TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH

KHOA CƠ KHÍ CHẾ TẠO MÁY

BỘ MÔN CƠ ĐIỆN TỬ

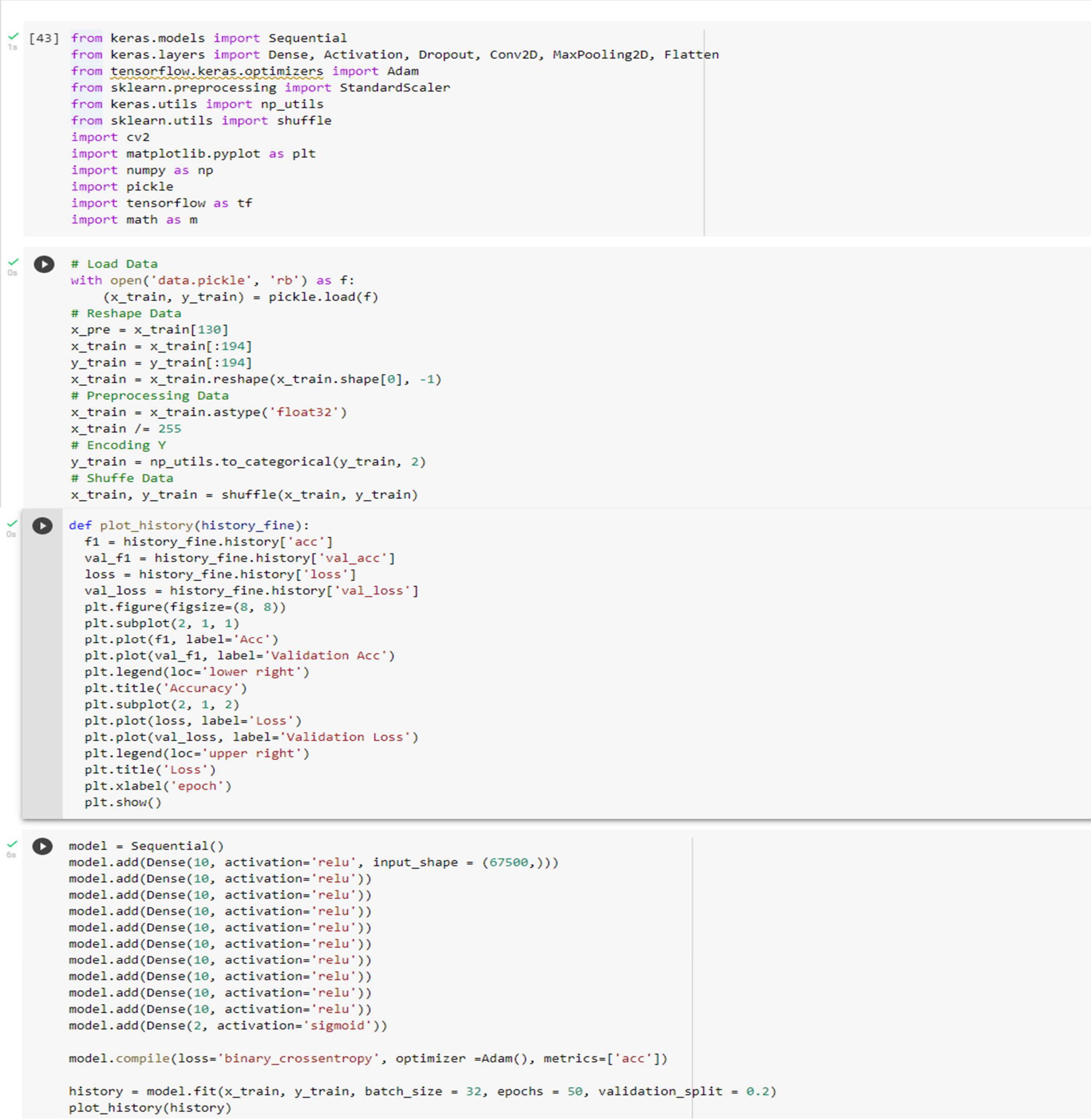
HCMUTE

BÀI TẬP VỀ NHÀ
MÔN AI TUẦN 12

GVHD:	**PGS.TS Nguyễn Trường Thịnh**
LỚP:	**Thứ 7**
TIẾT:	**3_6**
SVTH:	**Nguyễn Lê Vũ**
MSSV:	**19146428**

Tp. Hồ Chí Minh, tháng 05 năm 2022

1. Nhận diện khuôn mặt (dùng ANN)

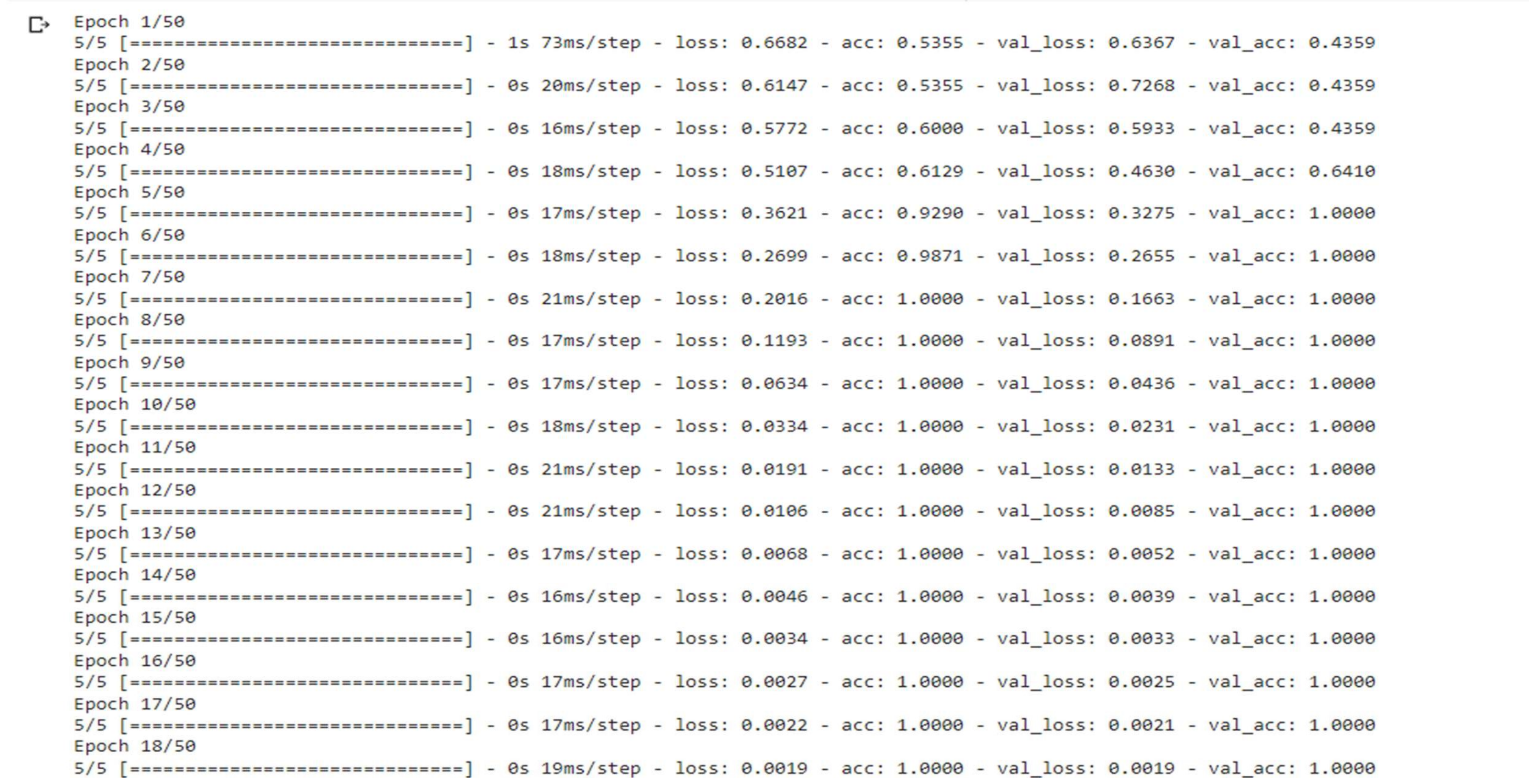


```python
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
import math as m
```

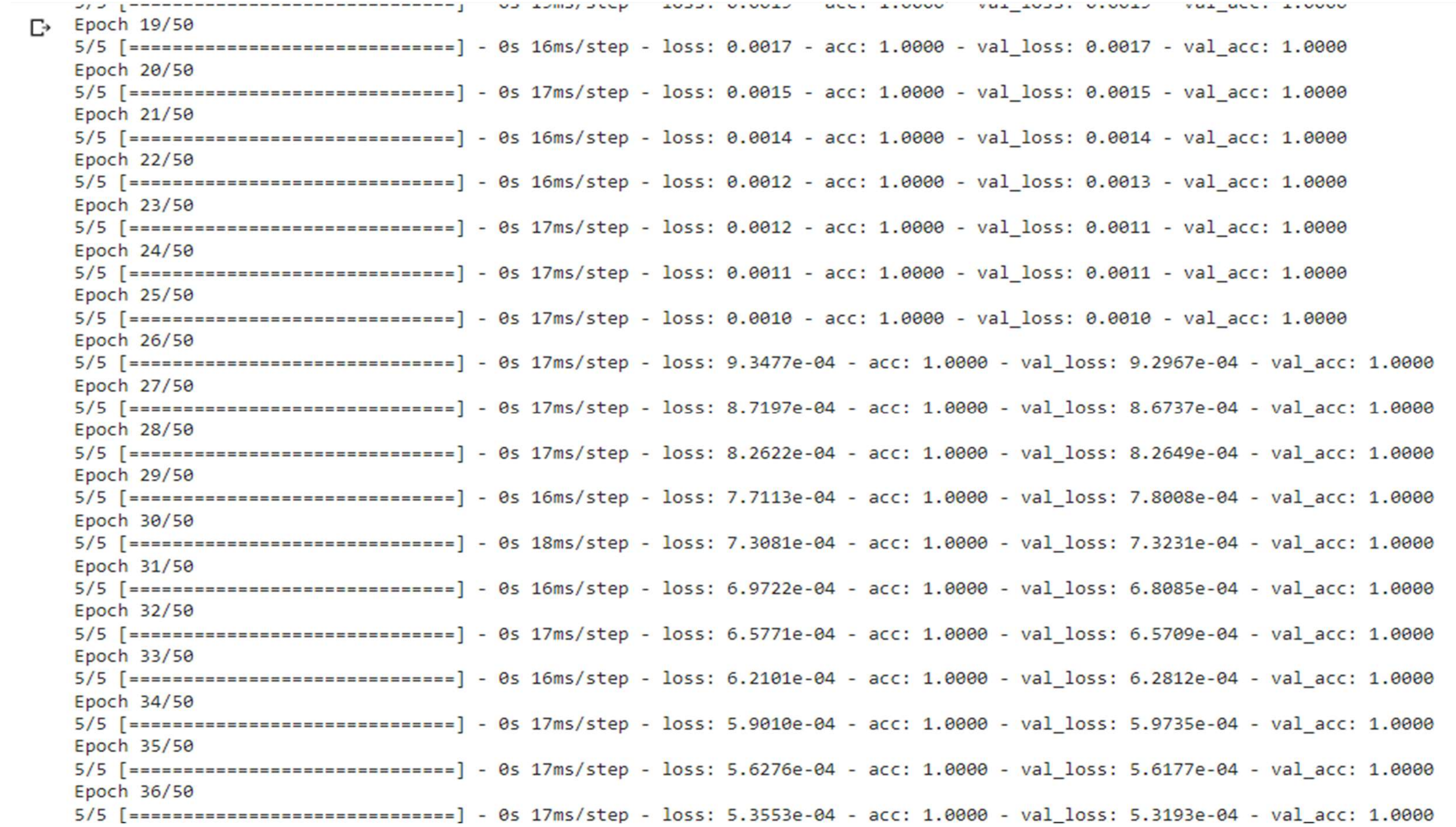
```python
# Load Data
with open('data.pickle', 'rb') as f:
    (x_train, y_train) = pickle.load(f)
# Reshape Data
x_pre = x_train[130]
x_train = x_train[:194]
y_train = y_train[:194]
x_train = x_train.reshape(x_train.shape[0], -1)
# Preprocessing Data
x_train = x_train.astype('float32')
x_train /= 255
# Encoding Y
y_train = np_utils.to_categorical(y_train, 2)
# Shuffe Data
x_train, y_train = shuffle(x_train, y_train)
```

```python
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()
```

