```
%cd
shutil.rmtree('/content/hands_dataset') #non usarloo
```

```
     /root
```

```
%cd
%cd ../content
!pwd
```

```
     /root
     /content
     /content
```

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dense, Flatten, BatchNormalization, Conv2D, M
from tensorflow.keras.optimizers import Adam, SGD, RMSprop
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix
from tensorflow.keras.models import Model
from sklearn.metrics import confusion_matrix
import itertools
import itertools
import os
import shutil
import random
import glob
import matplotlib.pyplot as plt
import warnings
from keras.callbacks import ModelCheckpoint
```

```
mobile = keras.applications.mobilenet.MobileNet()
```

```
#mobile.summary()
```

```
!pwd

!git clone https://github.com/ma-tesi/hands_dataset.git
```

```
     /content
     Cloning into 'hands_dataset'...
     remote: Enumerating objects: 2696, done.
     remote: Total 2696 (delta 0), reused 0 (delta 0), pack-reused 2696
     Receiving objects: 100% (2696/2696), 22.06 MiB | 21.95 MiB/s, done.
     Resolving deltas: 100% (863/863), done.
```

```
%cd hands_dataset
!ls
```

```
     /content/hands_dataset
     Dataset  Examples  LICENSE  README.md
```

```
!rm -rf Examples  LICENSE  README.md
!ls
```

```
     Dataset
```

```
dir_path = os.path.dirname(os.path.realpath('FT_mobilenet.ipynb'))
print(dir_path)
```

```
     /content/hands_dataset
```

```
!pwd
```

```
     /content/hands_dataset
```

```
%cd /content/hands_dataset/Dataset
%mkdir train
%mkdir test
%mkdir valid
%mv 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/ train/
```

```
     /content/hands_dataset/Dataset
     mv: cannot move '/' to 'train': Device or resource busy
```

```
#%cd -
!pwd
%cd valid
%mkdir 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/
%cd ../test
%mkdir 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/
```

```
     /content/hands_dataset/Dataset
     /content/hands_dataset/Dataset/valid
     mkdir: cannot create directory ‘/’: File exists
     /content/hands_dataset/Dataset/test
     mkdir: cannot create directory ‘/’: File exists
```

```
!pwd
```

```
     /content/hands_dataset/Dataset/test
```

```
%%bash
cd ../train
for ((i=0; i<=9; i++)); do
  a=$(find $i/ -type f | shuf -n 30)
  mv $a ../valid/$i/
  b=$(find $i/ -type f | shuf -n 5)
  mv $b ../test/$i/
done
```

```
%cd ../..
!pwd
```

```
     /content/hands_dataset
     /content/hands_dataset
```

```
train_path = 'Dataset/train'
valid_path = 'Dataset/valid'
test_path = 'Dataset/test'
```

```
train_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.prepro
    train_path, target_size=(224, 224), batch_size=16)
valid_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.prepro
    valid_path, target_size=(224, 224), batch_size=16)
test_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.preproc
    test_path, target_size=(224, 224), batch_size=10, shuffle=False)
```

```
     Found 2172 images belonging to 10 classes.
     Found 300 images belonging to 10 classes.
     Found 50 images belonging to 10 classes.
```

```
mobile = keras.applications.mobilenet.MobileNet()
mobile.summary()
```

```
     Model: "mobilenet_1.00_224"
     _________________________________________________________________
     Layer (type)                 Output Shape              Param #
     =================================================================
     input_4 (InputLayer)         [(None, 224, 224, 3)]     0
     _________________________________________________________________
     conv1 (Conv2D)               (None, 112, 112, 32)      864
     _________________________________________________________________
     conv1_bn (BatchNormalization (None, 112, 112, 32)      128
     _________________________________________________________________
     conv1_relu (ReLU)            (None, 112, 112, 32)      0
     _________________________________________________________________
     conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288
```

```
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512
_________________________________________________________________
conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768
```

```
x = mobile.layers[-6].output
# we take out only the last 6 layers
predictions = Dense(10, activation='softmax')(x) #si va ad aggiungere dopo l'ultimo global av
# we add the output layer
SGDmodel = Model(inputs=mobile.input, outputs=predictions)
```

