Build a CNN model of facial recognition

We use upper face images as training/validation dataset. The objective is to recognize masked faces face_recognition_CNN_n_squre_02.ipynb for processing non-square images 2021.01.30

Shengping Jiang

In [1]:
```
# face_recognition_CNN_n_squre_02.ipynb for processing non-square im
# 2021.01.30
# Shengping Jiang

import numpy as np
import pandas as pd
import os
import re
import matplotlib.pyplot as plt
import cv2
import random
from keras.utils import to_categorical
from keras.layers import Dense,Conv2D,Flatten,MaxPool2D,Dropout
from keras.models import Sequential
from keras.models import load_model
from keras.optimizers import SGD
from sklearn.model_selection import train_test_split
from keras import backend as K

np.random.seed(1)
```

```
Using TensorFlow backend.
```

In [2]:

```
# Process images before loading
extension = ['jpg','png','bmp','jpeg']
def load_data(path_name):
    images = []
    labels = []
    class_names = []
    for (root,dirs,files) in os.walk(path_name):
        #print('root:',root)
        #We only process images under ./data/train2
        pattern = '^\w+/train2/'
        if re.match(pattern, root):
            print('Read images from:',root)
            #print('files:',files)
            #label = root.split('/')[-1]
            for img in files:
                img = img.lower()
                if img.split('.')[1] in extension:
                    full_path = os.path.join(root, img)
                    #print('full_path:',full_path)
                    image = cv2.imread(full_path)
                    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    #print('image.shape, img_rows, img_cols:', image
                    label = full_path.split('/')[-2]
                    # Add image and label to images and lanels
                    images.append(image)
                    labels.append(label)
        else:   #no train folder
            print("Not include folder:",root)
            #break
    # Converting images to numpy
    images = np.array(images)
    #Change image data type as float and nomarize values to [0~1] fr
    images = images.astype('float32')
    images /= 255

    # This will get how many names in the labels (nb_classes)
    class_names = pd.get_dummies(labels).columns
    nb_classes = len(pd.get_dummies(labels).columns)

    # Create a name dictionary for prediction
    output = {i:class_names[i] for i in range(len(class_names))}

    # Convert class name to binary value
    labels = pd.get_dummies(labels).values
    # change binary values to float from integer
    labels = labels.astype('float32')
    print('image number:', len(images))
    return images,labels, nb_classes, output


```

In [3]:
```
# Example of pd.get_dummies()
names =['Caojun','Shengping','Shengping','Benyuan','Benyuan','Chensh
bivalue = pd.get_dummies(names).columns
print(pd.get_dummies(names))
print(bivalue)
```

```
   Benyuan  Caojun  Chenshu  Shengping
0        0       1        0          0
1        0       0        0          1
2        0       0        0          1
3        1       0        0          0
4        1       0        0          0
5        0       0        1          0
6        0       0        1          0
Index(['Benyuan', 'Caojun', 'Chenshu', 'Shengping'], dtype='object')
```

In [4]:
```
#准备训练与验证数据
train_path = 'data'
#IMAGE_SIZE = 64
images, labels, nb_classes, output = load_data(train_path)
# Separate images and labels to training group and validation group
train_images, valid_images, train_labels, valid_labels \
             = train_test_split(images, labels, test_size = 0.2, ran
```

```
Not include folder: data
Not include folder: data/train2
Read images from: data/train2/蔡江宸
Read images from: data/train2/胡潘
Read images from: data/train2/吴玟君
Read images from: data/train2/刘璐
Read images from: data/train2/汤振
Read images from: data/train2/罗卫
Not include folder: data/train
Not include folder: data/train/黄振
Not include folder: data/train/胡潘
Not include folder: data/train/刘璐
Not include folder: data/train/曾子寅
image number: 124
```

In [5]:
```
print('train_images.shape:',train_images.shape)
print(type(train_images))
print("train_images[0].shape:",train_images[0].shape)
#print("train_labels:",train_labels)
print("nb_classes:",nb_classes)
print("output:",output)
```

```
train_images.shape: (99, 64, 128, 3)
<class 'numpy.ndarray'>
train_images[0].shape: (64, 128, 3)
nb_classes: 6
output: {0: '刘璐', 1: '吴玟君', 2: '汤振', 3: '罗卫', 4: '胡潘', 5: '蔡江
宸'}
```