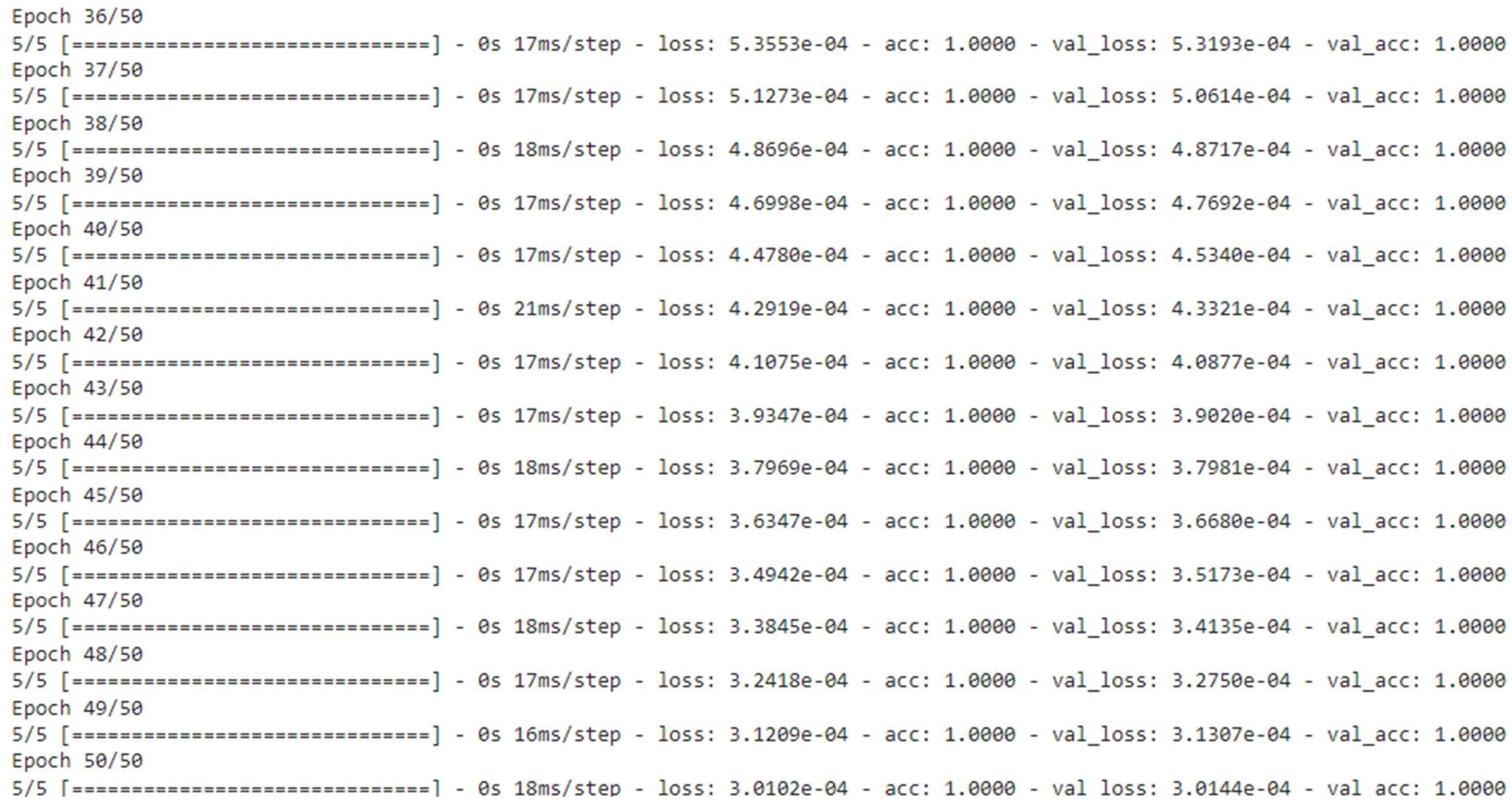
```python
model = Sequential()
model.add(Dense(10, activation='relu', input_shape = (67500,)))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(2, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer =Adam(), metrics=['acc'])

history = model.fit(x_train, y_train, batch_size = 32, epochs = 50, validation_split = 0.2)
plot_history(history)
```



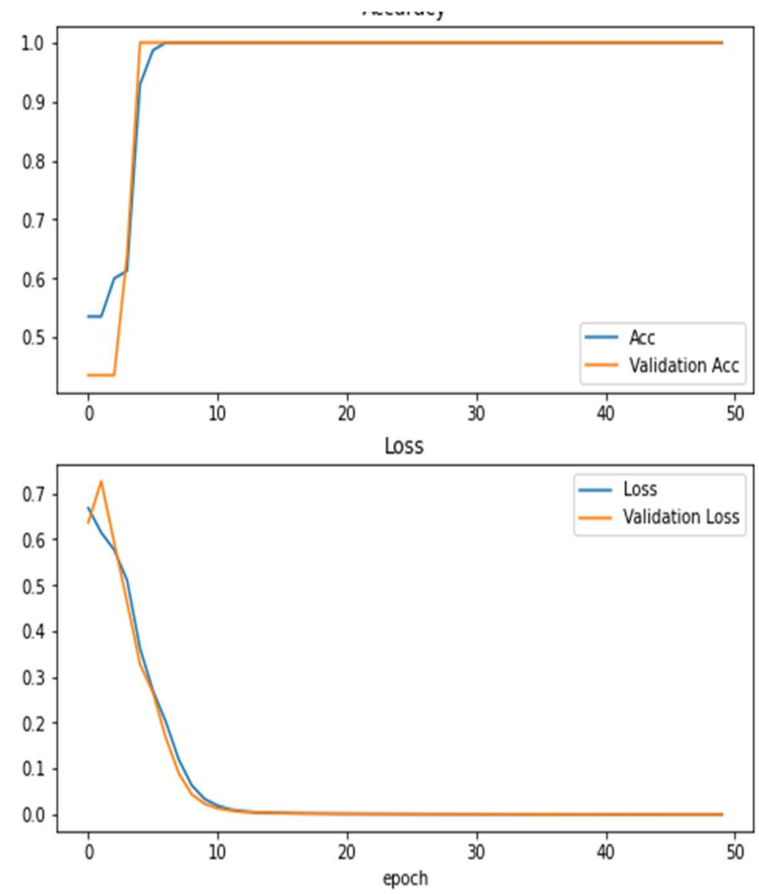
```
Epoch 1/50
5/5 [==============================] - 1s 73ms/step - loss: 0.6682 - acc: 0.5355 - val_loss: 0.6367 - val_acc: 0.4359
Epoch 2/50
5/5 [==============================] - 0s 20ms/step - loss: 0.6147 - acc: 0.5355 - val_loss: 0.7268 - val_acc: 0.4359
Epoch 3/50
5/5 [==============================] - 0s 16ms/step - loss: 0.5772 - acc: 0.6000 - val_loss: 0.5933 - val_acc: 0.4359
Epoch 4/50
5/5 [==============================] - 0s 18ms/step - loss: 0.5107 - acc: 0.6129 - val_loss: 0.4630 - val_acc: 0.6410
Epoch 5/50
5/5 [==============================] - 0s 17ms/step - loss: 0.3621 - acc: 0.9290 - val_loss: 0.3275 - val_acc: 1.0000
Epoch 6/50
5/5 [==============================] - 0s 18ms/step - loss: 0.2699 - acc: 0.9871 - val_loss: 0.2655 - val_acc: 1.0000
Epoch 7/50
5/5 [==============================] - 0s 21ms/step - loss: 0.2016 - acc: 1.0000 - val_loss: 0.1663 - val_acc: 1.0000
Epoch 8/50
5/5 [==============================] - 0s 17ms/step - loss: 0.1193 - acc: 1.0000 - val_loss: 0.0891 - val_acc: 1.0000
Epoch 9/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0634 - acc: 1.0000 - val_loss: 0.0436 - val_acc: 1.0000
Epoch 10/50
5/5 [==============================] - 0s 18ms/step - loss: 0.0334 - acc: 1.0000 - val_loss: 0.0231 - val_acc: 1.0000
Epoch 11/50
5/5 [==============================] - 0s 21ms/step - loss: 0.0191 - acc: 1.0000 - val_loss: 0.0133 - val_acc: 1.0000
Epoch 12/50
5/5 [==============================] - 0s 21ms/step - loss: 0.0106 - acc: 1.0000 - val_loss: 0.0085 - val_acc: 1.0000
Epoch 13/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0068 - acc: 1.0000 - val_loss: 0.0052 - val_acc: 1.0000
Epoch 14/50
5/5 [==============================] - 0s 16ms/step - loss: 0.0046 - acc: 1.0000 - val_loss: 0.0039 - val_acc: 1.0000
Epoch 15/50
5/5 [==============================] - 0s 16ms/step - loss: 0.0034 - acc: 1.0000 - val_loss: 0.0033 - val_acc: 1.0000
Epoch 16/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0027 - acc: 1.0000 - val_loss: 0.0025 - val_acc: 1.0000
Epoch 17/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0022 - acc: 1.0000 - val_loss: 0.0021 - val_acc: 1.0000
Epoch 18/50
5/5 [==============================] - 0s 19ms/step - loss: 0.0019 - acc: 1.0000 - val_loss: 0.0019 - val_acc: 1.0000
```

