Brain Tumor Detection using Deep Learning

April 25, 2024

[4]:
```
#Importing packages
from torch.utils.data import Dataset, DataLoader
from torchvision import transforms, models
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import os
import PIL
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
import tensorflow as tf
from tensorflow import keras
from keras.utils.np_utils import to_categorical # used for converting labels to␣
 ↪one-hot-encoding
from keras.models import Sequential
from keras import backend as K
import itertools
#from keras.layers.normalization import BatchNormalization
from keras.utils.np_utils import to_categorical # convert to one-hot-encoding
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau
from keras import regularizers
from keras.layers.core import Dense
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Conv2D, MaxPooling2D,␣
 ↪GlobalMaxPooling2D,AveragePooling2D, Flatten, Dropout, Input,␣
 ↪BatchNormalization
from sklearn.metrics import (accuracy_score, f1_score, precision_score,␣
 ↪recall_score, classification_report, confusion_matrix)
```

[5]:
```
!pip install numpy==1.20
```

```
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting numpy==1.20
  Downloading numpy-1.20.0-cp38-cp38-macosx_10_9_x86_64.whl.metadata (2.0 kB)
Downloading numpy-1.20.0-cp38-cp38-macosx_10_9_x86_64.whl (16.0 MB)
```

```
                                    16.0/16.0 MB
593.0 kB/s eta 0:00:0000:0100:01
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 1.20.3
    Uninstalling numpy-1.20.3:
      Successfully uninstalled numpy-1.20.3
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
rfpimp 1.3.2 requires sklearn, which is not installed.
wrf-python 1.3.4.1 requires basemap, which is not installed.
altair 5.2.0 requires typing-extensions>=4.0.1; python_version < "3.11", but you
have typing-extensions 3.7.4.3 which is incompatible.
bokeh 2.4.3 requires typing-extensions>=3.10.0, but you have typing-extensions
3.7.4.3 which is incompatible.
pandas 1.5.3 requires numpy>=1.20.3, but you have numpy 1.20.0 which is
incompatible.
pingouin 0.5.2 requires scikit-learn<1.1.0, but you have scikit-learn 1.1.3
which is incompatible.
pyportfolioopt 1.5.5 requires numpy<2.0.0,>=1.22.4, but you have numpy 1.20.0
which is incompatible.
sktime 0.14.0 requires numpy<1.23,>=1.21.0, but you have numpy 1.20.0 which is
incompatible.
tensorflow 2.4.1 requires numpy~=1.19.2, but you have numpy 1.20.0 which is
incompatible.
Successfully installed numpy-1.20.0
```

[7]:
```
import torch
```

1 Data pre-processing

[22]:
```
path = '/Users/kipkemoivincent/Desktop/Covid/Data2'
```

[37]:
```
IMG_WIDTH = 100
IMG_HEIGHT = 100
BATCH_SIZE= 7023
```

horizontal_flip=True, vertical_flip=True,zoom_range=0.3,

[38]:
```
train_datagen =tf.keras.preprocessing.image.ImageDataGenerator(rescale=1.0/
 ↪255,validation_split=0.0001)
train_generator = train_datagen.flow_from_directory(path,
                                                    target_size=(IMG_WIDTH,␣
 ↪IMG_HEIGHT),
                                                    batch_size=BATCH_SIZE,
                                                    class_mode='categorical',
                                                    shuffle=True,
                                                    subset='training')
```

```
Found 7023 images belonging to 4 classes.
```

[27]:
```
validation_generator = train_datagen.flow_from_directory(path,
                                                    target_size=(IMG_WIDTH,␣
 ↪IMG_HEIGHT),
                                                    batch_size=743,
                                                    class_mode='categorical',
                                                    shuffle=True,
                                                    subset='validation')
```

```
Found 0 images belonging to 4 classes.
```

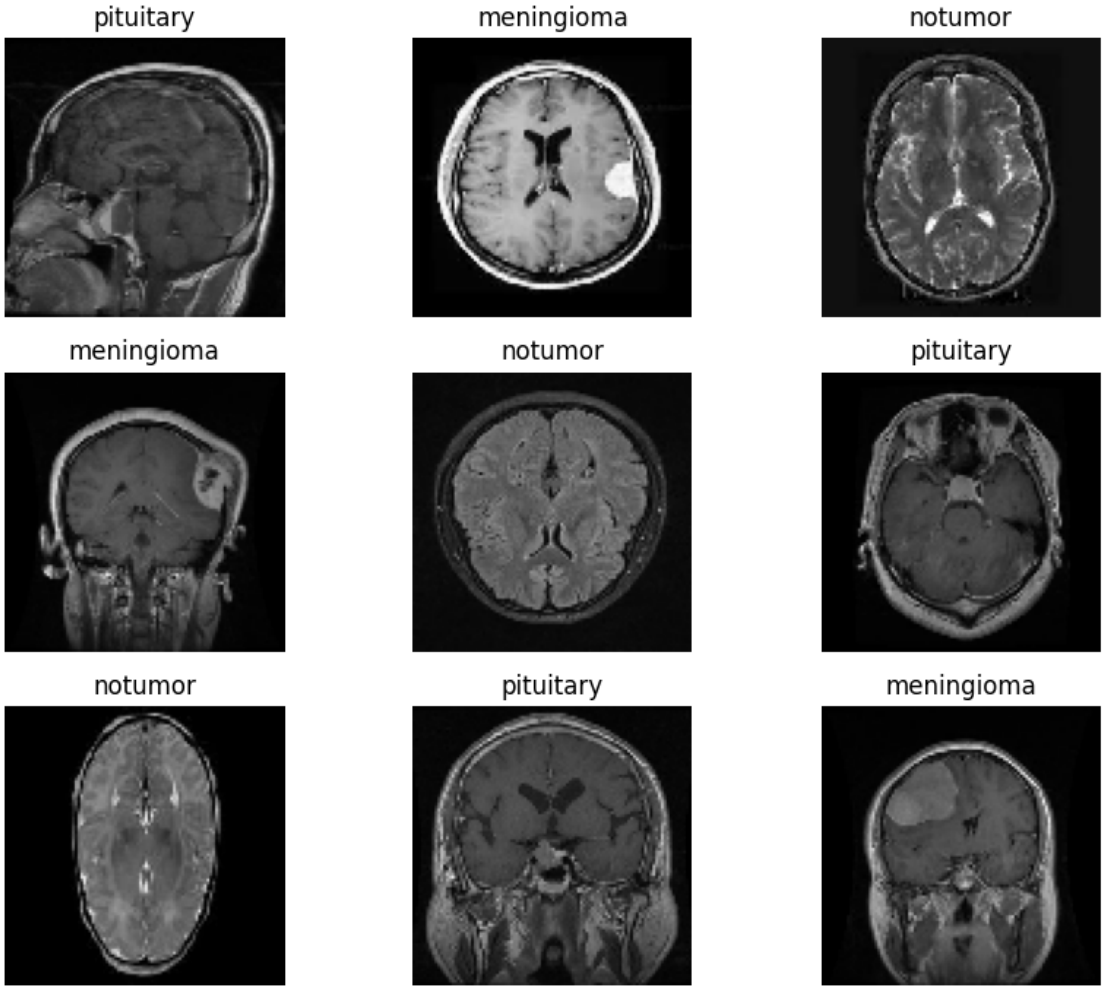
[39]:
```
labels = {value: key for key, value in train_generator.class_indices.items()}

print("Label Mappings for classes present in the training and validation␣
 ↪datasets\n")
for key, value in labels.items():
    print(f"{key} : {value}")
```

```
Label Mappings for classes present in the training and validation datasets

0 : glioma
1 : meningioma
2 : notumor
3 : pituitary
```

[29]:
```
#!pip install numpy==1.20
```

[40]:
```
def report(arr1,arr2):
    print ('\x1b[6;30;46m'+'Accuracy:'+str(np.round(accuracy_score(arr1,␣
 ↪arr2),4))+','+' Precision:'+str(np.round(precision_score(arr1, arr2),4))+','+
    ' Recall:'+str(np.round(recall_score(arr1, arr2),4))+','+' F1_score:
 ↪'+str(np.round(f1_score(arr1, arr2),4)))
    return
```

[31]:
```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(nrows=3, ncols=3, figsize=(9, 7))
idx = 0

for i in range(3):
    for j in range(3):
        label = labels[np.argmax(train_generator[0][1][idx])]
        ax[i, j].set_title(f"{label}")
        ax[i, j].imshow(train_generator[0][0][idx][:, :, :])
        ax[i, j].axis("off")
        idx += 1

plt.tight_layout()
#plt.suptitle("Sample Training Images", fontsize=21)
plt.show()
```



[41]:
```
X, y = next(train_generator)
X=(X-X.mean())/X.std()
#X_test, y_test = next(validation_generator)
```

[42]:
```
from sklearn.model_selection import train_test_split
X_train1, X_test, y_train1, y_test = train_test_split(X, y, test_size=0.10,␣
 ↪random_state=42, shuffle=True)
```

[43]:
```
X_train, X_val, y_train, y_val = train_test_split(X_train1, y_train1,␣
 ↪test_size=0.20, random_state=42, shuffle=True)
```

[44]:
```
X_train.shape,X_val.shape, X_test.shape
```

[44]:
```
((5056, 100, 100, 3), (1264, 100, 100, 3), (703, 100, 100, 3))
```

[45]:
```
IMPUT_SHAPE=(IMG_WIDTH, IMG_HEIGHT, 3)
```

2 Prediction using different Architectures

3 A. CustomCNN

[52]:
```
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.optimizers import Adam
initializer = tf.keras.initializers.HeNormal()
values = initializer(shape=(2, 2))
```

[53]:
```
from sklearn.utils import compute_class_weight
y=[np.argmax(i) for i in y_train]
class_weights = compute_class_weight(class_weight = "balanced",classes = np.
 ↪unique(y),y = y)
class_weights = dict(zip(np.unique(y), class_weights))
```

[54]:
```
input_data = Input(shape=IMPUT_SHAPE)

#Convolution
x = Conv2D(32, (3, 3), activation="relu")(input_data)

#Pooling
x = MaxPooling2D(pool_size = (4, 4), strides=(4, 4))(x)

#Dropout
x = Dropout(0.25)(x)

# 2nd Convolution
x = Conv2D(32, (3, 3), activation="relu")(x)

# 2nd Pooling layer
```

```
x = MaxPooling2D(pool_size = (2, 2))(x)

#Dropout
x = Dropout(0.3)(x)

#3rd Convolution
x = Conv2D(32, (3, 3), activation='relu')(x)

#3rd Pooling Layer
x = MaxPooling2D(pool_size=(2, 2))(x)

#Dropout
x = Dropout(0.3)(x)

# Flatten the layer
x = Flatten()(x)

# Fully Connected Layers
x =Dense(128, activation = 'relu')(x)
output = Dense(4, activation = 'softmax')(x)

cnn =keras.models.Model(inputs=input_data, outputs=output)

# Compile the Neural network
cnn.compile(optimizer =Adam(learning_rate=0.0001), loss =␣
 ↪'categorical_crossentropy',
            metrics = ['accuracy'])
```

[55]:
```
cnn.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 100, 100, 3)]     0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 98, 98, 32)        896
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 24, 24, 32)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 22, 22, 32)        9248
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
dropout_4 (Dropout)          (None, 11, 11, 32)        0
_________________________________________________________________
```

```
conv2d_5 (Conv2D)            (None, 9, 9, 32)          9248
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 4, 4, 32)          0
_________________________________________________________________
dropout_5 (Dropout)          (None, 4, 4, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 516
=================================================================
Total params: 85,572
Trainable params: 85,572
Non-trainable params: 0
_________________________________________________________________
```

[56]:
```
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from keras.callbacks import ReduceLROnPlateau
filepath11="weights.best_custom_cnn2.hdf5"
checkpoint1 = ModelCheckpoint(filepath11, monitor='val_accuracy', verbose=1,␣
 ↪save_best_only=True, mode='max')
es = EarlyStopping(monitor='val_accuracy', patience=20)
rlrop = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=10)
callbacks_list = [checkpoint1,es,rlrop]
```

[57]:
```
history1 = cnn.fit(X_train,y_train,epochs = 100,verbose = 1,batch_size=4,
                   validation_data =(X_test,y_test),callbacks=callbacks_list,
                  class_weight=class_weights)
```

```
Epoch 1/100
1264/1264 [==============================] - 24s 19ms/step - loss: 1.2916 -
accuracy: 0.4375 - val_loss: 0.8828 - val_accuracy: 0.6714

Epoch 00001: val_accuracy improved from -inf to 0.67141, saving model to
weights.best_custom_cnn2.hdf5
Epoch 2/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.8651 -
accuracy: 0.6635 - val_loss: 0.7001 - val_accuracy: 0.7710

Epoch 00002: val_accuracy improved from 0.67141 to 0.77098, saving model to
weights.best_custom_cnn2.hdf5
Epoch 3/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.7360 -
accuracy: 0.7175 - val_loss: 0.6294 - val_accuracy: 0.7553

Epoch 00003: val_accuracy did not improve from 0.77098
Epoch 4/100
```

```
1264/1264 [==============================] - 24s 19ms/step - loss: 0.6538 -
accuracy: 0.7470 - val_loss: 0.5994 - val_accuracy: 0.7411

Epoch 00004: val_accuracy did not improve from 0.77098
Epoch 5/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.6164 -
accuracy: 0.7628 - val_loss: 0.5024 - val_accuracy: 0.8137

Epoch 00005: val_accuracy improved from 0.77098 to 0.81366, saving model to
weights.best_custom_cnn2.hdf5
Epoch 6/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.5390 -
accuracy: 0.7902 - val_loss: 0.5118 - val_accuracy: 0.7752

Epoch 00006: val_accuracy did not improve from 0.81366
Epoch 7/100
1264/1264 [==============================] - 27s 21ms/step - loss: 0.4983 -
accuracy: 0.8165 - val_loss: 0.4584 - val_accuracy: 0.8165

Epoch 00007: val_accuracy improved from 0.81366 to 0.81650, saving model to
weights.best_custom_cnn2.hdf5
Epoch 8/100
1264/1264 [==============================] - 31s 24ms/step - loss: 0.4604 -
accuracy: 0.8241 - val_loss: 0.4091 - val_accuracy: 0.8592

Epoch 00008: val_accuracy improved from 0.81650 to 0.85917, saving model to
weights.best_custom_cnn2.hdf5
Epoch 9/100
1264/1264 [==============================] - 33s 26ms/step - loss: 0.4558 -
accuracy: 0.8242 - val_loss: 0.3671 - val_accuracy: 0.8848

Epoch 00009: val_accuracy improved from 0.85917 to 0.88478, saving model to
weights.best_custom_cnn2.hdf5
Epoch 10/100
1264/1264 [==============================] - 31s 25ms/step - loss: 0.4459 -
accuracy: 0.8327 - val_loss: 0.3606 - val_accuracy: 0.8777

Epoch 00010: val_accuracy did not improve from 0.88478
Epoch 11/100
1264/1264 [==============================] - 29s 23ms/step - loss: 0.4132 -
accuracy: 0.8516 - val_loss: 0.3486 - val_accuracy: 0.8919

Epoch 00011: val_accuracy improved from 0.88478 to 0.89189, saving model to
weights.best_custom_cnn2.hdf5
Epoch 12/100
1264/1264 [==============================] - 27s 22ms/step - loss: 0.3923 -
accuracy: 0.8576 - val_loss: 0.3328 - val_accuracy: 0.8848
```

```
Epoch 00012: val_accuracy did not improve from 0.89189
Epoch 13/100
1264/1264 [==============================] - 28s 22ms/step - loss: 0.3606 -
accuracy: 0.8654 - val_loss: 0.3292 - val_accuracy: 0.8848

Epoch 00013: val_accuracy did not improve from 0.89189
Epoch 14/100
1264/1264 [==============================] - 27s 22ms/step - loss: 0.3451 -
accuracy: 0.8708 - val_loss: 0.3236 - val_accuracy: 0.8890

Epoch 00014: val_accuracy did not improve from 0.89189
Epoch 15/100
1264/1264 [==============================] - 27s 21ms/step - loss: 0.3512 -
accuracy: 0.8655 - val_loss: 0.2995 - val_accuracy: 0.8947

Epoch 00015: val_accuracy improved from 0.89189 to 0.89474, saving model to
weights.best_custom_cnn2.hdf5
Epoch 16/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.3381 -
accuracy: 0.8728 - val_loss: 0.2848 - val_accuracy: 0.8976

Epoch 00016: val_accuracy improved from 0.89474 to 0.89758, saving model to
weights.best_custom_cnn2.hdf5
Epoch 17/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2880 -
accuracy: 0.8933 - val_loss: 0.2939 - val_accuracy: 0.8876

Epoch 00017: val_accuracy did not improve from 0.89758
Epoch 18/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.2968 -
accuracy: 0.8987 - val_loss: 0.2738 - val_accuracy: 0.9104

Epoch 00018: val_accuracy improved from 0.89758 to 0.91038, saving model to
weights.best_custom_cnn2.hdf5
Epoch 19/100
1264/1264 [==============================] - 26s 21ms/step - loss: 0.2767 -
accuracy: 0.8959 - val_loss: 0.3016 - val_accuracy: 0.8919

Epoch 00019: val_accuracy did not improve from 0.91038
Epoch 20/100
1264/1264 [==============================] - 26s 21ms/step - loss: 0.2735 -
accuracy: 0.9035 - val_loss: 0.2621 - val_accuracy: 0.9161

Epoch 00020: val_accuracy improved from 0.91038 to 0.91607, saving model to
weights.best_custom_cnn2.hdf5
Epoch 21/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.2729 -
accuracy: 0.8950 - val_loss: 0.2501 - val_accuracy: 0.9147
```

```
Epoch 00021: val_accuracy did not improve from 0.91607
Epoch 22/100
1264/1264 [==============================] - 27s 21ms/step - loss: 0.2643 -
accuracy: 0.9075 - val_loss: 0.2284 - val_accuracy: 0.9203

Epoch 00022: val_accuracy improved from 0.91607 to 0.92034, saving model to
weights.best_custom_cnn2.hdf5
Epoch 23/100
1264/1264 [==============================] - 26s 21ms/step - loss: 0.2501 -
accuracy: 0.9057 - val_loss: 0.2392 - val_accuracy: 0.9232

Epoch 00023: val_accuracy improved from 0.92034 to 0.92319, saving model to
weights.best_custom_cnn2.hdf5
Epoch 24/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2392 -
accuracy: 0.9091 - val_loss: 0.2157 - val_accuracy: 0.9289

Epoch 00024: val_accuracy improved from 0.92319 to 0.92888, saving model to
weights.best_custom_cnn2.hdf5
Epoch 25/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.2341 -
accuracy: 0.9208 - val_loss: 0.2173 - val_accuracy: 0.9260

Epoch 00025: val_accuracy did not improve from 0.92888
Epoch 26/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2318 -
accuracy: 0.9163 - val_loss: 0.2140 - val_accuracy: 0.9303

Epoch 00026: val_accuracy improved from 0.92888 to 0.93030, saving model to
weights.best_custom_cnn2.hdf5
Epoch 27/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2304 -
accuracy: 0.9129 - val_loss: 0.2297 - val_accuracy: 0.9118

Epoch 00027: val_accuracy did not improve from 0.93030
Epoch 28/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2178 -
accuracy: 0.9141 - val_loss: 0.1869 - val_accuracy: 0.9417

Epoch 00028: val_accuracy improved from 0.93030 to 0.94168, saving model to
weights.best_custom_cnn2.hdf5
Epoch 29/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.2097 -
accuracy: 0.9187 - val_loss: 0.2148 - val_accuracy: 0.9260

Epoch 00029: val_accuracy did not improve from 0.94168
Epoch 30/100
```

```
1264/1264 [==============================] - 26s 21ms/step - loss: 0.1907 -
accuracy: 0.9313 - val_loss: 0.1899 - val_accuracy: 0.9331

Epoch 00030: val_accuracy did not improve from 0.94168
Epoch 31/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.2070 -
accuracy: 0.9249 - val_loss: 0.1833 - val_accuracy: 0.9346

Epoch 00031: val_accuracy did not improve from 0.94168
Epoch 32/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.1948 -
accuracy: 0.9247 - val_loss: 0.1765 - val_accuracy: 0.9331

Epoch 00032: val_accuracy did not improve from 0.94168
Epoch 33/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1738 -
accuracy: 0.9349 - val_loss: 0.1699 - val_accuracy: 0.9445

Epoch 00033: val_accuracy improved from 0.94168 to 0.94452, saving model to
weights.best_custom_cnn2.hdf5
Epoch 34/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1831 -
accuracy: 0.9300 - val_loss: 0.1558 - val_accuracy: 0.9431

Epoch 00034: val_accuracy did not improve from 0.94452
Epoch 35/100
1264/1264 [==============================] - 26s 20ms/step - loss: 0.1830 -
accuracy: 0.9365 - val_loss: 0.1714 - val_accuracy: 0.9488

Epoch 00035: val_accuracy improved from 0.94452 to 0.94879, saving model to
weights.best_custom_cnn2.hdf5
Epoch 36/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1548 -
accuracy: 0.9431 - val_loss: 0.1587 - val_accuracy: 0.9488

Epoch 00036: val_accuracy did not improve from 0.94879
Epoch 37/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1790 -
accuracy: 0.9312 - val_loss: 0.1583 - val_accuracy: 0.9403

Epoch 00037: val_accuracy did not improve from 0.94879
Epoch 38/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1548 -
accuracy: 0.9408 - val_loss: 0.1555 - val_accuracy: 0.9459

Epoch 00038: val_accuracy did not improve from 0.94879
Epoch 39/100
1264/1264 [==============================] - 25s 19ms/step - loss: 0.1643 -
```

```
accuracy: 0.9402 - val_loss: 0.1590 - val_accuracy: 0.9445

Epoch 00039: val_accuracy did not improve from 0.94879
Epoch 40/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1487 -
accuracy: 0.9458 - val_loss: 0.1611 - val_accuracy: 0.9459

Epoch 00040: val_accuracy did not improve from 0.94879
Epoch 41/100
1264/1264 [==============================] - 28s 22ms/step - loss: 0.1403 -
accuracy: 0.9483 - val_loss: 0.1541 - val_accuracy: 0.9431

Epoch 00041: val_accuracy did not improve from 0.94879
Epoch 42/100
1264/1264 [==============================] - 27s 22ms/step - loss: 0.1406 -
accuracy: 0.9456 - val_loss: 0.1538 - val_accuracy: 0.9531

Epoch 00042: val_accuracy improved from 0.94879 to 0.95306, saving model to
weights.best_custom_cnn2.hdf5
Epoch 43/100
1264/1264 [==============================] - 27s 22ms/step - loss: 0.1331 -
accuracy: 0.9480 - val_loss: 0.1521 - val_accuracy: 0.9403

Epoch 00043: val_accuracy did not improve from 0.95306
Epoch 44/100
1264/1264 [==============================] - 27s 21ms/step - loss: 0.1346 -
accuracy: 0.9500 - val_loss: 0.1540 - val_accuracy: 0.9431

Epoch 00044: val_accuracy did not improve from 0.95306
Epoch 45/100
1264/1264 [==============================] - 27s 22ms/step - loss: 0.1338 -
accuracy: 0.9516 - val_loss: 0.1404 - val_accuracy: 0.9388

Epoch 00045: val_accuracy did not improve from 0.95306
Epoch 46/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1329 -
accuracy: 0.9538 - val_loss: 0.2203 - val_accuracy: 0.9189

Epoch 00046: val_accuracy did not improve from 0.95306
Epoch 47/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1217 -
accuracy: 0.9561 - val_loss: 0.1277 - val_accuracy: 0.9531

Epoch 00047: val_accuracy did not improve from 0.95306
Epoch 48/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1234 -
accuracy: 0.9569 - val_loss: 0.1353 - val_accuracy: 0.9559
```

```
Epoch 00048: val_accuracy improved from 0.95306 to 0.95590, saving model to
weights.best_custom_cnn2.hdf5
Epoch 49/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1266 -
accuracy: 0.9547 - val_loss: 0.1216 - val_accuracy: 0.9545

Epoch 00049: val_accuracy did not improve from 0.95590
Epoch 50/100
1264/1264 [==============================] - 26s 21ms/step - loss: 0.1285 -
accuracy: 0.9520 - val_loss: 0.1279 - val_accuracy: 0.9502

Epoch 00050: val_accuracy did not improve from 0.95590
Epoch 51/100
1264/1264 [==============================] - 27s 21ms/step - loss: 0.1017 -
accuracy: 0.9606 - val_loss: 0.1270 - val_accuracy: 0.9587

Epoch 00051: val_accuracy improved from 0.95590 to 0.95875, saving model to
weights.best_custom_cnn2.hdf5
Epoch 52/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1148 -
accuracy: 0.9575 - val_loss: 0.1722 - val_accuracy: 0.9360

Epoch 00052: val_accuracy did not improve from 0.95875
Epoch 53/100
1264/1264 [==============================] - 24s 19ms/step - loss: 0.1128 -
accuracy: 0.9587 - val_loss: 0.1178 - val_accuracy: 0.9602

Epoch 00053: val_accuracy improved from 0.95875 to 0.96017, saving model to
weights.best_custom_cnn2.hdf5
Epoch 54/100
1264/1264 [==============================] - 26s 21ms/step - loss: 0.1094 -
accuracy: 0.9593 - val_loss: 0.1314 - val_accuracy: 0.9488

Epoch 00054: val_accuracy did not improve from 0.96017
Epoch 55/100
1264/1264 [==============================] - 25s 20ms/step - loss: 0.1040 -
accuracy: 0.9628 - val_loss: 0.1209 - val_accuracy: 0.9630

Epoch 00055: val_accuracy improved from 0.96017 to 0.96302, saving model to
weights.best_custom_cnn2.hdf5
Epoch 56/100
1264/1264 [==============================] - 29s 23ms/step - loss: 0.1042 -
accuracy: 0.9604 - val_loss: 0.1196 - val_accuracy: 0.9587

Epoch 00056: val_accuracy did not improve from 0.96302
Epoch 57/100
1264/1264 [==============================] - 34s 27ms/step - loss: 0.1119 -
accuracy: 0.9574 - val_loss: 0.1091 - val_accuracy: 0.9587
```

