TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CƠ KHÍ CHẾ TẠO MÁY

BỘ MÔN CƠ ĐIỆN TỬ

HOMEWORK AI: WEEK 14

GVHD: TS. Nguyễn Trường Thịnh

SVTH: Phạm Phúc Chương

MSSV: 19146311

TP.Hồ Chí Minh, tháng năm 2022

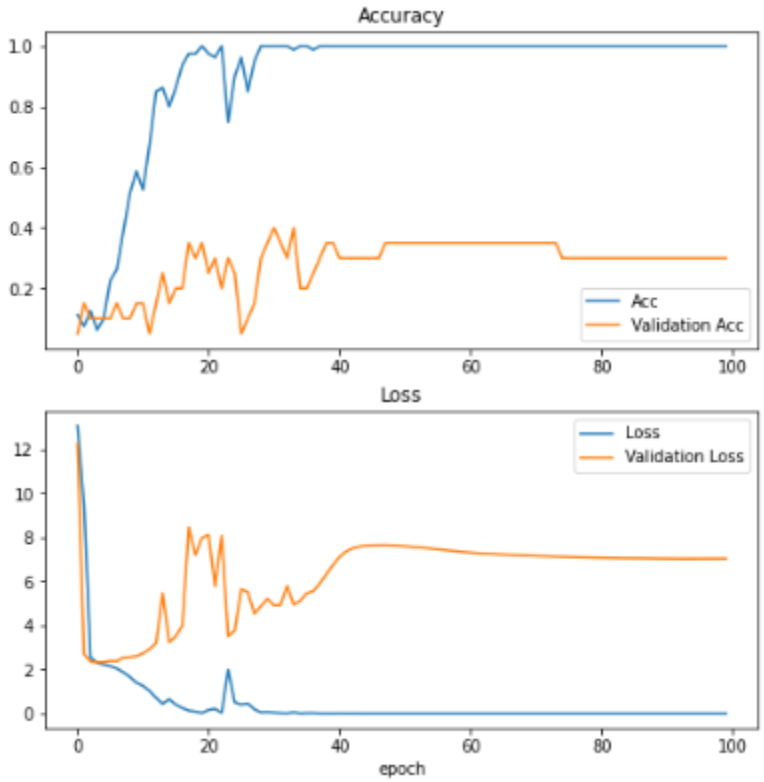
Mục Lục

1. Phân loại trái cây

Code:

```
#import thu vien
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D,
 Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
#dung file data up san tren google
from google.colab import drive
drive.mount('/content/drive')
#Ham ve do thi
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
# Load Data
with open('/content/drive/MyDrive/Colab Notebooks/Data/data_fruit.pickle',
 'rb') as f:
    (x_train, y_train) = pickle.load(f)
# Reshape Data
x_pre_1 = x_train[1]
x_pre_2 = x_train[13]
```

```
x_pre_3 = x_train[26]
x_pre_4 = x_train[38]
x_pre_5 = x_train[56]
 Preprocessing Data
x_train = x_train.astype('float32')
x_train /= 255
# Encoding Y
y_train = np_utils.to_categorical(y_train, 10)
# Shuffe Data
x_train, y_train = shuffle(x_train, y_train)
# Tao model
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same', input_shape = (150,150,3)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
model.summary()
# Training
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics
= ['acc'])
his = model.fit(x_train, y_train, epochs = 100, batch_size = 64, validatio
n_split = 0.2)
#Bieu do ket qua training
plot_history(his)
```



Hình 1 Kết quả training

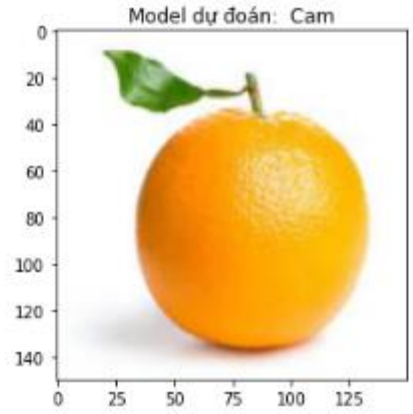
add label cho model

```
label = ['Cam','Dao','Dua hau','Du du','Khe','Le','Oi','Man','Sapoche','Xoai']
```