In [6]:
```
# Visualizing Training data
print(train_labels[1])
plt.imshow(train_images[1])
```

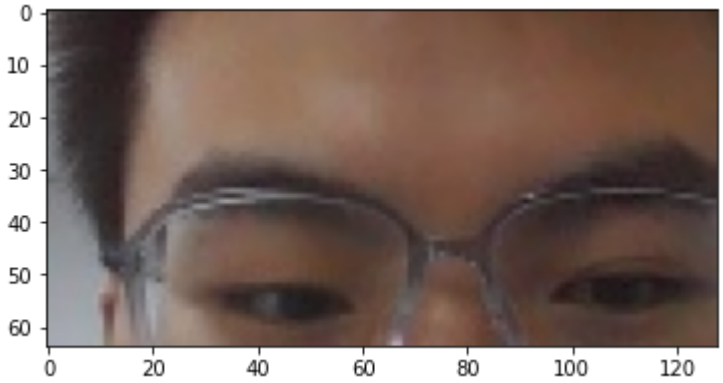
```
[1. 0. 0. 0. 0. 0.]
```

Out[6]: <matplotlib.image.AxesImage at 0x7f4e6ac688d0>

In [7]:
```
# Visualizing Training data
print(valid_labels[15])
plt.imshow(valid_images[15])
```

```
[0. 0. 1. 0. 0. 0.]
```

Out[7]: <matplotlib.image.AxesImage at 0x7f4e6ab58748>



In [8]:
```
def model_build(inputshape, nb_classes):
    model = Sequential()

    #以下代码将顺序添加CNN网络需要的各层，一个add就是一个网络层
    # 1 卷积层1
    model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='rel
    # 2 卷积层2
    model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='rel
    # 3 池化层1
    model.add(MaxPool2D(pool_size=(2, 2)))
    # 4 Dropout层1
    model.add(Dropout(0.25))
    # 5 卷积层3
    model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='rel
    # 6 卷积层4
    model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='rel
    # 7 池化层2
    model.add(MaxPool2D(pool_size=(2, 2)))
    # 8 Dropout层2
    model.add(Dropout(0.25))
    # 9 平化层1
    model.add(Flatten())
    #10 全连接层1
    model.add(Dense(512, activation='relu'))
    # 11 Dropout层3
    model.add(Dropout(0.5))
    # 12 全连接层2. 分类层，输出最终结果
    model.add(Dense(nb_classes, activation = 'softmax'))
    return model
```

In [9]:
```
#Define additional functions

def save_model(model, model_path):
    model.save(model_path)

def loadmodel(model_path):
    model = load_model(model_path)
    return model

def evalu_model(model, valid_images, valid_labels):
    score = model.evaluate(valid_images, valid_labels, verbose = 1)
    print("%s: %.2f%%" % (model.metrics_names[1], score[1] * 100))

```

In [10]:
```
# Train the model
# Images have been resized to 64 x 128 (h x w)
inputshape = (64,128,3)
# 训练样本的组数（图像样本的人数）
#nb_classes is calculated by load_data()
model = model_build(inputshape, nb_classes)
model.summary()
sgd = SGD(lr = 0.001, decay = 1e-6, momentum = 0.9, nesterov = True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metric
# 加载训练及验证样本，启动训练
history = model.fit(train_images, train_labels, batch_size = 12, epo
                            validation_data = (valid_images, valid_la
# 保存训练模型
model_path = './shengping_face_model2.h5'
save_model(model, model_path)