```
for layer in SGDmodel.layers:
    layer.trainable = True
```

```
SGDmodel.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
conv1 (Conv2D)               (None, 112, 112, 32)      864
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 32)      128
_________________________________________________________________
conv1_relu (ReLU)            (None, 112, 112, 32)      0
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0
```

```
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152

conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512

conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0

conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768
```

SGD OPT

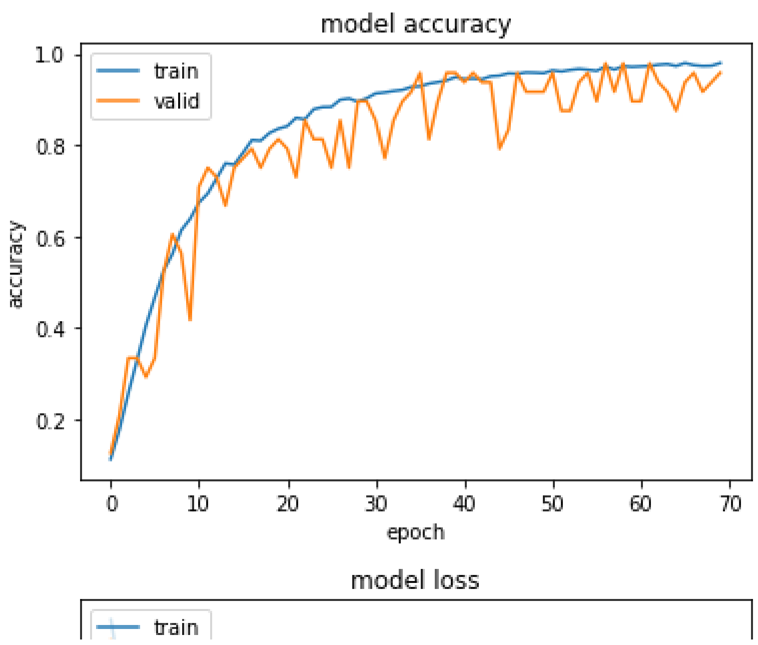
```
SGDmodel.compile(SGD(learning_rate=1e-5, momentum=0.9), loss='categorical_crossentropy', metr
steps=2172/16
```

```
mc = ModelCheckpoint('SGDbest_model.h5', monitor='val_loss', mode='min', save_best_only=True)
history = SGDmodel.fit_generator(train_batches, steps_per_epoch=steps,validation_data=valid_b
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:194
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/70
135/135 - 47s - loss: 2.7239 - accuracy: 0.1114 - val_loss: 2.5985 - val_accuracy: 0.
Epoch 2/70
135/135 - 12s - loss: 2.3809 - accuracy: 0.1763 - val_loss: 2.3019 - val_accuracy: 0.
Epoch 3/70
135/135 - 13s - loss: 2.1399 - accuracy: 0.2555 - val_loss: 2.1891 - val_accuracy: 0.
Epoch 4/70
135/135 - 12s - loss: 1.9531 - accuracy: 0.3297 - val_loss: 2.0128 - val_accuracy: 0.
Epoch 5/70
135/135 - 12s - loss: 1.7915 - accuracy: 0.4052 - val_loss: 1.8204 - val_accuracy: 0.
Epoch 6/70
135/135 - 12s - loss: 1.6647 - accuracy: 0.4664 - val_loss: 1.7655 - val_accuracy: 0.
Epoch 7/70
135/135 - 12s - loss: 1.5518 - accuracy: 0.5239 - val_loss: 1.5883 - val_accuracy: 0.
Epoch 8/70
135/135 - 12s - loss: 1.4736 - accuracy: 0.5608 - val_loss: 1.5839 - val_accuracy: 0.
Epoch 9/70
135/135 - 12s - loss: 1.3697 - accuracy: 0.6128 - val_loss: 1.3638 - val_accuracy: 0.
Epoch 10/70
135/135 - 12s - loss: 1.2893 - accuracy: 0.6377 - val_loss: 1.5037 - val_accuracy: 0.
Epoch 11/70
135/135 - 12s - loss: 1.2287 - accuracy: 0.6736 - val_loss: 1.1615 - val_accuracy: 0.
Epoch 12/70
135/135 - 12s - loss: 1.1599 - accuracy: 0.6934 - val_loss: 1.0313 - val_accuracy: 0.
Epoch 13/70
135/135 - 12s - loss: 1.0985 - accuracy: 0.7265 - val_loss: 1.1035 - val_accuracy: 0.
Epoch 14/70
135/135 - 12s - loss: 1.0382 - accuracy: 0.7597 - val_loss: 1.1577 - val_accuracy: 0.
Epoch 15/70
135/135 - 12s - loss: 1.0012 - accuracy: 0.7569 - val_loss: 1.0921 - val_accuracy: 0.
Epoch 16/70
135/135 - 12s - loss: 0.9490 - accuracy: 0.7831 - val_loss: 0.9486 - val_accuracy: 0.
Epoch 17/70
135/135 - 12s - loss: 0.8989 - accuracy: 0.8108 - val_loss: 0.9198 - val_accuracy: 0.
```