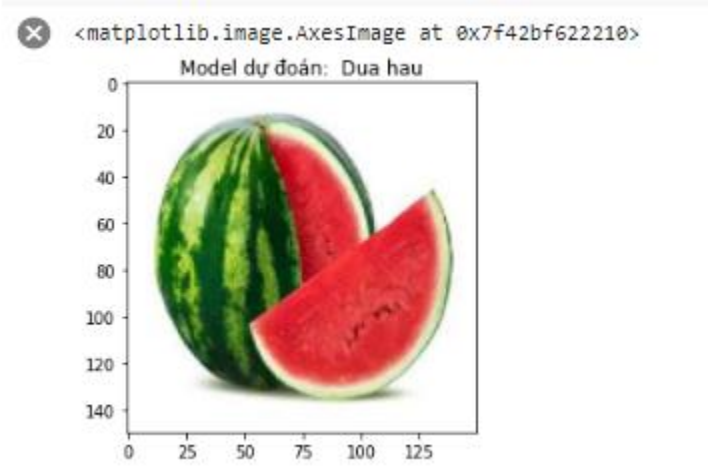
Xet lai

```
plt.title("Model dự đoán:  " + label[np.argmax(model.predict(x_pre_1.reshape(1,150,150,3)))])
plt.imshow(cv2.cvtColor(x_pre_1, cv2.COLOR_BGR2RGB), cmap=plt.get_cmap('gray'))
```

```
<matplotlib.image.AxesImage at 0x7f42be281e50>
```



