```
import numpy as np
import os

from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D
from keras.layers import Dropout
from keras.layers import Flatten
from keras.constraints import maxnorm
from tensorflow.keras.optimizers import Adam

from keras.layers.convolutional import Convolution2D
from keras.layers.convolutional import MaxPooling2D
from keras.callbacks import ModelCheckpoint, LearningRateScheduler
from keras.callbacks import ReduceLROnPlateau
from keras.callbacks import EarlyStopping


from keras.utils import np_utils
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator


from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
x_train= '/content/drive/MyDrive/Tiền/x_train'
x_test= '/content/drive/MyDrive/Tiền/x_test'
```

```
x_train = ImageDataGenerator(rescale=1/255)
x_test = ImageDataGenerator(rescale=1/255)
```

```
x_train_data = x_train.flow_from_directory(
    directory=r"/content/drive/MyDrive/Tiền/x_train",
    target_size=(224, 224),
    batch_size=3,
    class_mode='categorical',
    )
x_test_data = x_test.flow_from_directory(
    directory=r"/content/drive/MyDrive/Tiền/x_test",
    target_size=(224, 224),
    batch_size=3,
    class_mode= "categorical",
    )
```

```
Found 236 images belonging to 11 classes.
Found 142 images belonging to 11 classes.
```

```
x_train_data.class_indices
```

```
{'100k': 0,
 '10k': 1,
 '1k': 2,
 '200': 3,
 '200k': 4,
 '20k': 5,
 '2k': 6,
 '500': 7,
 '500k': 8,
 '50k': 9,
 '5k': 10}
```

```
model = Sequential([
    Conv2D(filters=32, kernel_size=(3, 3), activation='relu', padding = 'same', input_shap
    MaxPooling2D(pool_size=(2, 2), strides=2),
    Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'),
    MaxPooling2D(pool_size=(2, 2), strides=2),
    Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'same'),
    MaxPooling2D(pool_size=(2, 2), strides=2),
    Dropout(0.2),
    Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'same'),
    Flatten(),
    Dropout(0.5),
    Dense(units=11, activation='softmax')
])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 224, 224, 32)      896

 max_pooling2d (MaxPooling2D  (None, 112, 112, 32)     0
 )

 conv2d_3 (Conv2D)           (None, 112, 112, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 conv2d_4 (Conv2D)           (None, 56, 56, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 28, 28, 128)      0
 2D)

 dropout (Dropout)           (None, 28, 28, 128)       0

 conv2d_5 (Conv2D)           (None, 28, 28, 128)       147584

 flatten (Flatten)           (None, 100352)            0

 dropout_1 (Dropout)         (None, 100352)            0
```

```
 dense (Dense)               (None, 11)                1103883

=================================================================
Total params: 1,344,715
Trainable params: 1,344,715
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.optimizers import SGD
#opt = SGD(lr = 0.01, momentum= 0.9)
model.compile(optimizer=Adam(learning_rate=0.0005), loss='categorical_crossentropy', metri
history=model.fit(x_train_data,
                  epochs=1,
                  batch_size=32,
                  verbose=1,
                  validation_data=x_test_data)
```

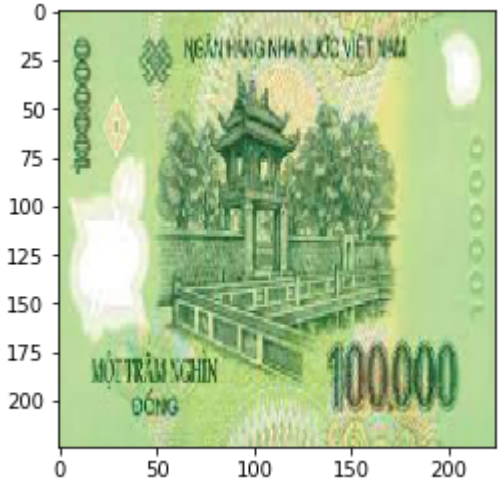
```
79/79 [==============================] - 3s 28ms/step - loss: 0.7349 - accuracy: 0.7
```

```
model.save('nhandangtientri.h5')
```

```
from keras.models import load_model
nhandangtien = load_model('nhandangtientri.h5')
```

```
from keras.preprocessing.image import load_img, img_to_array
img = load_img('/content/drive/MyDrive/Tiền/x_test/100k/100k (10).jpg', target_size =(224,
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,224,224,3)
img = img.astype('float32')
img = img/255
img.shape
```

```
(1, 224, 224, 3)
```




```
np.argmax(nhandangtien.predict(img),axis= 1)
```

```
array([0])
```

x_train_data.class_indices {'100k': 0, '10k': 1, '1k': 2, '200': 3, '200k': 4, '20k': 5, '2k': 6, '500': 7, '500k': 8, '50k': 9, '5k': 10}

✓ 0 giây hoàn thành lúc 16:44

Không thể kết nối với dịch vụ reCAPTCHA. Vui lòng kiểm tra kết nối internet của bạn và tải lại để nhận hình ảnh xác thực reCAPTCHA.