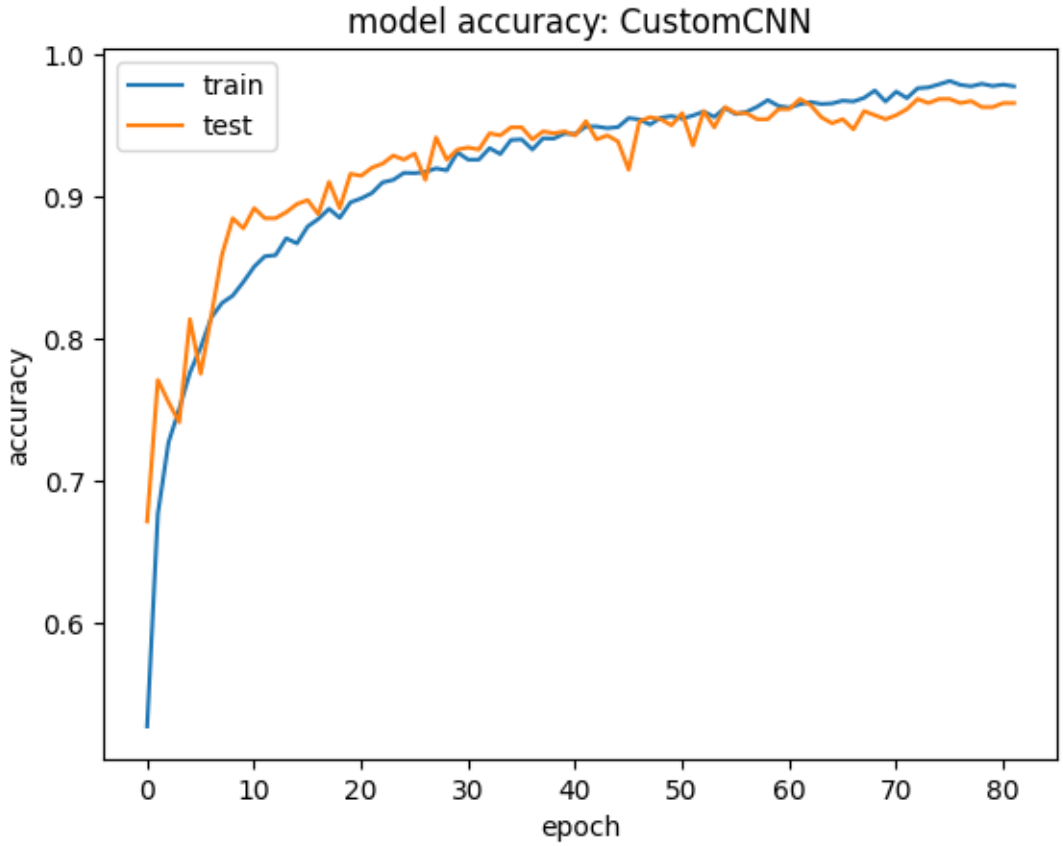
```
Epoch 00057: val_accuracy did not improve from 0.96302
Epoch 58/100
1264/1264 [==============================] - 37s 29ms/step - loss: 0.1033 -
accuracy: 0.9603 - val_loss: 0.1273 - val_accuracy: 0.9545

Epoch 00058: val_accuracy did not improve from 0.96302
Epoch 59/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.0973 -
accuracy: 0.9674 - val_loss: 0.1157 - val_accuracy: 0.9545

Epoch 00059: val_accuracy did not improve from 0.96302
Epoch 60/100
1264/1264 [==============================] - 35s 28ms/step - loss: 0.0969 -
accuracy: 0.9680 - val_loss: 0.1128 - val_accuracy: 0.9616

Epoch 00060: val_accuracy did not improve from 0.96302
Epoch 61/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.1015 -
accuracy: 0.9643 - val_loss: 0.1205 - val_accuracy: 0.9616

Epoch 00061: val_accuracy did not improve from 0.96302
Epoch 62/100
1264/1264 [==============================] - 35s 27ms/step - loss: 0.1035 -
accuracy: 0.9608 - val_loss: 0.1116 - val_accuracy: 0.9687

Epoch 00062: val_accuracy improved from 0.96302 to 0.96871, saving model to
weights.best_custom_cnn2.hdf5
Epoch 63/100
1264/1264 [==============================] - 34s 27ms/step - loss: 0.0895 -
accuracy: 0.9692 - val_loss: 0.1080 - val_accuracy: 0.9644

Epoch 00063: val_accuracy did not improve from 0.96871
Epoch 64/100
1264/1264 [==============================] - 35s 28ms/step - loss: 0.0926 -
accuracy: 0.9642 - val_loss: 0.1193 - val_accuracy: 0.9559

Epoch 00064: val_accuracy did not improve from 0.96871
Epoch 65/100
1264/1264 [==============================] - 31s 24ms/step - loss: 0.0950 -
accuracy: 0.9659 - val_loss: 0.1272 - val_accuracy: 0.9516

Epoch 00065: val_accuracy did not improve from 0.96871
Epoch 66/100
1264/1264 [==============================] - 29s 23ms/step - loss: 0.1012 -
accuracy: 0.9631 - val_loss: 0.1349 - val_accuracy: 0.9545

Epoch 00066: val_accuracy did not improve from 0.96871
```

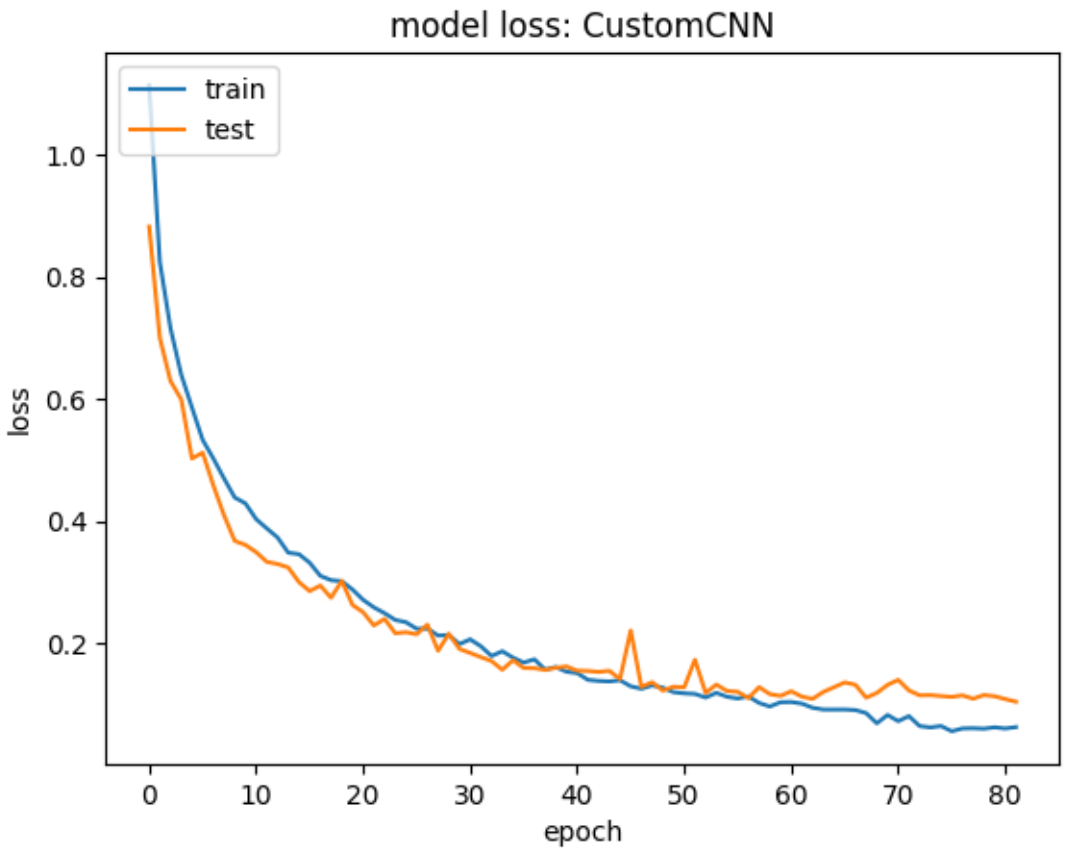
```
Epoch 67/100
1264/1264 [==============================] - 29s 23ms/step - loss: 0.0809 -
accuracy: 0.9713 - val_loss: 0.1311 - val_accuracy: 0.9474

Epoch 00067: val_accuracy did not improve from 0.96871
Epoch 68/100
1264/1264 [==============================] - 37s 29ms/step - loss: 0.0793 -
accuracy: 0.9719 - val_loss: 0.1100 - val_accuracy: 0.9602

Epoch 00068: val_accuracy did not improve from 0.96871
Epoch 69/100
1264/1264 [==============================] - 39s 31ms/step - loss: 0.0621 -
accuracy: 0.9758 - val_loss: 0.1176 - val_accuracy: 0.9573

Epoch 00069: val_accuracy did not improve from 0.96871
Epoch 70/100
1264/1264 [==============================] - 39s 31ms/step - loss: 0.0790 -
accuracy: 0.9682 - val_loss: 0.1309 - val_accuracy: 0.9545

Epoch 00070: val_accuracy did not improve from 0.96871
Epoch 71/100
1264/1264 [==============================] - 39s 31ms/step - loss: 0.0751 -
accuracy: 0.9702 - val_loss: 0.1392 - val_accuracy: 0.9573

Epoch 00071: val_accuracy did not improve from 0.96871
Epoch 72/100
1264/1264 [==============================] - 36s 29ms/step - loss: 0.0766 -
accuracy: 0.9713 - val_loss: 0.1217 - val_accuracy: 0.9616

Epoch 00072: val_accuracy did not improve from 0.96871
Epoch 73/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.0666 -
accuracy: 0.9765 - val_loss: 0.1138 - val_accuracy: 0.9687

Epoch 00073: val_accuracy did not improve from 0.96871
Epoch 74/100
1264/1264 [==============================] - 37s 29ms/step - loss: 0.0663 -
accuracy: 0.9763 - val_loss: 0.1142 - val_accuracy: 0.9659

Epoch 00074: val_accuracy did not improve from 0.96871
Epoch 75/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.0650 -
accuracy: 0.9789 - val_loss: 0.1126 - val_accuracy: 0.9687

Epoch 00075: val_accuracy did not improve from 0.96871
Epoch 76/100
1264/1264 [==============================] - 37s 29ms/step - loss: 0.0475 -
accuracy: 0.9837 - val_loss: 0.1112 - val_accuracy: 0.9687
```

```
Epoch 00076: val_accuracy did not improve from 0.96871
Epoch 77/100
1264/1264 [==============================] - 36s 29ms/step - loss: 0.0592 -
accuracy: 0.9783 - val_loss: 0.1137 - val_accuracy: 0.9659

Epoch 00077: val_accuracy did not improve from 0.96871
Epoch 78/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.0568 -
accuracy: 0.9804 - val_loss: 0.1082 - val_accuracy: 0.9673

Epoch 00078: val_accuracy did not improve from 0.96871
Epoch 79/100
1264/1264 [==============================] - 36s 28ms/step - loss: 0.0599 -
accuracy: 0.9798 - val_loss: 0.1142 - val_accuracy: 0.9630

Epoch 00079: val_accuracy did not improve from 0.96871
Epoch 80/100
1264/1264 [==============================] - 36s 29ms/step - loss: 0.0551 -
accuracy: 0.9816 - val_loss: 0.1122 - val_accuracy: 0.9630

Epoch 00080: val_accuracy did not improve from 0.96871
Epoch 81/100
1264/1264 [==============================] - 36s 29ms/step - loss: 0.0607 -
accuracy: 0.9773 - val_loss: 0.1076 - val_accuracy: 0.9659

Epoch 00081: val_accuracy did not improve from 0.96871
Epoch 82/100
1264/1264 [==============================] - 37s 29ms/step - loss: 0.0628 -
accuracy: 0.9784 - val_loss: 0.1033 - val_accuracy: 0.9659

Epoch 00082: val_accuracy did not improve from 0.96871
```

[58]:
```
# summarize history for accuracy
plt.plot(history1.history['accuracy'])
plt.plot(history1.history['val_accuracy'])
plt.title('model accuracy: CustomCNN')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history1.history['loss'])
plt.plot(history1.history['val_loss'])
plt.title('model loss: CustomCNN')
plt.ylabel('loss')
plt.xlabel('epoch')
```

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

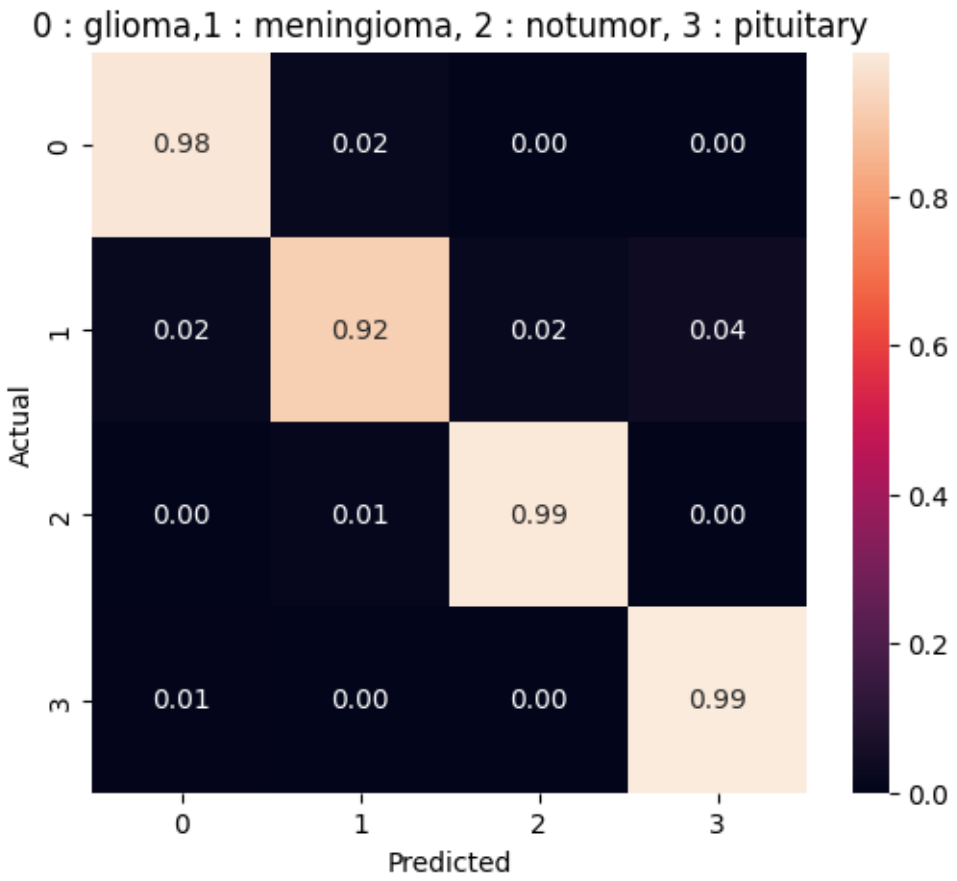




[101]:
```
# load the saved model
from keras.models import load_model
cnn=load_model('weights.best_custom_cnn2.hdf5')
```

[106]:
```
pred1=cnn.predict(X_test)
```

[102]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(Y_test, Pred)
# Normalise
cmn = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(cmn, annot=True, fmt='.2f')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('0 : glioma,1 : meningioma, 2 : notumor, 3 : pituitary')
plt.show(block=False)
```



[103]:
```
from keras import models
from numpy import loadtxt
from tensorflow.keras.models import save_model
save_model(cnn, "customCNN1.h5")
# load and evaluate a saved model
loaded_model = models.load_model('customCNN1.h5')
# summarize model.
model=loaded_model
train_pred_p=model.predict(X_train)
train_pred = np.argmax(train_pred_p, axis=1)
```

4 B. MobileNetV2

[64]:
```
from tensorflow.keras.layers.experimental.preprocessing import RandomFlip,␣
↪RandomRotation

def make_mobilenet_model(image_size, num_classes):
```

```
    input_shape = image_size

    base_model = tf.keras.applications.MobileNetV2(input_shape=input_shape,
                                                   include_top=False, # Do not␣
 ↪include the dense prediction layer
                                                   weights="imagenet") # Load␣
 ↪imageNet parameters

    # Freeze the base model by making it non trainable
    base_model.trainable = False

    # create the input layer (Same as the imageNetv2 input size)
    inputs = tf.keras.Input(shape=input_shape)

    # apply data augmentation to the inputs
    x = inputs

    # set training to False to avoid keeping track of statistics in the batch␣
 ↪norm layer
    x = base_model(x, training=False)

    # Add the new Binary classification layers
    # use global avg pooling to summarize the info in each channel
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    #include dropout with probability of 0.2 to avoid overfitting
    x = Dropout(0.3)(x)
    x = Flatten()(x)
# Fully Connected Layers
    x =Dense(128, activation = 'relu')(x)

    prediction_layer = Dense(4, activation='softmax')

    outputs = prediction_layer(x)

    model = keras.models.Model(inputs, outputs)

    return model
```

[65]:
```
filepath21="weights.best_mobile_net2.hdf5"
checkpoint2 = ModelCheckpoint(filepath21, monitor='val_accuracy', verbose=1,␣
 ↪save_best_only=True, mode='max')
es2 = EarlyStopping(monitor='val_accuracy', patience=20)
rlrop2 = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=10)
callbacks_list2 = [checkpoint2,es2,rlrop2]
```

[66]:
```
# Define a model using the make_model function
image_size = (100,100,3)
mobilenet_model = make_mobilenet_model(image_size, num_classes = 2)

# Preview the Model Summary
mobilenet_model.summary()
```

```
WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not in
[96, 128, 160, 192, 224]. Weights for input shape (224, 224) will be loaded as
the default.
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 100, 100, 3)]     0
_________________________________________________________________
mobilenetv2_1.00_224 (Functi (None, 4, 4, 1280)        2257984
_________________________________________________________________
global_average_pooling2d (Gl (None, 1280)              0
_________________________________________________________________
dropout_6 (Dropout)          (None, 1280)              0
_________________________________________________________________
flatten_2 (Flatten)          (None, 1280)              0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               163968
_________________________________________________________________
dense_5 (Dense)              (None, 4)                 516
=================================================================
Total params: 2,422,468
Trainable params: 164,484
Non-trainable params: 2,257,984
_________________________________________________________________
```

[67]:
```
base_learning_rate = 0.001
optimizer = Adam(learning_rate = base_learning_rate)
initial_epochs = 50
batch_size = 64
loss = 'categorical_crossentropy'
metrics = ['accuracy']

mobilenet_model.compile(optimizer =Adam(learning_rate=0.0001), loss =␣
↪'categorical_crossentropy',
            metrics = ['accuracy'])
```

[68]:
```
history2= mobilenet_model.fit(X_train, y_train,batch_size = 4, epochs = 100,␣
↪validation_data = (X_val, y_val),
```

```
                                                    callbacks=callbacks_list2,␣
↪class_weight=class_weights)
```

```
Epoch 1/100
1264/1264 [==============================] - 45s 34ms/step - loss: 1.0274 -
accuracy: 0.6334 - val_loss: 0.4062 - val_accuracy: 0.8465

Epoch 00001: val_accuracy improved from -inf to 0.84652, saving model to
weights.best_mobile_net2.hdf5
Epoch 2/100
1264/1264 [==============================] - 51s 40ms/step - loss: 0.4607 -
accuracy: 0.8346 - val_loss: 0.3407 - val_accuracy: 0.8758

Epoch 00002: val_accuracy improved from 0.84652 to 0.87579, saving model to
weights.best_mobile_net2.hdf5
Epoch 3/100
1264/1264 [==============================] - 45s 35ms/step - loss: 0.3998 -
accuracy: 0.8558 - val_loss: 0.3144 - val_accuracy: 0.8869

Epoch 00003: val_accuracy improved from 0.87579 to 0.88687, saving model to
weights.best_mobile_net2.hdf5
Epoch 4/100
1264/1264 [==============================] - 51s 41ms/step - loss: 0.3252 -
accuracy: 0.8820 - val_loss: 0.2843 - val_accuracy: 0.8900

Epoch 00004: val_accuracy improved from 0.88687 to 0.89003, saving model to
weights.best_mobile_net2.hdf5
Epoch 5/100
1264/1264 [==============================] - 46s 37ms/step - loss: 0.2957 -
accuracy: 0.8899 - val_loss: 0.2625 - val_accuracy: 0.8972

Epoch 00005: val_accuracy improved from 0.89003 to 0.89715, saving model to
weights.best_mobile_net2.hdf5
Epoch 6/100
1264/1264 [==============================] - 50s 39ms/step - loss: 0.2913 -
accuracy: 0.8968 - val_loss: 0.2514 - val_accuracy: 0.9066

Epoch 00006: val_accuracy improved from 0.89715 to 0.90665, saving model to
weights.best_mobile_net2.hdf5
Epoch 7/100
1264/1264 [==============================] - 51s 41ms/step - loss: 0.2449 -
accuracy: 0.9137 - val_loss: 0.2765 - val_accuracy: 0.8995

Epoch 00007: val_accuracy did not improve from 0.90665
Epoch 8/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.2381 -
accuracy: 0.9114 - val_loss: 0.2510 - val_accuracy: 0.9027
```

```
Epoch 00008: val_accuracy did not improve from 0.90665
Epoch 9/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.2349 -
accuracy: 0.9140 - val_loss: 0.2349 - val_accuracy: 0.9169

Epoch 00009: val_accuracy improved from 0.90665 to 0.91693, saving model to
weights.best_mobile_net2.hdf5
Epoch 10/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.2201 -
accuracy: 0.9169 - val_loss: 0.2089 - val_accuracy: 0.9185