Hình 2 Kết quả test model trên quả cam



Hình 3 Kết quả test model trên dưa hấu

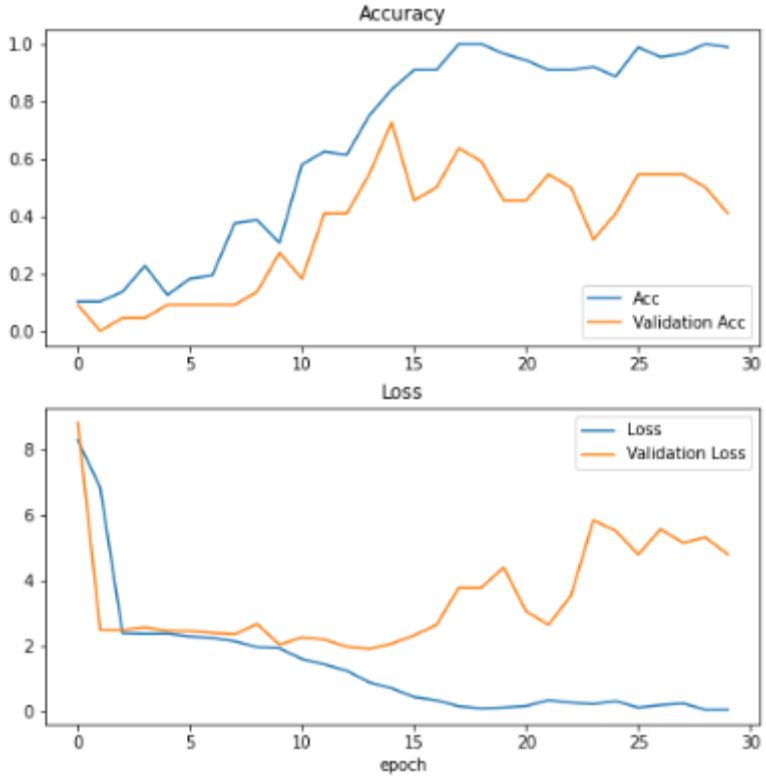
2. CNN_Nhận dạng 11 loại tiền Việt Nam

#Import lib

```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D,
 Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
#chuan bi ham ve bieu do
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
```

```
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
#add data in drive
from google.colab import drive
drive.mount('/content/drive')
# Load Data
with open('/content/drive/MyDrive/Colab Notebooks/Data/data_MoneyVN.pickle
', 'rb') as f:
    (x_train, y_train) = pickle.load(f)
# Reshape Data
x_pre_1 = x_train[1]
x_pre_2 = x_train[12]
x_pre_3 = x_train[25]
x_pre_4 = x_train[38]
x_pre_5 = x_train[56]
x_pre_6 = x_train[86]
# Preprocessing Data
x_train = x_train.astype('float32')
x_train /= 255
# Encoding Y
y_train = np_utils.to_categorical(y_train, 11)
# Shuffe Data
x_train, y_train = shuffle(x_train, y_train)
#tao model CNN
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same', input_shape = (150,150,3)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
```

```
model.add(Dense(11, activation='softmax'))
model.summary()
#Training
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics
= ['acc'])
his = model.fit(x_train, y_train, epochs = 30, batch_size = 64, validation
_split = 0.2)
#ve bieu do xem ket qua training
plot_history(his)
```



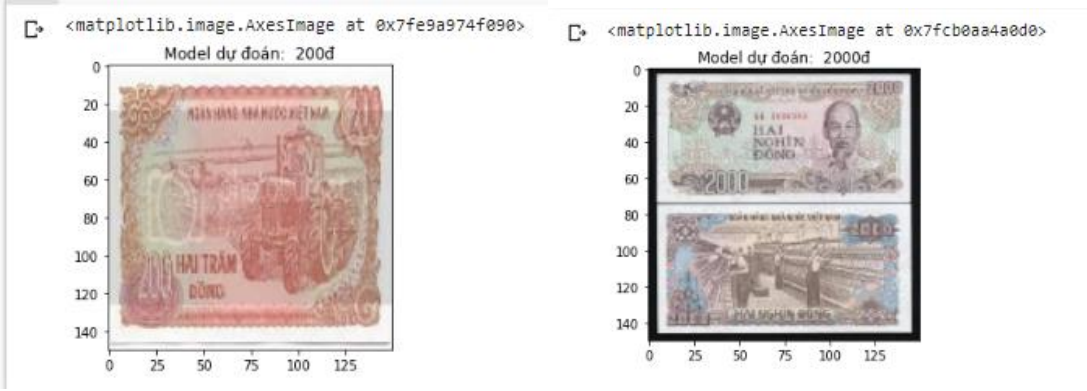
Hình 4 Kết quả training

Add label

```
label = ['200đ','500đ','1000đ','2000đ','5000đ','10000đ','20000đ','50000đ',
'100000đ','200000đ','500000']
```

model du doan

```
plt.title("Model dự đoán:  " + label[np.argmax(model.predict(x_pre_1.resha
pe(1,150,150,3)))])
plt.imshow(cv2.cvtColor(x_pre_1, cv2.COLOR_BGR2RGB), cmap=plt.get_cmap('gr
ay'))
```



Hình 5 Kết quả model dự đoán

3. Dự đoán món ăn Việt Nam

#Add lib

```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D,
 Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
from google.colab import drive
drive.mount('/content/drive')
```

#chuan bi ham ve bieu do

```
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
```

