In [1]:
```
import tensorflow as tf
print('版本:',tf.__version__)
import keras
print('版本:',keras.__version__)
```

```
版本: 2.1.0

Using TensorFlow backend.

版本: 2.3.1
```

In [2]:
```
import cv2
import numpy as np
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input
```

In [3]:
```
x_real = np.load('./figerPimg/datasetFP/x_real_fp.npz')['data']
y_real = np.load('./figerPimg/datasetFP/y_real_fp.npy')
x_easy = np.load('./figerPimg/datasetFP/x_easy_fp.npz')['data']
y_easy = np.load('./figerPimg/datasetFP/y_easy_fp.npy')
x_medium = np.load('./figerPimg/datasetFP/x_medium_fp.npz')['data']
y_medium = np.load('./figerPimg/datasetFP/y_medium_fp.npy')
x_hard = np.load('./figerPimg/datasetFP/x_hard_fp.npz')['data']
y_hard = np.load('./figerPimg/datasetFP/y_hard_fp.npy')
```

In [4]:
```
print(x_real.shape,y_real.shape)
print(x_easy.shape,y_easy.shape)
print(x_medium.shape,y_medium.shape)
print(x_hard.shape,y_hard.shape)
```

```
(6000, 96, 96, 3) (6000, 4)
(17931, 96, 96, 3) (17931, 4)
(17067, 96, 96, 3) (17067, 4)
(14272, 96, 96, 3) (14272, 4)
```

In [5]:
```
def restore_label(label):
    finger_list = ['thumb', 'index', 'middle', 'ring', 'little']
    label_list = list(label)
    label_list[1] = 'F' if label_list[1] else 'M'
    label_list[2] = 'Right' if label_list[2] else 'Left'
    label_list[3] = finger_list[label_list[3]]
    return label_list
```

In [6]:
```
# one-hot
id_label = to_categorical(y_real[:,0]-1)
gender_label = to_categorical(y_real[:,1])
LRhand_label = to_categorical(y_real[:,2])
finger_label = to_categorical(y_real[:,3])
```

In [7]:
```
print(type(id_label))
print(id_label.shape)
print(gender_label.shape)
print(LRhand_label.shape)
print(finger_label.shape)
```

```
<class 'numpy.ndarray'>
(6000, 600)
(6000, 2)
(6000, 2)
(6000, 5)
```

In [8]:
```
load model
_model = ResNet50(include_top=False, weights="imagenet",input_shape=(120,120,3))
```

In [9]:
```
id_model =  tf.keras.models.load_model('./figerPimg/rs_h5_all/resnet50_fpAll_id.h
gender_model =  tf.keras.models.load_model('./figerPimg/rs_h5_all/resnet50_fpAll_
LRhand_model =  tf.keras.models.load_model('./figerPimg/rs_h5_all/resnet50_fpAll_
finger_model =  tf.keras.models.load_model('./figerPimg/rs_h5_all/resnet50_fpAll_
```

In [10]:
```
img_path = "./figerPimg/fingerAltered/Easy/1__M_Left_middle_finger_CR.BMP"
img = cv2.imread(img_path)
img = cv2.resize(img, (120, 120))
np_img = np.array(img).reshape((1, 120, 120, 3))
np_img = np_img.astype(np.float32) / 255.
```

In [11]:
```
input = preprocess_input(np_img)
features = rs_model.predict(input, verbose=0)
```

In [12]:
```
# 指紋比對
id_pred = id_model.predict(features)
id_prob_list = np.argsort(id_pred[0], axis=0)

gender_pred = gender_model.predict(features)
gender_prob_list = np.argsort(gender_pred[0], axis=0)

LRhand_pred = LRhand_model.predict(features)
LR_prob_list = np.argsort(LRhand_pred[0], axis=0)

finger_pred = finger_model.predict(features)
finger_prob_list = np.argsort(finger_pred[0], axis=0)

print('輸入指紋:', img_path)
print('符合對象:', restore_label([id_prob_list[-1]+1, gender_prob_list[-1], LR_pr
print('符合機率:', [id_pred[0][id_prob_list[-1]], gender_pred[0][gender_prob_list
```

```
輸入指紋: ./figerPimg/fingerAltered/Easy/1__M_Left_middle_finger_CR.BMP
符合對象: [175, 'M', 'Right', 'middle']
符合機率: [0.2151452, 0.9215806, 0.7288998, 0.33929136]
```