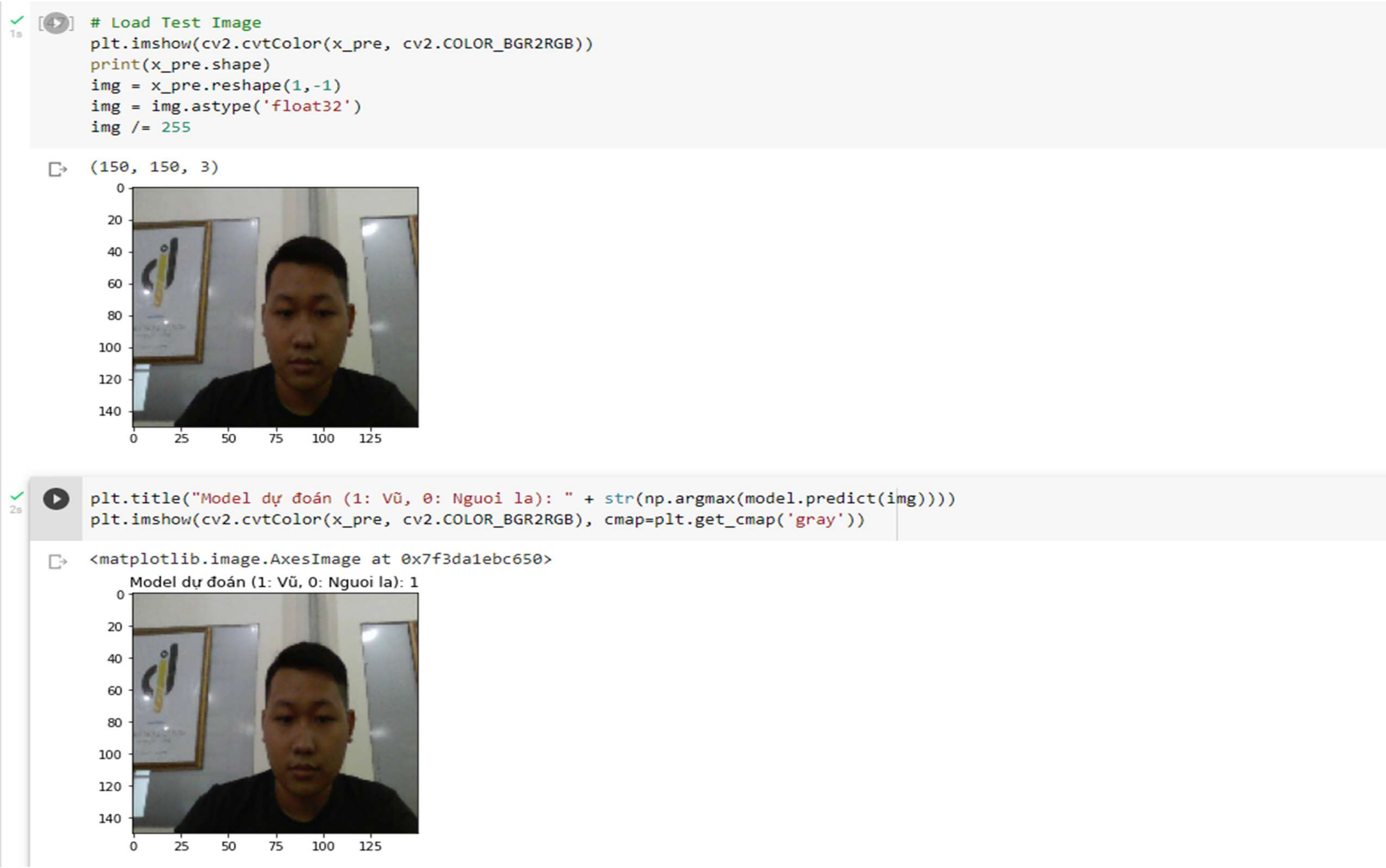


```
Epoch 19/50
5/5 [==============================] - 0s 16ms/step - loss: 0.0017 - acc: 1.0000 - val_loss: 0.0017 - val_acc: 1.0000
Epoch 20/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0015 - acc: 1.0000 - val_loss: 0.0015 - val_acc: 1.0000
Epoch 21/50
5/5 [==============================] - 0s 16ms/step - loss: 0.0014 - acc: 1.0000 - val_loss: 0.0014 - val_acc: 1.0000
Epoch 22/50
5/5 [==============================] - 0s 16ms/step - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0013 - val_acc: 1.0000
Epoch 23/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 24/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 25/50
5/5 [==============================] - 0s 17ms/step - loss: 0.0010 - acc: 1.0000 - val_loss: 0.0010 - val_acc: 1.0000
Epoch 26/50
5/5 [==============================] - 0s 17ms/step - loss: 9.3477e-04 - acc: 1.0000 - val_loss: 9.2967e-04 - val_acc: 1.0000
Epoch 27/50
5/5 [==============================] - 0s 17ms/step - loss: 8.7197e-04 - acc: 1.0000 - val_loss: 8.6737e-04 - val_acc: 1.0000
Epoch 28/50
5/5 [==============================] - 0s 17ms/step - loss: 8.2622e-04 - acc: 1.0000 - val_loss: 8.2649e-04 - val_acc: 1.0000
Epoch 29/50
5/5 [==============================] - 0s 16ms/step - loss: 7.7113e-04 - acc: 1.0000 - val_loss: 7.8008e-04 - val_acc: 1.0000
Epoch 30/50
5/5 [==============================] - 0s 18ms/step - loss: 7.3081e-04 - acc: 1.0000 - val_loss: 7.3231e-04 - val_acc: 1.0000
Epoch 31/50
5/5 [==============================] - 0s 16ms/step - loss: 6.9722e-04 - acc: 1.0000 - val_loss: 6.8085e-04 - val_acc: 1.0000
Epoch 32/50
5/5 [==============================] - 0s 17ms/step - loss: 6.5771e-04 - acc: 1.0000 - val_loss: 6.5709e-04 - val_acc: 1.0000
Epoch 33/50
5/5 [==============================] - 0s 16ms/step - loss: 6.2101e-04 - acc: 1.0000 - val_loss: 6.2812e-04 - val_acc: 1.0000
Epoch 34/50
5/5 [==============================] - 0s 17ms/step - loss: 5.9010e-04 - acc: 1.0000 - val_loss: 5.9735e-04 - val_acc: 1.0000
Epoch 35/50
5/5 [==============================] - 0s 17ms/step - loss: 5.6276e-04 - acc: 1.0000 - val_loss: 5.6177e-04 - val_acc: 1.0000
Epoch 36/50
5/5 [==============================] - 0s 17ms/step - loss: 5.3553e-04 - acc: 1.0000 - val_loss: 5.3193e-04 - val_acc: 1.0000
```



```
Epoch 36/50
5/5 [==============================] - 0s 17ms/step - loss: 5.3553e-04 - acc: 1.0000 - val_loss: 5.3193e-04 - val_acc: 1.0000
Epoch 37/50
5/5 [==============================] - 0s 17ms/step - loss: 5.1273e-04 - acc: 1.0000 - val_loss: 5.0614e-04 - val_acc: 1.0000
Epoch 38/50
5/5 [==============================] - 0s 18ms/step - loss: 4.8696e-04 - acc: 1.0000 - val_loss: 4.8717e-04 - val_acc: 1.0000
Epoch 39/50
5/5 [==============================] - 0s 17ms/step - loss: 4.6998e-04 - acc: 1.0000 - val_loss: 4.7692e-04 - val_acc: 1.0000
Epoch 40/50
5/5 [==============================] - 0s 17ms/step - loss: 4.4780e-04 - acc: 1.0000 - val_loss: 4.5340e-04 - val_acc: 1.0000
Epoch 41/50
5/5 [==============================] - 0s 21ms/step - loss: 4.2919e-04 - acc: 1.0000 - val_loss: 4.3321e-04 - val_acc: 1.0000
Epoch 42/50
5/5 [==============================] - 0s 17ms/step - loss: 4.1075e-04 - acc: 1.0000 - val_loss: 4.0877e-04 - val_acc: 1.0000
Epoch 43/50
5/5 [==============================] - 0s 17ms/step - loss: 3.9347e-04 - acc: 1.0000 - val_loss: 3.9020e-04 - val_acc: 1.0000
Epoch 44/50
5/5 [==============================] - 0s 18ms/step - loss: 3.7969e-04 - acc: 1.0000 - val_loss: 3.7981e-04 - val_acc: 1.0000
Epoch 45/50
5/5 [==============================] - 0s 17ms/step - loss: 3.6347e-04 - acc: 1.0000 - val_loss: 3.6680e-04 - val_acc: 1.0000
Epoch 46/50
5/5 [==============================] - 0s 17ms/step - loss: 3.4942e-04 - acc: 1.0000 - val_loss: 3.5173e-04 - val_acc: 1.0000
Epoch 47/50
5/5 [==============================] - 0s 18ms/step - loss: 3.3845e-04 - acc: 1.0000 - val_loss: 3.4135e-04 - val_acc: 1.0000
Epoch 48/50
5/5 [==============================] - 0s 17ms/step - loss: 3.2418e-04 - acc: 1.0000 - val_loss: 3.2750e-04 - val_acc: 1.0000
Epoch 49/50
5/5 [==============================] - 0s 16ms/step - loss: 3.1209e-04 - acc: 1.0000 - val_loss: 3.1307e-04 - val_acc: 1.0000
Epoch 50/50
5/5 [==============================] - 0s 18ms/step - loss: 3.0102e-04 - acc: 1.0000 - val_loss: 3.0144e-04 - val_acc: 1.0000
```





```
# Load Test Image
plt.imshow(cv2.cvtColor(x_pre, cv2.COLOR_BGR2RGB))
print(x_pre.shape)
img = x_pre.reshape(1,-1)
img = img.astype('float32')
img /= 255
```

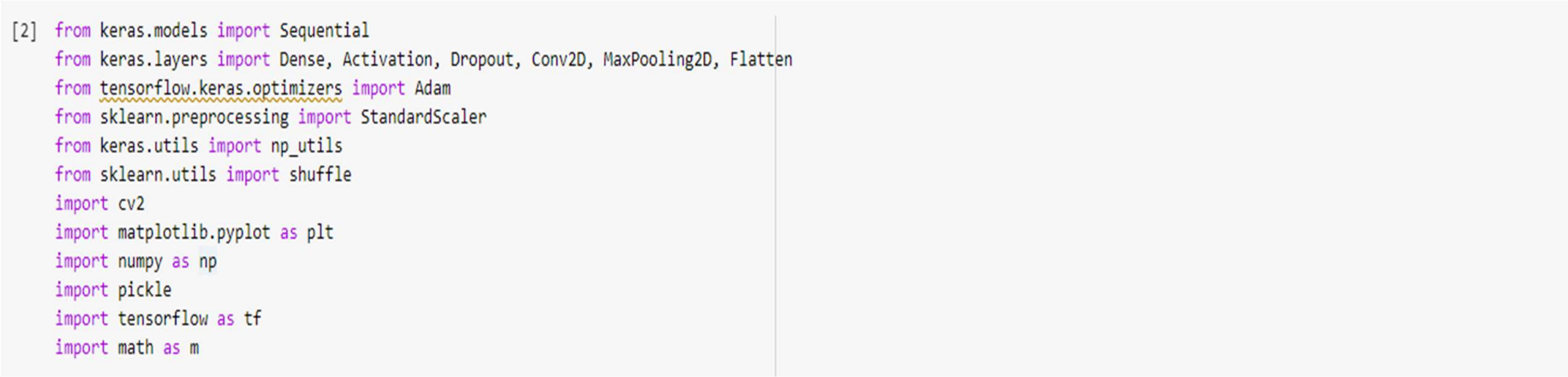
```
(150, 150, 3)
```

```
plt.title("Model dự đoán (1: Vũ, 0: Nguoi la): " + str(np.argmax(model.predict(img))))
plt.imshow(cv2.cvtColor(x_pre, cv2.COLOR_BGR2RGB), cmap=plt.get_cmap('gray'))
```