```

```
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:47: The name tf.g
et_default_graph is deprecated. Please use tf.compat.v1.get_default_g
raph instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:349: The name tf.
placeholder is deprecated. Please use tf.compat.v1.placeholder instea
d.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:3147: The name t
f.random_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:3014: The name t
f.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:2683: calling dro
pout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated
and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate
= 1 - keep_prob`.
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:1062: calling red
uce_prod_v1 (from tensorflow.python.ops.math_ops) with keep_dims is d
eprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 128, 32)       896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 62, 126, 32)       9248
_________________________________________________________________
```

```
max_pooling2d_1 (MaxPooling2 (None, 31, 63, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 31, 63, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 31, 63, 64)        18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 29, 61, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 30, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 14, 30, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 26880)             0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               13763072
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 3078
=================================================================
Total params: 13,831,718.0
Trainable params: 13,831,718.0
Non-trainable params: 0.0
_________________________________________________________________
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/optimizers.py:675: The name tf.train.Optimizer
is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:2550: calling red
uce_sum_v1 (from tensorflow.python.ops.math_ops) with keep_dims is de
precated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:2554: The name t
f.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/tensorflow_core/python/ops/math_grad.py:1424: where
(from tensorflow.python.ops.array_ops) is deprecated and will be remo
ved in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:Variable *= will be deprecated. Use `var.assign(va
r * other)` if you want assignment to the variable value or `x = x *
y` if you want a new python Tensor object.
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:766: The name tf.
assign_add is deprecated. Please use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:519: calling Cons
tant.__init__ (from tensorflow.python.ops.init_ops) with dtype is dep
recated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
```

```
it to the constructor
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:762: The name tf.
assign is deprecated. Please use tf.compat.v1.assign instead.

Train on 99 samples, validate on 25 samples
Epoch 1/30
WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:140: The name tf.
get_default_session is deprecated. Please use tf.compat.v1.get_defaul
t_session instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:145: The name tf.
ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instea
d.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:150: The name tf.
Session is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:298: The name tf.
global_variables is deprecated. Please use tf.compat.v1.global_variab
les instead.

WARNING:tensorflow:From /home/simon/.virtualenvs/codexe/lib/python3.
6/site-packages/keras/backend/tensorflow_backend.py:306: The name tf.
variables_initializer is deprecated. Please use tf.compat.v1.variable
s_initializer instead.

99/99 [==============================] - 1s - loss: 1.7843 - acc: 0.1
414 - val_loss: 1.7776 - val_acc: 0.3200
Epoch 2/30
99/99 [==============================] - 1s - loss: 1.7738 - acc: 0.1
717 - val_loss: 1.7694 - val_acc: 0.3200
Epoch 3/30
99/99 [==============================] - 1s - loss: 1.7447 - acc: 0.3
030 - val_loss: 1.7534 - val_acc: 0.3200
Epoch 4/30
99/99 [==============================] - 1s - loss: 1.7062 - acc: 0.3
636 - val_loss: 1.7202 - val_acc: 0.4000
Epoch 5/30
99/99 [==============================] - 1s - loss: 1.6585 - acc: 0.4
343 - val_loss: 1.6627 - val_acc: 0.5200
Epoch 6/30
99/99 [==============================] - 1s - loss: 1.5733 - acc: 0.4
747 - val_loss: 1.5376 - val_acc: 0.5200
Epoch 7/30
99/99 [==============================] - 1s - loss: 1.3865 - acc: 0.5
152 - val_loss: 1.3028 - val_acc: 0.5600
Epoch 8/30
99/99 [==============================] - 1s - loss: 1.1260 - acc: 0.5
253 - val_loss: 1.0171 - val_acc: 0.8400
Epoch 9/30
99/99 [==============================] - 1s - loss: 0.9219 - acc: 0.6
869 - val_loss: 0.9379 - val_acc: 0.5200
```

```
Epoch 10/30
99/99 [==============================] - 1s - loss: 0.8786 - acc: 0.6
566 - val_loss: 0.6766 - val_acc: 0.8400
Epoch 11/30
99/99 [==============================] - 1s - loss: 0.5576 - acc: 0.7
778 - val_loss: 0.4910 - val_acc: 0.9600
Epoch 12/30
99/99 [==============================] - 1s - loss: 0.5391 - acc: 0.8
384 - val_loss: 0.4176 - val_acc: 0.9200
Epoch 13/30
99/99 [==============================] - 1s - loss: 0.3914 - acc: 0.8
889 - val_loss: 0.3280 - val_acc: 0.9200
Epoch 14/30
99/99 [==============================] - 1s - loss: 0.3475 - acc: 0.8
788 - val_loss: 0.3136 - val_acc: 0.9600
Epoch 15/30
99/99 [==============================] - 1s - loss: 0.2777 - acc: 0.8
889 - val_loss: 0.2292 - val_acc: 0.9200
Epoch 16/30
99/99 [==============================] - 1s - loss: 0.2423 - acc: 0.8
990 - val_loss: 0.3307 - val_acc: 0.9600
Epoch 17/30
99/99 [==============================] - 1s - loss: 0.3127 - acc: 0.8
586 - val_loss: 0.2062 - val_acc: 0.9600
Epoch 18/30
99/99 [==============================] - 1s - loss: 0.2377 - acc: 0.9
192 - val_loss: 0.1734 - val_acc: 0.9600
Epoch 19/30
99/99 [==============================] - 1s - loss: 0.2082 - acc: 0.9
192 - val_loss: 0.1689 - val_acc: 0.9200
Epoch 20/30
99/99 [==============================] - 1s - loss: 0.1586 - acc: 0.9
697 - val_loss: 0.1307 - val_acc: 0.9600
Epoch 21/30
99/99 [==============================] - 1s - loss: 0.1515 - acc: 0.9
596 - val_loss: 0.1764 - val_acc: 0.9600
Epoch 22/30
99/99 [==============================] - 1s - loss: 0.1211 - acc: 0.9
596 - val_loss: 0.1323 - val_acc: 0.9200
Epoch 23/30
99/99 [==============================] - 1s - loss: 0.1258 - acc: 0.9
697 - val_loss: 0.1826 - val_acc: 0.9600
Epoch 24/30
99/99 [==============================] - 1s - loss: 0.0835 - acc: 0.9
899 - val_loss: 0.1253 - val_acc: 0.9200
Epoch 25/30
99/99 [==============================] - 1s - loss: 0.1325 - acc: 0.9
394 - val_loss: 0.1251 - val_acc: 0.9200
Epoch 26/30
99/99 [==============================] - 1s - loss: 0.2402 - acc: 0.9
192 - val_loss: 0.1469 - val_acc: 0.9600
Epoch 27/30
99/99 [==============================] - 1s - loss: 0.1223 - acc: 0.9
697 - val_loss: 0.1269 - val_acc: 0.9600
Epoch 28/30
99/99 [==============================] - 1s - loss: 0.0996 - acc: 0.9
```

```
697 - val_loss: 0.0911 - val_acc: 0.9600
Epoch 29/30
99/99 [==============================] - 1s - loss: 0.0991 - acc: 0.9
697 - val_loss: 0.0844 - val_acc: 0.9600
Epoch 30/30
99/99 [==============================] - 1s - loss: 0.0527 - acc: 0.9
899 - val_loss: 0.0865 - val_acc: 0.9600
```

