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4j_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4b_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6g_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block4d_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block2b_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6h_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block7d_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block2f_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6j_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block2c_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block6c_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block7c_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4e_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block5e_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6i_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block2c_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block2c_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block3d_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block3e_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block6l_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block1b_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_model_layer_call_and_return_conditi
WARNING:absl:Importing a function (__inference_block1a_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block4i_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_block4e_activation_layer_call_and_r
WARNING:absl:Importing a function (__inference_block6c_expand_activation_layer_cal
WARNING:absl:Importing a function (__inference_efficientnetb7_layer_call_and_retur
WARNING:absl:Importing a function (__inference_block4f_se_reduce_layer_call_and_re
WARNING:absl:Importing a function (__inference_block4d_activation_layer_call_and_r
```

```
##Plot results
import matplotlib.pyplot as plt

loss_train = hist.history['accuracy']
loss_val = hist.history['val_accuracy']
epochs = range(0,15)
plt.plot(epochs, loss_train, 'g', label='training accuracy')
plt.plot(epochs, loss_val, 'b', label='validation accuracy')
#plt.title('Training and Validation accuracy')
plt.xlabel('Epochs', fontsize=30, fontweight='bold')
plt.ylabel('Accuracy',fontsize=30,  fontweight='bold')
font = {'family' : 'normal',
        'weight' : 'bold',
        'size'   : 10}
```

```
    plt.rc('font', **font)

    plt.legend()
    plt.show()
```