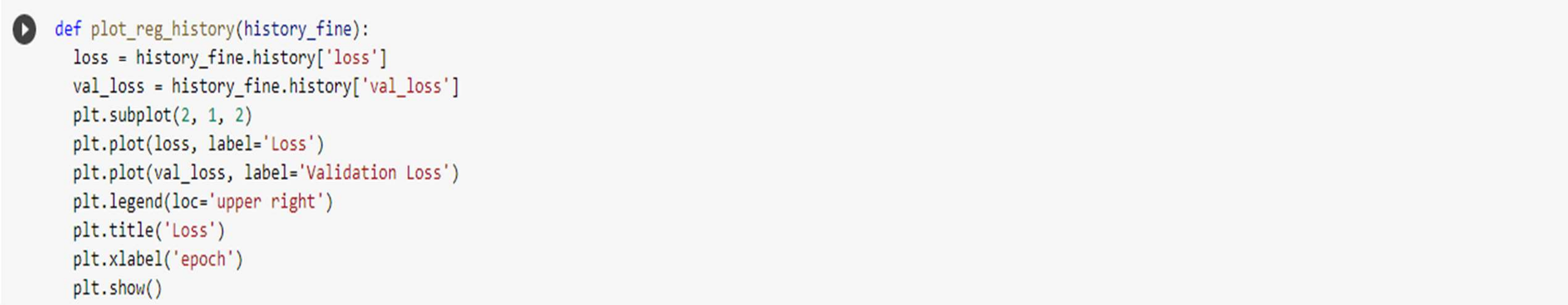
```
<matplotlib.image.AxesImage at 0x7f3da1ebc650>
```

Model dự đoán (1: Vũ, 0: Nguoi la): 1

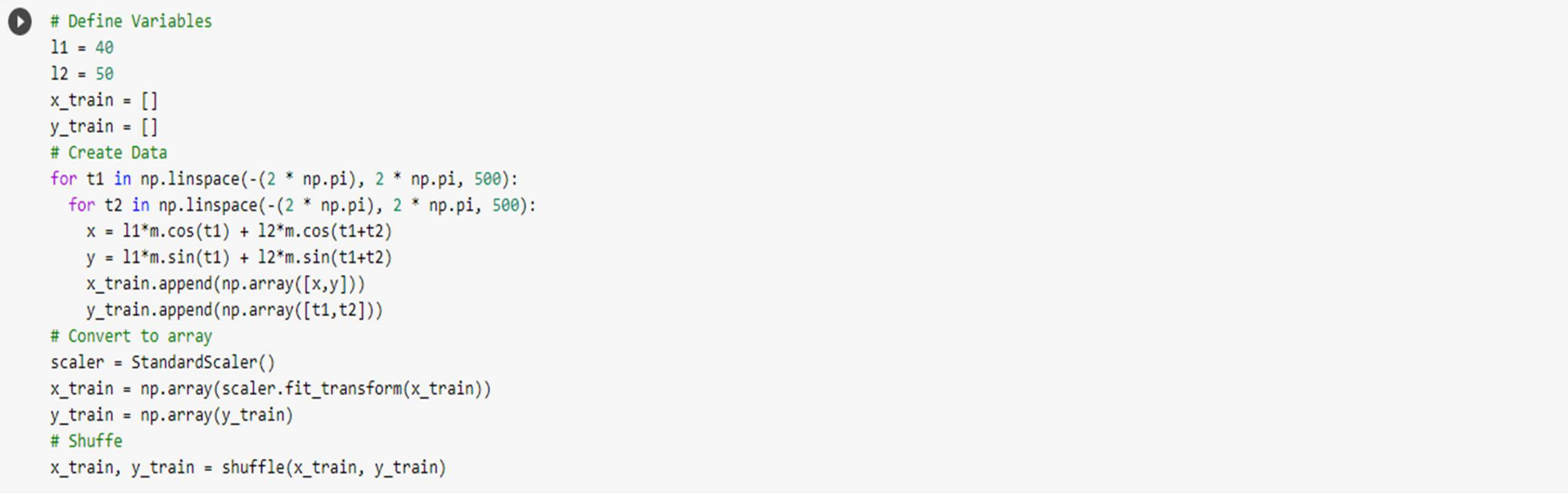
2. Robot 2 bậc



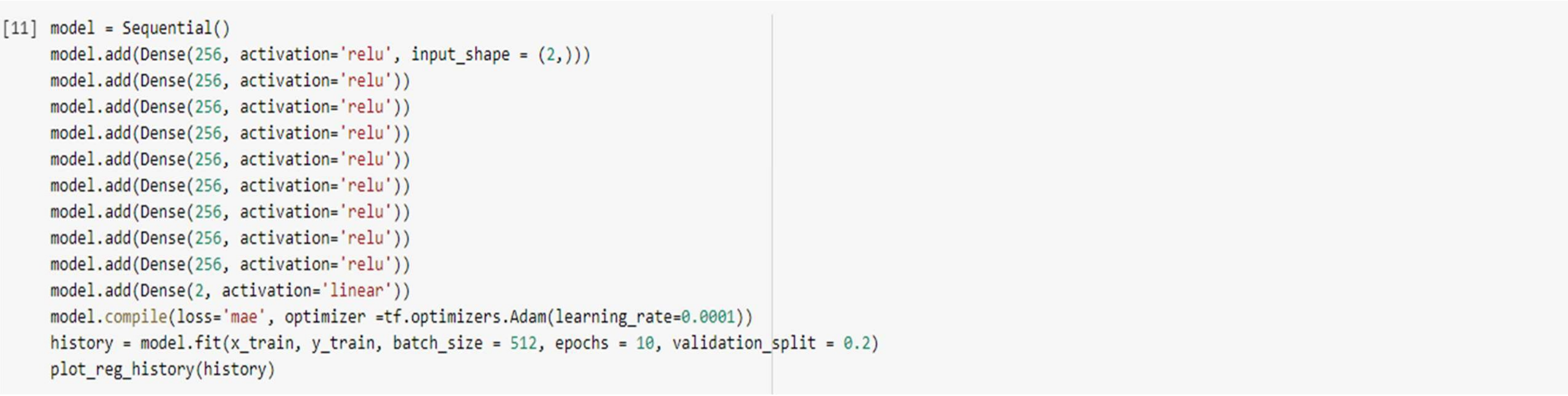
```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
import math as m
```



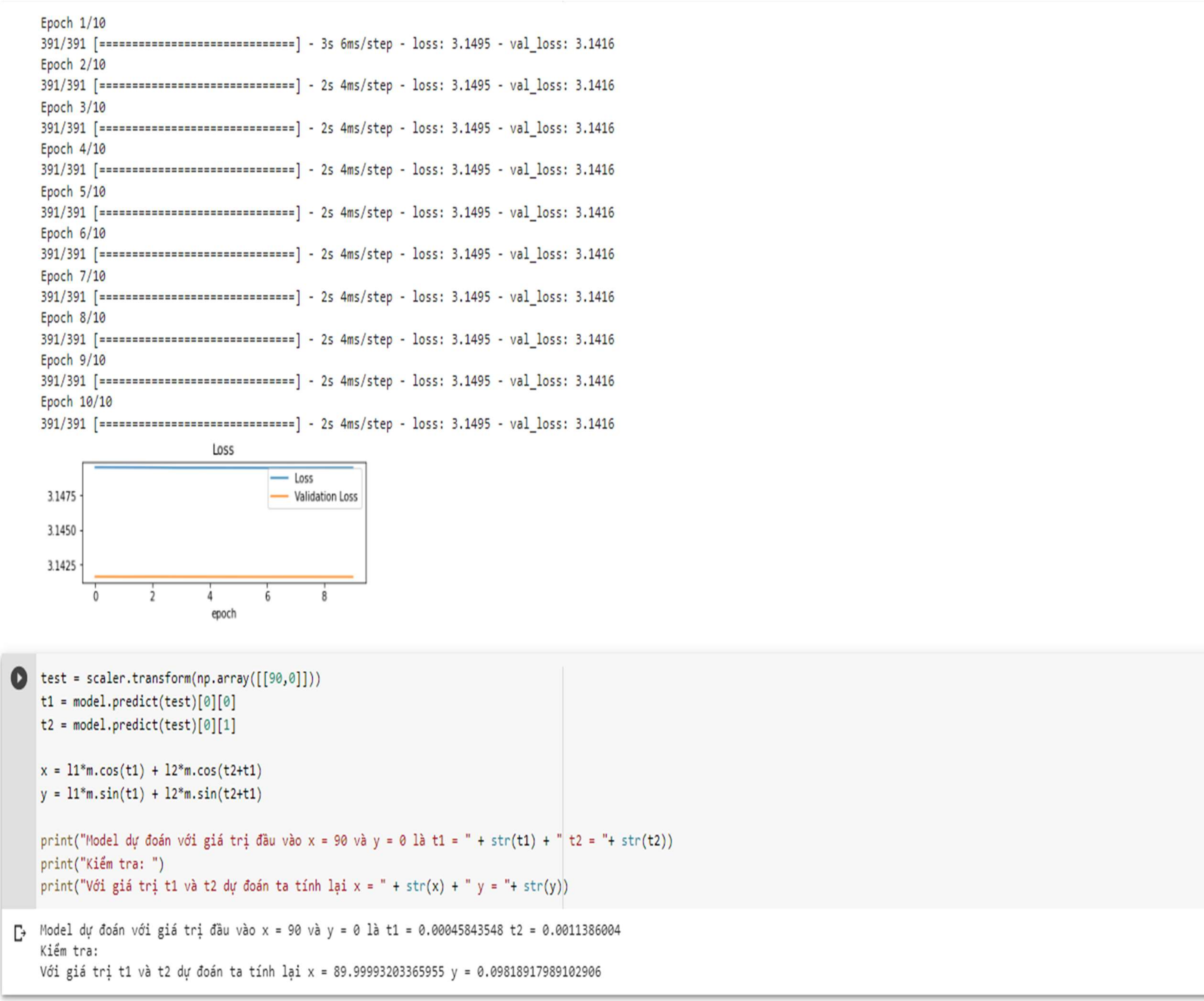
```
def plot_reg_history(history_fine):
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.subplot(2, 1, 2)
  plt.plot(loss, label='Loss')
  plt.plot(val_loss, label='Validation Loss')
  plt.legend(loc='upper right')
  plt.title('Loss')
  plt.xlabel('epoch')
  plt.show()
```



```
# Define Variables
l1 = 40
l2 = 50
x_train = []
y_train = []
# Create Data
for t1 in np.linspace(-(2 * np.pi), 2 * np.pi, 500):
  for t2 in np.linspace(-(2 * np.pi), 2 * np.pi, 500):
    x = l1*m.cos(t1) + l2*m.cos(t1+t2)
    y = l1*m.sin(t1) + l2*m.sin(t1+t2)
    x_train.append(np.array([x,y]))
    y_train.append(np.array([t1,t2]))
# Convert to array
scaler = StandardScaler()
x_train = np.array(scaler.fit_transform(x_train))
y_train = np.array(y_train)
# Shuffe
x_train, y_train = shuffle(x_train, y_train)
```



```
model = Sequential()
model.add(Dense(256, activation='relu', input_shape = (2,)))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(2, activation='linear'))
model.compile(loss='mae', optimizer =tf.optimizers.Adam(learning_rate=0.0001))
history = model.fit(x_train, y_train, batch_size = 512, epochs = 10, validation_split = 0.2)
plot_reg_history(history)
```