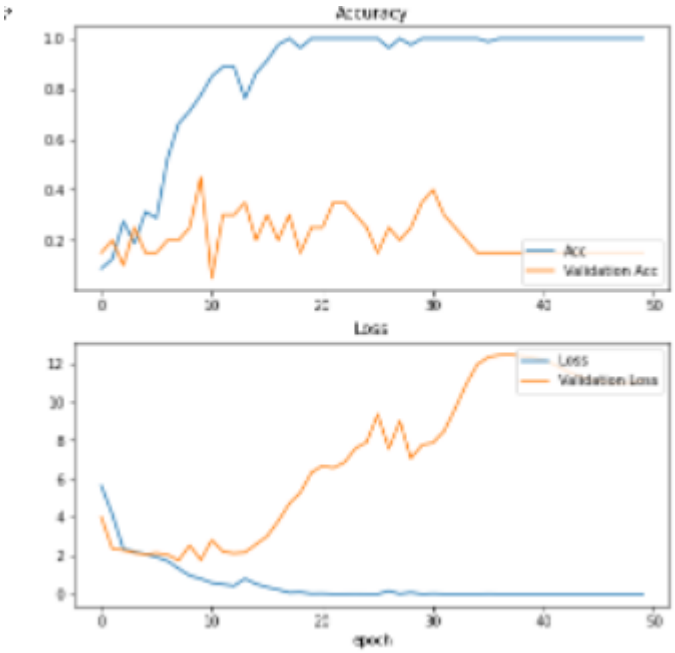
```
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
# Load Data
with open('/content/drive/MyDrive/Colab Notebooks/Data/data_food.pickle',
'rb') as f:
    (x_train, y_train) = pickle.load(f)

# Reshape Data
x_pre_1 = x_train[48]
x_pre_2 = x_train[22]
x_pre_3 = x_train[64]
x_pre_4 = x_train[75]
x_pre_5 = x_train[92]


# Preprocessing Data
x_train = x_train.astype('float32')
x_train /= 255

# Encoding Y
y_train = np_utils.to_categorical(y_train, 10)

# Shuffe Data
x_train, y_train = shuffle(x_train, y_train)
#chuan bi model
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same', input_shape = (150,150,3)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
```

```
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
model.summary()
```

#training

```
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics
= ['acc'])
his = model.fit(x_train, y_train, epochs = 50, batch_size = 64, validation
_split = 0.2)
```

#Draw bieu do



Hình 6 Biểu đồ training 10 món ăn

#add label

```
label = ['Banh_chung','Banh_day','Banh Mi','Bun dau mam tom','Che_buoi','c
om_tam','nem_ran','Pho','Thit kho hot vit','Vit lon']
```

test model

Hình 7 Kết quả training model

4. Training CNN MNIST

#add lib

```
from keras.models import Sequential, load_model
from keras.layers import Flatten,Dense
from tensorflow.keras.optimizers import Adam,SGD
from keras.callbacks import EarlyStopping
from keras.utils import np_utils
from keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
#chuan bi ham ve bieu do
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
```

```
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
# Load Data
(x_train,y_train),(x_test, y_test) = fashion_mnist.load_data()
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_test[i])
print(x_train.shape, y_train.shape)
x_train = x_train.reshape(60000,28,28,1)
x_test = x_test.reshape(10000,28,28,1)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /=255
x_test /=255
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
y_train.shape
```