```
Epoch 18/70
135/135 - 12s - loss: 0.8719 - accuracy: 0.8094 - val_loss: 0.8027 - val_accuracy: 0.
Epoch 19/70
135/135 - 12s - loss: 0.8332 - accuracy: 0.8264 - val_loss: 0.8231 - val_accuracy: 0.
Epoch 20/70
135/135 - 12s - loss: 0.7982 - accuracy: 0.8356 - val_loss: 0.8274 - val_accuracy: 0.
Epoch 21/70
135/135 - 12s - loss: 0.7756 - accuracy: 0.8412 - val_loss: 0.8810 - val_accuracy: 0.
Epoch 22/70
135/135 - 12s - loss: 0.7414 - accuracy: 0.8596 - val_loss: 0.8670 - val_accuracy: 0.
Epoch 23/70
135/135 - 12s - loss: 0.7139 - accuracy: 0.8564 - val_loss: 0.7427 - val_accuracy: 0.
Epoch 24/70
135/135 - 12s - loss: 0.6808 - accuracy: 0.8785 - val_loss: 0.7376 - val_accuracy: 0.
Epoch 25/70
135/135 - 12s - loss: 0.6616 - accuracy: 0.8831 - val_loss: 0.8006 - val_accuracy: 0.
Epoch 26/70
135/135 - 12s - loss: 0.6424 - accuracy: 0.8835 - val_loss: 0.7772 - val_accuracy: 0.
Epoch 27/70
135/135 - 12s - loss: 0.6117 - accuracy: 0.8992 - val_loss: 0.5743 - val_accuracy: 0.
Epoch 28/70
```

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



