jupyter-train2

March 28, 2023

[4]:
```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
# @Time    : 2023/3/27 21:48
# @Author  : Genius_limeng
# @FileName: train.py
# @Software: PyCharm
import random
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D, MaxPooling2D, AveragePooling2D
from keras.optimizers import SGD,adam
from keras.utils import np_utils
from keras.models import load_model
from keras import backend as K
from load_dataset import load_dataset, resize_image
from keras.models import Model
from sklearn.metrics import classification_report
import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix

IMAGE_SIZE = 128

#
class Dataset:
    def __init__(self, path_name):
        #
        self.train_images = None
        self.train_labels = None
        #
        self.valid_images = None
        self.valid_labels = None
        #
```

```
        self.test_images = None
        self.test_labels = None
        #
        self.path_name = path_name
        #
        self.user_num = len(os.listdir(path_name))
        #
        self.input_shape = None

    #
    def load(self, img_rows=IMAGE_SIZE, img_cols=IMAGE_SIZE, img_channels=3):
        #
        nb_classes = self.user_num
        #
        images, labels = load_dataset(self.path_name)

        train_images, valid_images, train_labels, valid_labels =␣
↪train_test_split(images, labels, test_size=0.3,
                                                            random_state=random.
↪randint(0, 100))
        _, test_images, _, test_labels = train_test_split(images, labels,␣
↪test_size=0.3,
                                                            random_state=random.
↪randint(0, 100))

        #       'th'        channels,rows,cols  :rows,cols,channels
        #      keras
        if K.image_dim_ordering() == 'th':
            train_images = train_images.reshape(train_images.shape[0],␣
↪img_channels, img_rows, img_cols)
            valid_images = valid_images.reshape(valid_images.shape[0],␣
↪img_channels, img_rows, img_cols)
            test_images = test_images.reshape(test_images.shape[0],␣
↪img_channels, img_rows, img_cols)
            self.input_shape = (img_channels, img_rows, img_cols)
        else:
            train_images = train_images.reshape(train_images.shape[0],␣
↪img_rows, img_cols, img_channels)
            valid_images = valid_images.reshape(valid_images.shape[0],␣
↪img_rows, img_cols, img_channels)
            test_images = test_images.reshape(test_images.shape[0], img_rows,␣
↪img_cols, img_channels)
            self.input_shape = (img_rows, img_cols, img_channels)

            #
            print(train_images.shape[0], 'train samples')
```

```
            print(valid_images.shape[0], 'valid samples')
            print(test_images.shape[0], 'test samples')

            #     categorical_crossentropy         nb_classes
            #    one-hot          4
            train_labels = np_utils.to_categorical(train_labels, nb_classes)
            valid_labels = np_utils.to_categorical(valid_labels, nb_classes)
            test_labels = np_utils.to_categorical(test_labels, nb_classes)

            #
            train_images = train_images.astype('float32')
            valid_images = valid_images.astype('float32')
            test_images = test_images.astype('float32')

            #    ,      0~1
            train_images /= 255
            valid_images /= 255
            test_images /= 255

            self.train_images = train_images
            self.valid_images = valid_images
            self.test_images = test_images
            self.train_labels = train_labels
            self.valid_labels = valid_labels
            self.test_labels = test_labels


# CNN
class Model:
    def __init__(self):
        self.model = None
        self.history = None

        #

    def build_model(self, dataset, nb_classes=4):
        #
        self.model = Sequential()
        #      CNN      add
        # --------------------------------1--------------------------------------
        self.model.add(Convolution2D(32, 3, 3, border_mode='same',
                                     input_shape=dataset.input_shape))  # 32
        self.model.add(Convolution2D(32, 3, 3, border_mode='same'))  # 32
        self.model.add(Activation('relu'))