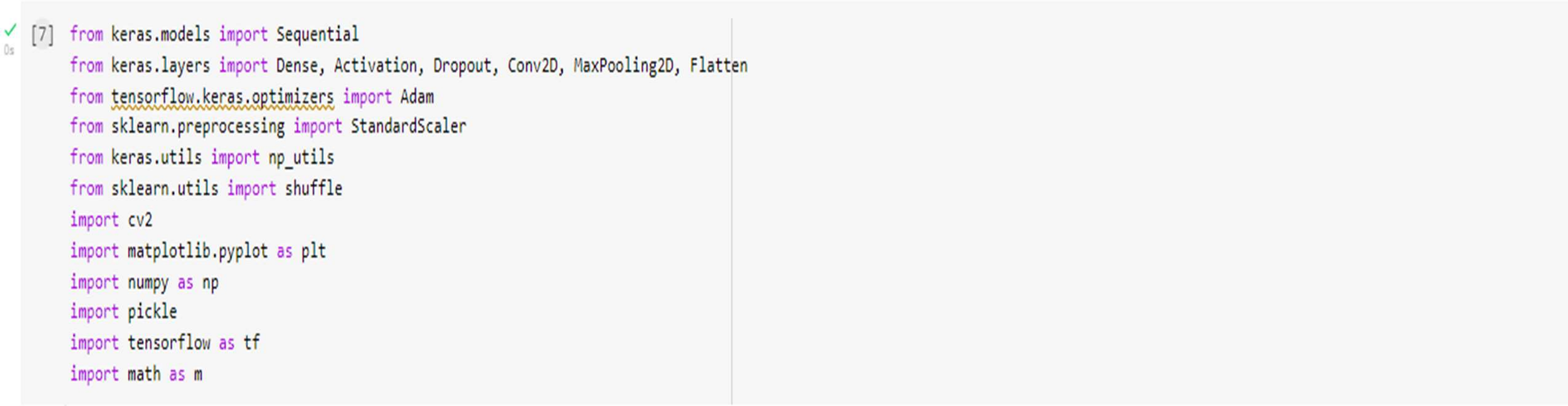


```
Epoch 1/10
391/391 [==============================] - 3s 6ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 2/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 3/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 4/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 5/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 6/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 7/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 8/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 9/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
Epoch 10/10
391/391 [==============================] - 2s 4ms/step - loss: 3.1495 - val_loss: 3.1416
```

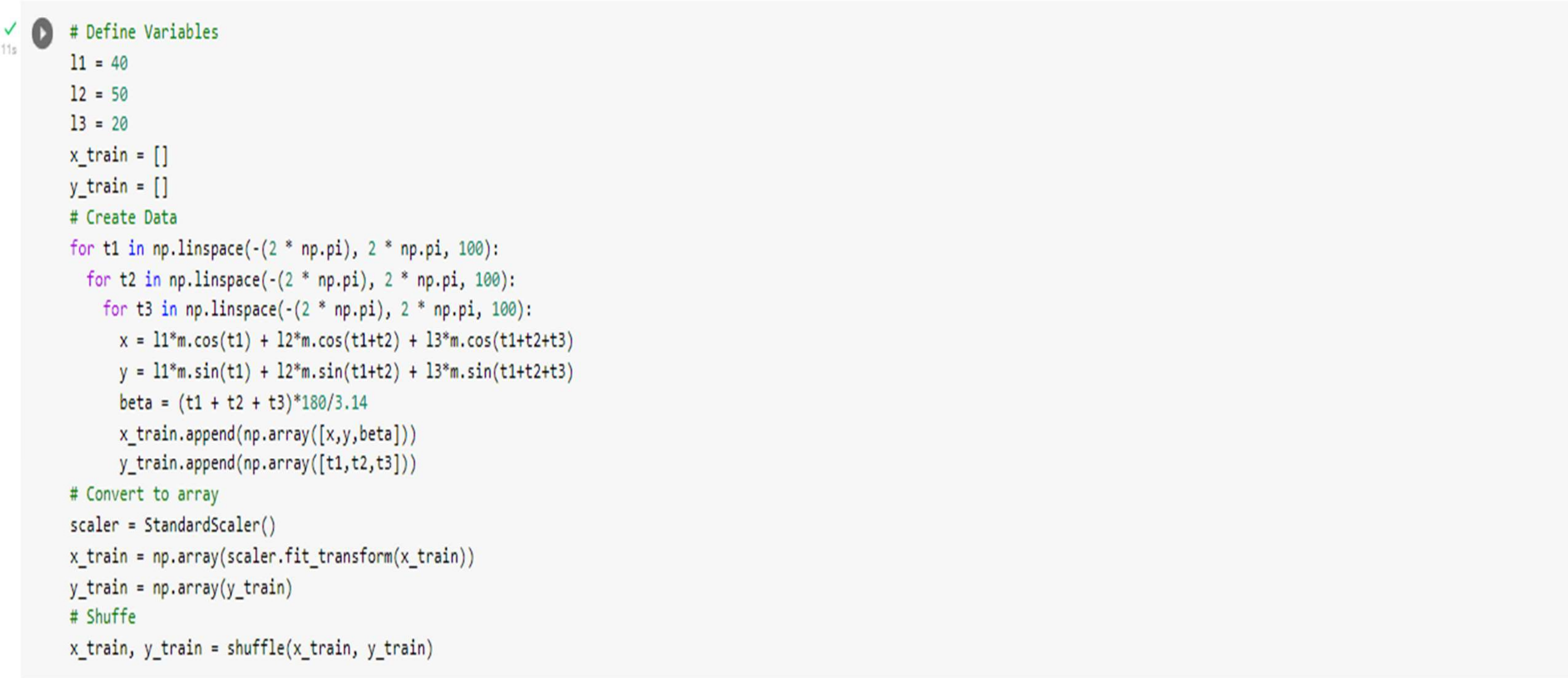
```python
test = scaler.transform(np.array([[90,0]]))
t1 = model.predict(test)[0][0]
t2 = model.predict(test)[0][1]

x = l1*m.cos(t1) + l2*m.cos(t2+t1)
y = l1*m.sin(t1) + l2*m.sin(t2+t1)

print("Model dự đoán với giá trị đầu vào x = 90 và y = 0 là t1 = " + str(t1) + " t2 = "+ str(t2))
print("Kiểm tra: ")
print("Với giá trị t1 và t2 dự đoán ta tính lại x = " + str(x) + " y = "+ str(y))
```

```
Model dự đoán với giá trị đầu vào x = 90 và y = 0 là t1 = 0.00045843548 t2 = 0.0011386004
Kiểm tra:
Với giá trị t1 và t2 dự đoán ta tính lại x = 89.99993203365955 y = 0.09818917989102906
```