#tao model

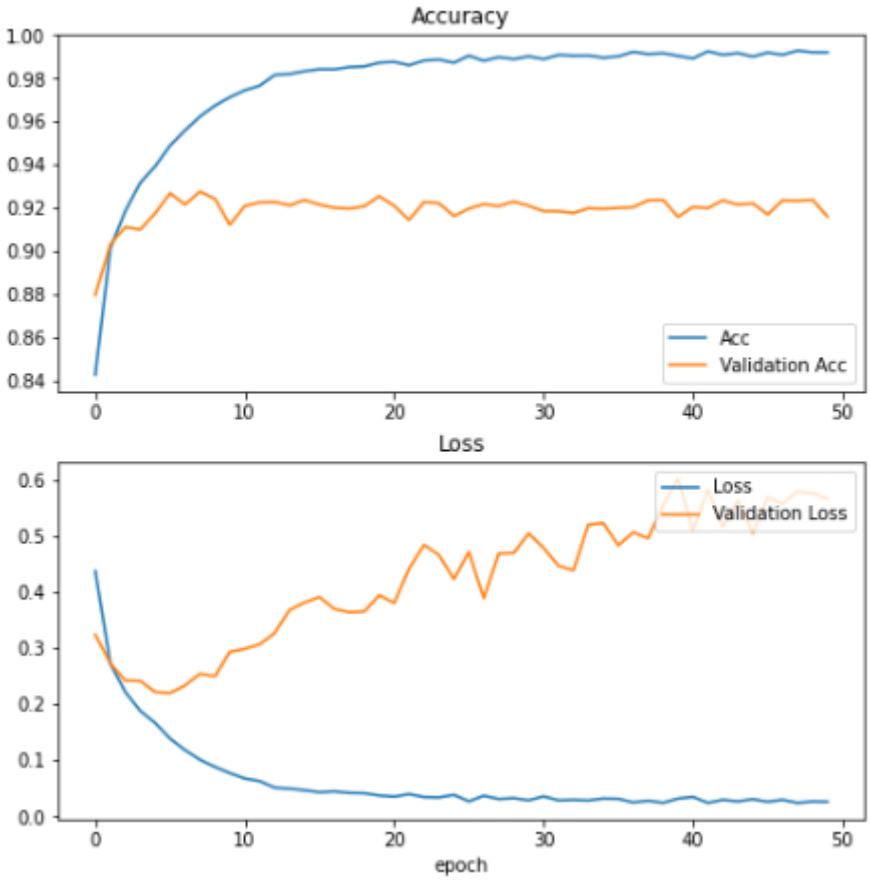
```
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same', input_shape = (28,28,1)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
model.summary()
```

#training

```
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics
= ['acc'])
```

```
his = model.fit(x_train, y_train, epochs = 50, batch_size = 64, validation
_split = 0.2)
```

#ket qua training



Hình 8 Ket qua training model

#save model

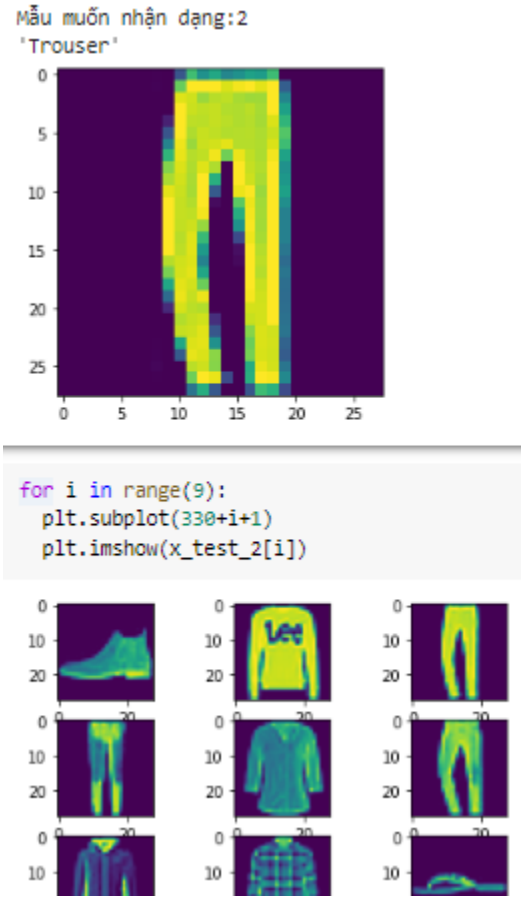
```
model.save('CNN_fashion_mnist.h5')
# chuan bi data test
x_test_2 = x_test.reshape(10000, 28,28)
```

label

```
label = ['T-
shirt/top','Trouser','Pullover','Dress','Coat','Sandal','Shirt','Sneaker',
'Bag','Ankle boot']
```

ket qua

```
b = int(input('Mẫu muốn nhận dạng:'))
plt.imshow(x_test_2[b])
label[np.argmax(model.predict(x_test_2)[b])]
```