Model parameters analysis

conv2d: (input_channels 3 x kernel_size 3 x 3+bians 1)x filters 32=896 parameters
conv2d_1: (Pre_layer filters 32 x kernel_size 3 x 3+bians 1) x filters 32=9248 parameters
conv2d_2: (Pre_layer filters 32 x kernel_size 3 x 3+bians 1) x filters 64=18496 parameters

conv2d_3: (Pre_layer filters 64 x kernel_size 3 x 3+bians 1) x filters 64=36928 parameters
max_pooling2d_1: output 64 images of 14x30. The flatten transfers the output to 14x30x64=26880 neurons
dense (Dense): (Pre_layer(flatten) outputs unit 26880 +bians 1) x units 512 = 13763072 parameters
dense_2: (Pre_layer(dense)outputs unit 512 +bians 1) x units 6 =3078 parameters

Image shape analysis

conv2d: input_shape: 64x128, 3 channels. Output: 64x128, 32 channels
--Because padding='same', input_shape(64x128)=Output(64x128)

conv2d_1: input_shape: 64x128, 32 channels. Output: 62x62, 32 channels
--Because padding is not defined，default is valid, output_shape = ceil((input_shape - (kernel_size - 1)) / stride)=((64x128- ((3,3) - (1,1))) / 1)=64x128-2x2=62x126

max_pooling2d: input_shape: 62x126, 32 channels. Output: 31x63, 32 channels
--Because pool_size=(2,2), no strides，strides=pool_size=(2,2). Not define padding，padding is valid. output_shape = (input_shape-pool_size + 1)/stride = (62x126-2x2+(1,1))/2=61x125/2 (Rounded up)=31x63

conv2d_2: input_shape: 31x63, 32 channels. Output: 31x63, 64 channels
--Because padding='same', input_shape(31x63)=Output(31x63)

conv2d_3: input_shape: 31x63, 64 channels. Output: 29x61, 64 channels
--Because padding is not given，it is valid. Output_shape = ((31x63- ((3,3) - (1,1))) / 1)=31x63-2x2=29x61

max_pooling2d_1: input_shape: 29x61, 64 channels. Output: 14x30, 64 channels
--Because pool_size=(2,2), no strides，strides=pool_size=(2,2)。 No padding is given，padding is valid. output_shape = (input_shape-pool_size + 1)/stride = (29x61-2x2+(1,1))/2=28x60/2 (rounded up)=14x30

flatten: input_shape: 14x30, 64 channels. Output: 14x30x64=26880, 1 dimension vector
--Flatten transfers all units to one dimension vector： pixel（14x30）x channel（64）=26880