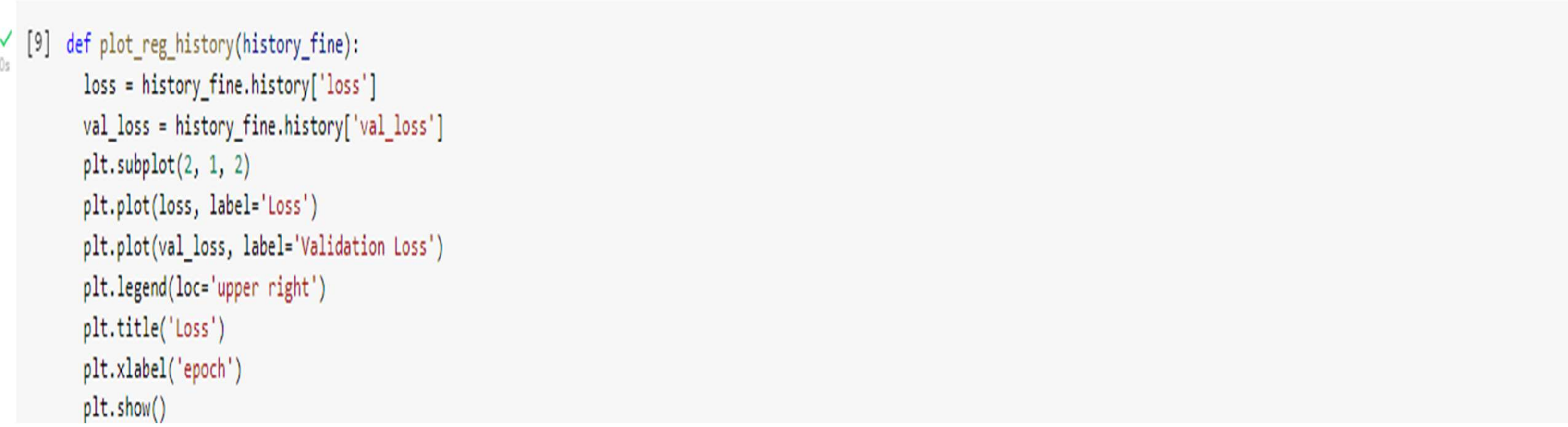
3. Robot 3 bậc



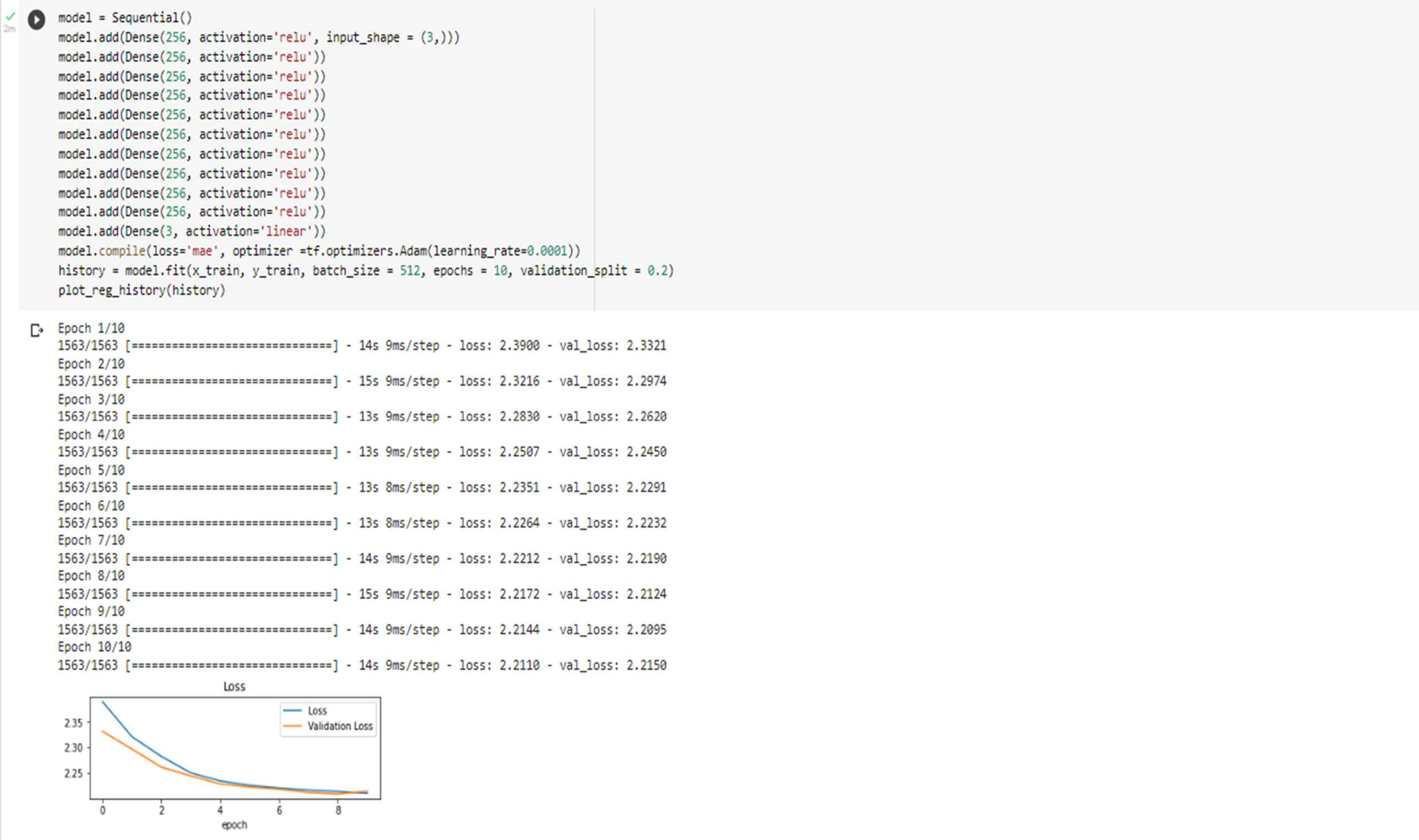
```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
import math as m
```



```
# Define Variables
l1 = 40
l2 = 50
l3 = 20
x_train = []
y_train = []
# Create Data
for t1 in np.linspace(-(2 * np.pi), 2 * np.pi, 100):
  for t2 in np.linspace(-(2 * np.pi), 2 * np.pi, 100):
    for t3 in np.linspace(-(2 * np.pi), 2 * np.pi, 100):
      x = l1*m.cos(t1) + l2*m.cos(t1+t2) + l3*m.cos(t1+t2+t3)
      y = l1*m.sin(t1) + l2*m.sin(t1+t2) + l3*m.sin(t1+t2+t3)
      beta = (t1 + t2 + t3)*180/3.14
      x_train.append(np.array([x,y,beta]))
      y_train.append(np.array([t1,t2,t3]))
# Convert to array
scaler = StandardScaler()
x_train = np.array(scaler.fit_transform(x_train))
y_train = np.array(y_train)
# Shuffe
x_train, y_train = shuffle(x_train, y_train)
```

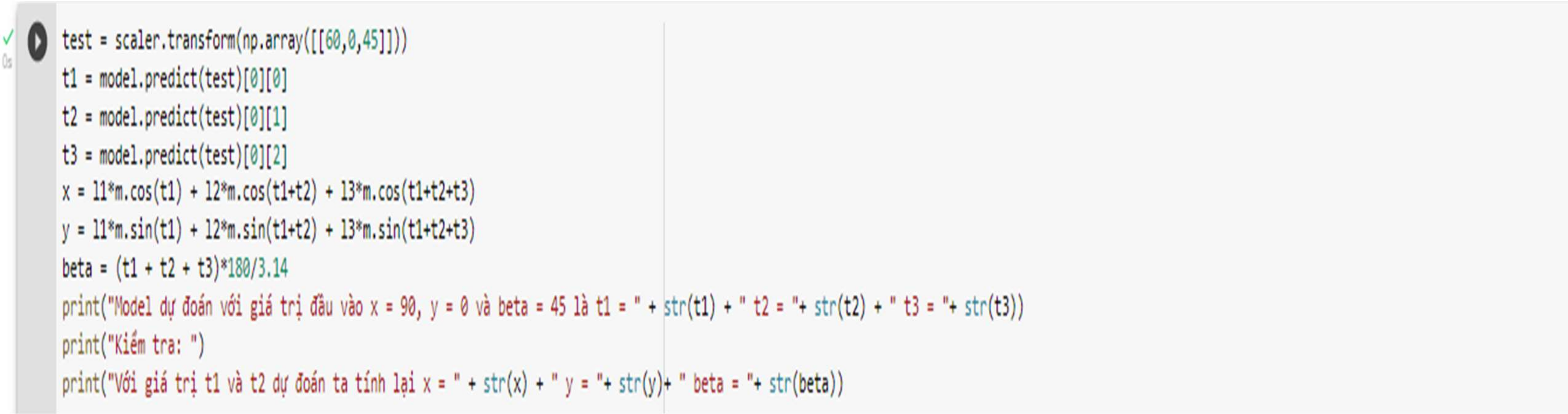


```
def plot_reg_history(history_fine):
    loss = history_fine.history['loss']
    val_loss = history_fine.history['val_loss']
    plt.subplot(2, 1, 2)
    plt.plot(loss, label='Loss')
    plt.plot(val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title('Loss')
    plt.xlabel('epoch')
    plt.show()
```



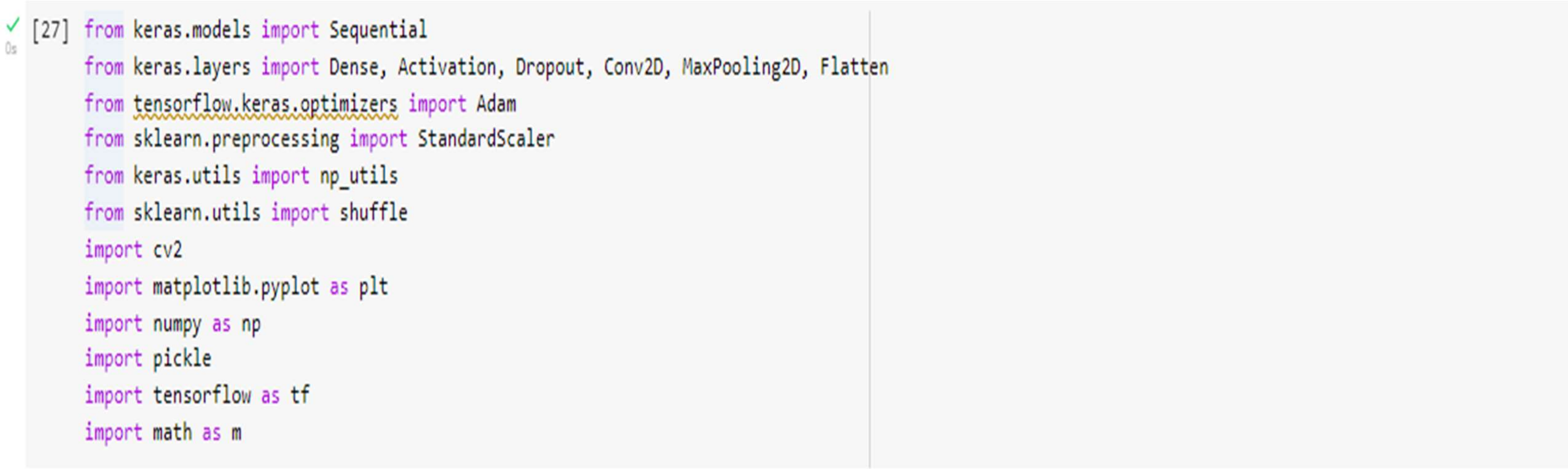
```
model = Sequential()
model.add(Dense(256, activation='relu', input_shape = (3,)))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(3, activation='linear'))
model.compile(loss='mae', optimizer =tf.optimizers.Adam(learning_rate=0.0001))
history = model.fit(x_train, y_train, batch_size = 512, epochs = 10, validation_split = 0.2)
plot_reg_history(history)
```

```
Epoch 1/10
1563/1563 [==============================] - 14s 9ms/step - loss: 2.3900 - val_loss: 2.3321
Epoch 2/10
1563/1563 [==============================] - 15s 9ms/step - loss: 2.3216 - val_loss: 2.2974
Epoch 3/10
1563/1563 [==============================] - 13s 9ms/step - loss: 2.2830 - val_loss: 2.2620
Epoch 4/10
1563/1563 [==============================] - 13s 9ms/step - loss: 2.2507 - val_loss: 2.2450
Epoch 5/10
1563/1563 [==============================] - 13s 8ms/step - loss: 2.2351 - val_loss: 2.2291
Epoch 6/10
1563/1563 [==============================] - 13s 8ms/step - loss: 2.2264 - val_loss: 2.2232
Epoch 7/10
1563/1563 [==============================] - 14s 9ms/step - loss: 2.2212 - val_loss: 2.2190
Epoch 8/10
1563/1563 [==============================] - 15s 9ms/step - loss: 2.2172 - val_loss: 2.2124
Epoch 9/10
1563/1563 [==============================] - 14s 9ms/step - loss: 2.2144 - val_loss: 2.2095
Epoch 10/10
1563/1563 [==============================] - 14s 9ms/step - loss: 2.2110 - val_loss: 2.2150
```