Epoch 00010: val_accuracy improved from 0.91693 to 0.91851, saving model to
weights.best_mobile_net2.hdf5
Epoch 11/100
1264/1264 [==============================] - 54s 42ms/step - loss: 0.2120 -
accuracy: 0.9229 - val_loss: 0.2056 - val_accuracy: 0.9241

Epoch 00011: val_accuracy improved from 0.91851 to 0.92405, saving model to
weights.best_mobile_net2.hdf5
Epoch 12/100
1264/1264 [==============================] - 51s 40ms/step - loss: 0.1824 -
accuracy: 0.9351 - val_loss: 0.1983 - val_accuracy: 0.9256

Epoch 00012: val_accuracy improved from 0.92405 to 0.92563, saving model to
weights.best_mobile_net2.hdf5
Epoch 13/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.1778 -
accuracy: 0.9360 - val_loss: 0.2047 - val_accuracy: 0.9248

Epoch 00013: val_accuracy did not improve from 0.92563
Epoch 14/100
1264/1264 [==============================] - 46s 37ms/step - loss: 0.1697 -
accuracy: 0.9336 - val_loss: 0.2089 - val_accuracy: 0.9209

Epoch 00014: val_accuracy did not improve from 0.92563
Epoch 15/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.1455 -
accuracy: 0.9521 - val_loss: 0.2095 - val_accuracy: 0.9209

Epoch 00015: val_accuracy did not improve from 0.92563
Epoch 16/100
1264/1264 [==============================] - 45s 36ms/step - loss: 0.1423 -
accuracy: 0.9460 - val_loss: 0.1874 - val_accuracy: 0.9280

Epoch 00016: val_accuracy improved from 0.92563 to 0.92801, saving model to
weights.best_mobile_net2.hdf5
Epoch 17/100
1264/1264 [==============================] - 55s 44ms/step - loss: 0.1384 -
```

```
accuracy: 0.9520 - val_loss: 0.2212 - val_accuracy: 0.9217

Epoch 00017: val_accuracy did not improve from 0.92801
Epoch 18/100
1264/1264 [==============================] - 45s 35ms/step - loss: 0.1382 -
accuracy: 0.9542 - val_loss: 0.1896 - val_accuracy: 0.9335

Epoch 00018: val_accuracy improved from 0.92801 to 0.93354, saving model to
weights.best_mobile_net2.hdf5
Epoch 19/100
1264/1264 [==============================] - 54s 43ms/step - loss: 0.1333 -
accuracy: 0.9505 - val_loss: 0.2098 - val_accuracy: 0.9280

Epoch 00019: val_accuracy did not improve from 0.93354
Epoch 20/100
1264/1264 [==============================] - 45s 36ms/step - loss: 0.1127 -
accuracy: 0.9615 - val_loss: 0.1812 - val_accuracy: 0.9343

Epoch 00020: val_accuracy improved from 0.93354 to 0.93434, saving model to
weights.best_mobile_net2.hdf5
Epoch 21/100
1264/1264 [==============================] - 54s 43ms/step - loss: 0.1194 -
accuracy: 0.9510 - val_loss: 0.1663 - val_accuracy: 0.9422

Epoch 00021: val_accuracy improved from 0.93434 to 0.94225, saving model to
weights.best_mobile_net2.hdf5
Epoch 22/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0993 -
accuracy: 0.9684 - val_loss: 0.1807 - val_accuracy: 0.9351

Epoch 00022: val_accuracy did not improve from 0.94225
Epoch 23/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.0997 -
accuracy: 0.9627 - val_loss: 0.1834 - val_accuracy: 0.9359

Epoch 00023: val_accuracy did not improve from 0.94225
Epoch 24/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0973 -
accuracy: 0.9633 - val_loss: 0.1689 - val_accuracy: 0.9391

Epoch 00024: val_accuracy did not improve from 0.94225
Epoch 25/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0910 -
accuracy: 0.9705 - val_loss: 0.1697 - val_accuracy: 0.9415

Epoch 00025: val_accuracy did not improve from 0.94225
Epoch 26/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0895 -
```

```
accuracy: 0.9711 - val_loss: 0.1555 - val_accuracy: 0.9422

Epoch 00026: val_accuracy did not improve from 0.94225
Epoch 27/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0840 -
accuracy: 0.9717 - val_loss: 0.1763 - val_accuracy: 0.9367

Epoch 00027: val_accuracy did not improve from 0.94225
Epoch 28/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0896 -
accuracy: 0.9706 - val_loss: 0.1640 - val_accuracy: 0.9430

Epoch 00028: val_accuracy improved from 0.94225 to 0.94304, saving model to
weights.best_mobile_net2.hdf5
Epoch 29/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.0724 -
accuracy: 0.9752 - val_loss: 0.1752 - val_accuracy: 0.9383

Epoch 00029: val_accuracy did not improve from 0.94304
Epoch 30/100
1264/1264 [==============================] - 49s 39ms/step - loss: 0.0664 -
accuracy: 0.9773 - val_loss: 0.1681 - val_accuracy: 0.9391

Epoch 00030: val_accuracy did not improve from 0.94304
Epoch 31/100
1264/1264 [==============================] - 51s 40ms/step - loss: 0.0707 -
accuracy: 0.9756 - val_loss: 0.1554 - val_accuracy: 0.9462

Epoch 00031: val_accuracy improved from 0.94304 to 0.94620, saving model to
weights.best_mobile_net2.hdf5
Epoch 32/100
1264/1264 [==============================] - 51s 40ms/step - loss: 0.0684 -
accuracy: 0.9778 - val_loss: 0.1549 - val_accuracy: 0.9438

Epoch 00032: val_accuracy did not improve from 0.94620
Epoch 33/100
1264/1264 [==============================] - 49s 39ms/step - loss: 0.0657 -
accuracy: 0.9787 - val_loss: 0.1662 - val_accuracy: 0.9367

Epoch 00033: val_accuracy did not improve from 0.94620
Epoch 34/100
1264/1264 [==============================] - 50s 39ms/step - loss: 0.0644 -
accuracy: 0.9807 - val_loss: 0.1603 - val_accuracy: 0.9478

Epoch 00034: val_accuracy improved from 0.94620 to 0.94778, saving model to
weights.best_mobile_net2.hdf5
Epoch 35/100
1264/1264 [==============================] - 45s 36ms/step - loss: 0.0720 -
```

```
accuracy: 0.9748 - val_loss: 0.1577 - val_accuracy: 0.9486

Epoch 00035: val_accuracy improved from 0.94778 to 0.94858, saving model to
weights.best_mobile_net2.hdf5
Epoch 36/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0490 -
accuracy: 0.9846 - val_loss: 0.1547 - val_accuracy: 0.9470

Epoch 00036: val_accuracy did not improve from 0.94858
Epoch 37/100
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0555 -
accuracy: 0.9821 - val_loss: 0.1580 - val_accuracy: 0.9430

Epoch 00037: val_accuracy did not improve from 0.94858
Epoch 38/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.0466 -
accuracy: 0.9867 - val_loss: 0.1595 - val_accuracy: 0.9502

Epoch 00038: val_accuracy improved from 0.94858 to 0.95016, saving model to
weights.best_mobile_net2.hdf5
Epoch 39/100
1264/1264 [==============================] - 45s 35ms/step - loss: 0.0509 -
accuracy: 0.9825 - val_loss: 0.1641 - val_accuracy: 0.9446

Epoch 00039: val_accuracy did not improve from 0.95016
Epoch 40/100
1264/1264 [==============================] - 58s 46ms/step - loss: 0.0464 -
accuracy: 0.9832 - val_loss: 0.1794 - val_accuracy: 0.9446

Epoch 00040: val_accuracy did not improve from 0.95016
Epoch 41/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0528 -
accuracy: 0.9837 - val_loss: 0.1550 - val_accuracy: 0.9407

Epoch 00041: val_accuracy did not improve from 0.95016
Epoch 42/100
1264/1264 [==============================] - 63s 50ms/step - loss: 0.0436 -
accuracy: 0.9878 - val_loss: 0.1534 - val_accuracy: 0.9438

Epoch 00042: val_accuracy did not improve from 0.95016
Epoch 43/100
1264/1264 [==============================] - 55s 44ms/step - loss: 0.0456 -
accuracy: 0.9832 - val_loss: 0.1671 - val_accuracy: 0.9470

Epoch 00043: val_accuracy did not improve from 0.95016
Epoch 44/100
1264/1264 [==============================] - 59s 47ms/step - loss: 0.0389 -
accuracy: 0.9876 - val_loss: 0.1746 - val_accuracy: 0.9383
```

```
Epoch 00044: val_accuracy did not improve from 0.95016
Epoch 45/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0428 -
accuracy: 0.9862 - val_loss: 0.1810 - val_accuracy: 0.9438

Epoch 00045: val_accuracy did not improve from 0.95016
Epoch 46/100
1264/1264 [==============================] - 59s 46ms/step - loss: 0.0376 -
accuracy: 0.9867 - val_loss: 0.1799 - val_accuracy: 0.9454

Epoch 00046: val_accuracy did not improve from 0.95016
Epoch 47/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0415 -
accuracy: 0.9864 - val_loss: 0.1677 - val_accuracy: 0.9509

Epoch 00047: val_accuracy improved from 0.95016 to 0.95095, saving model to
weights.best_mobile_net2.hdf5
Epoch 48/100
1264/1264 [==============================] - 60s 47ms/step - loss: 0.0322 -
accuracy: 0.9903 - val_loss: 0.1650 - val_accuracy: 0.9462

Epoch 00048: val_accuracy did not improve from 0.95095
Epoch 49/100
1264/1264 [==============================] - 45s 36ms/step - loss: 0.0320 -
accuracy: 0.9878 - val_loss: 0.1719 - val_accuracy: 0.9470

Epoch 00049: val_accuracy did not improve from 0.95095
Epoch 50/100
1264/1264 [==============================] - 57s 45ms/step - loss: 0.0357 -
accuracy: 0.9896 - val_loss: 0.1647 - val_accuracy: 0.9486

Epoch 00050: val_accuracy did not improve from 0.95095
Epoch 51/100
1264/1264 [==============================] - 46s 37ms/step - loss: 0.0354 -
accuracy: 0.9893 - val_loss: 0.1769 - val_accuracy: 0.9486

Epoch 00051: val_accuracy did not improve from 0.95095
Epoch 52/100
1264/1264 [==============================] - 57s 45ms/step - loss: 0.0294 -
accuracy: 0.9918 - val_loss: 0.1645 - val_accuracy: 0.9517

Epoch 00052: val_accuracy improved from 0.95095 to 0.95174, saving model to
weights.best_mobile_net2.hdf5
Epoch 53/100
1264/1264 [==============================] - 51s 41ms/step - loss: 0.0384 -
accuracy: 0.9882 - val_loss: 0.1771 - val_accuracy: 0.9462
```

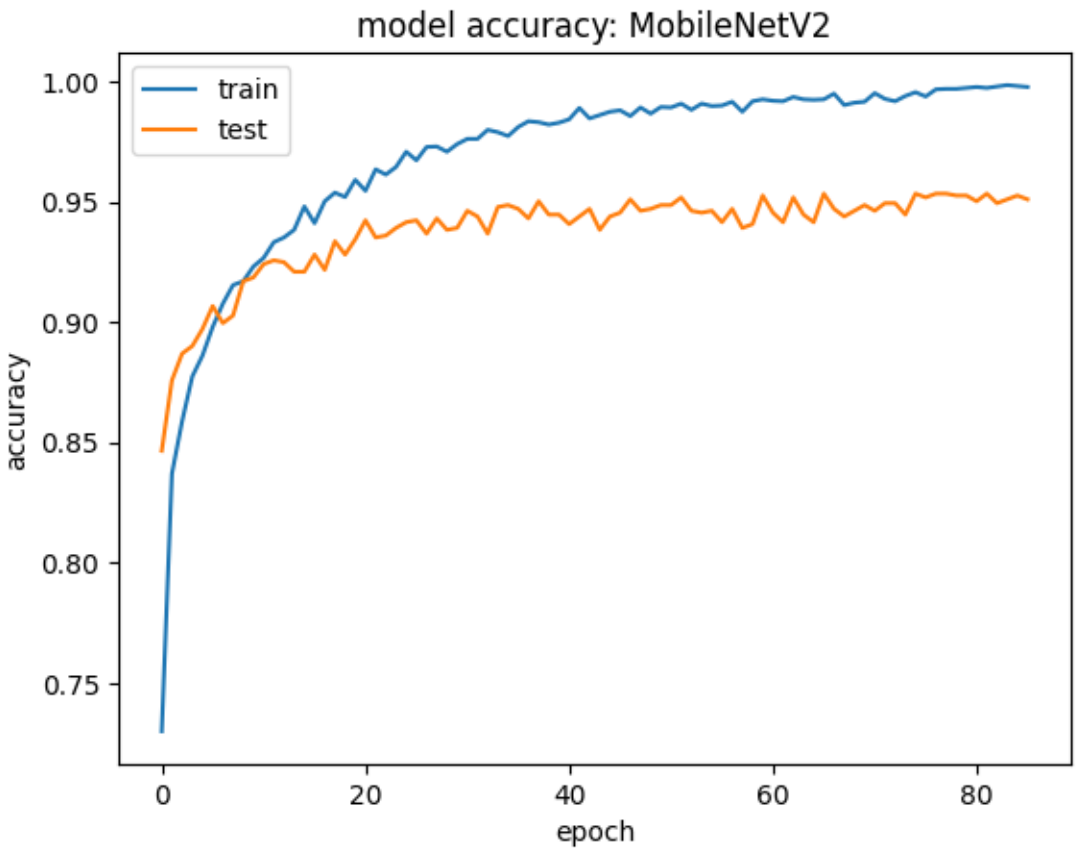
```
Epoch 00053: val_accuracy did not improve from 0.95174
Epoch 54/100
1264/1264 [==============================] - 62s 49ms/step - loss: 0.0234 -
accuracy: 0.9930 - val_loss: 0.1723 - val_accuracy: 0.9454

Epoch 00054: val_accuracy did not improve from 0.95174
Epoch 55/100
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0309 -
accuracy: 0.9908 - val_loss: 0.1734 - val_accuracy: 0.9462

Epoch 00055: val_accuracy did not improve from 0.95174
Epoch 56/100
1264/1264 [==============================] - 62s 49ms/step - loss: 0.0256 -
accuracy: 0.9928 - val_loss: 0.1782 - val_accuracy: 0.9415

Epoch 00056: val_accuracy did not improve from 0.95174
Epoch 57/100
1264/1264 [==============================] - 50s 39ms/step - loss: 0.0248 -
accuracy: 0.9916 - val_loss: 0.1682 - val_accuracy: 0.9470

Epoch 00057: val_accuracy did not improve from 0.95174
Epoch 58/100
1264/1264 [==============================] - 58s 46ms/step - loss: 0.0330 -
accuracy: 0.9873 - val_loss: 0.1941 - val_accuracy: 0.9391

Epoch 00058: val_accuracy did not improve from 0.95174
Epoch 59/100
1264/1264 [==============================] - 49s 39ms/step - loss: 0.0314 -
accuracy: 0.9902 - val_loss: 0.1728 - val_accuracy: 0.9407

Epoch 00059: val_accuracy did not improve from 0.95174
Epoch 60/100
1264/1264 [==============================] - 60s 48ms/step - loss: 0.0242 -
accuracy: 0.9919 - val_loss: 0.1519 - val_accuracy: 0.9525

Epoch 00060: val_accuracy improved from 0.95174 to 0.95253, saving model to
weights.best_mobile_net2.hdf5
Epoch 61/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0232 -
accuracy: 0.9928 - val_loss: 0.1915 - val_accuracy: 0.9454

Epoch 00061: val_accuracy did not improve from 0.95253
Epoch 62/100
1264/1264 [==============================] - 59s 47ms/step - loss: 0.0298 -
accuracy: 0.9899 - val_loss: 0.2006 - val_accuracy: 0.9415

Epoch 00062: val_accuracy did not improve from 0.95253
Epoch 63/100
```

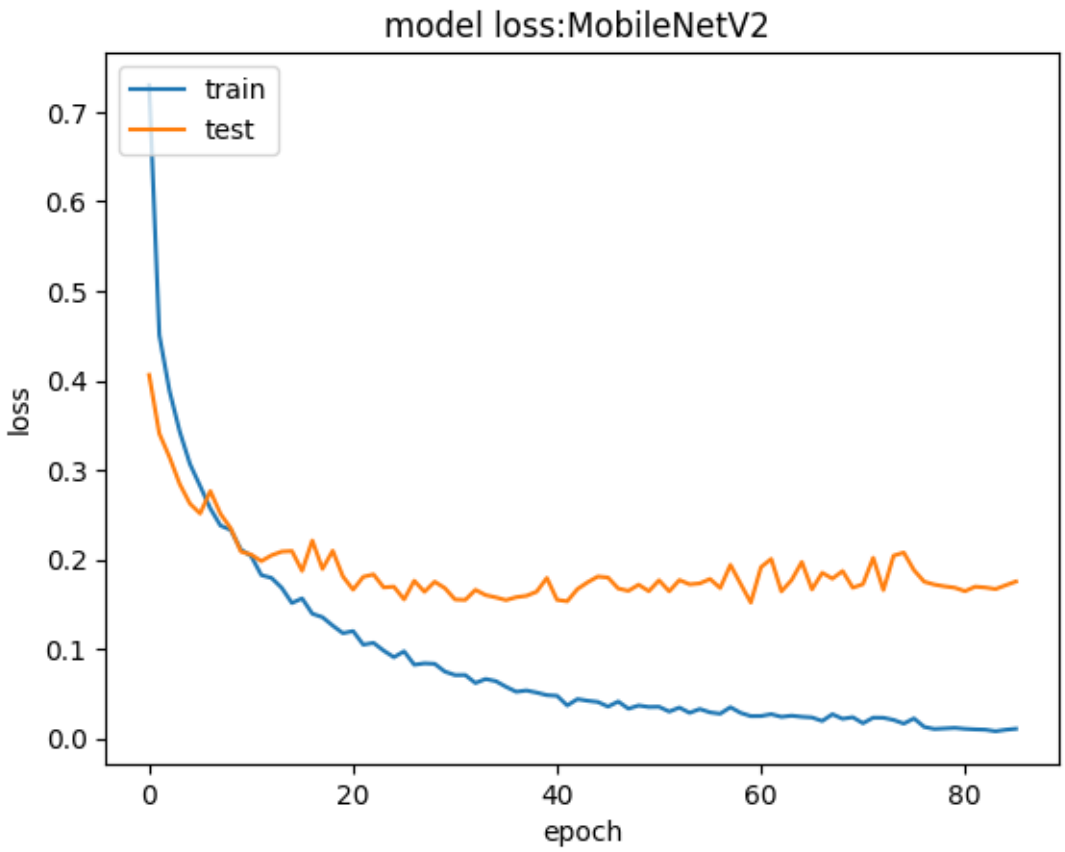
```
1264/1264 [==============================] - 49s 39ms/step - loss: 0.0229 -
accuracy: 0.9941 - val_loss: 0.1645 - val_accuracy: 0.9517

Epoch 00063: val_accuracy did not improve from 0.95253
Epoch 64/100
1264/1264 [==============================] - 60s 47ms/step - loss: 0.0211 -
accuracy: 0.9944 - val_loss: 0.1771 - val_accuracy: 0.9446

Epoch 00064: val_accuracy did not improve from 0.95253
Epoch 65/100
1264/1264 [==============================] - 50s 39ms/step - loss: 0.0226 -
accuracy: 0.9930 - val_loss: 0.1973 - val_accuracy: 0.9415

Epoch 00065: val_accuracy did not improve from 0.95253
Epoch 66/100
1264/1264 [==============================] - 61s 48ms/step - loss: 0.0178 -
accuracy: 0.9929 - val_loss: 0.1667 - val_accuracy: 0.9533

Epoch 00066: val_accuracy improved from 0.95253 to 0.95332, saving model to
weights.best_mobile_net2.hdf5
Epoch 67/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0194 -
accuracy: 0.9945 - val_loss: 0.1852 - val_accuracy: 0.9470

Epoch 00067: val_accuracy did not improve from 0.95332
Epoch 68/100
1264/1264 [==============================] - 56s 44ms/step - loss: 0.0223 -
accuracy: 0.9916 - val_loss: 0.1787 - val_accuracy: 0.9438

Epoch 00068: val_accuracy did not improve from 0.95332
Epoch 69/100
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0193 -
accuracy: 0.9918 - val_loss: 0.1871 - val_accuracy: 0.9462

Epoch 00069: val_accuracy did not improve from 0.95332
Epoch 70/100
1264/1264 [==============================] - 51s 40ms/step - loss: 0.0256 -
accuracy: 0.9908 - val_loss: 0.1686 - val_accuracy: 0.9486

Epoch 00070: val_accuracy did not improve from 0.95332
Epoch 71/100
1264/1264 [==============================] - 45s 36ms/step - loss: 0.0161 -
accuracy: 0.9947 - val_loss: 0.1725 - val_accuracy: 0.9462

Epoch 00071: val_accuracy did not improve from 0.95332
Epoch 72/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0265 -
accuracy: 0.9927 - val_loss: 0.2020 - val_accuracy: 0.9494
```

```
Epoch 00072: val_accuracy did not improve from 0.95332
Epoch 73/100
1264/1264 [==============================] - 46s 37ms/step - loss: 0.0220 -
accuracy: 0.9929 - val_loss: 0.1659 - val_accuracy: 0.9494

Epoch 00073: val_accuracy did not improve from 0.95332
Epoch 74/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.0212 -
accuracy: 0.9944 - val_loss: 0.2043 - val_accuracy: 0.9446

Epoch 00074: val_accuracy did not improve from 0.95332
Epoch 75/100
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0166 -
accuracy: 0.9949 - val_loss: 0.2078 - val_accuracy: 0.9533

Epoch 00075: val_accuracy did not improve from 0.95332
Epoch 76/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0254 -
accuracy: 0.9920 - val_loss: 0.1881 - val_accuracy: 0.9517

Epoch 00076: val_accuracy did not improve from 0.95332
Epoch 77/100
1264/1264 [==============================] - 43s 34ms/step - loss: 0.0142 -
accuracy: 0.9970 - val_loss: 0.1753 - val_accuracy: 0.9533

Epoch 00077: val_accuracy did not improve from 0.95332
Epoch 78/100
1264/1264 [==============================] - 54s 43ms/step - loss: 0.0094 -
accuracy: 0.9973 - val_loss: 0.1721 - val_accuracy: 0.9533

Epoch 00078: val_accuracy did not improve from 0.95332
Epoch 79/100
1264/1264 [==============================] - 44s 35ms/step - loss: 0.0098 -
accuracy: 0.9975 - val_loss: 0.1701 - val_accuracy: 0.9525

Epoch 00079: val_accuracy did not improve from 0.95332
Epoch 80/100
1264/1264 [==============================] - 52s 41ms/step - loss: 0.0115 -
accuracy: 0.9975 - val_loss: 0.1686 - val_accuracy: 0.9525

Epoch 00080: val_accuracy did not improve from 0.95332
Epoch 81/100
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0114 -
accuracy: 0.9975 - val_loss: 0.1647 - val_accuracy: 0.9502

Epoch 00081: val_accuracy did not improve from 0.95332
Epoch 82/100
```

```
1264/1264 [==============================] - 55s 43ms/step - loss: 0.0099 -
accuracy: 0.9969 - val_loss: 0.1697 - val_accuracy: 0.9533

Epoch 00082: val_accuracy did not improve from 0.95332
Epoch 83/100
1264/1264 [==============================] - 53s 42ms/step - loss: 0.0102 -
accuracy: 0.9982 - val_loss: 0.1688 - val_accuracy: 0.9494

Epoch 00083: val_accuracy did not improve from 0.95332
Epoch 84/100
1264/1264 [==============================] - 54s 43ms/step - loss: 0.0069 -
accuracy: 0.9983 - val_loss: 0.1670 - val_accuracy: 0.9509