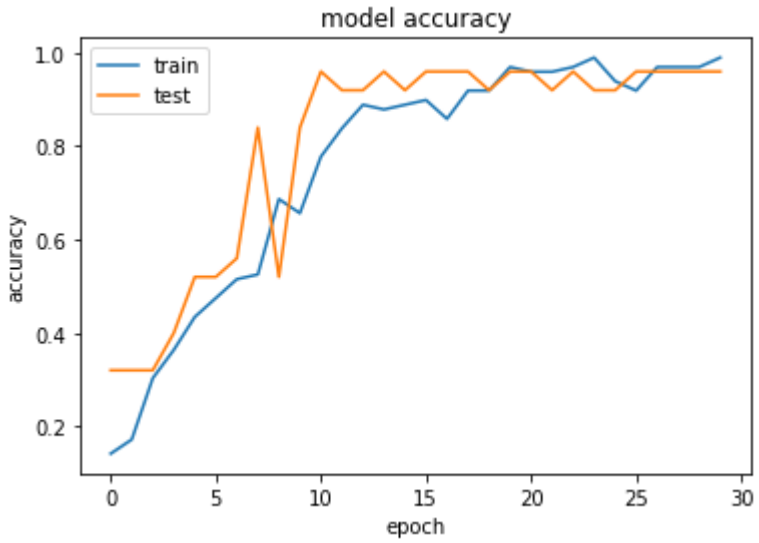
dense：input_shape: 26880, one dimension. Output: 512, one dimension
--dense connects 26880 units to each output unit. output units are 512

dense_1：input_shape: 512, one dimension. Output: 6, one dimension
--dense connects 512 units to each output unit. output units are 6

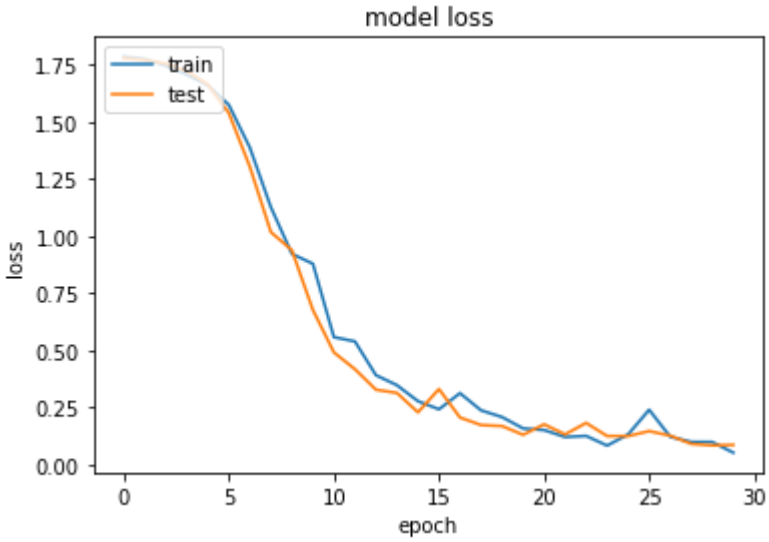
In [11]:
```
# 读出保存的模型并进行验证
model_path = './shengping_face_model2.h5'
model2 = loadmodel(model_path)
evalu_model(model2, valid_images, valid_labels)
eva_result = model2.evaluate(valid_images,valid_labels)
print('eva_result:',eva_result)
```

```
25/25 [==============================] - 0s
acc: 96.00%
25/25 [==============================] - 0s
eva_result: [0.08653917908668518, 0.9599999785423279]
```

In [12]:
```
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [13]:
```
# 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', 'test'], loc='upper left')
plt.show()
```



In [14]:
```
checkImage = valid_images[14:15]
checklabel = valid_labels[14:15]
#print('checkImage:',checkImage)
predicts = model.predict(checkImage)
print("predicts:",predicts)
print("checklabel:",checklabel)
```

```
predicts: [[2.0491282e-06 1.6856793e-06 9.8777175e-01 2.0218498e-05 1.
1815689e-02
  3.8861079e-04]]
checklabel: [[0. 0. 1. 0. 0. 0.]]
```

In [15]:

```
# Testing predictions and compare to actual label

def index_predict(n):
    checkImage = valid_images[n:n+1]
    checklabel = valid_labels[n:n+1]
    #print('checkImage:',checkImage)
    predicts = model.predict(checkImage)
    print("predicts:",predicts)

    print("Actual binary, name: ",checklabel[0], output[int(np.argma
    for i in range(len(predicts)):
        print("Predicted :- ",output[int(np.argmax(predicts[i]))])
        #print('predict:',predict[i])
```

In [16]:
```
for i in range(len(valid_labels)):
    index_predict(i)
```

```
predicts: [[9.8787097e-04 8.9721994e-05 3.7342799e-03 3.3260537e-03
9.8862737e-01
  3.2347268e-03]]
Actual binary, name:  [0. 0. 0. 0. 1. 0.] 胡潘
Predicted :-  胡潘
predicts: [[0.01379944 0.0062185  0.0835894  0.172029   0.6970752  0.
02728842]]
Actual binary, name:  [0. 0. 0. 0. 1. 0.] 胡潘
Predicted :-  胡潘
predicts: [[9.9999774e-01 4.3099405e-07 1.5926870e-09 1.2036265e-14
3.2242593e-08
  1.7944269e-06]]
Actual binary, name:  [1. 0. 0. 0. 0. 0.] 刘璐
Predicted :-  刘璐
predicts: [[5.0224871e-03 9.9459672e-01 3.5654526e-07 1.6302594e-05
9.7832555e-05
  2.6628820e-04]]
Actual binary, name:  [0. 1. 0. 0. 0. 0.] 吴玟君
Predicted :-  吴玟君
predicts: [[1.3158612e-07 9.9999785e-01 3.9044394e-14 1.8744620e-07
4.1303676e-09
  1.7519769e-06]]
Actual binary, name:  [0. 1. 0. 0. 0. 0.] 吴玟君
Predicted :-  吴玟君
predicts: [[7.4822474e-06 1.5423456e-07 7.6317250e-05 6.9135058e-09
7.3153578e-06
  9.9990869e-01]]
Actual binary, name:  [0. 0. 0. 0. 0. 1.] 蔡江宸
Predicted :-  蔡江宸
predicts: [[7.5593976e-05 1.5592390e-04 3.1925845e-01 6.1089249e-04
6.7039531e-01
  9.5038889e-03]]
Actual binary, name:  [0. 0. 0. 0. 1. 0.] 胡潘
Predicted :-  胡潘
predicts: [[5.5492474e-05 4.4090531e-05 2.8060514e-01 1.9161115e-04
7.1754968e-01
  1.5539433e-03]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  胡潘
predicts: [[5.2539972e-06 1.0214385e-04 5.5464305e-05 9.8813802e-01
1.1452772e-02
  2.4628398e-04]]
Actual binary, name:  [0. 0. 0. 1. 0. 0.] 罗卫
Predicted :-  罗卫
predicts: [[2.0502725e-06 2.9544822e-08 9.9965024e-01 1.6773693e-07
2.7637204e-04
  7.1231421e-05]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[9.9999499e-01 4.2410156e-06 9.7286606e-12 2.9399673e-15
3.6305801e-09
  7.2935194e-07]]
Actual binary, name:  [1. 0. 0. 0. 0. 0.] 刘璐
```