Hình 9 Kết quả nhận diện

5. Training FASION_MNIST

#import thư viện

```
from keras.models import Sequential, load_model
from keras.layers import Flatten,Dense
from tensorflow.keras.optimizers import Adam,SGD
from keras.callbacks import EarlyStopping
from keras.utils import np_utils
from keras.datasets import mnist
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
```

chuan bi ham ve bieu do

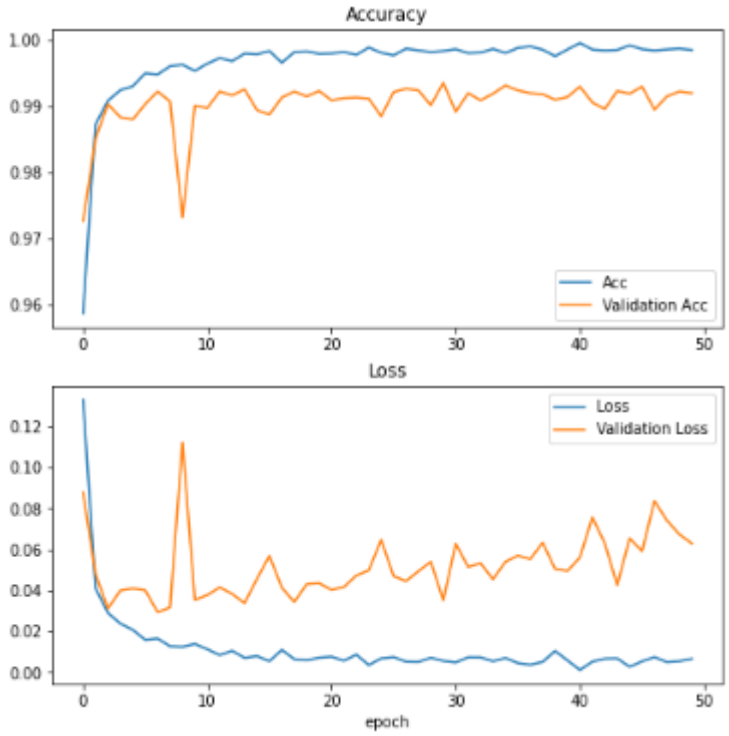
```
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
```

```
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
# Load Data
(x_train,y_train),(x_test, y_test) = mnist.load_data()
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_test[i])
print(x_train.shape, y_train.shape)
x_train = x_train.reshape(60000,28,28,1)
x_test = x_test.reshape(10000,28,28,1)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /=255
x_test /=255
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
y_train.shape
```

#chuan bi model

```
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same', input_shape = (28,28,1)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_unifo
rm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
```

```
model.add(Conv2D(128, (3,3), activation='relu',kernel_initializer='he_unif
orm', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
model.summary()
#training
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics
= ['acc'])
his = model.fit(x_train, y_train, epochs = 50, batch_size = 64, validation
_split = 0.2)
#ve bieu do
plot_history(his)
```



Hình 10 Kết quả training model

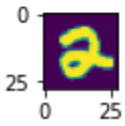
test

```
a = int(input('Random phan tu muon test:'))
print('Model nhan dang ra= ')
print(np.argmax(model.predict(x_test),axis=1)[a])
```

#ket qua

```
Random phan tu muon test:222
Model nhan dang ra=
2
```

```
(x_train,y_train),(x_test, y_test) = mnist.load_data()
for i in range(1):
  plt.subplot(330+a+1)
  plt.imshow(x_test[a])
```