Epoch 00084: val_accuracy did not improve from 0.95332
Epoch 85/100
1264/1264 [==============================] - 47s 38ms/step - loss: 0.0095 -
accuracy: 0.9978 - val_loss: 0.1714 - val_accuracy: 0.9525

Epoch 00085: val_accuracy did not improve from 0.95332
Epoch 86/100
1264/1264 [==============================] - 54s 43ms/step - loss: 0.0102 -
accuracy: 0.9977 - val_loss: 0.1755 - val_accuracy: 0.9509

Epoch 00086: val_accuracy did not improve from 0.95332
```

[69]:
```python
# summarize history for accuracy
plt.plot(history2.history['accuracy'])
plt.plot(history2.history['val_accuracy'])
plt.title('model accuracy: MobileNetV2')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history2.history['loss'])
plt.plot(history2.history['val_loss'])
plt.title('model loss:MobileNetV2 ')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

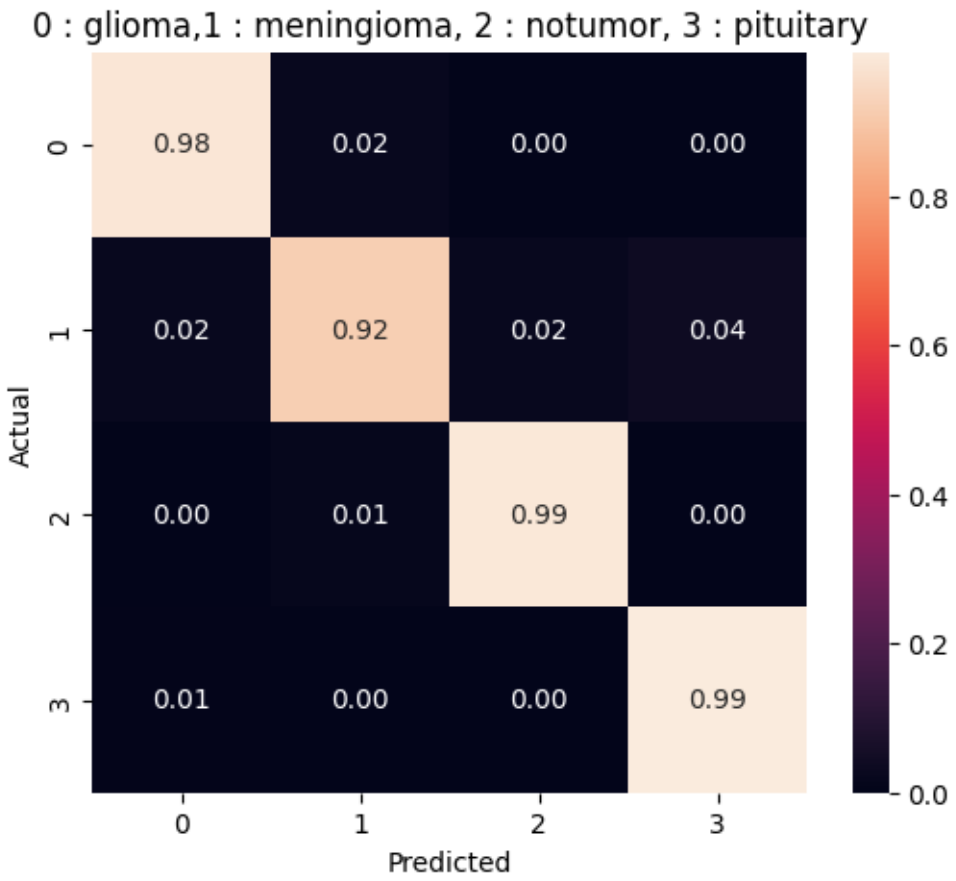

model accuracy: MobileNetV2
train
test
1.00
0.95
0.90
0.85
0.80
0.75
accuracy
0
20
40
60
80
epoch



[104]:
```
mobilenet_model=load_model('weights.best_mobile_net2.hdf5')
```

[107]:
```
pred2=mobilenet_model.predict(X_test)
```

[71]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(Y_test, Pred)
# Normalise
cmn = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(cmn, annot=True, fmt='.2f')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('0 : glioma,1 : meningioma, 2 : notumor, 3 : pituitary')
plt.show(block=False)
```



[72]:
```
from keras import models
from numpy import loadtxt
from tensorflow.keras.models import save_model
save_model(cnn, "Mobilenetv2.h5")
# load and evaluate a saved model
loaded_model = models.load_model('Mobilenetv2.h5')
# summarize model.
model=loaded_model
train_pred_p=model.predict(X_train)
train_pred = np.argmax(train_pred_p, axis=1)
```

5 C: DenseNet169

[73]:
```
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
```

[74]:
```
from tensorflow.keras.applications import DenseNet169
```

[78]:
```
def make_densenet_model(image_size, num_classes):

    input_shape = image_size

    base_model = tf.keras.applications.DenseNet169(input_shape=input_shape,
                                                   include_top=False, # Do not␣
 ↪include the dense prediction layer
                                                   weights="imagenet") # Load␣
 ↪imageNet parameters

    # Freeze the base model by making it non trainable
    base_model.trainable = False

    # create the input layer (Same as the imageNetv2 input size)
    inputs = tf.keras.Input(shape=input_shape)

    # apply data augmentation to the inputs
    x = inputs

    # set training to False to avoid keeping track of statistics in the batch␣
 ↪norm layer
    x = base_model(x, training=False)

    # Add the new Binary classification layers
    # use global avg pooling to summarize the info in each channel
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    #include dropout with probability of 0.2 to avoid overfitting
    x = Dropout(0.3)(x)
    x = Flatten()(x)
    # Fully Connected Layers
    x =Dense(128, activation = 'relu')(x)

    prediction_layer = Dense(4, activation='softmax')

    outputs = prediction_layer(x)

    model = keras.models.Model(inputs, outputs)

    return model
```

[79]:
```
image_size = (100,100,3)
densenet_model = make_densenet_model(image_size, num_classes = 2)

# Preview the Model Summary
densenet_model.summary()
```

```
Model: "model_4"
_________________________________________________________________
```

```
Layer (type)                 Output Shape              Param #
=================================================================
input_8 (InputLayer)         [(None, 100, 100, 3)]     0
_________________________________________________________________
densenet169 (Functional)     (None, 3, 3, 1664)        12642880
_________________________________________________________________
global_average_pooling2d_2 ( (None, 1664)              0
_________________________________________________________________
dropout_8 (Dropout)          (None, 1664)              0
_________________________________________________________________
flatten_4 (Flatten)          (None, 1664)              0
_________________________________________________________________
dense_8 (Dense)              (None, 128)               213120
_________________________________________________________________
dense_9 (Dense)              (None, 4)                 516
=================================================================
Total params: 12,856,516
Trainable params: 213,636
Non-trainable params: 12,642,880
_________________________________________________________________
```

[80]:
```
filepath31="weights.best_densenet1692.hdf5"
checkpoint3 = ModelCheckpoint(filepath31, monitor='val_accuracy', verbose=1,␣
 ↪save_best_only=True, mode='max')
es3 = EarlyStopping(monitor='val_accuracy', patience=20)
rlrop3 = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=10)
callbacks_list3 = [checkpoint3,es3,rlrop3]
```

[81]:
```
densenet_model.compile(optimizer =Adam(learning_rate=0.0001), loss =␣
 ↪'categorical_crossentropy',
           metrics = ['accuracy'])
```

[84]:
```
history3 = densenet_model.fit(X_train, y_train,epochs = 100,verbose =␣
 ↪1,batch_size=2,validation_data =(X_val,y_val)
                   ,callbacks=callbacks_list3,class_weight=class_weights)
```

```
Epoch 1/100
2528/2528 [==============================] - 229s 90ms/step - loss: 0.4918 -
accuracy: 0.8218 - val_loss: 0.3327 - val_accuracy: 0.8774

Epoch 00001: val_accuracy improved from -inf to 0.87737, saving model to
weights.best_densenet1692.hdf5
Epoch 2/100
2528/2528 [==============================] - 243s 96ms/step - loss: 0.3994 -
accuracy: 0.8513 - val_loss: 0.3493 - val_accuracy: 0.8608

Epoch 00002: val_accuracy did not improve from 0.87737
Epoch 3/100
```

```
2528/2528 [==============================] - 245s 97ms/step - loss: 0.3589 -
accuracy: 0.8645 - val_loss: 0.2915 - val_accuracy: 0.8837

Epoch 00003: val_accuracy improved from 0.87737 to 0.88370, saving model to
weights.best_densenet1692.hdf5
Epoch 4/100
2528/2528 [==============================] - 234s 92ms/step - loss: 0.3055 -
accuracy: 0.8859 - val_loss: 0.2871 - val_accuracy: 0.8979

Epoch 00004: val_accuracy improved from 0.88370 to 0.89794, saving model to
weights.best_densenet1692.hdf5
Epoch 5/100
2528/2528 [==============================] - 237s 94ms/step - loss: 0.2906 -
accuracy: 0.8928 - val_loss: 0.2345 - val_accuracy: 0.9122

Epoch 00005: val_accuracy improved from 0.89794 to 0.91218, saving model to
weights.best_densenet1692.hdf5
Epoch 6/100
2528/2528 [==============================] - 240s 95ms/step - loss: 0.2770 -
accuracy: 0.8995 - val_loss: 0.2601 - val_accuracy: 0.8956

Epoch 00006: val_accuracy did not improve from 0.91218
Epoch 7/100
2528/2528 [==============================] - 239s 95ms/step - loss: 0.2579 -
accuracy: 0.9059 - val_loss: 0.2303 - val_accuracy: 0.9177

Epoch 00007: val_accuracy improved from 0.91218 to 0.91772, saving model to
weights.best_densenet1692.hdf5
Epoch 8/100
2528/2528 [==============================] - 238s 94ms/step - loss: 0.2366 -
accuracy: 0.9098 - val_loss: 0.2164 - val_accuracy: 0.9201

Epoch 00008: val_accuracy improved from 0.91772 to 0.92009, saving model to
weights.best_densenet1692.hdf5
Epoch 9/100
2528/2528 [==============================] - 237s 94ms/step - loss: 0.2216 -
accuracy: 0.9189 - val_loss: 0.2234 - val_accuracy: 0.9209

Epoch 00009: val_accuracy improved from 0.92009 to 0.92089, saving model to
weights.best_densenet1692.hdf5
Epoch 10/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.2014 -
accuracy: 0.9266 - val_loss: 0.2410 - val_accuracy: 0.9090

Epoch 00010: val_accuracy did not improve from 0.92089
Epoch 11/100
2528/2528 [==============================] - 242s 96ms/step - loss: 0.1896 -
accuracy: 0.9326 - val_loss: 0.1794 - val_accuracy: 0.9320
```

```
Epoch 00011: val_accuracy improved from 0.92089 to 0.93196, saving model to
weights.best_densenet1692.hdf5
Epoch 12/100
2528/2528 [==============================] - 236s 94ms/step - loss: 0.1774 -
accuracy: 0.9330 - val_loss: 0.1971 - val_accuracy: 0.9248

Epoch 00012: val_accuracy did not improve from 0.93196
Epoch 13/100
2528/2528 [==============================] - 252s 100ms/step - loss: 0.1725 -
accuracy: 0.9381 - val_loss: 0.1816 - val_accuracy: 0.9335

Epoch 00013: val_accuracy improved from 0.93196 to 0.93354, saving model to
weights.best_densenet1692.hdf5
Epoch 14/100
2528/2528 [==============================] - 241s 95ms/step - loss: 0.1549 -
accuracy: 0.9450 - val_loss: 0.1838 - val_accuracy: 0.9280

Epoch 00014: val_accuracy did not improve from 0.93354
Epoch 15/100
2528/2528 [==============================] - 238s 94ms/step - loss: 0.1542 -
accuracy: 0.9417 - val_loss: 0.1878 - val_accuracy: 0.9328

Epoch 00015: val_accuracy did not improve from 0.93354
Epoch 16/100
2528/2528 [==============================] - 247s 98ms/step - loss: 0.1465 -
accuracy: 0.9436 - val_loss: 0.1724 - val_accuracy: 0.9399

Epoch 00016: val_accuracy improved from 0.93354 to 0.93987, saving model to
weights.best_densenet1692.hdf5
Epoch 17/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.1382 -
accuracy: 0.9506 - val_loss: 0.1641 - val_accuracy: 0.9478

Epoch 00017: val_accuracy improved from 0.93987 to 0.94778, saving model to
weights.best_densenet1692.hdf5
Epoch 18/100
2528/2528 [==============================] - 237s 94ms/step - loss: 0.1274 -
accuracy: 0.9551 - val_loss: 0.1755 - val_accuracy: 0.9407

Epoch 00018: val_accuracy did not improve from 0.94778
Epoch 19/100
2528/2528 [==============================] - 250s 99ms/step - loss: 0.1275 -
accuracy: 0.9508 - val_loss: 0.1662 - val_accuracy: 0.9438

Epoch 00019: val_accuracy did not improve from 0.94778
Epoch 20/100
2528/2528 [==============================] - 242s 96ms/step - loss: 0.1162 -
```

```
accuracy: 0.9583 - val_loss: 0.1710 - val_accuracy: 0.9391

Epoch 00020: val_accuracy did not improve from 0.94778
Epoch 21/100
2528/2528 [==============================] - 243s 96ms/step - loss: 0.1076 -
accuracy: 0.9612 - val_loss: 0.1581 - val_accuracy: 0.9470

Epoch 00021: val_accuracy did not improve from 0.94778
Epoch 22/100
2528/2528 [==============================] - 237s 94ms/step - loss: 0.1130 -
accuracy: 0.9604 - val_loss: 0.1469 - val_accuracy: 0.9494

Epoch 00022: val_accuracy improved from 0.94778 to 0.94937, saving model to
weights.best_densenet1692.hdf5
Epoch 23/100
2528/2528 [==============================] - 413s 163ms/step - loss: 0.1035 -
accuracy: 0.9604 - val_loss: 0.1733 - val_accuracy: 0.9407

Epoch 00023: val_accuracy did not improve from 0.94937
Epoch 24/100
2528/2528 [==============================] - 228s 90ms/step - loss: 0.1017 -
accuracy: 0.9634 - val_loss: 0.1473 - val_accuracy: 0.9525

Epoch 00024: val_accuracy improved from 0.94937 to 0.95253, saving model to
weights.best_densenet1692.hdf5
Epoch 25/100
2528/2528 [==============================] - 231s 91ms/step - loss: 0.0970 -
accuracy: 0.9638 - val_loss: 0.1592 - val_accuracy: 0.9486

Epoch 00025: val_accuracy did not improve from 0.95253
Epoch 26/100
2528/2528 [==============================] - 2151s 851ms/step - loss: 0.0961 -
accuracy: 0.9646 - val_loss: 0.1486 - val_accuracy: 0.9509

Epoch 00026: val_accuracy did not improve from 0.95253
Epoch 27/100
2528/2528 [==============================] - 236s 93ms/step - loss: 0.0884 -
accuracy: 0.9688 - val_loss: 0.1726 - val_accuracy: 0.9391

Epoch 00027: val_accuracy did not improve from 0.95253
Epoch 28/100
2528/2528 [==============================] - 235s 93ms/step - loss: 0.0808 -
accuracy: 0.9709 - val_loss: 0.1569 - val_accuracy: 0.9525

Epoch 00028: val_accuracy did not improve from 0.95253
Epoch 29/100
2528/2528 [==============================] - 250s 99ms/step - loss: 0.0881 -
accuracy: 0.9686 - val_loss: 0.1449 - val_accuracy: 0.9525
```

```
Epoch 00029: val_accuracy did not improve from 0.95253
Epoch 30/100
2528/2528 [==============================] - 266s 105ms/step - loss: 0.0787 -
accuracy: 0.9713 - val_loss: 0.1647 - val_accuracy: 0.9422

Epoch 00030: val_accuracy did not improve from 0.95253
Epoch 31/100
2528/2528 [==============================] - 251s 99ms/step - loss: 0.0742 -
accuracy: 0.9749 - val_loss: 0.1606 - val_accuracy: 0.9486

Epoch 00031: val_accuracy did not improve from 0.95253
Epoch 32/100
2528/2528 [==============================] - 223s 88ms/step - loss: 0.0718 -
accuracy: 0.9759 - val_loss: 0.1532 - val_accuracy: 0.9581

Epoch 00032: val_accuracy improved from 0.95253 to 0.95807, saving model to
weights.best_densenet1692.hdf5
Epoch 33/100
2528/2528 [==============================] - 220s 87ms/step - loss: 0.0748 -
accuracy: 0.9735 - val_loss: 0.1677 - val_accuracy: 0.9438

Epoch 00033: val_accuracy did not improve from 0.95807
Epoch 34/100
2528/2528 [==============================] - 1923s 761ms/step - loss: 0.0754 -
accuracy: 0.9733 - val_loss: 0.1576 - val_accuracy: 0.9502

Epoch 00034: val_accuracy did not improve from 0.95807
Epoch 35/100
2528/2528 [==============================] - 226s 89ms/step - loss: 0.0679 -
accuracy: 0.9739 - val_loss: 0.1725 - val_accuracy: 0.9430

Epoch 00035: val_accuracy did not improve from 0.95807
Epoch 36/100
2528/2528 [==============================] - 213s 84ms/step - loss: 0.0626 -
accuracy: 0.9765 - val_loss: 0.1512 - val_accuracy: 0.9525

Epoch 00036: val_accuracy did not improve from 0.95807
Epoch 37/100
2528/2528 [==============================] - 1496s 592ms/step - loss: 0.0659 -
accuracy: 0.9759 - val_loss: 0.1593 - val_accuracy: 0.9494

Epoch 00037: val_accuracy did not improve from 0.95807
Epoch 38/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.0618 -
accuracy: 0.9775 - val_loss: 0.1812 - val_accuracy: 0.9407

Epoch 00038: val_accuracy did not improve from 0.95807
```

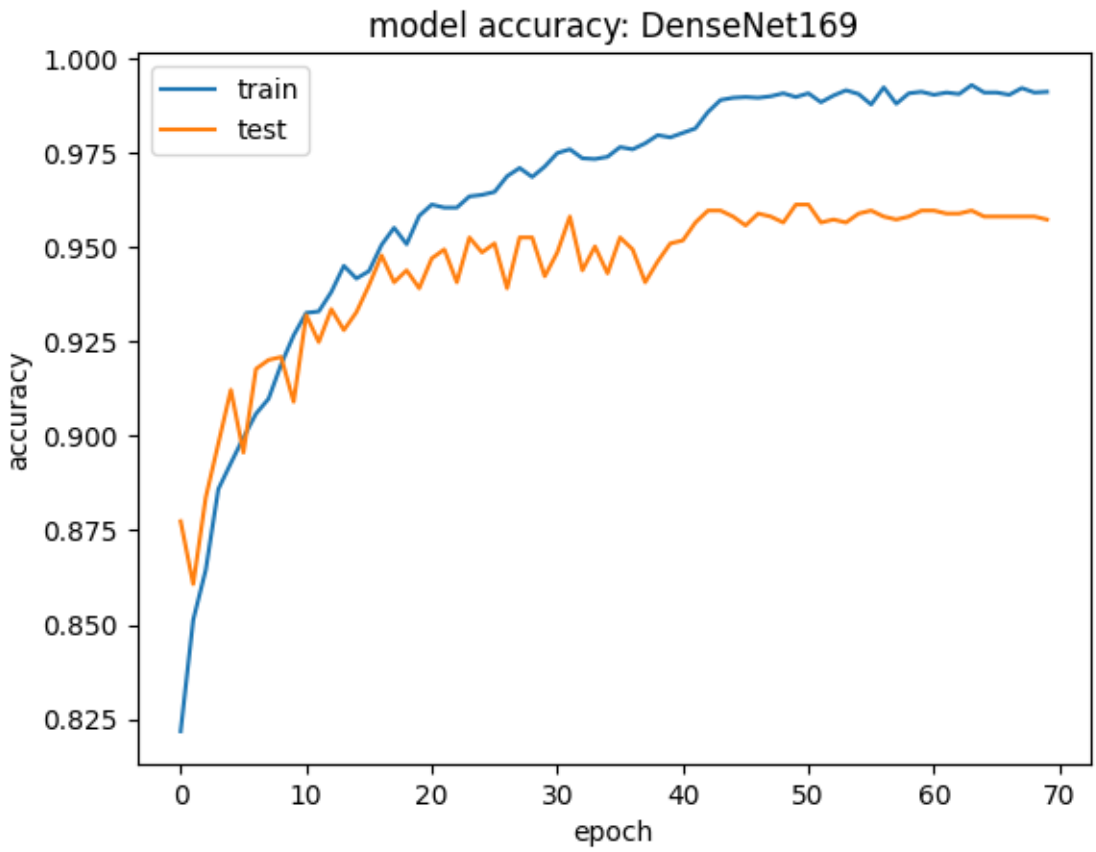
```
Epoch 39/100
2528/2528 [==============================] - 235s 93ms/step - loss: 0.0585 -
accuracy: 0.9796 - val_loss: 0.1716 - val_accuracy: 0.9462

Epoch 00039: val_accuracy did not improve from 0.95807
Epoch 40/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.0631 -
accuracy: 0.9790 - val_loss: 0.1543 - val_accuracy: 0.9509

Epoch 00040: val_accuracy did not improve from 0.95807
Epoch 41/100
2528/2528 [==============================] - 254s 100ms/step - loss: 0.0587 -
accuracy: 0.9802 - val_loss: 0.1839 - val_accuracy: 0.9517

Epoch 00041: val_accuracy did not improve from 0.95807
Epoch 42/100
2528/2528 [==============================] - 217s 86ms/step - loss: 0.0531 -
accuracy: 0.9814 - val_loss: 0.1504 - val_accuracy: 0.9565

Epoch 00042: val_accuracy did not improve from 0.95807
Epoch 43/100
2528/2528 [==============================] - 219s 87ms/step - loss: 0.0432 -
accuracy: 0.9858 - val_loss: 0.1413 - val_accuracy: 0.9597

Epoch 00043: val_accuracy improved from 0.95807 to 0.95965, saving model to
weights.best_densenet1692.hdf5
Epoch 44/100
2528/2528 [==============================] - 1558s 617ms/step - loss: 0.0335 -
accuracy: 0.9889 - val_loss: 0.1412 - val_accuracy: 0.9597

Epoch 00044: val_accuracy did not improve from 0.95965
Epoch 45/100
2528/2528 [==============================] - 253s 100ms/step - loss: 0.0336 -
accuracy: 0.9895 - val_loss: 0.1496 - val_accuracy: 0.9581

Epoch 00045: val_accuracy did not improve from 0.95965
Epoch 46/100
2528/2528 [==============================] - 234s 93ms/step - loss: 0.0317 -
accuracy: 0.9897 - val_loss: 0.1494 - val_accuracy: 0.9557

Epoch 00046: val_accuracy did not improve from 0.95965
Epoch 47/100
2528/2528 [==============================] - 262s 104ms/step - loss: 0.0355 -
accuracy: 0.9895 - val_loss: 0.1438 - val_accuracy: 0.9589