        self.model.add(MaxPooling2D(pool_size=(2, 2)))  #
        self.model.add(Dropout(0.25))  # Dropout
        # --------------------------------2--------------------------------------
```

```
        self.model.add(Convolution2D(64, 3, 3, border_mode='same'))  # 64
        self.model.add(Convolution2D(64, 3, 3, border_mode='same'))  # 64
        self.model.add(Activation('relu'))
        self.model.add(MaxPooling2D(pool_size=(2, 2)))  #
        self.model.add(Dropout(0.25))  # Dropout
        # ---------------------------------3-------------------------------------
        self.model.add(Convolution2D(128, 3, 3, border_mode='same'))  # 128
        self.model.add(Convolution2D(128, 3, 3, border_mode='same'))  # 128
        self.model.add(Convolution2D(128, 3, 3, border_mode='same'))  # 128
        self.model.add(Activation('relu'))
        self.model.add(MaxPooling2D(pool_size=(2, 2)))  #
        self.model.add(Dropout(0.25))  # Dropout
        # ---------------------------------4-------------------------------------
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Activation('relu'))
        self.model.add(MaxPooling2D(pool_size=(2, 2)))  #
        self.model.add(Dropout(0.25))  # Dropout
        # ---------------------------------5-------------------------------------
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Convolution2D(256, 3, 3, border_mode='same'))  # 256
        self.model.add(Activation('relu'))
        self.model.add(MaxPooling2D(pool_size=(2, 2)))  #
#         self.model.add(AveragePooling2D(pool_size=(2, 2), strides=2))  #
        self.model.add(Dropout(0.25))  # Dropout
        # ---------------------------------6-------------------------------------
        self.model.add(Flatten())  # Flatten
        self.model.add(Dense(256))  # Dense ,
        self.model.add(Activation('relu'))
        self.model.add(Dropout(0.5))  # Dropout
        self.model.add(Dense(nb_classes))  # Dense
        self.model.add(Activation('softmax'))  #

        #
        self.model.summary()

    #
    def compile(self):
        #  SGD              (learning_rate)
        # 0.01     (decay)
        #                 (momentum),
        #
        #
        sgd = SGD(lr=0.0001, decay=1e-6,  # 10 -6
                  momentum=0.8, nesterov=True)  #  SGD+momentum
```

```
        Adam = adam(lr=0.0001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.
↪0, amsgrad=False)
        self.model.compile(loss='categorical_crossentropy',
                           optimizer=sgd,
                           metrics=['accuracy'])  #

    #
    def train(self, dataset, batch_size=20, nb_epoch=20,␣
↪data_augmentation=True):

        if not data_augmentation:
            self.history = self.model.fit(dataset.train_images,
                                          dataset.train_labels,
                                          batch_size=batch_size,
                                          nb_epoch=nb_epoch,
                                          validation_data=(dataset.
↪valid_images, dataset.valid_labels),
                                          shuffle=True)
        #
        else:
            #              datagen datagen
            #              python
            datagen = ImageDataGenerator(
                featurewise_center=False,  #         0
                samplewise_center=False,  #         0
                featurewise_std_normalization=False,  #
                samplewise_std_normalization=False,  #
                zca_whitening=False,  #      ZCA
                rotation_range=20,  #         (  0 180)
                width_shift_range=0.2,  #               0~1
                height_shift_range=0.2,  #
                horizontal_flip=True,  #
                vertical_flip=False)  #

            #             ZCA
            datagen.fit(dataset.train_images)

            #
            self.history = self.model.fit_generator(datagen.flow(dataset.
↪train_images, dataset.train_labels,
                                                                 ␣
↪batch_size=batch_size),
                                                    samples_per_epoch=dataset.
↪train_images.shape[0],
                                                    nb_epoch=nb_epoch,
```

```
                                                    validation_data=(dataset.