```
test_labels = test_batches.classes
predictions = SGDmodel.predict_generator(test_batches, steps=5, verbose=0)
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:2001:
  warnings.warn('`Model.predict_generator` is deprecated and '
```

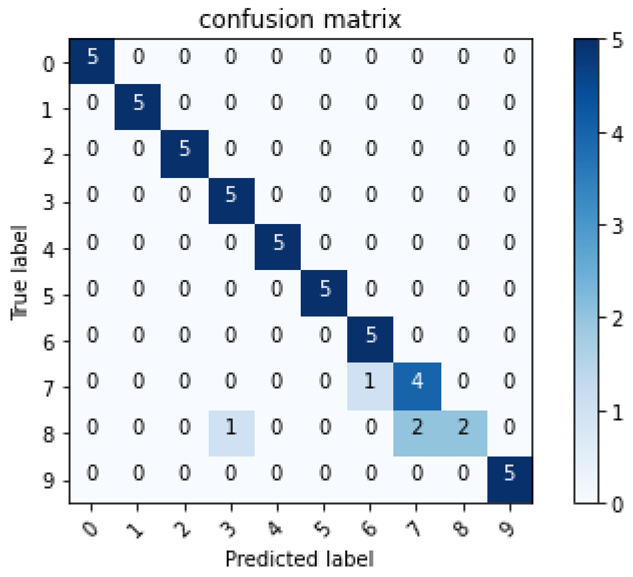
```
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

```
test_batches.class_indices
```

```
{'0': 0,
 '1': 1,
 '2': 2,
 '3': 3,
 '4': 4,
 '5': 5,
 '6': 6,
 '7': 7,
 '8': 8,
 '9': 9}
```

```
def plot_confusion_matrix(cm, classes,
                        normalize=False,
                        title='Confusion matrix',
                        cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
            horizontalalignment="center",
            color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


cm_plot_labels = ['0','1','2','3','4','5','6','7','8','9']
plot_confusion_matrix(cm, cm_plot_labels, title='confusion matrix')
```

```
Confusion matrix, without normalization
[[5 0 0 0 0 0 0 0 0 0]
 [0 5 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0]
 [0 0 0 0 0 5 0 0 0 0]
 [0 0 0 0 0 0 5 0 0 0]
 [0 0 0 0 0 0 1 4 0 0]
 [0 0 0 1 0 0 0 2 2 0]
 [0 0 0 0 0 0 0 0 0 5]]
```



```
SGDmodel.save('BDmobile_handsRGB_model.h5')
```

```
test_labels
```

```
array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4,
       4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8,
       8, 9, 9, 9, 9, 9], dtype=int32)
```

```
mobile = keras.applications.mobilenet.MobileNet()
mobile.summary()
```

```
Model: "mobilenet_1.00_224"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_5 (InputLayer)         [(None, 224, 224, 3)]     0         
_________________________________________________________________
conv1 (Conv2D)               (None, 112, 112, 32)      864       
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 32)      128       
_________________________________________________________________
conv1_relu (ReLU)            (None, 112, 112, 32)      0         
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288       
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128       
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0         
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048      
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256       
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0         
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0         
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576       
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256       
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0         
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192      
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512       
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0         
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152      
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512       
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0         
```

```
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384     
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512       
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0         
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0         
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152      
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512       
_________________________________________________________________
conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0         
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768     
```

```
x = mobile.layers[-6].output
# we take out only the last 6 layers
predictions = Dense(10, activation='softmax')(x) #si va ad aggiungere dopo l'ultimo global av
# we add the output layer
RMSmodel = Model(inputs=mobile.input, outputs=predictions)
```

```
for layer in RMSmodel.layers:
  layer.trainable = True
```

```
RMSmodel.compile(RMSprop(
    learning_rate=1e-7, rho=0.9, momentum=0.95, epsilon=1e-07), loss='categorical_crossentrop
steps=2172/16
```