Epoch 00047: val_accuracy did not improve from 0.95965
Epoch 48/100
2528/2528 [==============================] - 246s 97ms/step - loss: 0.0330 -
```

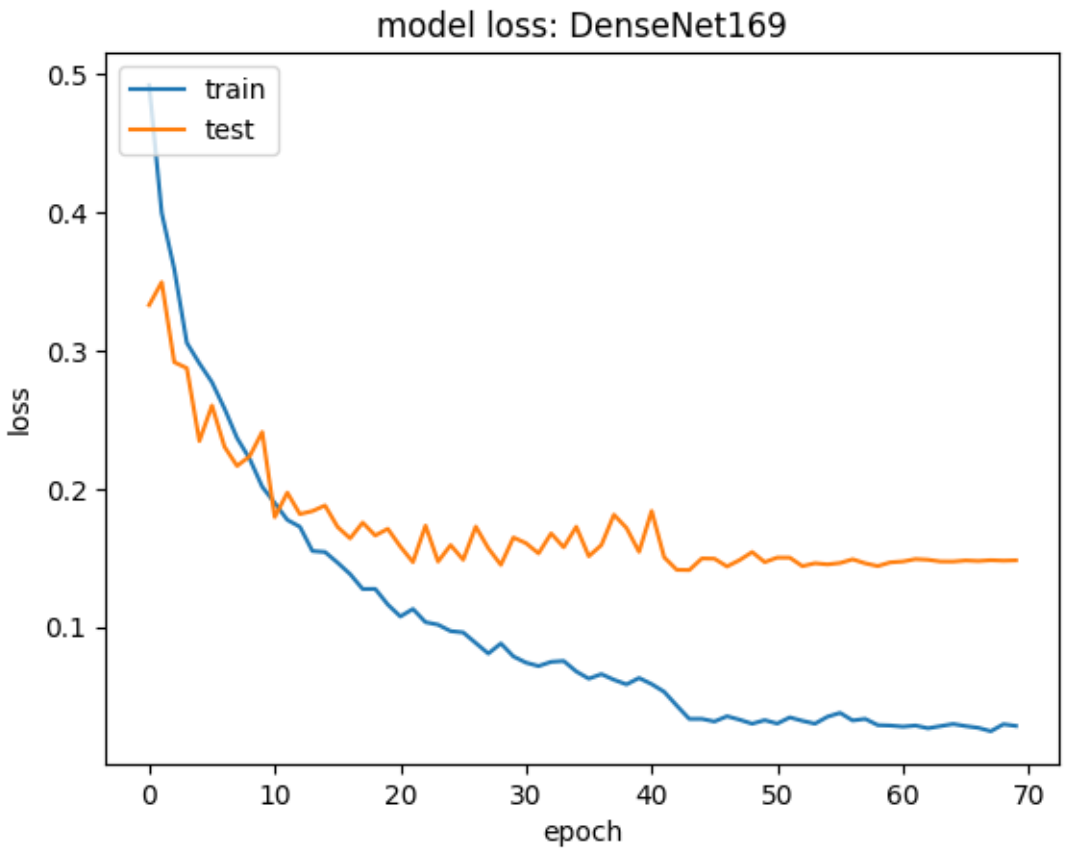
```
accuracy: 0.9899 - val_loss: 0.1484 - val_accuracy: 0.9581

Epoch 00048: val_accuracy did not improve from 0.95965
Epoch 49/100
2528/2528 [==============================] - 267s 106ms/step - loss: 0.0301 -
accuracy: 0.9907 - val_loss: 0.1542 - val_accuracy: 0.9565

Epoch 00049: val_accuracy did not improve from 0.95965
Epoch 50/100
2528/2528 [==============================] - 303s 120ms/step - loss: 0.0326 -
accuracy: 0.9897 - val_loss: 0.1469 - val_accuracy: 0.9612

Epoch 00050: val_accuracy improved from 0.95965 to 0.96123, saving model to
weights.best_densenet1692.hdf5
Epoch 51/100
2528/2528 [==============================] - 294s 116ms/step - loss: 0.0302 -
accuracy: 0.9907 - val_loss: 0.1502 - val_accuracy: 0.9612

Epoch 00051: val_accuracy did not improve from 0.96123
Epoch 52/100
2528/2528 [==============================] - 214s 85ms/step - loss: 0.0346 -
accuracy: 0.9883 - val_loss: 0.1501 - val_accuracy: 0.9565

Epoch 00052: val_accuracy did not improve from 0.96123
Epoch 53/100
2528/2528 [==============================] - 214s 85ms/step - loss: 0.0322 -
accuracy: 0.9901 - val_loss: 0.1440 - val_accuracy: 0.9573

Epoch 00053: val_accuracy did not improve from 0.96123
Epoch 54/100
2528/2528 [==============================] - 293s 116ms/step - loss: 0.0301 -
accuracy: 0.9915 - val_loss: 0.1461 - val_accuracy: 0.9565

Epoch 00054: val_accuracy did not improve from 0.96123
Epoch 55/100
2528/2528 [==============================] - 317s 125ms/step - loss: 0.0351 -
accuracy: 0.9905 - val_loss: 0.1453 - val_accuracy: 0.9589

Epoch 00055: val_accuracy did not improve from 0.96123
Epoch 56/100
2528/2528 [==============================] - 260s 103ms/step - loss: 0.0379 -
accuracy: 0.9877 - val_loss: 0.1463 - val_accuracy: 0.9597

Epoch 00056: val_accuracy did not improve from 0.96123
Epoch 57/100
2528/2528 [==============================] - 258s 102ms/step - loss: 0.0324 -
accuracy: 0.9923 - val_loss: 0.1489 - val_accuracy: 0.9581
```

```
Epoch 00057: val_accuracy did not improve from 0.96123
Epoch 58/100
2528/2528 [==============================] - 261s 103ms/step - loss: 0.0336 -
accuracy: 0.9879 - val_loss: 0.1460 - val_accuracy: 0.9573

Epoch 00058: val_accuracy did not improve from 0.96123
Epoch 59/100
2528/2528 [==============================] - 263s 104ms/step - loss: 0.0290 -
accuracy: 0.9907 - val_loss: 0.1441 - val_accuracy: 0.9581

Epoch 00059: val_accuracy did not improve from 0.96123
Epoch 60/100
2528/2528 [==============================] - 263s 104ms/step - loss: 0.0288 -
accuracy: 0.9911 - val_loss: 0.1467 - val_accuracy: 0.9597

Epoch 00060: val_accuracy did not improve from 0.96123
Epoch 61/100
2528/2528 [==============================] - 259s 103ms/step - loss: 0.0280 -
accuracy: 0.9903 - val_loss: 0.1474 - val_accuracy: 0.9597

Epoch 00061: val_accuracy did not improve from 0.96123
Epoch 62/100
2528/2528 [==============================] - 337s 133ms/step - loss: 0.0287 -
accuracy: 0.9909 - val_loss: 0.1491 - val_accuracy: 0.9589

Epoch 00062: val_accuracy did not improve from 0.96123
Epoch 63/100
2528/2528 [==============================] - 319s 126ms/step - loss: 0.0268 -
accuracy: 0.9905 - val_loss: 0.1486 - val_accuracy: 0.9589

Epoch 00063: val_accuracy did not improve from 0.96123
Epoch 64/100
2528/2528 [==============================] - 233s 92ms/step - loss: 0.0283 -
accuracy: 0.9929 - val_loss: 0.1474 - val_accuracy: 0.9597

Epoch 00064: val_accuracy did not improve from 0.96123
Epoch 65/100
2528/2528 [==============================] - 261s 103ms/step - loss: 0.0299 -
accuracy: 0.9909 - val_loss: 0.1473 - val_accuracy: 0.9581

Epoch 00065: val_accuracy did not improve from 0.96123
Epoch 66/100
2528/2528 [==============================] - 335s 133ms/step - loss: 0.0283 -
accuracy: 0.9909 - val_loss: 0.1481 - val_accuracy: 0.9581

Epoch 00066: val_accuracy did not improve from 0.96123
Epoch 67/100
2528/2528 [==============================] - 341s 135ms/step - loss: 0.0271 -
```

```
accuracy: 0.9903 - val_loss: 0.1477 - val_accuracy: 0.9581

Epoch 00067: val_accuracy did not improve from 0.96123
Epoch 68/100
2528/2528 [==============================] - 264s 104ms/step - loss: 0.0246 -
accuracy: 0.9921 - val_loss: 0.1483 - val_accuracy: 0.9581

Epoch 00068: val_accuracy did not improve from 0.96123
Epoch 69/100
2528/2528 [==============================] - 251s 99ms/step - loss: 0.0297 -
accuracy: 0.9909 - val_loss: 0.1480 - val_accuracy: 0.9581

Epoch 00069: val_accuracy did not improve from 0.96123
Epoch 70/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.0285 -
accuracy: 0.9911 - val_loss: 0.1482 - val_accuracy: 0.9573

Epoch 00070: val_accuracy did not improve from 0.96123
```

[85]:
```
# summarize history for accuracy
plt.plot(history3.history['accuracy'])
plt.plot(history3.history['val_accuracy'])
plt.title('model accuracy: DenseNet169')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history3.history['loss'])
plt.plot(history3.history['val_loss'])
plt.title('model loss: DenseNet169')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

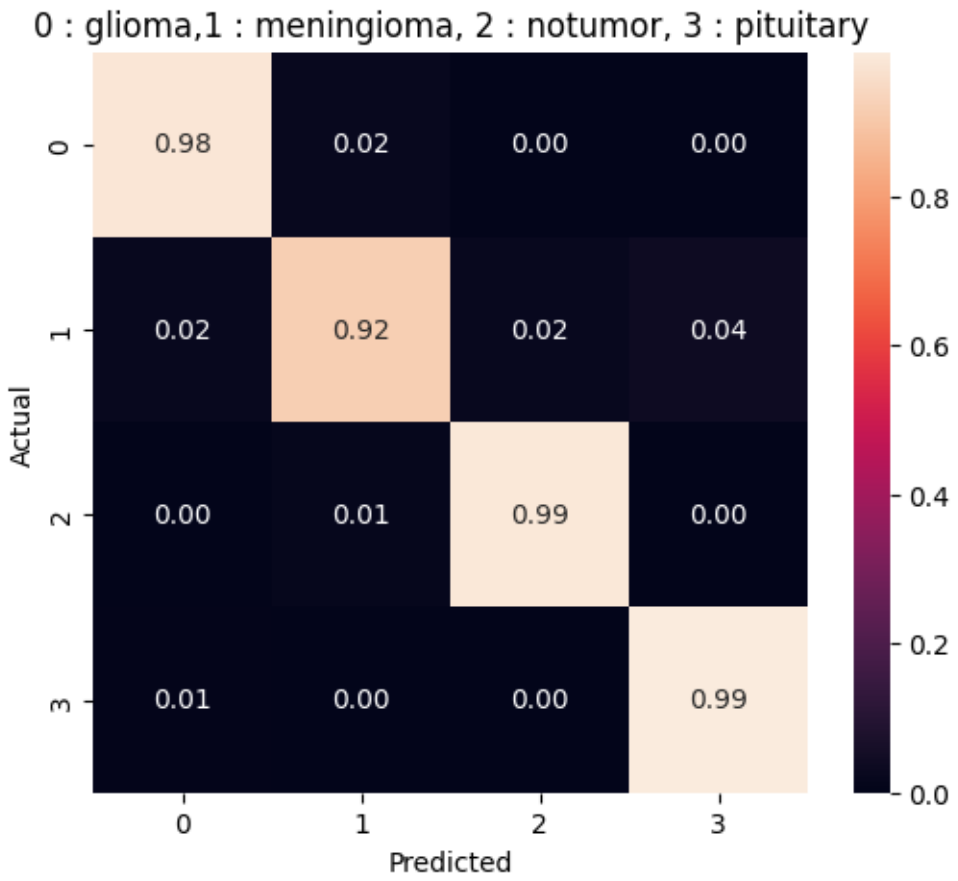

model accuracy: DenseNet169
train
test
accuracy
epoch
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0
10
20
30
40
50
60
70



[108]:
```
model=load_model('weights.best_densenet1692.hdf5')
```

[117]:
```
pred3=model.predict(X_test)
```

[87]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(Y_test, Pred)
# Normalise
cmn = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(cmn, annot=True, fmt='.2f')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('0 : glioma,1 : meningioma, 2 : notumor, 3 : pituitary')
plt.show(block=False)
```



[88]:
```
from keras import models
from numpy import loadtxt
from tensorflow.keras.models import save_model
save_model(cnn, "DenseNet1692.h5")
# load and evaluate a saved model
loaded_model = models.load_model('DenseNet1692.h5')
# summarize model.
model=loaded_model
train_pred_p=model.predict(X_train)
train_pred = np.argmax(train_pred_p, axis=1)
```

6 D: ResNet50

[89]:
```
def make_resnet_model(image_size, num_classes):

    input_shape = image_size

    base_model = tf.keras.applications.ResNet50(input_shape=input_shape,
```

```
                                                    include_top=False, # Do not␣
↪include the dense prediction layer
                                                    weights="imagenet") # Load␣
↪imageNet parameters

    # Freeze the base model by making it non trainable
    base_model.trainable = False

    # create the input layer (Same as the imageNetv2 input size)
    inputs = tf.keras.Input(shape=input_shape)

    # apply data augmentation to the inputs
    x = inputs

    # set training to False to avoid keeping track of statistics in the batch␣
↪norm layer
    x = base_model(x, training=False)

    # Add the new Binary classification layers
    # use global avg pooling to summarize the info in each channel
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    #include dropout with probability of 0.2 to avoid overfitting
    x = Dropout(0.3)(x)
    x = Flatten()(x)
    # Fully Connected Layers
    x =Dense(128, activation = 'relu')(x)

    prediction_layer = Dense(4, activation='softmax')

    outputs = prediction_layer(x)

    model = keras.models.Model(inputs, outputs)

    return model
```

[90]:
```
image_size = (100,100,3)
resnet_model = make_resnet_model(image_size, num_classes = 2)

# Preview the Model Summary
resnet_model.summary()
```

```
Model: "model_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        [(None, 100, 100, 3)]     0
_________________________________________________________________
```

```
resnet50 (Functional)        (None, 4, 4, 2048)        23587712
_________________________________________________________________
global_average_pooling2d_3 ( (None, 2048)              0
_________________________________________________________________
dropout_9 (Dropout)          (None, 2048)              0
_________________________________________________________________
flatten_5 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_10 (Dense)             (None, 128)               262272
_________________________________________________________________
dense_11 (Dense)             (None, 4)                 516
=================================================================
Total params: 23,850,500
Trainable params: 262,788
Non-trainable params: 23,587,712
_________________________________________________________________
```

[91]:
```
filepath51="weights.best_ResNet502.hdf5"
checkpoint4 = ModelCheckpoint(filepath51, monitor='val_accuracy', verbose=1,␣
 ↪save_best_only=True, mode='max')
es4 = EarlyStopping(monitor='val_accuracy', patience=20)
rlrop4 = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=10)
callbacks_list4 = [checkpoint4,es4,rlrop4]
```

[92]:
```
base_learning_rate = 0.001
optimizer = Adam(learning_rate = base_learning_rate)
initial_epochs = 50
batch_size = 64
loss = 'categorical_crossentropy'
metrics = ['accuracy']

resnet_model.compile(optimizer =Adam(learning_rate=0.0001), loss =␣
 ↪'categorical_crossentropy', metrics = ['accuracy'])
```

[93]:
```
history4 = resnet_model.fit(X_train, y_train,epochs = 100,verbose =␣
 ↪1,batch_size=2,validation_data =(X_val,y_val)
                    ,callbacks=callbacks_list4,class_weight=class_weights)
```

```
Epoch 1/100
2528/2528 [==============================] - 283s 110ms/step - loss: 1.0110 -
accuracy: 0.5737 - val_loss: 0.5416 - val_accuracy: 0.8062

Epoch 00001: val_accuracy improved from -inf to 0.80617, saving model to
weights.best_ResNet502.hdf5
Epoch 2/100
2528/2528 [==============================] - 2463s 975ms/step - loss: 0.6110 -
accuracy: 0.7605 - val_loss: 0.4703 - val_accuracy: 0.8228
```

```
Epoch 00002: val_accuracy improved from 0.80617 to 0.82278, saving model to
weights.best_ResNet502.hdf5
Epoch 3/100
2528/2528 [==============================] - 334s 132ms/step - loss: 0.5222 -
accuracy: 0.7965 - val_loss: 0.4245 - val_accuracy: 0.8347

Epoch 00003: val_accuracy improved from 0.82278 to 0.83465, saving model to
weights.best_ResNet502.hdf5
Epoch 4/100
2528/2528 [==============================] - 556s 220ms/step - loss: 0.4793 -
accuracy: 0.8174 - val_loss: 0.3879 - val_accuracy: 0.8536

Epoch 00004: val_accuracy improved from 0.83465 to 0.85364, saving model to
weights.best_ResNet502.hdf5
Epoch 5/100
2528/2528 [==============================] - 746s 295ms/step - loss: 0.4430 -
accuracy: 0.8284 - val_loss: 0.3642 - val_accuracy: 0.8608

Epoch 00005: val_accuracy improved from 0.85364 to 0.86076, saving model to
weights.best_ResNet502.hdf5
Epoch 6/100
2528/2528 [==============================] - 1199s 475ms/step - loss: 0.4298 -
accuracy: 0.8356 - val_loss: 0.3499 - val_accuracy: 0.8663

Epoch 00006: val_accuracy improved from 0.86076 to 0.86630, saving model to
weights.best_ResNet502.hdf5
Epoch 7/100
2528/2528 [==============================] - 1050s 415ms/step - loss: 0.4284 -
accuracy: 0.8363 - val_loss: 0.4024 - val_accuracy: 0.8434

Epoch 00007: val_accuracy did not improve from 0.86630
Epoch 8/100
2528/2528 [==============================] - 11022s 4s/step - loss: 0.3970 -
accuracy: 0.8451 - val_loss: 0.3354 - val_accuracy: 0.8837

Epoch 00008: val_accuracy improved from 0.86630 to 0.88370, saving model to
weights.best_ResNet502.hdf5
Epoch 9/100
2528/2528 [==============================] - 218s 86ms/step - loss: 0.3897 -
accuracy: 0.8471 - val_loss: 0.3477 - val_accuracy: 0.8663

Epoch 00009: val_accuracy did not improve from 0.88370
Epoch 10/100
2528/2528 [==============================] - 211s 84ms/step - loss: 0.3978 -
accuracy: 0.8439 - val_loss: 0.3250 - val_accuracy: 0.8805

Epoch 00010: val_accuracy did not improve from 0.88370
Epoch 11/100
```

```
2528/2528 [==============================] - 215s 85ms/step - loss: 0.3708 -
accuracy: 0.8583 - val_loss: 0.3265 - val_accuracy: 0.8813

Epoch 00011: val_accuracy did not improve from 0.88370
Epoch 12/100
2528/2528 [==============================] - 217s 86ms/step - loss: 0.3439 -
accuracy: 0.8667 - val_loss: 0.3299 - val_accuracy: 0.8790

Epoch 00012: val_accuracy did not improve from 0.88370
Epoch 13/100
2528/2528 [==============================] - 218s 86ms/step - loss: 0.3462 -
accuracy: 0.8674 - val_loss: 0.3202 - val_accuracy: 0.8813

Epoch 00013: val_accuracy did not improve from 0.88370
Epoch 14/100
2528/2528 [==============================] - 224s 89ms/step - loss: 0.3356 -
accuracy: 0.8769 - val_loss: 0.3134 - val_accuracy: 0.8908

Epoch 00014: val_accuracy improved from 0.88370 to 0.89082, saving model to
weights.best_ResNet502.hdf5
Epoch 15/100
2528/2528 [==============================] - 222s 88ms/step - loss: 0.3175 -
accuracy: 0.8824 - val_loss: 0.3213 - val_accuracy: 0.8790

Epoch 00015: val_accuracy did not improve from 0.89082
Epoch 16/100
2528/2528 [==============================] - 225s 89ms/step - loss: 0.3243 -
accuracy: 0.8763 - val_loss: 0.3218 - val_accuracy: 0.8805

Epoch 00016: val_accuracy did not improve from 0.89082
Epoch 17/100
2528/2528 [==============================] - 231s 91ms/step - loss: 0.3201 -
accuracy: 0.8758 - val_loss: 0.3487 - val_accuracy: 0.8726

Epoch 00017: val_accuracy did not improve from 0.89082
Epoch 18/100
2528/2528 [==============================] - 233s 92ms/step - loss: 0.3070 -
accuracy: 0.8829 - val_loss: 0.3130 - val_accuracy: 0.8829

Epoch 00018: val_accuracy did not improve from 0.89082
Epoch 19/100
2528/2528 [==============================] - 236s 93ms/step - loss: 0.2913 -
accuracy: 0.8888 - val_loss: 0.3232 - val_accuracy: 0.8750

Epoch 00019: val_accuracy did not improve from 0.89082
Epoch 20/100
2528/2528 [==============================] - 240s 95ms/step - loss: 0.2770 -
accuracy: 0.8944 - val_loss: 0.2837 - val_accuracy: 0.8940
```

```
Epoch 00020: val_accuracy improved from 0.89082 to 0.89399, saving model to
weights.best_ResNet502.hdf5
Epoch 21/100
2528/2528 [==============================] - 242s 96ms/step - loss: 0.2757 -
accuracy: 0.8933 - val_loss: 0.2837 - val_accuracy: 0.8892

Epoch 00021: val_accuracy did not improve from 0.89399
Epoch 22/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.2741 -
accuracy: 0.8926 - val_loss: 0.2653 - val_accuracy: 0.9027

Epoch 00022: val_accuracy improved from 0.89399 to 0.90269, saving model to
weights.best_ResNet502.hdf5
Epoch 23/100
2528/2528 [==============================] - 257s 101ms/step - loss: 0.2672 -
accuracy: 0.8956 - val_loss: 0.2767 - val_accuracy: 0.9066

Epoch 00023: val_accuracy improved from 0.90269 to 0.90665, saving model to
weights.best_ResNet502.hdf5
Epoch 24/100
2528/2528 [==============================] - 242s 96ms/step - loss: 0.2639 -
accuracy: 0.9024 - val_loss: 0.2824 - val_accuracy: 0.8995

Epoch 00024: val_accuracy did not improve from 0.90665
Epoch 25/100
2528/2528 [==============================] - 244s 96ms/step - loss: 0.2635 -
accuracy: 0.9015 - val_loss: 0.2617 - val_accuracy: 0.9074

Epoch 00025: val_accuracy improved from 0.90665 to 0.90744, saving model to
weights.best_ResNet502.hdf5
Epoch 26/100
2528/2528 [==============================] - 239s 95ms/step - loss: 0.2532 -
accuracy: 0.8978 - val_loss: 0.2714 - val_accuracy: 0.9043

Epoch 00026: val_accuracy did not improve from 0.90744
Epoch 27/100
2528/2528 [==============================] - 240s 95ms/step - loss: 0.2593 -
accuracy: 0.9002 - val_loss: 0.2615 - val_accuracy: 0.9098

Epoch 00027: val_accuracy improved from 0.90744 to 0.90981, saving model to
weights.best_ResNet502.hdf5
Epoch 28/100
2528/2528 [==============================] - 245s 97ms/step - loss: 0.2380 -
accuracy: 0.9144 - val_loss: 0.2911 - val_accuracy: 0.8869

Epoch 00028: val_accuracy did not improve from 0.90981
Epoch 29/100
```

```
2528/2528 [==============================] - 250s 99ms/step - loss: 0.2408 -
accuracy: 0.9091 - val_loss: 0.2735 - val_accuracy: 0.9074

Epoch 00029: val_accuracy did not improve from 0.90981
Epoch 30/100
2528/2528 [==============================] - 363s 144ms/step - loss: 0.2433 -
accuracy: 0.9031 - val_loss: 0.2474 - val_accuracy: 0.9130

Epoch 00030: val_accuracy improved from 0.90981 to 0.91297, saving model to
weights.best_ResNet502.hdf5
Epoch 31/100
2528/2528 [==============================] - 564s 223ms/step - loss: 0.2436 -
accuracy: 0.9031 - val_loss: 0.2979 - val_accuracy: 0.9122

Epoch 00031: val_accuracy did not improve from 0.91297
Epoch 32/100
2528/2528 [==============================] - 252s 100ms/step - loss: 0.2274 -
accuracy: 0.9138 - val_loss: 0.2571 - val_accuracy: 0.9106

Epoch 00032: val_accuracy did not improve from 0.91297
Epoch 33/100
2528/2528 [==============================] - 251s 99ms/step - loss: 0.2351 -
accuracy: 0.9125 - val_loss: 0.2530 - val_accuracy: 0.9122