↪valid_images, dataset.valid_labels))

    MODEL_PATH = './model/CNN-FaceRecognitionModel.h5'

    #
    def save_model(self, file_path=MODEL_PATH):
        self.model.save(file_path)

    #
    def load_model(self, file_path=MODEL_PATH):
        self.model = load_model(file_path)

    #
    def evaluate(self, dataset):
        score = self.model.evaluate(dataset.test_images, dataset.test_labels,␣
↪batch_size=20, verbose=1)
        print("%s: %.2f%%" % (self.model.metrics_names[1], score[1] * 100))
        print("score:", score)
        plt.plot(self.history.history['acc'])
        plt.plot(self.history.history['val_acc'])
        plt.title('model accuracy')
        plt.ylabel('accuracy')
        plt.xlabel('epoch')
        plt.legend(['train', 'val'], loc='upper left')
        plt.show()

        plt.plot(self.history.history['loss'])
        plt.plot(self.history.history['val_loss'])
        plt.title('model loss')
        plt.ylabel('loss')
        plt.xlabel('epoch')
        plt.legend(['train', 'val'], loc='upper left')
        plt.show()

    #
    def predict(self, dataset):
        prob = self.model.predict(dataset.test_images)
        predIdxs = np.argmax(prob, axis=1)
        print('\n')
        print(classification_report(dataset.test_labels.argmax(axis=1),␣
↪predIdxs,
                                    =['faces_other''LiMeng', 'ShiWei',␣
↪'YangXinYe', 'ZhengLiWei'], digits=5))

    #
    def heatmap(self, dataset):
```

```
        prob = self.model.predict(dataset.test_images)
        predIdxs = np.argmax(prob, axis=1)
        cm = confusion_matrix(dataset.test_labels.argmax(axis=1), predIdxs)
        sns.heatmap(cm, annot=True)
        plt.title('Confusion matrix')
        plt.ylabel('Actual label')
        plt.xlabel('Predicted label')
        plt.show()
    #
    def face_predict(self, image):
        #
        if K.image_dim_ordering() == 'th' and image.shape != (1, 3, IMAGE_SIZE,␣
↪IMAGE_SIZE):
            image = resize_image(image)  #         IMAGE_SIZE x IMAGE_SIZE
            image = image.reshape((1, 3, IMAGE_SIZE, IMAGE_SIZE))  #␣
↪        1
        elif K.image_dim_ordering() == 'tf' and image.shape != (1, IMAGE_SIZE,␣
↪IMAGE_SIZE, 3):
            image = resize_image(image)
            image = image.reshape((1, IMAGE_SIZE, IMAGE_SIZE, 3))

        #
        image = image.astype('float32')
        image /= 255

        #
        result_probability = self.model.predict_proba(image)
        print('result:', result_probability)

        #    (
        if max(result_probability[0]) >= 0.8:
            result = self.model.predict_classes(image)
            print('result:', result)
            #
            return result[0]
        else:
            print('result:none')
            return -1
```

[5]:
```
if __name__ == '__main__':
    user_num = len(os.listdir('./dataset/'))
    # print("    ", user_num, " ")
    dataset = Dataset('./dataset/')
    dataset.load()
    cnn = Model()
    cnn.build_model(dataset, nb_classes=user_num)
    cnn.compile()  #
```

```
    #
    cnn.train(dataset)
    #
    cnn.save_model(file_path='./model/CNN-FaceRecognitionModel.h5')
```

```
36373 train samples
15589 valid samples
15589 test samples
```

```
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:115:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(32, (3, 3),
input_shape=(128, 128,…, padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:116:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(32, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:121:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(64, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:122:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(64, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:127:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:128:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:129:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:134:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:135:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:136:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:141:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:142:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:143:
UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(256, (3, 3),
padding="same")`
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_14 (Conv2D)           (None, 128, 128, 32)      896
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 128, 128, 32)      9248
_________________________________________________________________
activation_2 (Activation)    (None, 128, 128, 32)      0
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
dropout_7 (Dropout)          (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 64, 64, 64)        18496
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 64, 64, 64)        36928
_________________________________________________________________
activation_3 (Activation)    (None, 64, 64, 64)        0
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 32, 32, 64)        0
_________________________________________________________________
dropout_8 (Dropout)          (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 32, 32, 128)       73856
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 32, 32, 128)       147584
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 32, 32, 128)       147584
_________________________________________________________________
activation_4 (Activation)    (None, 32, 32, 128)       0
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 16, 16, 128)       0
_________________________________________________________________
dropout_9 (Dropout)          (None, 16, 16, 128)       0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 16, 16, 256)       295168
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 16, 16, 256)       590080
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 16, 16, 256)       590080
_________________________________________________________________
activation_5 (Activation)    (None, 16, 16, 256)       0
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 8, 8, 256)         0
_________________________________________________________________
dropout_10 (Dropout)         (None, 8, 8, 256)         0
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 8, 8, 256)         590080
```

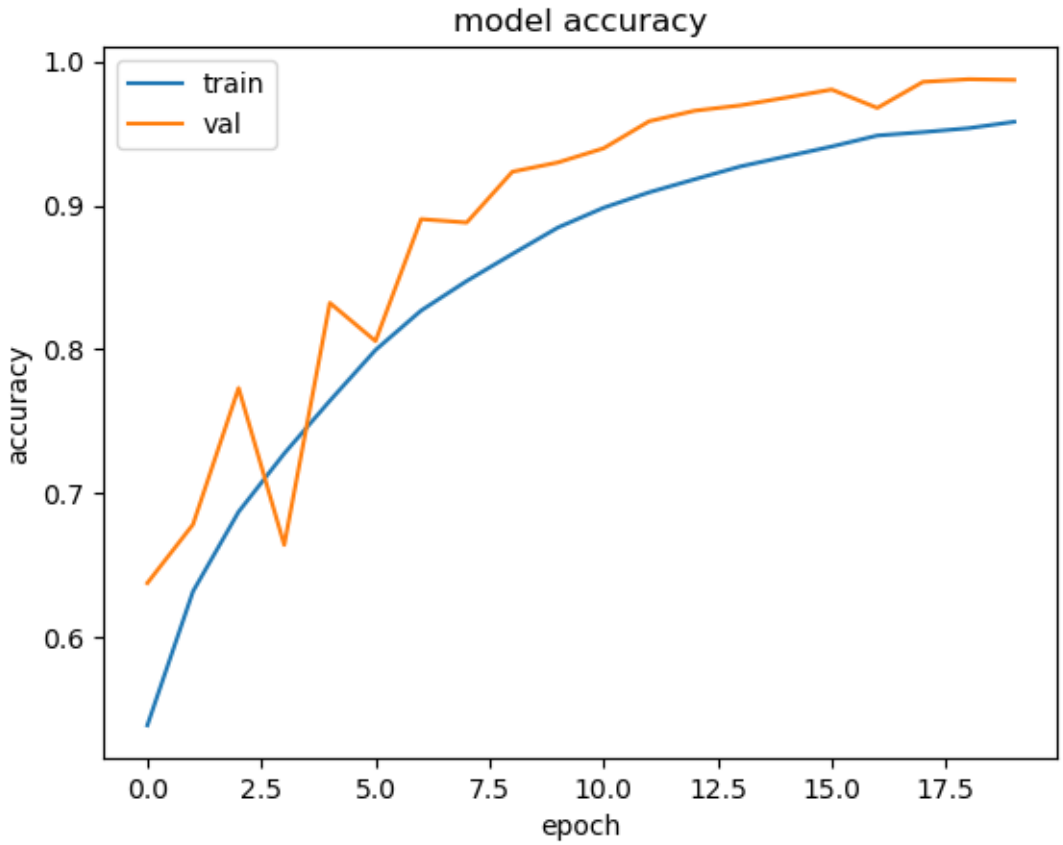
```
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 8, 8, 256)         590080
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 8, 8, 256)         590080
_________________________________________________________________
activation_6 (Activation)    (None, 8, 8, 256)         0
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 4, 4, 256)         0
_________________________________________________________________
dropout_11 (Dropout)         (None, 4, 4, 256)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 4096)              0
_________________________________________________________________
dense_3 (Dense)              (None, 256)               1048832
_________________________________________________________________
activation_7 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_12 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_4 (Dense)              (None, 5)                 1285
_________________________________________________________________
activation_8 (Activation)    (None, 5)                 0
=================================================================
Total params: 4,730,277
Trainable params: 4,730,277
Non-trainable params: 0
_________________________________________________________________
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:207:
UserWarning: The semantics of the Keras 2 argument `steps_per_epoch` is not the
same as the Keras 1 argument `samples_per_epoch`. `steps_per_epoch` is the
number of batches to draw from the generator at each epoch. Basically
steps_per_epoch = samples_per_epoch/batch_size. Similarly
`nb_val_samples`->`validation_steps` and `val_samples`->`steps` arguments have
changed. Update your method calls accordingly.