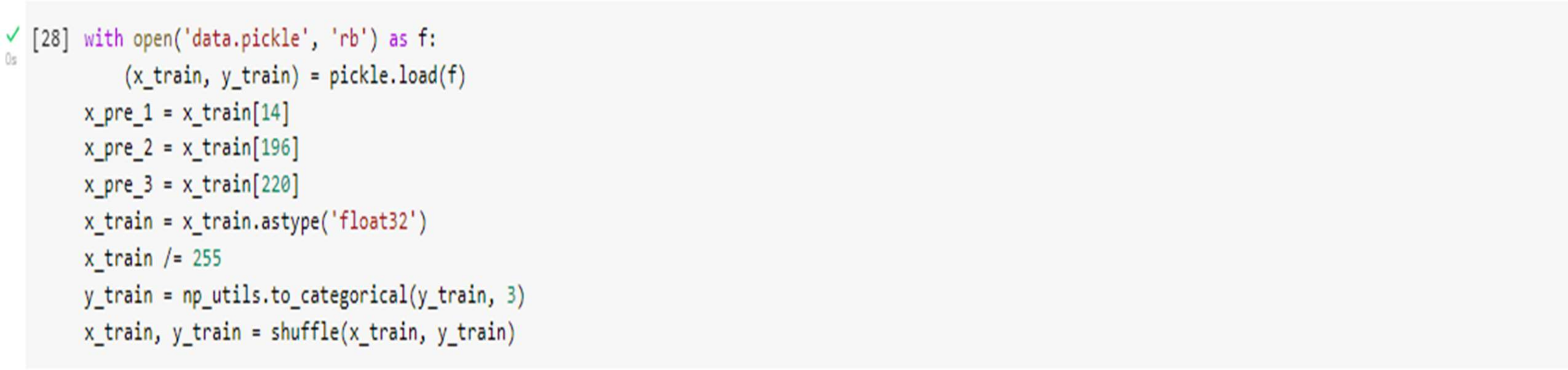


```
test = scaler.transform(np.array([[60,0,45]]))
t1 = model.predict(test)[0][0]
t2 = model.predict(test)[0][1]
t3 = model.predict(test)[0][2]
x = l1*m.cos(t1) + l2*m.cos(t1+t2) + l3*m.cos(t1+t2+t3)
y = l1*m.sin(t1) + l2*m.sin(t1+t2) + l3*m.sin(t1+t2+t3)
beta = (t1 + t2 + t3)*180/3.14
print("Model dự đoán với giá trị đầu vào x = 90, y = 0 và beta = 45 là t1 = " + str(t1) + " t2 = "+ str(t2) + " t3 = "+ str(t3))
print("Kiểm tra: ")
print("Với giá trị t1 và t2 dự đoán ta tính lại x = " + str(x) + " y = "+ str(y)+ " beta = "+ str(beta))
```

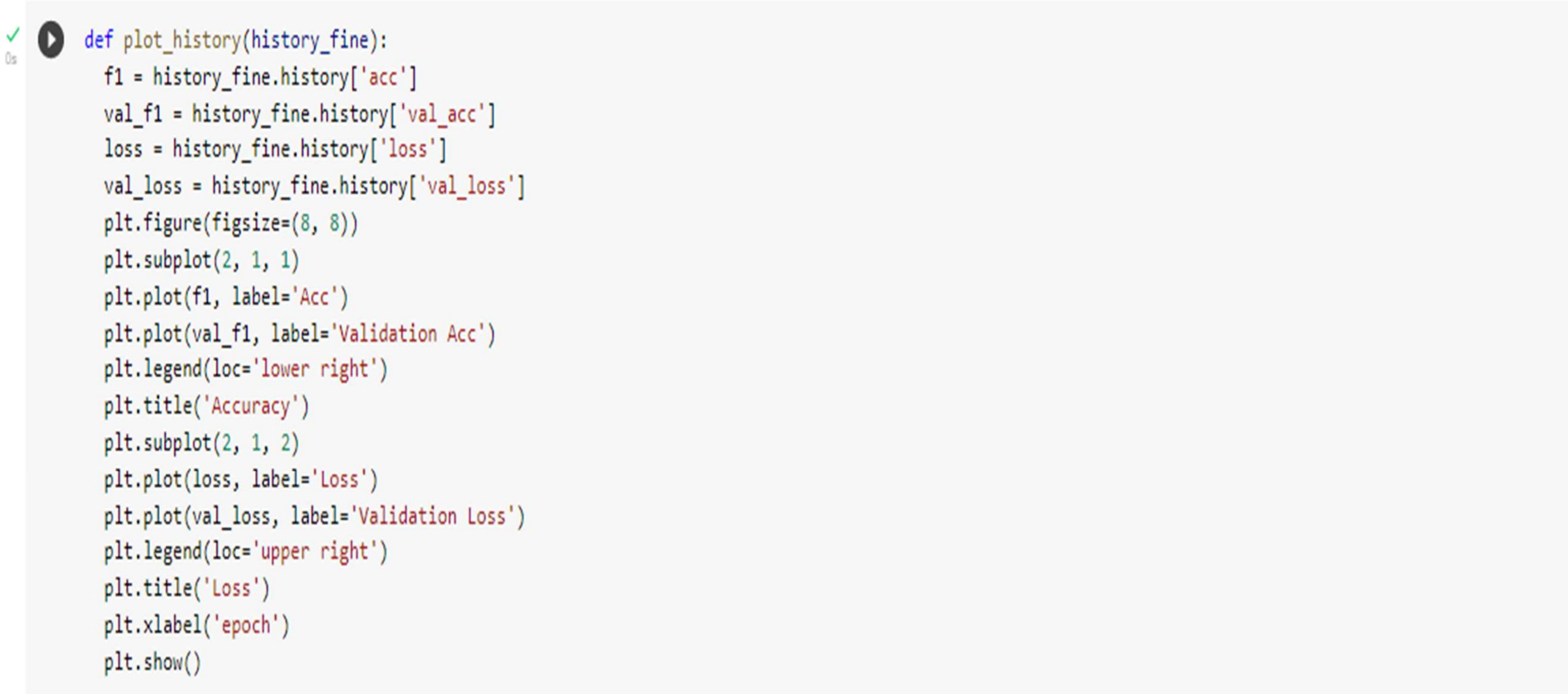
4. Nhận diện khuôn mặt (dùng CNN)



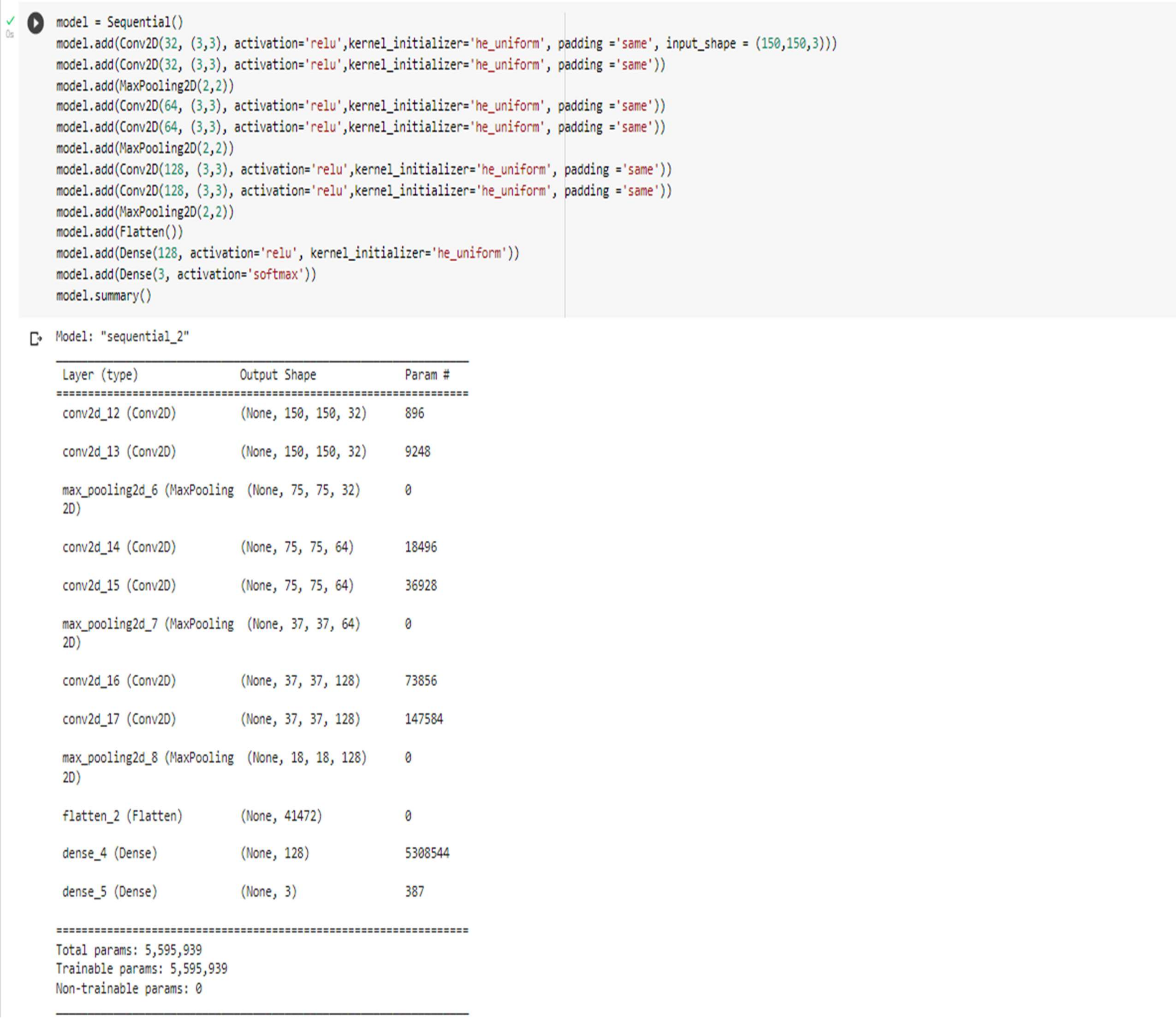
```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pickle
import tensorflow as tf
import math as m
```



```
with open('data.pickle', 'rb') as f:
    (x_train, y_train) = pickle.load(f)
x_pre_1 = x_train[14]
x_pre_2 = x_train[196]
x_pre_3 = x_train[220]
x_train = x_train.astype('float32')
x_train /= 255
y_train = np_utils.to_categorical(y_train, 3)
x_train, y_train = shuffle(x_train, y_train)
```