Epoch 00033: val_accuracy did not improve from 0.91297
Epoch 34/100
2528/2528 [==============================] - 265s 105ms/step - loss: 0.2165 -
accuracy: 0.9180 - val_loss: 0.2371 - val_accuracy: 0.9233

Epoch 00034: val_accuracy improved from 0.91297 to 0.92326, saving model to
weights.best_ResNet502.hdf5
Epoch 35/100
2528/2528 [==============================] - 263s 104ms/step - loss: 0.2320 -
accuracy: 0.9087 - val_loss: 0.2604 - val_accuracy: 0.9177

Epoch 00035: val_accuracy did not improve from 0.92326
Epoch 36/100
2528/2528 [==============================] - 313s 124ms/step - loss: 0.2115 -
accuracy: 0.9185 - val_loss: 0.2789 - val_accuracy: 0.8979

Epoch 00036: val_accuracy did not improve from 0.92326
Epoch 37/100
2528/2528 [==============================] - 301s 119ms/step - loss: 0.2205 -
accuracy: 0.9115 - val_loss: 0.2861 - val_accuracy: 0.9035

Epoch 00037: val_accuracy did not improve from 0.92326
Epoch 38/100
2528/2528 [==============================] - 358s 142ms/step - loss: 0.2048 -
```

```
accuracy: 0.9228 - val_loss: 0.2565 - val_accuracy: 0.9106

Epoch 00038: val_accuracy did not improve from 0.92326
Epoch 39/100
2528/2528 [==============================] - 376s 149ms/step - loss: 0.1983 -
accuracy: 0.9253 - val_loss: 0.2629 - val_accuracy: 0.9130

Epoch 00039: val_accuracy did not improve from 0.92326
Epoch 40/100
2528/2528 [==============================] - 268s 106ms/step - loss: 0.1993 -
accuracy: 0.9281 - val_loss: 0.2598 - val_accuracy: 0.9122

Epoch 00040: val_accuracy did not improve from 0.92326
Epoch 41/100
2528/2528 [==============================] - 265s 105ms/step - loss: 0.1848 -
accuracy: 0.9328 - val_loss: 0.2458 - val_accuracy: 0.9161

Epoch 00041: val_accuracy did not improve from 0.92326
Epoch 42/100
2528/2528 [==============================] - 369s 146ms/step - loss: 0.1982 -
accuracy: 0.9213 - val_loss: 0.2961 - val_accuracy: 0.8964

Epoch 00042: val_accuracy did not improve from 0.92326
Epoch 43/100
2528/2528 [==============================] - 426s 168ms/step - loss: 0.1833 -
accuracy: 0.9375 - val_loss: 0.2510 - val_accuracy: 0.9043

Epoch 00043: val_accuracy did not improve from 0.92326
Epoch 44/100
2528/2528 [==============================] - 310s 123ms/step - loss: 0.1907 -
accuracy: 0.9273 - val_loss: 0.2620 - val_accuracy: 0.9114

Epoch 00044: val_accuracy did not improve from 0.92326
Epoch 45/100
2528/2528 [==============================] - 300s 119ms/step - loss: 0.1718 -
accuracy: 0.9309 - val_loss: 0.2231 - val_accuracy: 0.9201

Epoch 00045: val_accuracy did not improve from 0.92326
Epoch 46/100
2528/2528 [==============================] - 289s 114ms/step - loss: 0.1545 -
accuracy: 0.9476 - val_loss: 0.2169 - val_accuracy: 0.9233

Epoch 00046: val_accuracy did not improve from 0.92326
Epoch 47/100
2528/2528 [==============================] - 328s 130ms/step - loss: 0.1630 -
accuracy: 0.9417 - val_loss: 0.2200 - val_accuracy: 0.9280

Epoch 00047: val_accuracy improved from 0.92326 to 0.92801, saving model to
```

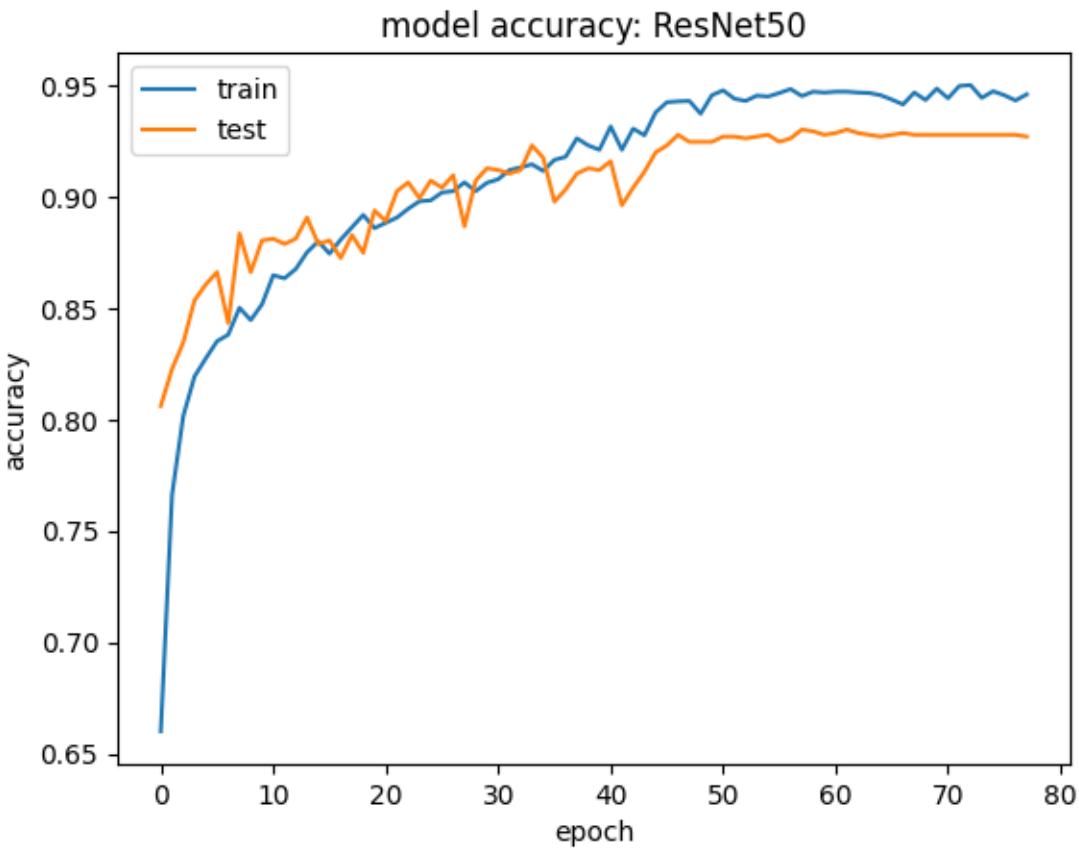
```
weights.best_ResNet502.hdf5
Epoch 48/100
2528/2528 [==============================] - 311s 123ms/step - loss: 0.1458 -
accuracy: 0.9497 - val_loss: 0.2222 - val_accuracy: 0.9248

Epoch 00048: val_accuracy did not improve from 0.92801
Epoch 49/100
2528/2528 [==============================] - 288s 114ms/step - loss: 0.1662 -
accuracy: 0.9322 - val_loss: 0.2164 - val_accuracy: 0.9248

Epoch 00049: val_accuracy did not improve from 0.92801
Epoch 50/100
2528/2528 [==============================] - 282s 112ms/step - loss: 0.1556 -
accuracy: 0.9474 - val_loss: 0.2193 - val_accuracy: 0.9248

Epoch 00050: val_accuracy did not improve from 0.92801
Epoch 51/100
2528/2528 [==============================] - 272s 108ms/step - loss: 0.1477 -
accuracy: 0.9515 - val_loss: 0.2176 - val_accuracy: 0.9272

Epoch 00051: val_accuracy did not improve from 0.92801
Epoch 52/100
2528/2528 [==============================] - 261s 103ms/step - loss: 0.1610 -
accuracy: 0.9415 - val_loss: 0.2199 - val_accuracy: 0.9272

Epoch 00052: val_accuracy did not improve from 0.92801
Epoch 53/100
2528/2528 [==============================] - 272s 107ms/step - loss: 0.1450 -
accuracy: 0.9440 - val_loss: 0.2174 - val_accuracy: 0.9264

Epoch 00053: val_accuracy did not improve from 0.92801
Epoch 54/100
2528/2528 [==============================] - 266s 105ms/step - loss: 0.1556 -
accuracy: 0.9404 - val_loss: 0.2252 - val_accuracy: 0.9272

Epoch 00054: val_accuracy did not improve from 0.92801
Epoch 55/100
2528/2528 [==============================] - 265s 105ms/step - loss: 0.1566 -
accuracy: 0.9425 - val_loss: 0.2225 - val_accuracy: 0.9280

Epoch 00055: val_accuracy did not improve from 0.92801
Epoch 56/100
2528/2528 [==============================] - 265s 105ms/step - loss: 0.1559 -
accuracy: 0.9414 - val_loss: 0.2213 - val_accuracy: 0.9248

Epoch 00056: val_accuracy did not improve from 0.92801
Epoch 57/100
2528/2528 [==============================] - 264s 104ms/step - loss: 0.1458 -
```

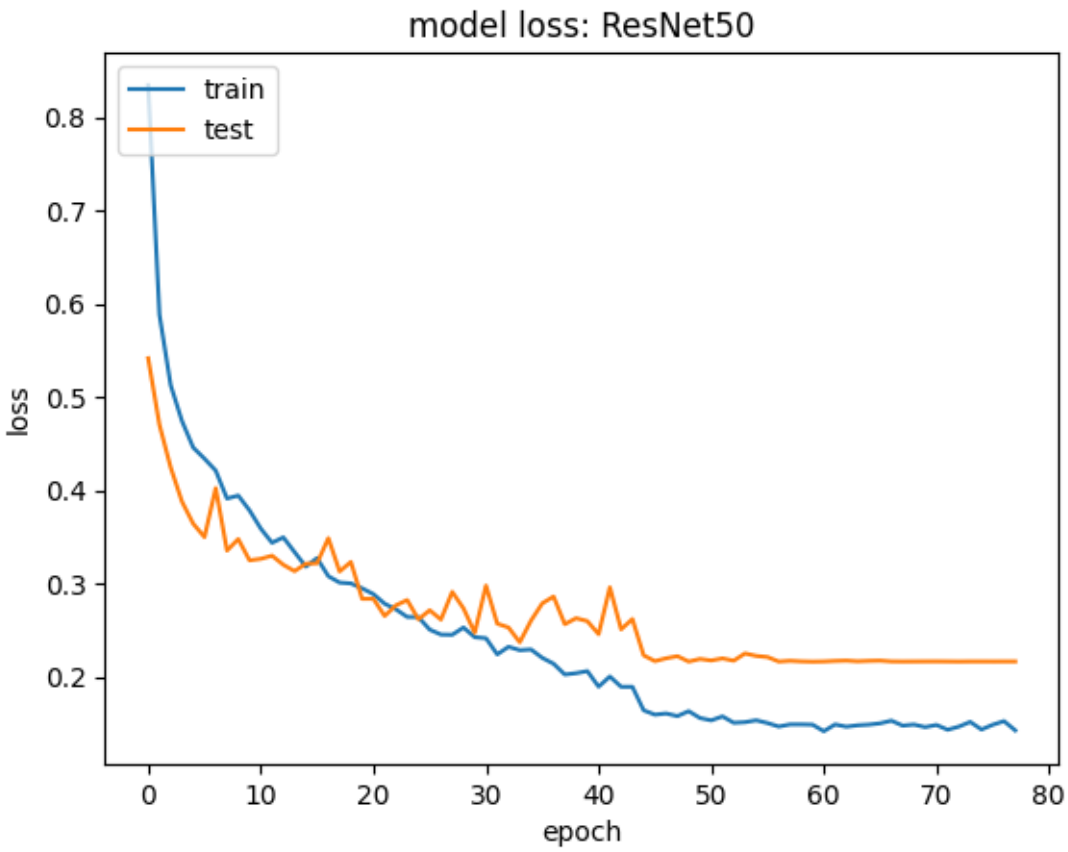
```
accuracy: 0.9502 - val_loss: 0.2165 - val_accuracy: 0.9264

Epoch 00057: val_accuracy did not improve from 0.92801
Epoch 58/100
2528/2528 [==============================] - 285s 113ms/step - loss: 0.1462 -
accuracy: 0.9477 - val_loss: 0.2173 - val_accuracy: 0.9304

Epoch 00058: val_accuracy improved from 0.92801 to 0.93038, saving model to
weights.best_ResNet502.hdf5
Epoch 59/100
2528/2528 [==============================] - 263s 104ms/step - loss: 0.1494 -
accuracy: 0.9476 - val_loss: 0.2167 - val_accuracy: 0.9296

Epoch 00059: val_accuracy did not improve from 0.93038
Epoch 60/100
2528/2528 [==============================] - 290s 115ms/step - loss: 0.1550 -
accuracy: 0.9440 - val_loss: 0.2164 - val_accuracy: 0.9280

Epoch 00060: val_accuracy did not improve from 0.93038
Epoch 61/100
2528/2528 [==============================] - 347s 137ms/step - loss: 0.1369 -
accuracy: 0.9462 - val_loss: 0.2166 - val_accuracy: 0.9288

Epoch 00061: val_accuracy did not improve from 0.93038
Epoch 62/100
2528/2528 [==============================] - 351s 139ms/step - loss: 0.1617 -
accuracy: 0.9477 - val_loss: 0.2171 - val_accuracy: 0.9304

Epoch 00062: val_accuracy did not improve from 0.93038
Epoch 63/100
2528/2528 [==============================] - 258s 102ms/step - loss: 0.1459 -
accuracy: 0.9488 - val_loss: 0.2175 - val_accuracy: 0.9288

Epoch 00063: val_accuracy did not improve from 0.93038
Epoch 64/100
2528/2528 [==============================] - 242s 96ms/step - loss: 0.1509 -
accuracy: 0.9448 - val_loss: 0.2168 - val_accuracy: 0.9280

Epoch 00064: val_accuracy did not improve from 0.93038
Epoch 65/100
2528/2528 [==============================] - 3361s 1s/step - loss: 0.1546 -
accuracy: 0.9454 - val_loss: 0.2172 - val_accuracy: 0.9272

Epoch 00065: val_accuracy did not improve from 0.93038
Epoch 66/100
2528/2528 [==============================] - 263s 104ms/step - loss: 0.1555 -
accuracy: 0.9376 - val_loss: 0.2175 - val_accuracy: 0.9280
```

```
Epoch 00066: val_accuracy did not improve from 0.93038
Epoch 67/100
2528/2528 [==============================] - 250s 99ms/step - loss: 0.1613 -
accuracy: 0.9400 - val_loss: 0.2166 - val_accuracy: 0.9288

Epoch 00067: val_accuracy did not improve from 0.93038
Epoch 68/100
2528/2528 [==============================] - 267s 106ms/step - loss: 0.1476 -
accuracy: 0.9494 - val_loss: 0.2165 - val_accuracy: 0.9280

Epoch 00068: val_accuracy did not improve from 0.93038
Epoch 69/100
2528/2528 [==============================] - 290s 115ms/step - loss: 0.1510 -
accuracy: 0.9423 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00069: val_accuracy did not improve from 0.93038
Epoch 70/100
2528/2528 [==============================] - 306s 121ms/step - loss: 0.1458 -
accuracy: 0.9498 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00070: val_accuracy did not improve from 0.93038
Epoch 71/100
2528/2528 [==============================] - 306s 121ms/step - loss: 0.1468 -
accuracy: 0.9443 - val_loss: 0.2167 - val_accuracy: 0.9280

Epoch 00071: val_accuracy did not improve from 0.93038
Epoch 72/100
2528/2528 [==============================] - 305s 121ms/step - loss: 0.1332 -
accuracy: 0.9515 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00072: val_accuracy did not improve from 0.93038
Epoch 73/100
2528/2528 [==============================] - 285s 113ms/step - loss: 0.1431 -
accuracy: 0.9520 - val_loss: 0.2165 - val_accuracy: 0.9280

Epoch 00073: val_accuracy did not improve from 0.93038
Epoch 74/100
2528/2528 [==============================] - 287s 113ms/step - loss: 0.1407 -
accuracy: 0.9459 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00074: val_accuracy did not improve from 0.93038
Epoch 75/100
2528/2528 [==============================] - 291s 115ms/step - loss: 0.1456 -
accuracy: 0.9456 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00075: val_accuracy did not improve from 0.93038
Epoch 76/100
2528/2528 [==============================] - 292s 115ms/step - loss: 0.1524 -
```

```
accuracy: 0.9436 - val_loss: 0.2165 - val_accuracy: 0.9280

Epoch 00076: val_accuracy did not improve from 0.93038
Epoch 77/100
2528/2528 [==============================] - 293s 116ms/step - loss: 0.1469 -
accuracy: 0.9454 - val_loss: 0.2166 - val_accuracy: 0.9280

Epoch 00077: val_accuracy did not improve from 0.93038
Epoch 78/100
2528/2528 [==============================] - 304s 120ms/step - loss: 0.1406 -
accuracy: 0.9444 - val_loss: 0.2166 - val_accuracy: 0.9272

Epoch 00078: val_accuracy did not improve from 0.93038
```

[94]:
```
# summarize history for accuracy
plt.plot(history4.history['accuracy'])
plt.plot(history4.history['val_accuracy'])
plt.title('model accuracy: ResNet50')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history4.history['loss'])
plt.plot(history4.history['val_loss'])
plt.title('model loss: ResNet50')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

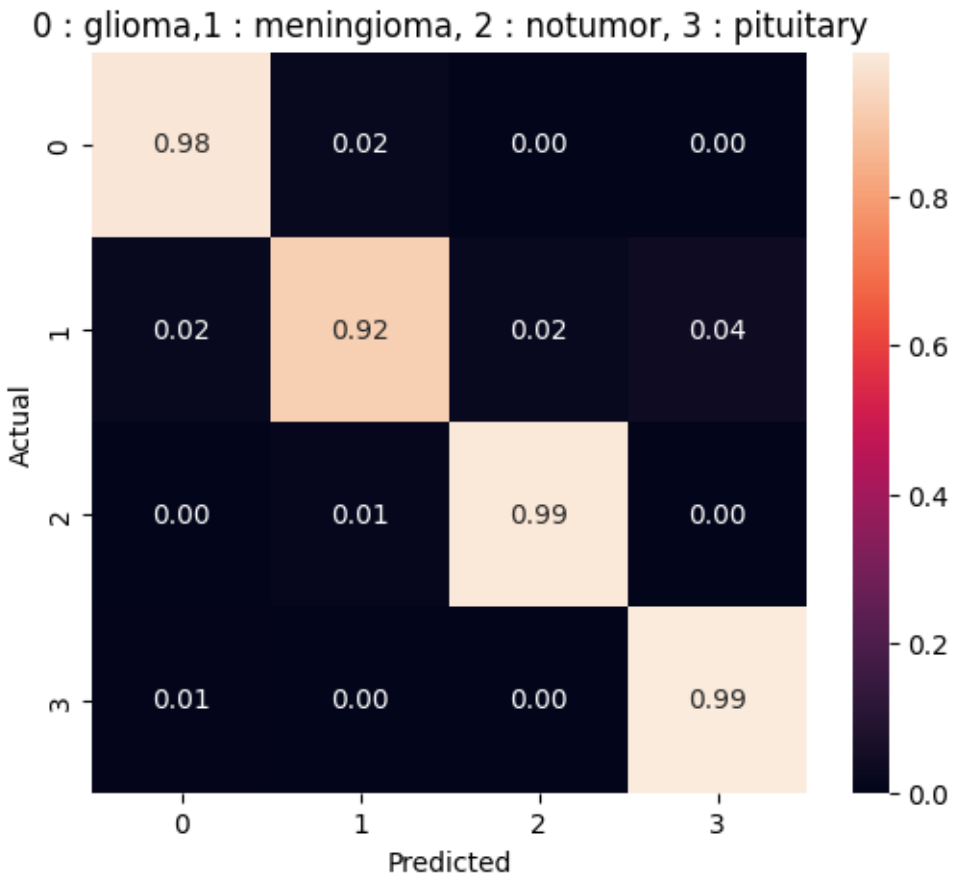

model accuracy: ResNet50
train
test
accuracy
epoch
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
10
20
30
40
50
60
70
80



[110]:
```
model=load_model('weights.best_ResNet502.hdf5')
```

[111]:
```
pred4=model.predict(X_test)
```

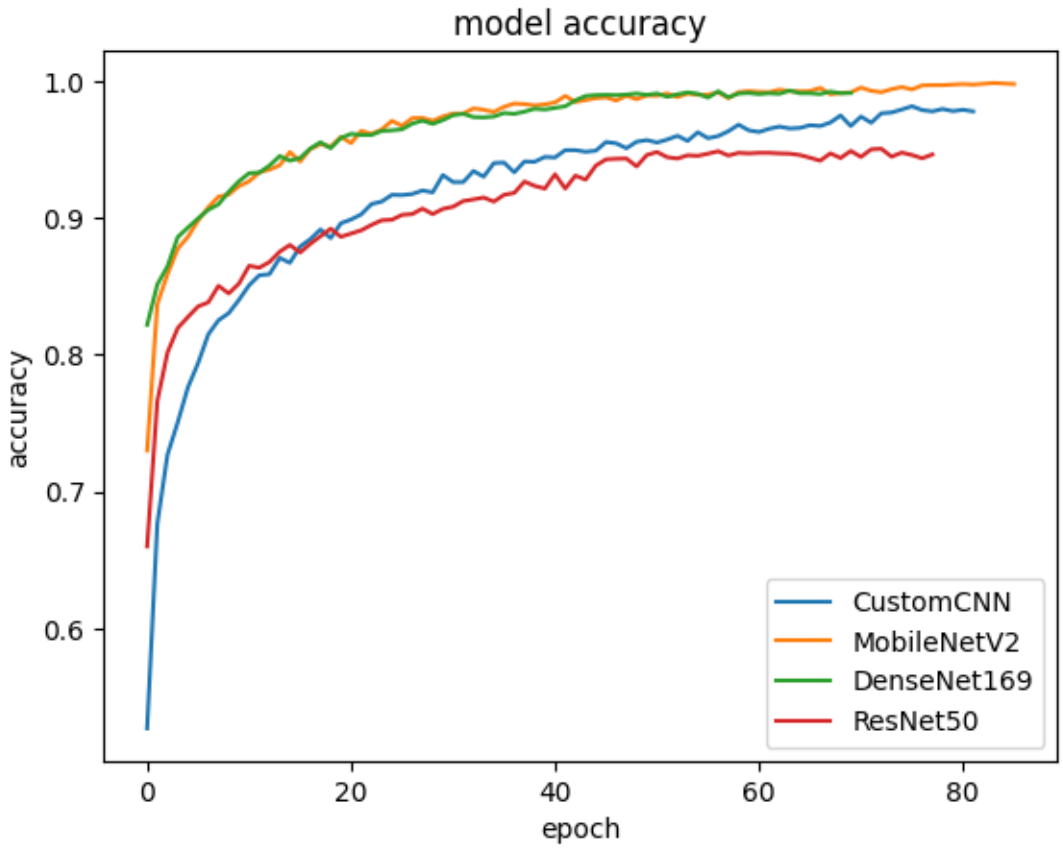
[96]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(Y_test, Pred)
# Normalise
cmn = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(cmn, annot=True, fmt='.2f')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('0 : glioma,1 : meningioma, 2 : notumor, 3 : pituitary')
plt.show(block=False)
```

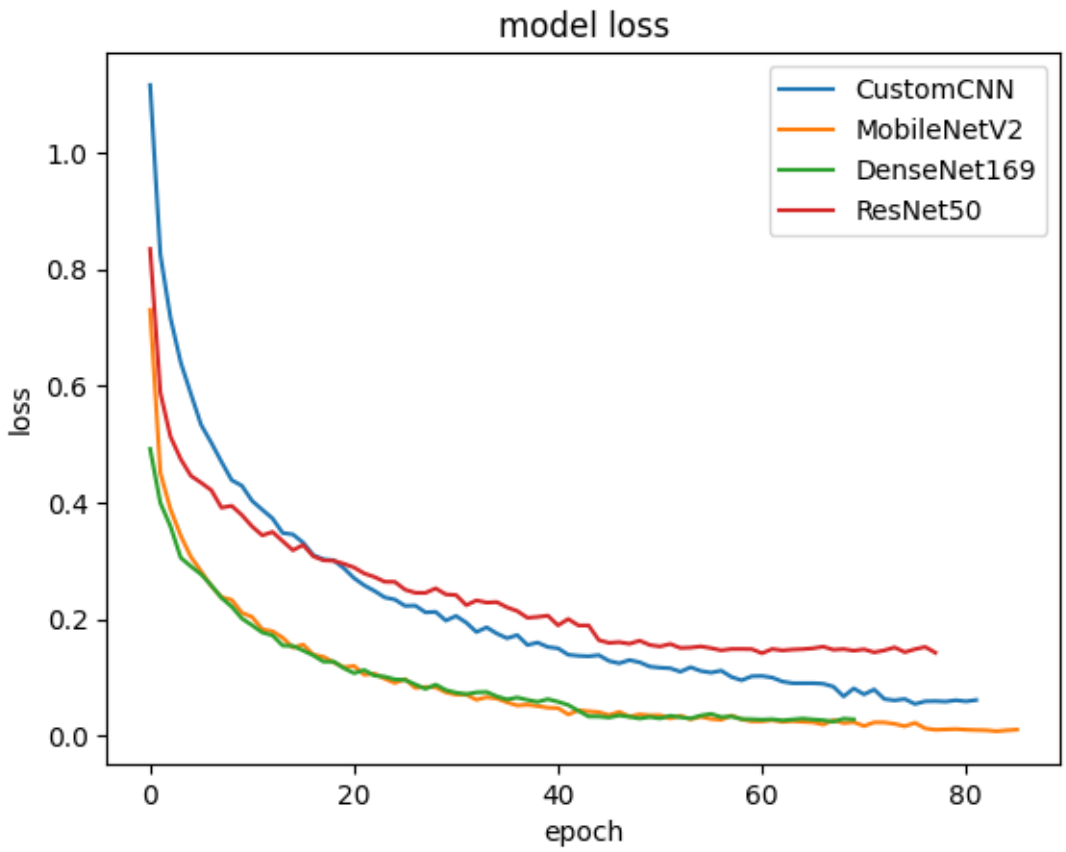


[98]:
```
from keras import models
from numpy import loadtxt
from tensorflow.keras.models import save_model
save_model(cnn, "ResNet502.h5")
# load and evaluate a saved model
loaded_model = models.load_model('ResNet502.h5')
# summarize model.
model=loaded_model
train_pred_p=model.predict(X_train)
train_pred = np.argmax(train_pred_p, axis=1)
```

[99]:
```
# summarize history for accuracy
plt.plot(history1.history['accuracy'])
plt.plot(history2.history['accuracy'])
plt.plot(history3.history['accuracy'])
plt.plot(history4.history['accuracy'])