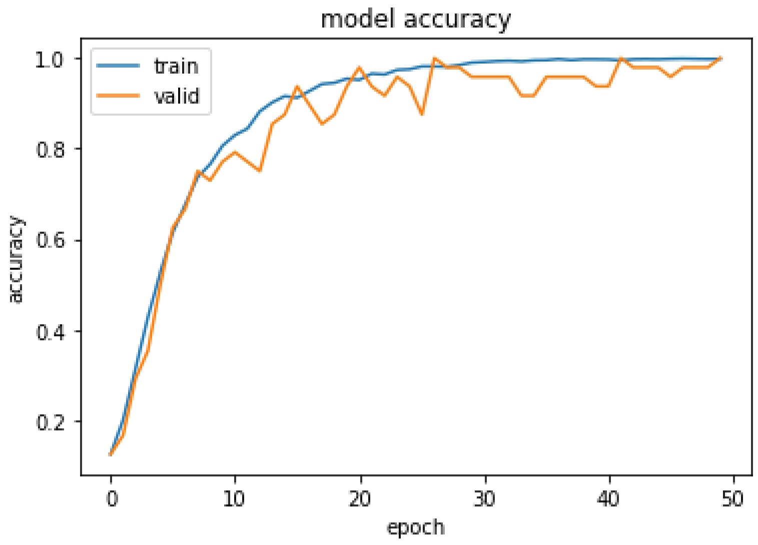
RMSprop

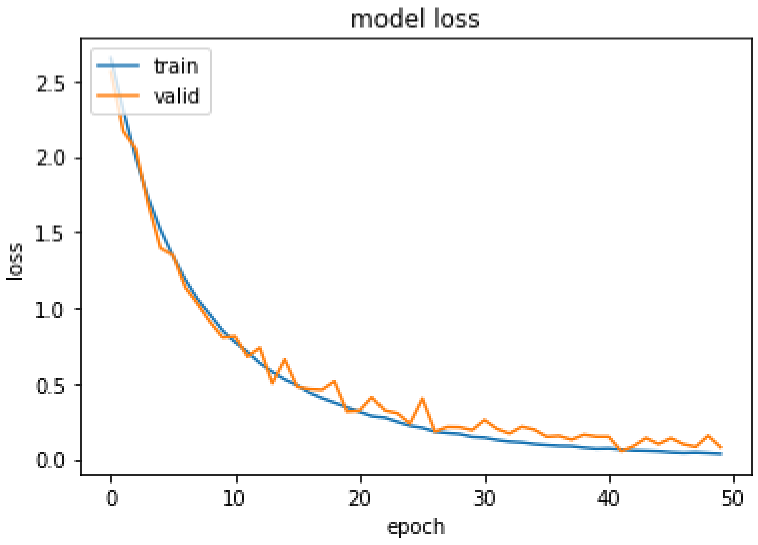
```
mc = ModelCheckpoint('RMSbest_model.h5', monitor='val_loss', mode='min', save_best_only=True)
history = RMSmodel.fit_generator(train_batches, steps_per_epoch=steps,validation_data=valid_b
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:194
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/50
135/135 - 15s - loss: 2.6544 - accuracy: 0.1252 - val_loss: 2.5616 - val_accuracy: 0.
Epoch 2/50
135/135 - 13s - loss: 2.3081 - accuracy: 0.2007 - val_loss: 2.1687 - val_accuracy: 0.
Epoch 3/50
135/135 - 13s - loss: 1.9949 - accuracy: 0.3140 - val_loss: 2.0536 - val_accuracy: 0.
Epoch 4/50
135/135 - 13s - loss: 1.7305 - accuracy: 0.4282 - val_loss: 1.6915 - val_accuracy: 0.
Epoch 5/50
135/135 - 13s - loss: 1.5167 - accuracy: 0.5267 - val_loss: 1.3957 - val_accuracy: 0.
Epoch 6/50
```

```
135/135 - 13s - loss: 1.3445 - accuracy: 0.6142 - val_loss: 1.3502 - val_accuracy: 0.
Epoch 7/50
135/135 - 13s - loss: 1.1892 - accuracy: 0.6763 - val_loss: 1.1368 - val_accuracy: 0.
Epoch 8/50
135/135 - 13s - loss: 1.0575 - accuracy: 0.7366 - val_loss: 1.0270 - val_accuracy: 0.
Epoch 9/50
135/135 - 13s - loss: 0.9569 - accuracy: 0.7652 - val_loss: 0.9077 - val_accuracy: 0.
Epoch 10/50
135/135 - 13s - loss: 0.8517 - accuracy: 0.8062 - val_loss: 0.8069 - val_accuracy: 0.
Epoch 11/50
135/135 - 13s - loss: 0.7751 - accuracy: 0.8297 - val_loss: 0.8147 - val_accuracy: 0.
Epoch 12/50
135/135 - 13s - loss: 0.7108 - accuracy: 0.8439 - val_loss: 0.6797 - val_accuracy: 0.
Epoch 13/50
135/135 - 13s - loss: 0.6362 - accuracy: 0.8821 - val_loss: 0.7386 - val_accuracy: 0.
Epoch 14/50
135/135 - 13s - loss: 0.5787 - accuracy: 0.9019 - val_loss: 0.5021 - val_accuracy: 0.
Epoch 15/50
135/135 - 13s - loss: 0.5283 - accuracy: 0.9157 - val_loss: 0.6615 - val_accuracy: 0.
Epoch 16/50
135/135 - 13s - loss: 0.4895 - accuracy: 0.9121 - val_loss: 0.4797 - val_accuracy: 0.
Epoch 17/50
135/135 - 13s - loss: 0.4398 - accuracy: 0.9273 - val_loss: 0.4642 - val_accuracy: 0.
Epoch 18/50
135/135 - 13s - loss: 0.4041 - accuracy: 0.9424 - val_loss: 0.4591 - val_accuracy: 0.
Epoch 19/50
135/135 - 13s - loss: 0.3741 - accuracy: 0.9452 - val_loss: 0.5163 - val_accuracy: 0.
Epoch 20/50
135/135 - 13s - loss: 0.3441 - accuracy: 0.9544 - val_loss: 0.3150 - val_accuracy: 0.
Epoch 21/50
135/135 - 13s - loss: 0.3143 - accuracy: 0.9517 - val_loss: 0.3208 - val_accuracy: 0.
Epoch 22/50
135/135 - 13s - loss: 0.2853 - accuracy: 0.9655 - val_loss: 0.4098 - val_accuracy: 0.
Epoch 23/50
135/135 - 13s - loss: 0.2742 - accuracy: 0.9636 - val_loss: 0.3220 - val_accuracy: 0.
Epoch 24/50
135/135 - 13s - loss: 0.2471 - accuracy: 0.9738 - val_loss: 0.3037 - val_accuracy: 0.
Epoch 25/50
135/135 - 13s - loss: 0.2214 - accuracy: 0.9751 - val_loss: 0.2354 - val_accuracy: 0.
Epoch 26/50
135/135 - 13s - loss: 0.2074 - accuracy: 0.9816 - val_loss: 0.4019 - val_accuracy: 0.
Epoch 27/50
135/135 - 12s - loss: 0.1830 - accuracy: 0.9816 - val_loss: 0.1827 - val_accuracy: 1.
Epoch 28/50
135/135 - 13s - loss: 0.1737 - accuracy: 0.9807 - val_loss: 0.2136 - val_accuracy: 0.
```