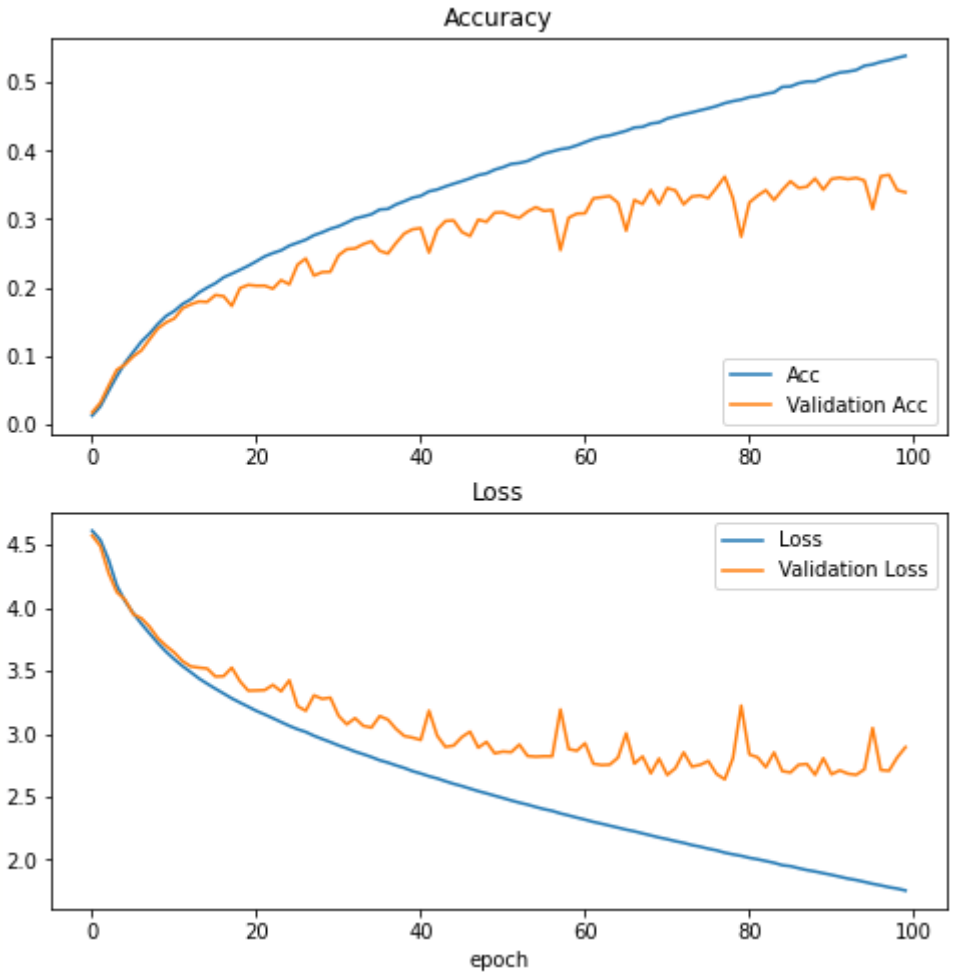
Hình 11 Kết quả test model

6. Training Cifar-100

```
# add lib
from keras.models import Sequential, load_model

from keras.layers import Flatten,Dense
from tensorflow.keras.optimizers import Adam,SGD
from keras.callbacks import EarlyStopping
from keras.utils import np_utils
from keras.datasets import  cifar100
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
#chuan bi ham ve bieu do
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation Acc')
  plt.legend(loc='lower right')
  plt.title('Accuracy')
  plt.subplot(2, 1, 2)
```

```
  plt.plot(loss, label='Loss')
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
# Load Data
(x_train,y_train),(x_test, y_test) = cifar100.load_data()
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_test[i])
print(x_train.shape, y_train.shape)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /=255
x_test /=255
y_train = np_utils.to_categorical(y_train,100)
y_test = np_utils.to_categorical(y_test,100)
#chuan bi model
model = Sequential()
model.add(Conv2D(32,(3,3),activation = 'relu',kernel_initializer='he_unifo
rm',padding ='same',input_shape = (32,32,3)))
model.add(Conv2D(32,(3,3),activation = 'relu',kernel_initializer='he_unifo
rm',padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3),activation = 'relu',kernel_initializer='he_unifo
rm',padding ='same'))
model.add(Conv2D(64,(3,3),activation = 'relu',kernel_initializer='he_unifo
rm',padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3),activation = 'relu',kernel_initializer='he_unif
orm',padding ='same'))
model.add(Conv2D(128,(3,3),activation = 'relu',kernel_initializer='he_unif
orm',padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(512, activation ='relu',kernel_initializer='he_uniform'))
model.add(Dense(100,activation ='softmax'))
model.summary()
#training
opt = SGD(lr = 0.001)
model.compile(optimizer = opt, loss ='categorical_crossentropy', metrics =
 ['acc'])
his = model.fit(x_train, y_train, batch_size = 64, epochs = 100, validatio
n_data = (x_test,y_test))
```

```
#bieu do model training duoc
plot_history(his)
```