plt.title('model accuracy')
plt.ylabel('accuracy')
```

```
plt.xlabel('epoch')
plt.legend(['CustomCNN', 'MobileNetV2','DenseNet169','ResNet50'], loc='lower␣
 ↪right')
plt.show()
# summarize history for loss
plt.plot(history1.history['loss'])
plt.plot(history2.history['loss'])
plt.plot(history3.history['loss'])
plt.plot(history4.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['CustomCNN', 'MobileNetV2','DenseNet169','ResNet50'], loc='upper␣
 ↪right')
plt.show()
```





7 ENSEMBLE

To create an ensemble of the four models, we will stack their predictions and use Microsoft FLAML AutoML to find an optimal combiner.

[112]:
```
X_data=np.concatenate([pred1,pred2,pred3,pred4], axis=1)
y_data= np.argmax(y_test, axis=1)
X_data.shape, y_data.shape
```

[112]:
```
((703, 16), (703,))
```

[120]:
```
acc1=accuracy_score(np.argmax(pred1, axis=1),np.argmax(y_test, axis=1))
acc2=accuracy_score(np.argmax(pred2, axis=1),np.argmax(y_test, axis=1))
acc3=accuracy_score(np.argmax(pred3, axis=1),np.argmax(y_test, axis=1))
acc4=accuracy_score(np.argmax(pred4, axis=1),np.argmax(y_test, axis=1))
accuracy_score(np.argmax(pred1, axis=1),np.argmax(y_test, axis=1))
```

[120]:
```
0.968705547652916
```

[119]:
```
y_test
```

[119]:
```
array([[1., 0., 0., 0.],
       [0., 0., 0., 1.],
       [0., 0., 1., 0.],
       …,
       [0., 0., 1., 0.],
       [0., 0., 1., 0.],
       [0., 1., 0., 0.]], dtype=float32)
```

[113]:
```
from flaml import AutoML
automl = AutoML()
# Specify automl goal and constraint
automl_settings = {
    "time_budget": 1000,  # total running time in seconds
    "task": 'classification',  # task type
    "seed": 24545678,  # random seed
    "metric" : 'accuracy'}

automl.fit(X_train=X_data, y_train=y_data, **automl_settings)
```

```
[flaml.automl.logger: 04-25 17:06:59] {1679} INFO - task = classification
[flaml.automl.logger: 04-25 17:06:59] {1690} INFO - Evaluation method: cv
[flaml.automl.logger: 04-25 17:06:59] {1788} INFO - Minimizing error metric:
1-accuracy
[flaml.automl.logger: 04-25 17:06:59] {1900} INFO - List of ML learners in
AutoML Run: ['lgbm', 'rf', 'xgboost', 'extra_tree', 'xgb_limitdepth', 'lrl1']
[flaml.automl.logger: 04-25 17:06:59] {2218} INFO - iteration 0, current learner
lgbm
[flaml.automl.logger: 04-25 17:06:59] {2344} INFO - Estimated sufficient time
budget=789s. Estimated necessary time budget=18s.
[flaml.automl.logger: 04-25 17:06:59] {2391} INFO -  at 0.1s,   estimator lgbm's
best error=0.0398,     best estimator lgbm's best error=0.0398
[flaml.automl.logger: 04-25 17:06:59] {2218} INFO - iteration 1, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.2s,   estimator lgbm's
best error=0.0398,     best estimator lgbm's best error=0.0398
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 2, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.2s,   estimator lgbm's
best error=0.0398,     best estimator lgbm's best error=0.0398
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 3, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.3s,   estimator lgbm's
best error=0.0356,     best estimator lgbm's best error=0.0356
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 4, current learner
xgboost
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.4s,   estimator
xgboost's best error=0.0342,  best estimator xgboost's best error=0.0342
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 5, current learner
```

```
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.4s,   estimator lgbm's
best error=0.0342,     best estimator xgboost's best error=0.0342
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 6, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.5s,   estimator lgbm's
best error=0.0299,     best estimator lgbm's best error=0.0299
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 7, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.5s,   estimator lgbm's
best error=0.0299,     best estimator lgbm's best error=0.0299
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 8, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.6s,   estimator lgbm's
best error=0.0299,     best estimator lgbm's best error=0.0299
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 9, current learner
lgbm
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.7s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 10, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.8s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 11, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 0.9s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 12, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:00] {2391} INFO -  at 1.0s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:00] {2218} INFO - iteration 13, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.1s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 14, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.3s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 15, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.3s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 16, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.5s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 17, current
```

```
learner lgbm
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.5s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 18, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.7s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 19, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.8s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 20, current
learner rf
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 1.9s,   estimator rf's
best error=0.0455,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 21, current
learner rf
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 2.0s,   estimator rf's
best error=0.0455,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 22, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:01] {2391} INFO -  at 2.1s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:01] {2218} INFO - iteration 23, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.2s,   estimator lgbm's
best error=0.0256,     best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 24, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.2s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 25, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.4s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 26, current
learner rf
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.6s,   estimator rf's
best error=0.0370,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 27, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.7s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 28, current
learner extra_tree
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 2.8s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 29, current
```

```
learner extra_tree
[flaml.automl.logger: 04-25 17:07:02] {2391} INFO -  at 3.0s,   estimator
extra_tree's best error=0.0313,       best estimator lgbm's best error=0.0256
[flaml.automl.logger: 04-25 17:07:02] {2218} INFO - iteration 30, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.1s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 31, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.2s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 32, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.4s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 33, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.5s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 34, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.8s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 35, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 3.9s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 36, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 4.0s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 37, current
learner rf
[flaml.automl.logger: 04-25 17:07:03] {2391} INFO -  at 4.1s,   estimator rf's
best error=0.0370,       best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:03] {2218} INFO - iteration 38, current
learner rf
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.3s,   estimator rf's
best error=0.0370,       best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 39, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.4s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 40, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.5s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 41, current
```

```
learner xgboost
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.6s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 42, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.8s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 43, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 4.9s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 44, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:04] {2391} INFO -  at 5.0s,   estimator
xgboost's best error=0.0284,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:04] {2218} INFO - iteration 45, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.1s,   estimator
xgboost's best error=0.0256,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 46, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.3s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 47, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.4s,   estimator
xgboost's best error=0.0256,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 48, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.5s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 49, current
learner rf
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.6s,   estimator rf's
best error=0.0370,       best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 50, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 5.9s,   estimator
xgboost's best error=0.0256,  best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 51, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:05] {2391} INFO -  at 6.0s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:05] {2218} INFO - iteration 52, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:06] {2391} INFO -  at 6.1s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:06] {2218} INFO - iteration 53, current
```

```
learner lgbm
[flaml.automl.logger: 04-25 17:07:06] {2391} INFO -  at 6.2s,   estimator lgbm's
best error=0.0242,     best estimator lgbm's best error=0.0242
[flaml.automl.logger: 04-25 17:07:06] {2218} INFO - iteration 54, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:06] {2391} INFO -  at 6.4s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:06] {2218} INFO - iteration 55, current
learner rf
[flaml.automl.logger: 04-25 17:07:06] {2391} INFO -  at 6.7s,   estimator rf's
best error=0.0370,       best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:06] {2218} INFO - iteration 56, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:07] {2391} INFO -  at 7.2s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:07] {2218} INFO - iteration 57, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:07] {2391} INFO -  at 7.4s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:07] {2218} INFO - iteration 58, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:07] {2391} INFO -  at 7.5s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:07] {2218} INFO - iteration 59, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:07] {2391} INFO -  at 7.9s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:07] {2218} INFO - iteration 60, current
learner rf
[flaml.automl.logger: 04-25 17:07:08] {2391} INFO -  at 8.1s,   estimator rf's
best error=0.0327,       best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:08] {2218} INFO - iteration 61, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:08] {2391} INFO -  at 8.3s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:08] {2218} INFO - iteration 62, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:08] {2391} INFO -  at 8.5s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:08] {2218} INFO - iteration 63, current
learner rf
[flaml.automl.logger: 04-25 17:07:08] {2391} INFO -  at 8.7s,   estimator rf's
best error=0.0327,       best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:08] {2218} INFO - iteration 64, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:08] {2391} INFO -  at 8.8s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:08] {2218} INFO - iteration 65, current
```

```
learner lgbm
[flaml.automl.logger: 04-25 17:07:09] {2391} INFO -  at 9.4s,   estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:09] {2218} INFO - iteration 66, current
learner rf
[flaml.automl.logger: 04-25 17:07:09] {2391} INFO -  at 9.6s,   estimator rf's
best error=0.0327,       best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:09] {2218} INFO - iteration 67, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:09] {2391} INFO -  at 10.0s,  estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:09] {2218} INFO - iteration 68, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:10] {2391} INFO -  at 10.2s,  estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:10] {2218} INFO - iteration 69, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:10] {2391} INFO -  at 10.5s,  estimator lgbm's
best error=0.0213,     best estimator lgbm's best error=0.0213
[flaml.automl.logger: 04-25 17:07:10] {2218} INFO - iteration 70, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:10] {2391} INFO -  at 10.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:10] {2218} INFO - iteration 71, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:10] {2391} INFO -  at 10.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:10] {2218} INFO - iteration 72, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:10] {2391} INFO -  at 11.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:10] {2218} INFO - iteration 73, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:11] {2391} INFO -  at 11.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:11] {2218} INFO - iteration 74, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:11] {2391} INFO -  at 11.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:11] {2218} INFO - iteration 75, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:11] {2391} INFO -  at 11.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:11] {2218} INFO - iteration 76, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:11] {2391} INFO -  at 11.7s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:11] {2218} INFO - iteration 77, current
```

```
learner xgboost
[flaml.automl.logger: 04-25 17:07:11] {2391} INFO -  at 12.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:11] {2218} INFO - iteration 78, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:12] {2391} INFO -  at 12.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:12] {2218} INFO - iteration 79, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:12] {2391} INFO -  at 12.6s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:12] {2218} INFO - iteration 80, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:12] {2391} INFO -  at 12.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:12] {2218} INFO - iteration 81, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:12] {2391} INFO -  at 13.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:12] {2218} INFO - iteration 82, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:13] {2391} INFO -  at 13.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:13] {2218} INFO - iteration 83, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:13] {2391} INFO -  at 13.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:13] {2218} INFO - iteration 84, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:13] {2391} INFO -  at 13.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:13] {2218} INFO - iteration 85, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:13] {2391} INFO -  at 14.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:13] {2218} INFO - iteration 86, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:14] {2391} INFO -  at 14.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:14] {2218} INFO - iteration 87, current
learner rf
[flaml.automl.logger: 04-25 17:07:14] {2391} INFO -  at 14.4s,  estimator rf's
best error=0.0327,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:14] {2218} INFO - iteration 88, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:14] {2391} INFO -  at 14.5s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:14] {2218} INFO - iteration 89, current
```

```
learner xgboost
[flaml.automl.logger: 04-25 17:07:14] {2391} INFO -  at 14.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:14] {2218} INFO - iteration 90, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:14] {2391} INFO -  at 15.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:14] {2218} INFO - iteration 91, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:15] {2391} INFO -  at 15.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:15] {2218} INFO - iteration 92, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:15] {2391} INFO -  at 15.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:15] {2218} INFO - iteration 93, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:15] {2391} INFO -  at 15.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:15] {2218} INFO - iteration 94, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:15] {2391} INFO -  at 15.9s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:15] {2218} INFO - iteration 95, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:16] {2391} INFO -  at 16.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:16] {2218} INFO - iteration 96, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:16] {2391} INFO -  at 16.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:16] {2218} INFO - iteration 97, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:16] {2391} INFO -  at 16.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:16] {2218} INFO - iteration 98, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:16] {2391} INFO -  at 17.0s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:16] {2218} INFO - iteration 99, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:17] {2391} INFO -  at 17.7s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:17] {2218} INFO - iteration 100, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:17] {2391} INFO -  at 17.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:17] {2218} INFO - iteration 101, current
```

```
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:17] {2391} INFO -  at 18.1s,  estimator
xgb_limitdepth's best error=0.0327,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:17] {2218} INFO - iteration 102, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 18.2s,  estimator
xgb_limitdepth's best error=0.0299,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 103, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 18.4s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 104, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 18.6s,  estimator
xgb_limitdepth's best error=0.0299,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 105, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 18.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 106, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 18.8s,  estimator
xgb_limitdepth's best error=0.0299,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 107, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:18] {2391} INFO -  at 19.1s,  estimator
xgb_limitdepth's best error=0.0284,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:18] {2218} INFO - iteration 108, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:19] {2391} INFO -  at 19.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:19] {2218} INFO - iteration 109, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:19] {2391} INFO -  at 19.6s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:19] {2218} INFO - iteration 110, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:19] {2391} INFO -  at 19.9s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:19] {2218} INFO - iteration 111, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:20] {2391} INFO -  at 20.3s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:20] {2218} INFO - iteration 112, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:20] {2391} INFO -  at 20.6s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:20] {2218} INFO - iteration 113, current
```

```
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:20] {2391} INFO -  at 20.8s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:20] {2218} INFO - iteration 114, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:21] {2391} INFO -  at 21.1s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:21] {2218} INFO - iteration 115, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:21] {2391} INFO -  at 21.3s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:21] {2218} INFO - iteration 116, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:21] {2391} INFO -  at 21.7s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:21] {2218} INFO - iteration 117, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:21] {2391} INFO -  at 21.9s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:21] {2218} INFO - iteration 118, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:22] {2391} INFO -  at 22.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:22] {2218} INFO - iteration 119, current
learner lgbm
[flaml.automl.logger: 04-25 17:07:22] {2391} INFO -  at 22.5s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:22] {2218} INFO - iteration 120, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:22] {2391} INFO -  at 22.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:22] {2218} INFO - iteration 121, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:22] {2391} INFO -  at 22.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:22] {2218} INFO - iteration 122, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:23] {2391} INFO -  at 23.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:23] {2218} INFO - iteration 123, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:23] {2391} INFO -  at 23.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:23] {2218} INFO - iteration 124, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:23] {2391} INFO -  at 23.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:23] {2218} INFO - iteration 125, current
```

```
learner xgboost
[flaml.automl.logger: 04-25 17:07:23] {2391} INFO -  at 23.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:23] {2218} INFO - iteration 126, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:23] {2391} INFO -  at 24.1s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:23] {2218} INFO - iteration 127, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:24] {2391} INFO -  at 24.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:24] {2218} INFO - iteration 128, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:24] {2391} INFO -  at 24.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:24] {2218} INFO - iteration 129, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:24] {2391} INFO -  at 24.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:24] {2218} INFO - iteration 130, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:25] {2391} INFO -  at 25.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:25] {2218} INFO - iteration 131, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:25] {2391} INFO -  at 25.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:25] {2218} INFO - iteration 132, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:25] {2391} INFO -  at 25.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:25] {2218} INFO - iteration 133, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:25] {2391} INFO -  at 25.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:25] {2218} INFO - iteration 134, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:25] {2391} INFO -  at 26.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:25] {2218} INFO - iteration 135, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:26] {2391} INFO -  at 26.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:26] {2218} INFO - iteration 136, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:26] {2391} INFO -  at 26.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:26] {2218} INFO - iteration 137, current
```

```
learner xgboost
[flaml.automl.logger: 04-25 17:07:26] {2391} INFO -  at 26.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:26] {2218} INFO - iteration 138, current
learner rf
[flaml.automl.logger: 04-25 17:07:26] {2391} INFO -  at 27.0s,  estimator rf's
best error=0.0284,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:26] {2218} INFO - iteration 139, current
learner rf
[flaml.automl.logger: 04-25 17:07:27] {2391} INFO -  at 27.2s,  estimator rf's
best error=0.0284,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:27] {2218} INFO - iteration 140, current
learner rf
[flaml.automl.logger: 04-25 17:07:27] {2391} INFO -  at 27.4s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:27] {2218} INFO - iteration 141, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:27] {2391} INFO -  at 27.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:27] {2218} INFO - iteration 142, current
learner rf
[flaml.automl.logger: 04-25 17:07:27] {2391} INFO -  at 27.9s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:27] {2218} INFO - iteration 143, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:27] {2391} INFO -  at 28.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:27] {2218} INFO - iteration 144, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:28] {2391} INFO -  at 28.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:28] {2218} INFO - iteration 145, current
learner rf
[flaml.automl.logger: 04-25 17:07:28] {2391} INFO -  at 28.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:28] {2218} INFO - iteration 146, current
learner rf
[flaml.automl.logger: 04-25 17:07:28] {2391} INFO -  at 28.7s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:28] {2218} INFO - iteration 147, current
learner rf
[flaml.automl.logger: 04-25 17:07:28] {2391} INFO -  at 28.9s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:28] {2218} INFO - iteration 148, current
learner rf
[flaml.automl.logger: 04-25 17:07:28] {2391} INFO -  at 29.0s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:28] {2218} INFO - iteration 149, current
```

```
learner rf
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 29.2s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 150, current
learner rf
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 29.4s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 151, current
learner rf
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 29.6s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 152, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 29.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 153, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 29.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 154, current
learner rf
[flaml.automl.logger: 04-25 17:07:29] {2391} INFO -  at 30.1s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:29] {2218} INFO - iteration 155, current
learner rf
[flaml.automl.logger: 04-25 17:07:30] {2391} INFO -  at 30.3s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:30] {2218} INFO - iteration 156, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:30] {2391} INFO -  at 30.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:30] {2218} INFO - iteration 157, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:30] {2391} INFO -  at 30.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:30] {2218} INFO - iteration 158, current
learner rf
[flaml.automl.logger: 04-25 17:07:30] {2391} INFO -  at 30.9s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:30] {2218} INFO - iteration 159, current
learner rf
[flaml.automl.logger: 04-25 17:07:30] {2391} INFO -  at 31.1s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:30] {2218} INFO - iteration 160, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:31] {2391} INFO -  at 31.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:31] {2218} INFO - iteration 161, current
```

```
learner rf
[flaml.automl.logger: 04-25 17:07:31] {2391} INFO -  at 31.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:31] {2218} INFO - iteration 162, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:31] {2391} INFO -  at 31.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:31] {2218} INFO - iteration 163, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:32] {2391} INFO -  at 32.1s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:32] {2218} INFO - iteration 164, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:32] {2391} INFO -  at 32.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:32] {2218} INFO - iteration 165, current
learner rf
[flaml.automl.logger: 04-25 17:07:32] {2391} INFO -  at 32.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:32] {2218} INFO - iteration 166, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:32] {2391} INFO -  at 32.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:32] {2218} INFO - iteration 167, current
learner lrl1
[flaml.automl.logger: 04-25 17:07:32] {2391} INFO -  at 33.0s,  estimator lrl1's
best error=0.0270,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:32] {2218} INFO - iteration 168, current
learner lrl1

/opt/anaconda3/envs/Project/lib/python3.8/site-
packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was
reached which means the coef_ did not converge
  warnings.warn(
/opt/anaconda3/envs/Project/lib/python3.8/site-
packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was
reached which means the coef_ did not converge
  warnings.warn(
/opt/anaconda3/envs/Project/lib/python3.8/site-
packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was
reached which means the coef_ did not converge
  warnings.warn(
/opt/anaconda3/envs/Project/lib/python3.8/site-
packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was
reached which means the coef_ did not converge
  warnings.warn(