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





```
test_labels = test_batches.classes
predictions = RMSmodel.predict_generator(test_batches, steps=5, verbose=0)
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:2001:
  warnings.warn('`Model.predict_generator` is deprecated and '
```

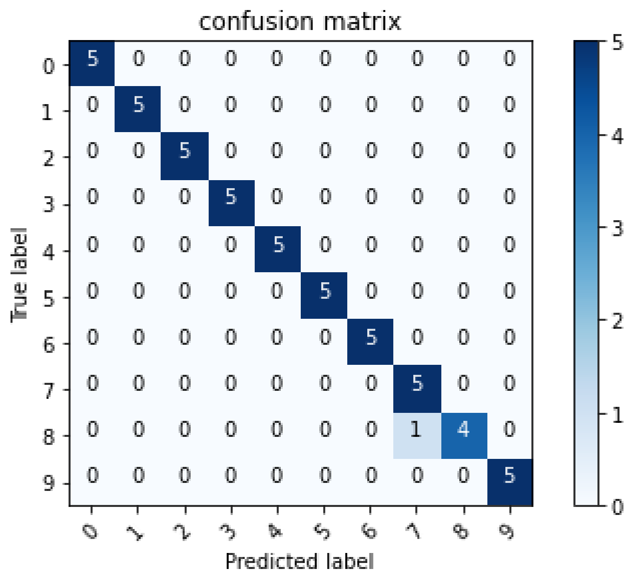
```
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

```
test_batches.class_indices
```

```
{'0': 0,
 '1': 1,
 '2': 2,
 '3': 3,
 '4': 4,
 '5': 5,
 '6': 6,
 '7': 7,
 '8': 8,
 '9': 9}
```

```
cm_plot_labels = ['0','1','2','3','4','5','6','7','8','9']
plot_confusion_matrix(cm, cm_plot_labels, title='confusion matrix')
```

```
Confusion matrix, without normalization
[[5 0 0 0 0 0 0 0 0 0]
 [0 5 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0]
 [0 0 0 0 0 5 0 0 0 0]
 [0 0 0 0 0 0 5 0 0 0]
 [0 0 0 0 0 0 0 5 0 0]
 [0 0 0 0 0 0 0 1 4 0]
 [0 0 0 0 0 0 0 0 0 5]]
```



```
RMSmodel.save('RMS_last.h5')
```

END