C:\Users\limeng\anaconda3\envs\py37\lib\site-packages\ipykernel_launcher.py:207:
UserWarning: Update your `fit_generator` call to the Keras 2 API:
`fit_generator(<keras_pre…, validation_data=(array([[[…,
steps_per_epoch=1818, epochs=20)`

Epoch 1/20
1818/1818 [==============================] - 164s 90ms/step - loss: 1.2440 -
acc: 0.5390 - val_loss: 1.2831 - val_acc: 0.6378
Epoch 2/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.9507 -
acc: 0.6319 - val_loss: 0.9635 - val_acc: 0.6784
Epoch 3/20
1818/1818 [==============================] - 150s 83ms/step - loss: 0.7966 -
acc: 0.6872 - val_loss: 0.7528 - val_acc: 0.7732
```

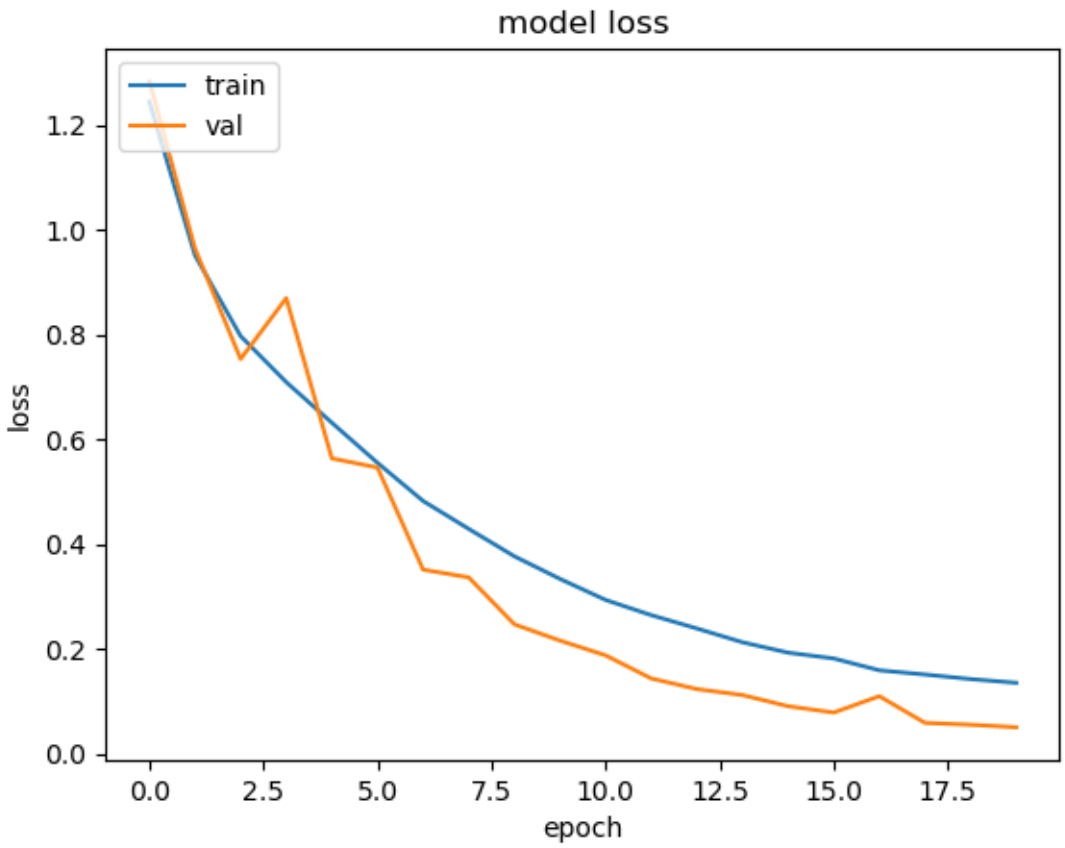
```
Epoch 4/20
1818/1818 [==============================] - 160s 88ms/step - loss: 0.7087 -
acc: 0.7277 - val_loss: 0.8690 - val_acc: 0.6642
Epoch 5/20
1818/1818 [==============================] - 155s 85ms/step - loss: 0.6311 -
acc: 0.7644 - val_loss: 0.5631 - val_acc: 0.8323
Epoch 6/20
1818/1818 [==============================] - 153s 84ms/step - loss: 0.5546 -
acc: 0.7996 - val_loss: 0.5457 - val_acc: 0.8058
Epoch 7/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.4817 -
acc: 0.8270 - val_loss: 0.3507 - val_acc: 0.8905
Epoch 8/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.4285 -