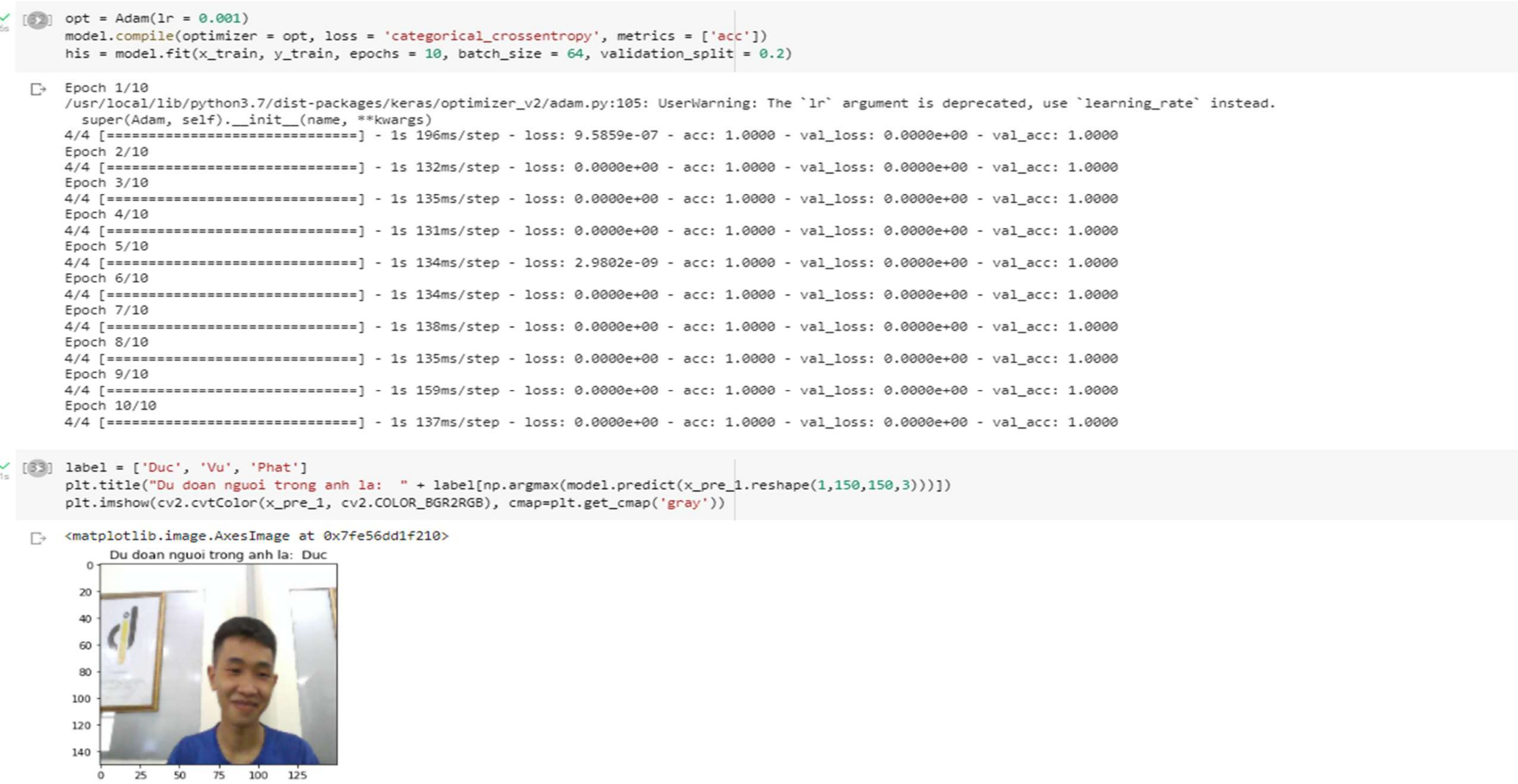


```
def plot_history(history_fine):
  f1 = history_fine.history['acc']
  val_f1 = history_fine.history['val_acc']
  loss = history_fine.history['loss']
  val_loss = history_fine.history['val_loss']
  plt.figure(figsize=(8, 8))
  plt.subplot(2, 1, 1)
  plt.plot(f1, label='Acc')
  plt.plot(val_f1, label='Validation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



```
model = Sequential()
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_uniform', padding ='same', input_shape = (150,150,3)))
model.add(Conv2D(32, (3,3), activation='relu',kernel_initializer='he_uniform', padding ='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64, (3,3), activation='relu',kernel_initializer='he_uniform', padding ='same'))
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(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.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 150, 150, 32)      896

 conv2d_13 (Conv2D)          (None, 150, 150, 32)      9248

 max_pooling2d_6 (MaxPooling  (None, 75, 75, 32)       0
 2D)

 conv2d_14 (Conv2D)          (None, 75, 75, 64)        18496

 conv2d_15 (Conv2D)          (None, 75, 75, 64)        36928

 max_pooling2d_7 (MaxPooling  (None, 37, 37, 64)       0
 2D)

 conv2d_16 (Conv2D)          (None, 37, 37, 128)       73856

 conv2d_17 (Conv2D)          (None, 37, 37, 128)       147584

 max_pooling2d_8 (MaxPooling  (None, 18, 18, 128)      0
 2D)

 flatten_2 (Flatten)         (None, 41472)             0

 dense_4 (Dense)             (None, 128)               5308544

 dense_5 (Dense)             (None, 3)                 387

=================================================================
Total params: 5,595,939
Trainable params: 5,595,939
Non-trainable params: 0
_________________________________________________________________
```



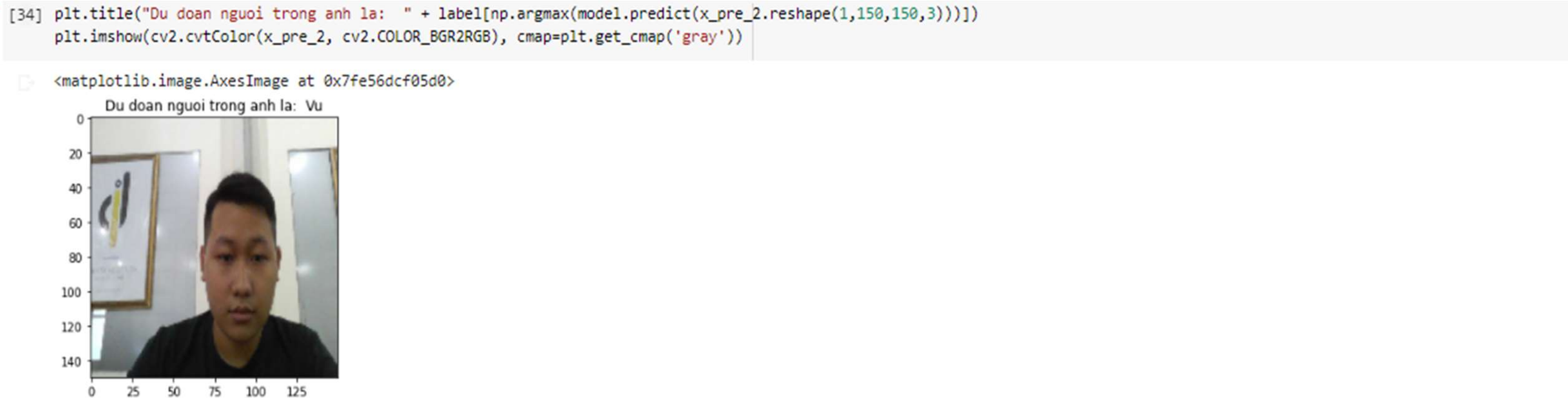
```
opt = Adam(lr = 0.001)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['acc'])
his = model.fit(x_train, y_train, epochs = 10, batch_size = 64, validation_split = 0.2)
```

```
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
4/4 [==============================] - 1s 196ms/step - loss: 9.5859e-07 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 2/10
4/4 [==============================] - 1s 132ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 3/10
4/4 [==============================] - 1s 135ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 4/10
4/4 [==============================] - 1s 131ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 5/10
4/4 [==============================] - 1s 134ms/step - loss: 2.9802e-09 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 6/10
4/4 [==============================] - 1s 134ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 7/10
4/4 [==============================] - 1s 138ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 8/10
4/4 [==============================] - 1s 135ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 9/10
4/4 [==============================] - 1s 159ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
Epoch 10/10
4/4 [==============================] - 1s 137ms/step - loss: 0.0000e+00 - acc: 1.0000 - val_loss: 0.0000e+00 - val_acc: 1.0000
```

```
[33]
label = ['Duc', 'Vu', 'Phat']
plt.title("Du doan nguoi trong anh la:  " + 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 0x7fe56dd1f210>
```

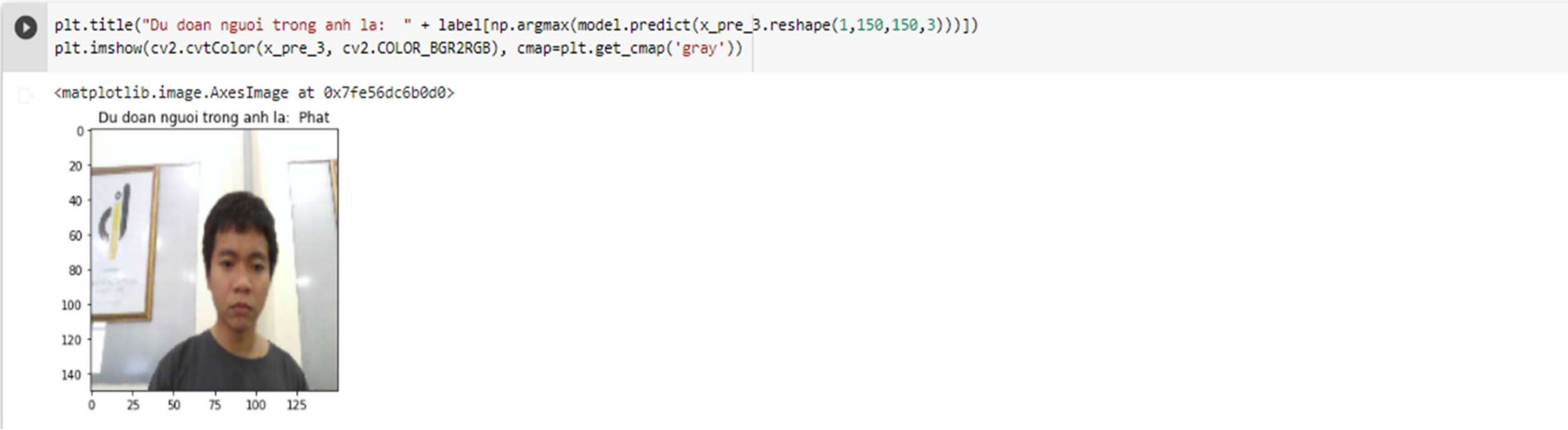
Du doan nguoi trong anh la: Duc



```
[34]
plt.title("Du doan nguoi trong anh la:  " + label[np.argmax(model.predict(x_pre_2.reshape(1,150,150,3)))])
plt.imshow(cv2.cvtColor(x_pre_2, cv2.COLOR_BGR2RGB), cmap=plt.get_cmap('gray'))
```

```
<matplotlib.image.AxesImage at 0x7fe56dcf05d0>
```

Du doan nguoi trong anh la: Vu



```
plt.title("Du doan nguoi trong anh la:  " + label[np.argmax(model.predict(x_pre_3.reshape(1,150,150,3)))])
plt.imshow(cv2.cvtColor(x_pre_3, cv2.COLOR_BGR2RGB), cmap=plt.get_cmap('gray'))
```

```
<matplotlib.image.AxesImage at 0x7fe56dc6b0d0>
```

Du doan nguoi trong anh la: Phat