[flaml.automl.logger: 04-25 17:07:33] {2391} INFO -  at 33.2s,  estimator lrl1's
best error=0.0256,     best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:33] {2218} INFO - iteration 169, current
learner lrl1
[flaml.automl.logger: 04-25 17:07:33] {2391} INFO -  at 33.3s,  estimator lrl1's
best error=0.0256,     best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:33] {2218} INFO - iteration 170, current
learner rf
[flaml.automl.logger: 04-25 17:07:33] {2391} INFO -  at 33.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:33] {2218} INFO - iteration 171, current
learner rf
[flaml.automl.logger: 04-25 17:07:33] {2391} INFO -  at 33.6s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:33] {2218} INFO - iteration 172, current
learner rf
[flaml.automl.logger: 04-25 17:07:33] {2391} INFO -  at 33.9s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:33] {2218} INFO - iteration 173, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:34] {2391} INFO -  at 34.1s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:34] {2218} INFO - iteration 174, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:34] {2391} INFO -  at 34.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:34] {2218} INFO - iteration 175, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:34] {2391} INFO -  at 34.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:34] {2218} INFO - iteration 176, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:34] {2391} INFO -  at 34.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:34] {2218} INFO - iteration 177, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:07:34] {2391} INFO -  at 34.9s,  estimator
xgb_limitdepth's best error=0.0242,   best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:34] {2218} INFO - iteration 178, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:35] {2391} INFO -  at 35.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:35] {2218} INFO - iteration 179, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:35] {2391} INFO -  at 35.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:35] {2218} INFO - iteration 180, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:36] {2391} INFO -  at 36.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:36] {2218} INFO - iteration 181, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:36] {2391} INFO -  at 36.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:36] {2218} INFO - iteration 182, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:36] {2391} INFO -  at 36.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:36] {2218} INFO - iteration 183, current
learner rf
[flaml.automl.logger: 04-25 17:07:36] {2391} INFO -  at 37.0s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:36] {2218} INFO - iteration 184, current
learner rf
[flaml.automl.logger: 04-25 17:07:37] {2391} INFO -  at 37.2s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:37] {2218} INFO - iteration 185, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:37] {2391} INFO -  at 37.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:37] {2218} INFO - iteration 186, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:37] {2391} INFO -  at 37.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:37] {2218} INFO - iteration 187, current
learner rf
[flaml.automl.logger: 04-25 17:07:37] {2391} INFO -  at 38.0s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:37] {2218} INFO - iteration 188, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:38] {2391} INFO -  at 38.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:38] {2218} INFO - iteration 189, current
learner rf
[flaml.automl.logger: 04-25 17:07:38] {2391} INFO -  at 38.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:38] {2218} INFO - iteration 190, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:38] {2391} INFO -  at 38.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:38] {2218} INFO - iteration 191, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:39] {2391} INFO -  at 39.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:39] {2218} INFO - iteration 192, current
learner rf
[flaml.automl.logger: 04-25 17:07:39] {2391} INFO -  at 39.5s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:39] {2218} INFO - iteration 193, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:39] {2391} INFO -  at 39.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:39] {2218} INFO - iteration 194, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:40] {2391} INFO -  at 40.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:40] {2218} INFO - iteration 195, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:40] {2391} INFO -  at 40.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:40] {2218} INFO - iteration 196, current
learner rf
[flaml.automl.logger: 04-25 17:07:40] {2391} INFO -  at 40.7s,  estimator rf's
best error=0.0270,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:40] {2218} INFO - iteration 197, current
learner rf
[flaml.automl.logger: 04-25 17:07:40] {2391} INFO -  at 41.0s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:40] {2218} INFO - iteration 198, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:41] {2391} INFO -  at 41.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:41] {2218} INFO - iteration 199, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:41] {2391} INFO -  at 41.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:41] {2218} INFO - iteration 200, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:41] {2391} INFO -  at 41.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:41] {2218} INFO - iteration 201, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:42] {2391} INFO -  at 42.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:42] {2218} INFO - iteration 202, current
learner rf
[flaml.automl.logger: 04-25 17:07:42] {2391} INFO -  at 42.7s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:42] {2218} INFO - iteration 203, current
learner rf
[flaml.automl.logger: 04-25 17:07:42] {2391} INFO -  at 43.1s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:42] {2218} INFO - iteration 204, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:43] {2391} INFO -  at 43.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:43] {2218} INFO - iteration 205, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:43] {2391} INFO -  at 43.9s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:43] {2218} INFO - iteration 206, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:44] {2391} INFO -  at 44.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:44] {2218} INFO - iteration 207, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:44] {2391} INFO -  at 44.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:44] {2218} INFO - iteration 208, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:44] {2391} INFO -  at 45.0s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:44] {2218} INFO - iteration 209, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:45] {2391} INFO -  at 45.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:45] {2218} INFO - iteration 210, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:45] {2391} INFO -  at 45.5s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:45] {2218} INFO - iteration 211, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:46] {2391} INFO -  at 46.2s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:46] {2218} INFO - iteration 212, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:46] {2391} INFO -  at 46.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:46] {2218} INFO - iteration 213, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:46] {2391} INFO -  at 46.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:46] {2218} INFO - iteration 214, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:46] {2391} INFO -  at 46.8s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:46] {2218} INFO - iteration 215, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:47] {2391} INFO -  at 47.1s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:47] {2218} INFO - iteration 216, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:47] {2391} INFO -  at 47.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:47] {2218} INFO - iteration 217, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:47] {2391} INFO -  at 47.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:47] {2218} INFO - iteration 218, current
learner rf
[flaml.automl.logger: 04-25 17:07:47] {2391} INFO -  at 47.8s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:47] {2218} INFO - iteration 219, current
learner rf
[flaml.automl.logger: 04-25 17:07:47] {2391} INFO -  at 48.1s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:47] {2218} INFO - iteration 220, current
learner rf
[flaml.automl.logger: 04-25 17:07:48] {2391} INFO -  at 48.3s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:48] {2218} INFO - iteration 221, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:48] {2391} INFO -  at 48.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:48] {2218} INFO - iteration 222, current
learner rf
[flaml.automl.logger: 04-25 17:07:48] {2391} INFO -  at 49.0s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:48] {2218} INFO - iteration 223, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:49] {2391} INFO -  at 49.3s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:49] {2218} INFO - iteration 224, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:49] {2391} INFO -  at 49.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:49] {2218} INFO - iteration 225, current
learner rf
[flaml.automl.logger: 04-25 17:07:49] {2391} INFO -  at 49.9s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:49] {2218} INFO - iteration 226, current
learner rf
[flaml.automl.logger: 04-25 17:07:50] {2391} INFO -  at 50.1s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:50] {2218} INFO - iteration 227, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:50] {2391} INFO -  at 50.6s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:50] {2218} INFO - iteration 228, current
learner rf
[flaml.automl.logger: 04-25 17:07:50] {2391} INFO -  at 50.9s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0199
```

```
[flaml.automl.logger: 04-25 17:07:50] {2218} INFO - iteration 229, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:51] {2391} INFO -  at 51.4s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:51] {2218} INFO - iteration 230, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:51] {2391} INFO -  at 51.7s,  estimator
xgboost's best error=0.0199,  best estimator xgboost's best error=0.0199
[flaml.automl.logger: 04-25 17:07:51] {2218} INFO - iteration 231, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:52] {2391} INFO -  at 52.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:52] {2218} INFO - iteration 232, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:52] {2391} INFO -  at 52.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:52] {2218} INFO - iteration 233, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:52] {2391} INFO -  at 52.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:52] {2218} INFO - iteration 234, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:53] {2391} INFO -  at 53.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:53] {2218} INFO - iteration 235, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:53] {2391} INFO -  at 53.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:53] {2218} INFO - iteration 236, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:54] {2391} INFO -  at 54.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:54] {2218} INFO - iteration 237, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:54] {2391} INFO -  at 54.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:54] {2218} INFO - iteration 238, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:55] {2391} INFO -  at 56.0s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:55] {2218} INFO - iteration 239, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:56] {2391} INFO -  at 56.4s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:56] {2218} INFO - iteration 240, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:56] {2391} INFO -  at 56.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:07:56] {2218} INFO - iteration 241, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:57] {2391} INFO -  at 57.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:57] {2218} INFO - iteration 242, current
learner rf
[flaml.automl.logger: 04-25 17:07:57] {2391} INFO -  at 57.4s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:57] {2218} INFO - iteration 243, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:57] {2391} INFO -  at 57.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:57] {2218} INFO - iteration 244, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:58] {2391} INFO -  at 58.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:58] {2218} INFO - iteration 245, current
learner xgboost
[flaml.automl.logger: 04-25 17:07:58] {2391} INFO -  at 59.0s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:07:58] {2218} INFO - iteration 246, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:00] {2391} INFO -  at 60.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:00] {2218} INFO - iteration 247, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:00] {2391} INFO -  at 60.4s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:00] {2218} INFO - iteration 248, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:00] {2391} INFO -  at 60.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:00] {2218} INFO - iteration 249, current
learner rf
[flaml.automl.logger: 04-25 17:08:01] {2391} INFO -  at 61.2s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:01] {2218} INFO - iteration 250, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:01] {2391} INFO -  at 61.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:01] {2218} INFO - iteration 251, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:01] {2391} INFO -  at 62.1s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:01] {2218} INFO - iteration 252, current
learner rf
[flaml.automl.logger: 04-25 17:08:02] {2391} INFO -  at 62.4s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:02] {2218} INFO - iteration 253, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:02] {2391} INFO -  at 62.7s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:02] {2218} INFO - iteration 254, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:03] {2391} INFO -  at 63.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:03] {2218} INFO - iteration 255, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:03] {2391} INFO -  at 63.7s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:03] {2218} INFO - iteration 256, current
learner lgbm
[flaml.automl.logger: 04-25 17:08:03] {2391} INFO -  at 63.8s,  estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:03] {2218} INFO - iteration 257, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:04] {2391} INFO -  at 64.5s,  estimator
xgb_limitdepth's best error=0.0228,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:04] {2218} INFO - iteration 258, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:04] {2391} INFO -  at 64.8s,  estimator
xgb_limitdepth's best error=0.0228,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:04] {2218} INFO - iteration 259, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:05] {2391} INFO -  at 65.8s,  estimator
xgb_limitdepth's best error=0.0228,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:05] {2218} INFO - iteration 260, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:06] {2391} INFO -  at 66.5s,  estimator
xgb_limitdepth's best error=0.0228,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:06] {2218} INFO - iteration 261, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:07] {2391} INFO -  at 67.1s,  estimator
xgb_limitdepth's best error=0.0228,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:07] {2218} INFO - iteration 262, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:07] {2391} INFO -  at 67.8s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:07] {2218} INFO - iteration 263, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:08] {2391} INFO -  at 68.4s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:08] {2218} INFO - iteration 264, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:08] {2391} INFO -  at 69.0s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:08] {2218} INFO - iteration 265, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:09] {2391} INFO -  at 69.4s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:09] {2218} INFO - iteration 266, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:09] {2391} INFO -  at 69.8s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:09] {2218} INFO - iteration 267, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:10] {2391} INFO -  at 70.2s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:10] {2218} INFO - iteration 268, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:10] {2391} INFO -  at 70.5s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:10] {2218} INFO - iteration 269, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:10] {2391} INFO -  at 70.8s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:10] {2218} INFO - iteration 270, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:11] {2391} INFO -  at 71.5s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:11] {2218} INFO - iteration 271, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:11] {2391} INFO -  at 71.7s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:11] {2218} INFO - iteration 272, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:12] {2391} INFO -  at 72.7s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:12] {2218} INFO - iteration 273, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:12] {2391} INFO -  at 72.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:12] {2218} INFO - iteration 274, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:13] {2391} INFO -  at 73.8s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:13] {2218} INFO - iteration 275, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:14] {2391} INFO -  at 74.8s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:14] {2218} INFO - iteration 276, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:14] {2391} INFO -  at 75.1s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:14] {2218} INFO - iteration 277, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:15] {2391} INFO -  at 75.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:15] {2218} INFO - iteration 278, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:15] {2391} INFO -  at 75.7s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:15] {2218} INFO - iteration 279, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:16] {2391} INFO -  at 76.8s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:16] {2218} INFO - iteration 280, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:17] {2391} INFO -  at 77.5s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:17] {2218} INFO - iteration 281, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:17] {2391} INFO -  at 77.9s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:17] {2218} INFO - iteration 282, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:18] {2391} INFO -  at 78.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:18] {2218} INFO - iteration 283, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:18] {2391} INFO -  at 79.1s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:18] {2218} INFO - iteration 284, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:19] {2391} INFO -  at 79.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:19] {2218} INFO - iteration 285, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:19] {2391} INFO -  at 79.8s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:19] {2218} INFO - iteration 286, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:20] {2391} INFO -  at 80.2s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:20] {2218} INFO - iteration 287, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:20] {2391} INFO -  at 80.6s,  estimator
xgb_limitdepth's best error=0.0213,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:20] {2218} INFO - iteration 288, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:21] {2391} INFO -  at 81.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:21] {2218} INFO - iteration 289, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:21] {2391} INFO -  at 81.8s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:21] {2218} INFO - iteration 290, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:22] {2391} INFO -  at 82.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:22] {2218} INFO - iteration 291, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:22] {2391} INFO -  at 82.7s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:22] {2218} INFO - iteration 292, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:23] {2391} INFO -  at 83.4s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:23] {2218} INFO - iteration 293, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:23] {2391} INFO -  at 83.7s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:23] {2218} INFO - iteration 294, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:24] {2391} INFO -  at 84.2s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:24] {2218} INFO - iteration 295, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:24] {2391} INFO -  at 84.8s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:24] {2218} INFO - iteration 296, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:25] {2391} INFO -  at 85.8s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:25] {2218} INFO - iteration 297, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:26] {2391} INFO -  at 86.3s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:26] {2218} INFO - iteration 298, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:26] {2391} INFO -  at 86.7s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:26] {2218} INFO - iteration 299, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:27] {2391} INFO -  at 87.2s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:27] {2218} INFO - iteration 300, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:27] {2391} INFO -  at 87.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:27] {2218} INFO - iteration 301, current
learner rf
[flaml.automl.logger: 04-25 17:08:27] {2391} INFO -  at 88.0s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:27] {2218} INFO - iteration 302, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:28] {2391} INFO -  at 88.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:28] {2218} INFO - iteration 303, current
learner rf
[flaml.automl.logger: 04-25 17:08:28] {2391} INFO -  at 88.6s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:28] {2218} INFO - iteration 304, current
learner xgb_limitdepth
[flaml.automl.logger: 04-25 17:08:29] {2391} INFO -  at 89.4s,  estimator
xgb_limitdepth's best error=0.0199,   best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:29] {2218} INFO - iteration 305, current
learner rf
[flaml.automl.logger: 04-25 17:08:29] {2391} INFO -  at 89.7s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:29] {2218} INFO - iteration 306, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:30] {2391} INFO -  at 90.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:30] {2218} INFO - iteration 307, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:30] {2391} INFO -  at 90.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:30] {2218} INFO - iteration 308, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:30] {2391} INFO -  at 91.1s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:30] {2218} INFO - iteration 309, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:31] {2391} INFO -  at 91.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:31] {2218} INFO - iteration 310, current
learner rf
[flaml.automl.logger: 04-25 17:08:31] {2391} INFO -  at 91.9s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:31] {2218} INFO - iteration 311, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:32] {2391} INFO -  at 92.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:32] {2218} INFO - iteration 312, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:32] {2391} INFO -  at 92.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

```
[flaml.automl.logger: 04-25 17:08:32] {2218} INFO - iteration 313, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:33] {2391} INFO -  at 93.4s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:33] {2218} INFO - iteration 314, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:33] {2391} INFO -  at 93.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:33] {2218} INFO - iteration 315, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:34] {2391} INFO -  at 94.3s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:34] {2218} INFO - iteration 316, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:34] {2391} INFO -  at 94.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:34] {2218} INFO - iteration 317, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:34] {2391} INFO -  at 95.0s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:34] {2218} INFO - iteration 318, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:35] {2391} INFO -  at 95.4s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:35] {2218} INFO - iteration 319, current
learner rf
[flaml.automl.logger: 04-25 17:08:35] {2391} INFO -  at 95.6s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:35] {2218} INFO - iteration 320, current
learner rf
[flaml.automl.logger: 04-25 17:08:35] {2391} INFO -  at 95.9s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:35] {2218} INFO - iteration 321, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:36] {2391} INFO -  at 96.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:36] {2218} INFO - iteration 322, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:36] {2391} INFO -  at 96.9s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:36] {2218} INFO - iteration 323, current
learner rf
[flaml.automl.logger: 04-25 17:08:37] {2391} INFO -  at 97.2s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:37] {2218} INFO - iteration 324, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:37] {2391} INFO -  at 97.6s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
```

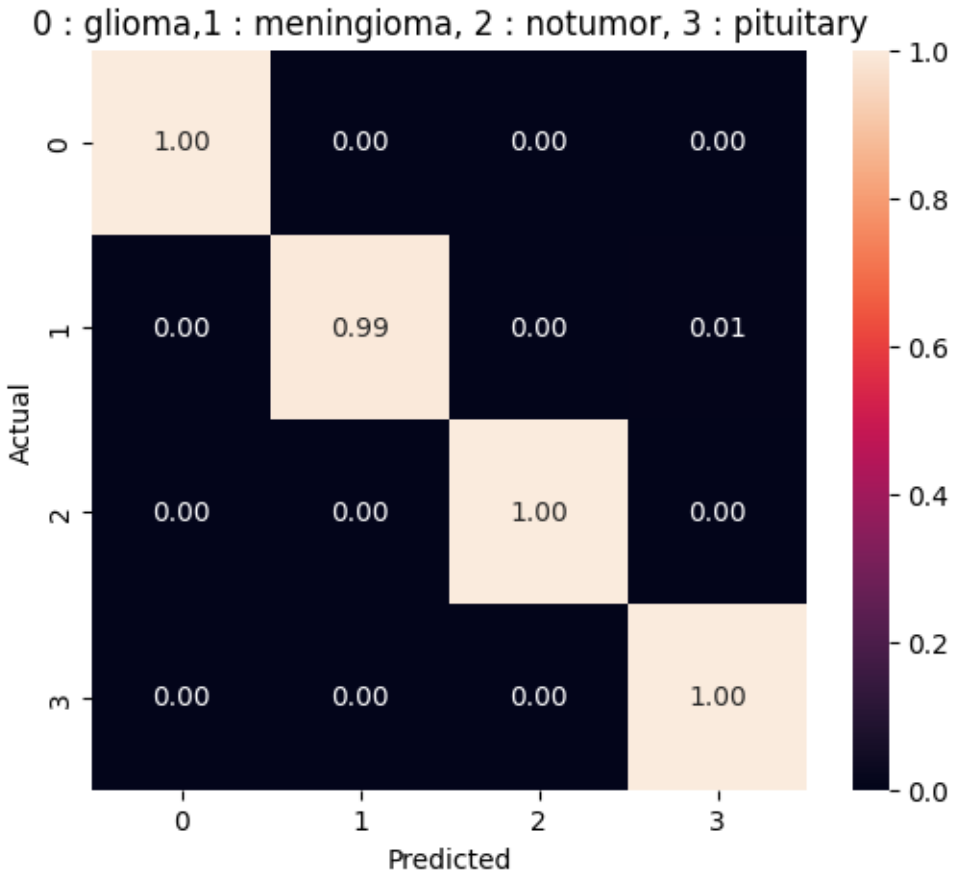
```
[flaml.automl.logger: 04-25 17:08:37] {2218} INFO - iteration 325, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:38] {2391} INFO -  at 98.4s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:38] {2218} INFO - iteration 326, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:38] {2391} INFO -  at 98.7s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:38] {2218} INFO - iteration 327, current
learner xgboost
[flaml.automl.logger: 04-25 17:08:39] {2391} INFO -  at 99.5s,  estimator
xgboost's best error=0.0185,  best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:39] {2218} INFO - iteration 328, current
learner rf
[flaml.automl.logger: 04-25 17:08:39] {2391} INFO -  at 99.9s,  estimator rf's
best error=0.0256,       best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:39] {2218} INFO - iteration 329, current
learner lgbm
[flaml.automl.logger: 04-25 17:08:39] {2391} INFO -  at 100.0s, estimator lgbm's
best error=0.0213,     best estimator xgboost's best error=0.0185
[flaml.automl.logger: 04-25 17:08:39] {2627} INFO - retrain xgboost for 0.1s
[flaml.automl.logger: 04-25 17:08:40] {2630} INFO - retrained model:
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=[],
              colsample_bylevel=0.7222010785416154, colsample_bynode=1,
              colsample_bytree=0.8600840124935673, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=0, gpu_id=-1, grow_policy='lossguide', importance_type=None,
              interaction_constraints='', learning_rate=0.055437673600423176,
              max_bin=256, max_cat_threshold=64, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=0, max_leaves=14,
              min_child_weight=0.05664523153071309, missing=nan,
              monotone_constraints='()', n_estimators=18, n_jobs=-1,
              num_parallel_tree=1, objective='multi:softprob', predictor='auto',
…)
[flaml.automl.logger: 04-25 17:08:40] {1930} INFO - fit succeeded
[flaml.automl.logger: 04-25 17:08:40] {1931} INFO - Time taken to find the best
model: 52.18394494056702
```

[114]:
```
pred5=automl.predict(X_data)
```

[124]:
```
acc1=accuracy_score(np.argmax(pred1, axis=1),np.argmax(y_test, axis=1))
acc2=accuracy_score(np.argmax(pred2, axis=1),np.argmax(y_test, axis=1))
acc3=accuracy_score(np.argmax(pred3, axis=1),np.argmax(y_test, axis=1))
acc4=accuracy_score(np.argmax(pred4, axis=1),np.argmax(y_test, axis=1))
acc5=accuracy_score(pred5,np.argmax(y_test, axis=1))
acc1,acc2,acc3,acc4,acc5
```

[124]:
```
(0.968705547652916,
 0.9445234708392604,
 0.9274537695590327,
 0.9274537695590327,
 0.9985775248933144)
```

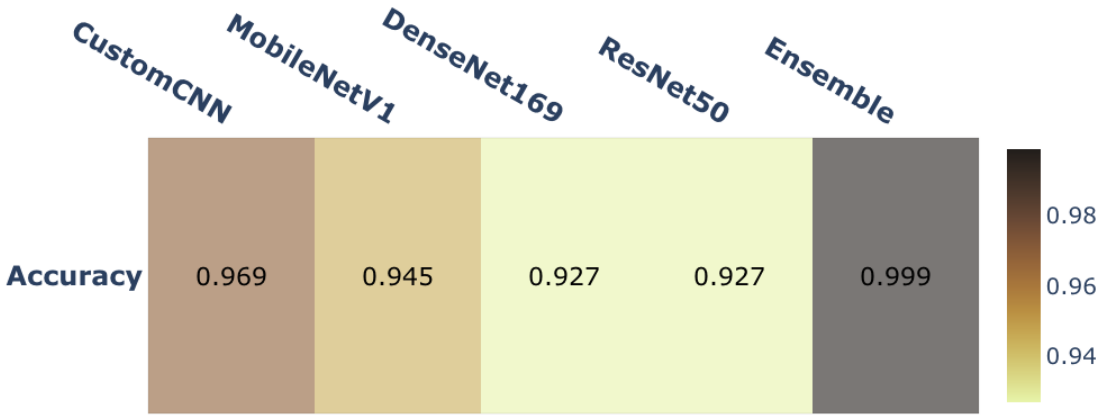
[115]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(y_data, pred5)
# Normalise
cmn = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(cmn, annot=True, fmt='.2f')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('0 : glioma,1 : meningioma, 2 : notumor, 3 : pituitary')
plt.show(block=False)
```



[129]:
```
import numpy as np
import plotly.graph_objects as go
from functools import reduce
from itertools import product
from IPython.display import Image
SUB = str.maketrans("0123456789", "      ")
SUP = str.maketrans("0123456789", " ¹²³    ")
z=[ [np.round(acc1,3),np.round(acc2,3),np.round(acc3,3),np.round(acc4,3),np.
 ↪round(acc5,3)]]
x=['<b>CustomCNN</b>', '<b>MobileNetV1</b>', '<b>DenseNet169</b>',␣
 ↪'<b>ResNet50</b>', '<b>Ensemble</b>']
y=['<b>Accuracy</b>']

def get_anno_text(z_value):
    annotations=[]
    a, b = len(z_value), len(z_value[0])
    flat_z = reduce(lambda x,y: x+y, z_value) # z_value.flat if you deal with␣
 ↪numpy
    coords = product(range(a), range(b))
    for pos, elem in zip(coords, flat_z):
        annotations.append({'font': {'color': 'black'},
                    'showarrow': False,
                    'text': str(elem),
                    'x': pos[1],
                    'y': pos[0],
                        'font.size':22   })
    return annotations

fig = go.Figure(data=go.Heatmap(
                    z=z,
                    x=x,
                    y=y,
                    hoverongaps = True, colorscale ='turbid',
    opacity=0.6,colorbar=dict(tickfont=dict(size=20)) ))#matter#

fig.update_layout(title={'text': "",
        'y':0.8,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
           plot_bgcolor='rgba(0,0,0,0)',
    annotations = get_anno_text(z),
                  width=1000,
height=400,xaxis={'side': 'top'},margin=dict(l=20, r=20, t=20, b=20))

fig.update_xaxes(tickfont = dict(size=24),linewidth=0.1, linecolor='black',
```

```
                    mirror=True)
fig.update_yaxes(tickfont = dict(size=24),linewidth=0.1, linecolor='black',

                    mirror=True)
fig.write_image("table2b.png",engine="kaleido")
#plt.savefig("table2a.pdf", format="pdf", bbox_inches="tight")
fig.show()
Image('table2b.png')
```

[129]:



[ ]: