

```

import numpy as np
from tensorflow import keras
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense, Activation, Dropout, LSTM, BatchNormalization
from keras.layers import Flatten
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.utils import to_categorical
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D

```

```

trainset='/content/drive/MyDrive/Anh faceID group/train'
validationset='/content/drive/MyDrive/Anh faceID group/validation'
train=ImageDataGenerator(rescale=1/255.0, validation_split=0)
validation=ImageDataGenerator(rescale=1/255.0, validation_split=0.9)

```

```

train_data=train.flow_from_directory(trainset, target_size=(150,150), batch_size=10, class_mode=
validation_set=validation.flow_from_directory(validationset, target_size=(150,150), batch_size=

```

```

    Found 138 images belonging to 3 classes.
    Found 30 images belonging to 3 classes.

```

```

print(train_data.class_indices)
print(validation_set.class_indices)

```

```

{'binh': 0, 'tien': 1, 'trieu': 2}
{'binh': 0, 'tien': 1, 'trieu': 2}

```

```

from keras.backend import conv2d
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',in
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))

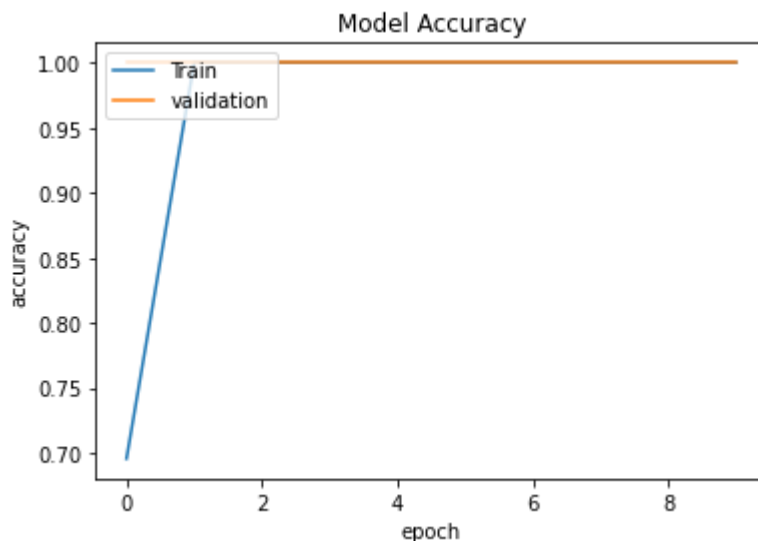
model.add(Flatten())
model.add(Dense(128,activation='relu',kernel_initializer='he_uniform'))
model.add(Dense(3,activation='softmax'))

```

```
model.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy'])
history=model.fit(train_data,batch_size=10,epochs=10,verbose=1,validation_data=validation_set)
```

```
Epoch 1/10
14/14 [=====] - 35s 2s/step - loss: 6.2755 - accuracy: 0.6957 -
Epoch 2/10
14/14 [=====] - 32s 2s/step - loss: 0.0030 - accuracy: 1.0000 -
Epoch 3/10
14/14 [=====] - 33s 2s/step - loss: 1.2161e-04 - accuracy: 1.0000 -
Epoch 4/10
14/14 [=====] - 32s 2s/step - loss: 2.7321e-05 - accuracy: 1.0000 -
Epoch 5/10
14/14 [=====] - 32s 2s/step - loss: 1.3488e-05 - accuracy: 1.0000 -
Epoch 6/10
14/14 [=====] - 32s 2s/step - loss: 4.7441e-06 - accuracy: 1.0000 -
Epoch 7/10
14/14 [=====] - 33s 2s/step - loss: 1.9963e-06 - accuracy: 1.0000 -
Epoch 8/10
14/14 [=====] - 33s 2s/step - loss: 1.1135e-06 - accuracy: 1.0000 -
Epoch 9/10
14/14 [=====] - 32s 2s/step - loss: 4.7597e-07 - accuracy: 1.0000 -
Epoch 10/10
14/14 [=====] - 32s 2s/step - loss: 2.8334e-07 - accuracy: 1.0000 -
```

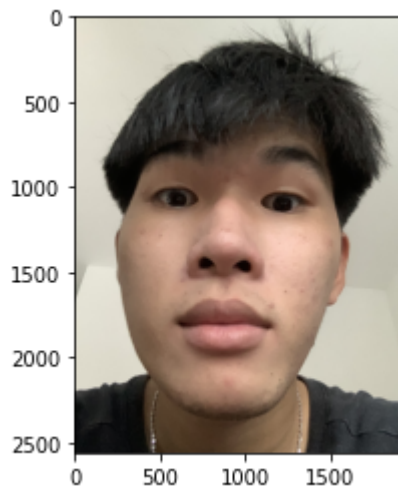
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train','validation'],loc='upper left')
plt.show()
```



```
model.save('/content/drive/MyDrive/BT AI/FaceIDGr.h5')
```

```
url='/content/drive/MyDrive/Anh test/z3406326496487_e8d490855dd13b379a5431743d73007c.jpg'  
plt.imshow(load_img(url))
```

<matplotlib.image.AxesImage at 0x7f93c6bab950>



```
img=load_img('/content/drive/MyDrive/Anh test/z3406326496487_e8d490855dd13b379a5431743d73007c  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=-1)  
  
array([2])
```