Hình 12 Kết quả training model

#lưu model

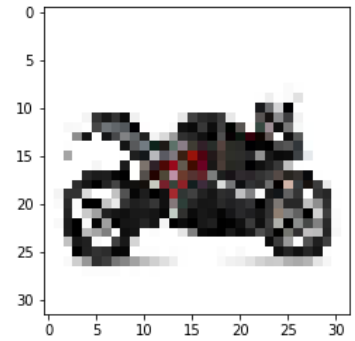
```
model.save('CNN_cifar100.h5')
```

creat label<99>

```
dict = {0: 'apple',1: 'aquarium_fish',2: 'baby',3: 'bear',4: 'beaver',5: '
bed',6: 'bee',7: 'beetle',8: 'bicycle',9: 'bottle',10:'bowl',11: 'boy',12:
 'bridge',13: 'bus',14: 'butterfly',15: 'camel',16: 'can',17: 'castle',18:
 'caterpillar',19: 'cattle',20: 'chair',21: 'chimpanzee',22: 'clock',23: '
cloud',24: 'cockroach',25:'couch',26: 'cra',27: 'crocodile',28:'cup',29: '
dinosaur',30: 'dolphin',31: 'elephant',32: 'flatfish',33: 'forest',34: 'fo
x',35: 'girl',36: 'hamster',37: 'house',38: 'kangaroo',39: 'keyboard',40:
'lamp',41: 'lawn_mower',42: 'leopard',43: 'lion',44: 'lizard',45: 'lobster
',46: 'man',47: 'maple_tree',48: 'motorcycle',49: 'mountain',50: 'mouse',5
1: 'mushroom',52: 'oak_tree',53: 'orange',54: 'orchid',55: 'otter',56: 'pa
```

```
lm_tree',57: 'pear',58: 'pickup_truck',59: 'pine_tree',60: 'plain',61: 'pl
ate',62: 'poppy',63: 'porcupine',64: 'possum',65: 'rabbit',66: 'raccoon',
67: 'ray',68: 'road',69: 'rocket',70: 'rose',71: 'sea',72: 'seal',3: 'shar
k',74: 'shrew',75: 'skunk',76: 'skyscraper',77: 'snail',
78: 'snake',79: 'spider',80: 'squirrel',81: 'streetcar',82: 'sunflower',
83: 'sweet_pepper',84: 'table',85: 'tank',86: 'telephone',87: 'television'
,88: 'tiger',89: 'tractor',90: 'train',91: 'trout',92: 'tulip',93: 'turtle
',94: 'wardrobe',95: 'whale',96: 'willow_tree',97: 'wolf',98: 'woman',99:
'worm'}
# Test
img = tf.keras.utils.load_img("testmt.jpg", target_size = (32,32))
plt.imshow(img)
plt.show()
img = tf.keras.utils.img_to_array(img)
img = img.reshape(1,32,32,3)
img = img.astype('float32')
img /=255
print(dict[np.argmax(model.predict(img),axis = 1)[0]])
#import ảnh ngoài
```

Hình 13 Anh motor dung de test



Hình 14 Ket qua training