acc: 0.8476 - val_loss: 0.3360 - val_acc: 0.8881
Epoch 9/20
1818/1818 [==============================] - 151s 83ms/step - loss: 0.3762 -
acc: 0.8663 - val_loss: 0.2464 - val_acc: 0.9234
Epoch 10/20
1818/1818 [==============================] - 148s 82ms/step - loss: 0.3331 -
acc: 0.8847 - val_loss: 0.2153 - val_acc: 0.9299
Epoch 11/20
1818/1818 [==============================] - 148s 82ms/step - loss: 0.2930 -
acc: 0.8984 - val_loss: 0.1871 - val_acc: 0.9397
Epoch 12/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.2639 -
acc: 0.9091 - val_loss: 0.1430 - val_acc: 0.9585
Epoch 13/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.2384 -
acc: 0.9182 - val_loss: 0.1227 - val_acc: 0.9658
Epoch 14/20
1818/1818 [==============================] - 152s 84ms/step - loss: 0.2121 -
acc: 0.9271 - val_loss: 0.1113 - val_acc: 0.9695
Epoch 15/20
1818/1818 [==============================] - 155s 85ms/step - loss: 0.1922 -
acc: 0.9340 - val_loss: 0.0900 - val_acc: 0.9749
Epoch 16/20
1818/1818 [==============================] - 157s 86ms/step - loss: 0.1812 -
acc: 0.9409 - val_loss: 0.0779 - val_acc: 0.9805
Epoch 17/20
1818/1818 [==============================] - 173s 95ms/step - loss: 0.1586 -
acc: 0.9486 - val_loss: 0.1092 - val_acc: 0.9677
