```
!pip3 install autokeras
```

```
    Requirement already satisfied: autokeras in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: keras-tuner<1.1,>=1.0.2 in /usr/local/lib/python3.7
    Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (f
    Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: tensorflow<2.6,>=2.3.0 in /usr/local/lib/python3.7/
    Requirement already satisfied: tensorboard in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (fr
    Requirement already satisfied: kt-legacy in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages 
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (fr
    Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python
    Requirement already satisfied: gast==0.4.0 in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist-packag
    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: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-pack
    Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.7/dist-p
    Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0 in /usr/local/li
    Requirement already satisfied: keras-nightly~=2.5.0.dev in /usr/local/lib/python3.
    Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: grpcio~=1.34.0 in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.7/dis
    Requirement already satisfied: typing-extensions~=3.7.4 in /usr/local/lib/python3.
    Requirement already satisfied: h5py~=3.1.0 in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/
    Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/d
    Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist
    Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local
    Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/pyt
    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 in /usr/local/lib/python3.7/dist-pack
    Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.
    Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7
    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: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
    Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/pyth
    Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages 
    Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (
```

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

```
    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
from numpy import mean
from numpy import std
import numpy as np
from matplotlib import pyplot
from sklearn.model_selection import KFold
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import BatchNormalization
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.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
import os
from sklearn.model_selection import train_test_split
import autokeras as ak
```

```
# x = list()
# data = list()
# y = list()
# z = 0
# ##Class-1 images##
# folder_path_class1 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/STFT_Githu
# #folder_path_class2 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_
# #folder_path_class3 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_
# #folder_path_class4 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_
# #folder_path_class5 = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physionet_

# #folder_path_class1b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physione
# #folder_path_class2b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physione
# #folder_path_class3b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physione
# #folder_path_class4b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physione
# #folder_path_class5b = ('/content/drive/MyDrive/PCG_signal_time_frequency_image/Physione
# paths = [folder_path_class1]

# class_types = {'MVP':0,'MR':1,'MS':2,'normal':3,'AS':4}

# for p in paths:
#   #print(p)
#   for image in os.walk(p):
#     data.append(image[2])
#     #print(image[2])

#   for i in range(len(data[0])):
#     name = data[0][i].split('_')[0]
#     #print(name)
#     y.append(class_types[str(name)])
#     str_complete = p + data[0][i]
#     #print(str_complete)
#     img = cv2.imread(str_complete)
#     img = cv2.resize(img, (128, 128))
#     x.append(img)#Ensure all images are loaded
#   data = []

# data_x = np.asarray(x)
# y = np.asarray(y)
# np.save('/content/drive/MyDrive/PCG_signal_time_frequency_image/STFT_Github/x',data_x)
# np.save('/content/drive/MyDrive/PCG_signal_time_frequency_image/STFT_Github/y',y)
```

```
x  = np.load("/content/drive/MyDrive/PCG_signal_time_frequency_image/CWT_Github/x.npy")
y = np.load("/content/drive/MyDrive/PCG_signal_time_frequency_image/CWT_Github/y.npy")
```

```
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

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

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
```

AutoKeras Code

```
input_node = ak.ImageInput()
```

```
output_node = ak.Normalization()(input_node)
output_node1 = ak.ConvBlock()(output_node)
output_node2 = ak.ConvBlock(max_pooling=True)(output_node1)
output_node = ak.ClassificationHead()(output_node)

auto_model = ak.AutoModel(
    inputs=input_node, outputs=output_node, overwrite=True, max_trials=1
)

print(x_train.shape)
print(y_tr_one_hot.shape)

# Feed the AutoModel with training data.
auto_model.fit(x_train, y_train, epochs=10)
# Predict with the best model.
predicted_y = auto_model.predict(x_test)
# Evaluate the best model with testing data.
print(auto_model.evaluate(x_test, y_test))
```

```
Trial 1 Complete [00h 00m 12s]
val_loss: 0.12087678909301758

Best val_loss So Far: 0.12087678909301758
Total elapsed time: 00h 00m 12s
INFO:tensorflow:Oracle triggered exit
Epoch 1/10
25/25 [==============================] - 1s 41ms/step - loss: 24.6282 - accuracy: 0.3
Epoch 2/10
25/25 [==============================] - 1s 40ms/step - loss: 2.2449 - accuracy: 0.77
Epoch 3/10
25/25 [==============================] - 1s 41ms/step - loss: 0.3660 - accuracy: 0.93
Epoch 4/10
25/25 [==============================] - 1s 41ms/step - loss: 0.2739 - accuracy: 0.94
Epoch 5/10
25/25 [==============================] - 1s 41ms/step - loss: 0.1519 - accuracy: 0.96
Epoch 6/10
25/25 [==============================] - 1s 40ms/step - loss: 0.1515 - accuracy: 0.95
Epoch 7/10
25/25 [==============================] - 1s 40ms/step - loss: 0.0972 - accuracy: 0.97
Epoch 8/10
25/25 [==============================] - 1s 41ms/step - loss: 0.1860 - accuracy: 0.95
Epoch 9/10
25/25 [==============================] - 1s 40ms/step - loss: 0.3473 - accuracy: 0.94
Epoch 10/10
25/25 [==============================] - 1s 40ms/step - loss: 0.0307 - accuracy: 0.99
INFO:tensorflow:Assets written to: ./auto_model/best_model/assets
4/4 [==============================] - 0s 15ms/step
4/4 [==============================] - 0s 14ms/step - loss: 0.4353 - accuracy: 0.9500
[0.43529921770095825, 0.949999988079071]
```

```
predicted_y = auto_model.predict(x_test)
```

```
4/4 [==============================] - 0s 14ms/step
```

```
p_list = predicted_y.reshape(100).tolist()
```

```
p_list_int = []
for i in p_list:
  p_list_int.append(int(i))
```

```
from sklearn.metrics import confusion_matrix
cm1 = confusion_matrix(y_test,p_list_int)
print("confusion matrix \n",cm1)
```

```
confusion matrix
 [[14  2  2  0  0]
 [ 0 20  1  0  0]
 [ 0  0 26  0  0]
 [ 0  0  0 15  0]
 [ 0  0  0  0 20]]
```

```
precision = sklearn.metrics.precision_score(y_test,p_list_int,average='micro')
print(precision)
```

```
0.95
```

```
accuracy=np.diag(cm1).sum()/cm1.sum().sum()
print(accuracy)
```

```
0.95
```

```
  recall = sklearn.metrics.recall_score(y_test,p_list_int,average='micro')
  print(recall)
```

```
0.95
```

```
F1 = sklearn.metrics.f1_score(y_test,p_list_int,average='micro')
print(F1)
```

```
0.9500000000000001
```

```
K_cappa = sklearn.metrics.cohen_kappa_score(y_test,p_list_int)
print(K_cappa)
```

```
0.9367648918679651
```

● ✕