```
Predicted :-  刘璐
predicts: [[1.3795243e-05 9.9997723e-01 1.2531072e-11 3.0065672e-07
1.2772068e-06
  7.2973521e-06]]
Actual binary, name:  [0. 1. 0. 0. 0. 0.] 吴玟君
Predicted :-  吴玟君
predicts: [[7.2040139e-06 1.7859188e-07 9.9615508e-01 2.9848431e-06
3.5697147e-03
  2.6483263e-04]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[5.0115813e-07 4.8307154e-08 9.9935335e-01 1.0921337e-06
6.0370937e-04
  4.1210744e-05]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[2.0491282e-06 1.6856793e-06 9.8777175e-01 2.0218498e-05
1.1815689e-02
  3.8861079e-04]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[1.8987097e-05 1.2039476e-06 9.8543108e-01 2.5053219e-05
1.4206828e-02
  3.1696452e-04]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[2.6513455e-06 1.6776983e-04 3.4698495e-04 9.7680640e-01
2.2110252e-02
  5.6606013e-04]]
Actual binary, name:  [0. 0. 0. 1. 0. 0.] 罗卫
Predicted :-  罗卫
predicts: [[2.2742568e-06 7.5841999e-05 6.7673514e-05 9.8732007e-01
1.2372936e-02
  1.6110601e-04]]
Actual binary, name:  [0. 0. 0. 1. 0. 0.] 罗卫
Predicted :-  罗卫
predicts: [[2.3127021e-05 8.3085788e-06 4.0442496e-06 1.1634216e-06
8.7443390e-04
  9.9908888e-01]]
Actual binary, name:  [0. 0. 0. 0. 0. 1.] 蔡江宸
Predicted :-  蔡江宸
predicts: [[5.3416265e-06 2.2987253e-07 2.7305944e-04 2.2093610e-08
3.9871054e-05
  9.9968147e-01]]
Actual binary, name:  [0. 0. 0. 0. 0. 1.] 蔡江宸
Predicted :-  蔡江宸
predicts: [[3.8484282e-06 1.1713930e-04 6.6535542e-04 5.0491234e-03
9.9239606e-01
  1.7684379e-03]]
Actual binary, name:  [0. 0. 0. 0. 1. 0.] 胡潘
Predicted :-  胡潘
predicts: [[4.3506166e-06 6.5332047e-06 1.9676934e-05 3.0994611e-06
2.8017085e-04
  9.9968624e-01]]
Actual binary, name:  [0. 0. 0. 0. 0. 1.] 蔡江宸
Predicted :-  蔡江宸
predicts: [[2.6747308e-05 1.8773864e-06 9.8036915e-01 3.2552882e-05
```

```
1.9175127e-02
  3.9447338e-04]]
Actual binary, name:  [0. 0. 1. 0. 0. 0.] 汤振
Predicted :-  汤振
predicts: [[9.9999201e-01 6.9059211e-06 2.2567129e-11 7.5456626e-15
5.7915059e-09
  1.1313166e-06]]
Actual binary, name:  [1. 0. 0. 0. 0. 0.] 刘璐
Predicted :-  刘璐
predicts: [[7.01824320e-05 1.04334084e-07 3.85002140e-03 1.08676206e-
08
  2.00488768e-03 9.94074881e-01]]
Actual binary, name:  [0. 0. 0. 0. 0. 1.] 蔡江宸
Predicted :-  蔡江宸
```

In [17]:

```
for i in range(len(valid_labels)):
    print('i, valid_labels[i]:',i,valid_labels[i] )
#plt.imshow(valid_images[20])
```

```
i, valid_labels[i]: 0 [0. 0. 0. 0. 1. 0.]
i, valid_labels[i]: 1 [0. 0. 0. 0. 1. 0.]
i, valid_labels[i]: 2 [1. 0. 0. 0. 0. 0.]
i, valid_labels[i]: 3 [0. 1. 0. 0. 0. 0.]
i, valid_labels[i]: 4 [0. 1. 0. 0. 0. 0.]
i, valid_labels[i]: 5 [0. 0. 0. 0. 0. 1.]
i, valid_labels[i]: 6 [0. 0. 0. 0. 1. 0.]
i, valid_labels[i]: 7 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 8 [0. 0. 0. 1. 0. 0.]
i, valid_labels[i]: 9 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 10 [1. 0. 0. 0. 0. 0.]
i, valid_labels[i]: 11 [0. 1. 0. 0. 0. 0.]
i, valid_labels[i]: 12 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 13 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 14 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 15 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 16 [0. 0. 0. 1. 0. 0.]
i, valid_labels[i]: 17 [0. 0. 0. 1. 0. 0.]
i, valid_labels[i]: 18 [0. 0. 0. 0. 0. 1.]
i, valid_labels[i]: 19 [0. 0. 0. 0. 0. 1.]
i, valid_labels[i]: 20 [0. 0. 0. 0. 1. 0.]
i, valid_labels[i]: 21 [0. 0. 0. 0. 0. 1.]
i, valid_labels[i]: 22 [0. 0. 1. 0. 0. 0.]
i, valid_labels[i]: 23 [1. 0. 0. 0. 0. 0.]
i, valid_labels[i]: 24 [0. 0. 0. 0. 0. 1.]
```

In [ ]:

```
1
```