Epoch 18/20
1818/1818 [==============================] - 155s 85ms/step - loss: 0.1505 -
acc: 0.9509 - val_loss: 0.0582 - val_acc: 0.9858
Epoch 19/20
1818/1818 [==============================] - 155s 85ms/step - loss: 0.1418 -
acc: 0.9537 - val_loss: 0.0547 - val_acc: 0.9877
```

```
Epoch 20/20
1818/1818 [==============================] - 174s 96ms/step - loss: 0.1347 -
acc: 0.9581 - val_loss: 0.0499 - val_acc: 0.9872
```

[11]:
```
#
cnn.evaluate(dataset)
```

```
15589/15589 [==============================] - 18s 1ms/step
acc: 98.73%
score: [0.04902047472949644, 0.9872987334131315]
```

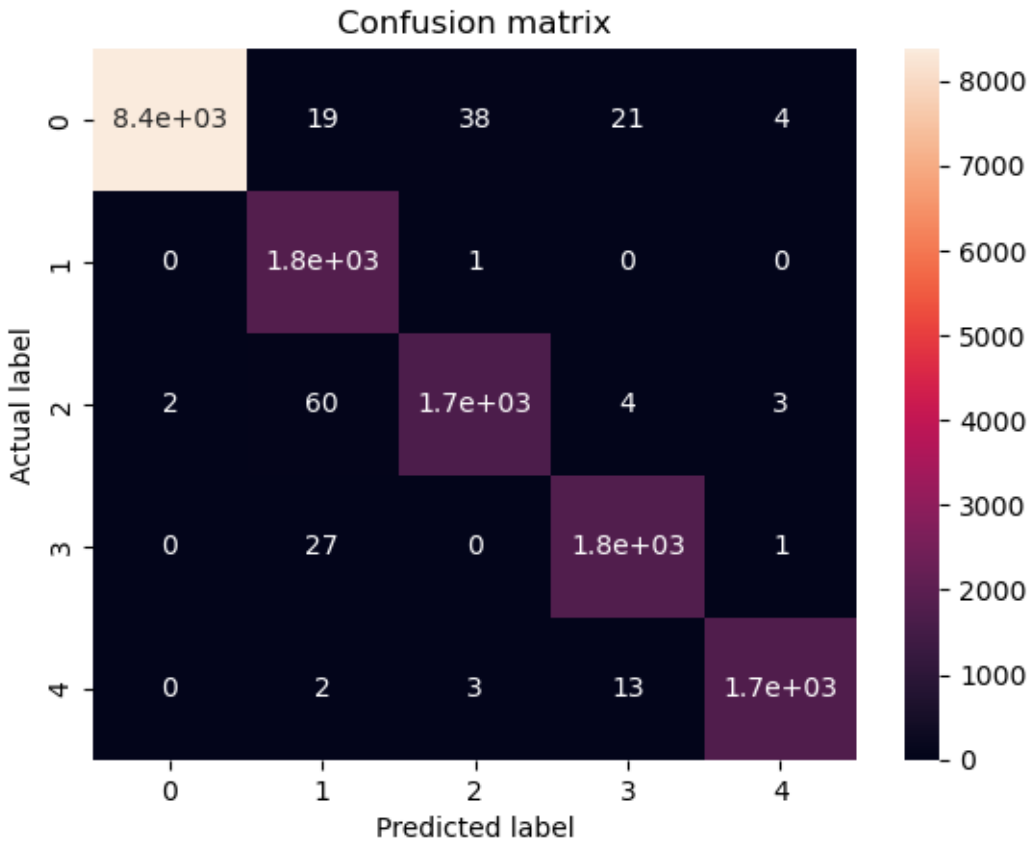




[9]:
```
#
cnn.predict(dataset)
```

	precision	recall	f1-score	support
LiMeng	0.99976	0.99030	0.99501	8455
ShiWei	0.94357	0.99945	0.97071	1807
YangXinYe	0.97567	0.96064	0.96809	1753
ZhengLiWei	0.97920	0.98459	0.98189	1817
faces_other	0.99542	0.98976	0.99258	1757
accuracy			0.98730	15589
macro avg	0.97872	0.98495	0.98166	15589
weighted avg	0.98765	0.98730	0.98736	15589

[10]:
```
#
cnn.heatmap(dataset)
```


Confusion matrix
8.4e+03
19
38
21
4
0
1.8e+03
1
0
0
2
60
1.7e+03
4
3
0
27
0
1.8e+03
1
0
2
3
13
1.7e+03
Actual label
Predicted label
0
1
2
3
4
8000
7000
6000
5000
4000
3000
2000
1000
0

[ ]: