**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**



**PROJECTS OF MACHINE DESIGN**

**Exercises for processing image data by using CNN**

**LE HOANG HOA 19146122**

**Teacher: NGUYEN TRUONG THINH**

**Ho Chi Minh City, 22th of May 2022**

**Bài 1: Cifa100**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.utils import to\_categorical

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import OneHotEncoder

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from keras.datasets import cifar100

(X\_train,Y\_train),(X\_test,Y\_test) = cifar100.load\_data()

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz>

169009152/169001437 [==============================] - 4s 0us/step

169017344/169001437 [==============================] - 4s 0us/step

X\_train.shape

(50000, 32, 32, 3)

labels = ['apple', 'aquarium\_fish', 'baby', 'bear', 'beaver', 'bed', 'bee', 'beetle', 'bicycle', 'bottle', 'bowl', 'boy', 'bridge', 'bus', 'butterfly', 'camel', 'can', 'castle', 'caterpillar', 'cattle', 'chair', 'chimpanzee', 'clock', 'cloud', 'cockroach', 'couch', 'crab', 'crocodile', 'cup', 'dinosaur', 'dolphin', 'elephant', 'flatfish', 'forest', 'fox', 'girl', 'hamster', 'house', 'kangaroo', 'keyboard', 'lamp', 'lawn\_mower', 'leopard', 'lion', 'lizard', 'lobster', 'man', 'maple\_tree', 'motorcycle', 'mountain', 'mouse', 'mushroom', 'oak\_tree', 'orange', 'orchid', 'otter', 'palm\_tree', 'pear', 'pickup\_truck', 'pine\_tree', 'plain', 'plate', 'poppy', 'porcupine', 'possum', 'rabbit', 'raccoon', 'ray', 'road', 'rocket', 'rose', 'sea', 'seal', 'shark', 'shrew', 'skunk', 'skyscraper', 'snail', 'snake', 'spider', 'squirrel', 'streetcar', 'sunflower', 'sweet\_pepper', 'table', 'tank', 'telephone', 'television', 'tiger', 'tractor', 'train', 'trout', 'tulip', 'turtle', 'wardrobe', 'whale', 'willow\_tree', 'wolf', 'woman', 'worm']

fig, axes = plt.subplots(ncols=10, nrows=5, figsize=(17, 8))

index = 0

for i in range(5):

    for j in range(10):

        axes[i,j].set\_title(labels[Y\_train[index][0]])

        axes[i,j].imshow(X\_train[index])

        axes[i,j].get\_xaxis().set\_visible(False)

        axes[i,j].get\_yaxis().set\_visible(False)

        index += 1

plt.show()

**Ảnh có chứa văn bản, cũ

Mô tả được tạo tự động**

X\_train = X\_train.astype('float32')

X\_test = X\_test.astype('float32')

X\_train = X\_train/255

X\_test = X\_test/255

Y\_train = to\_categorical (Y\_train,100)

Y\_test = to\_categorical (Y\_test,100)

X\_train.shape

(50000, 32, 32, 3)

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16,(3,3), padding='same', activation='relu', input\_shape=(32, 32, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(100, activation='softmax')

])

model.summary()

**Ảnh có chứa bàn

Mô tả được tạo tự động**

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(X\_train,Y\_train,epochs=100,batch\_size=64,verbose=1,validation\_data=(X\_test,Y\_test))

Ảnh có chứa văn bản

Mô tả được tạo tự động

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

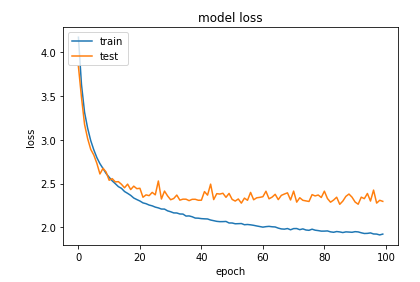
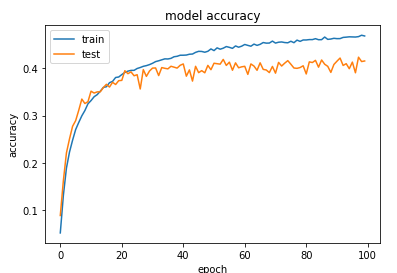
plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

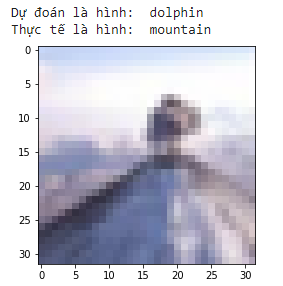


y\_pred = model5.predict(X\_test[[0]])

plt.imshow(X\_test[0])

print('Dự đoán là hình: ', labels[np.argmax(y\_pred)])

print('Thực tế là hình: ', labels[np.argmax(Y\_test[0])])

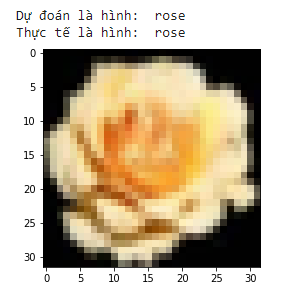


y\_pred = model5.predict(X\_test[[20]])

plt.imshow(X\_test[20])

print('Dự đoán là hình: ', labels[np.argmax(y\_pred)])

print('Thực tế là hình: ', labels[np.argmax(Y\_test[20])])



**Bài 2: Nhận diện khuôn mặt:**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.utils import to\_categorical

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from google.colab import drive

drive.mount('/content/drive')

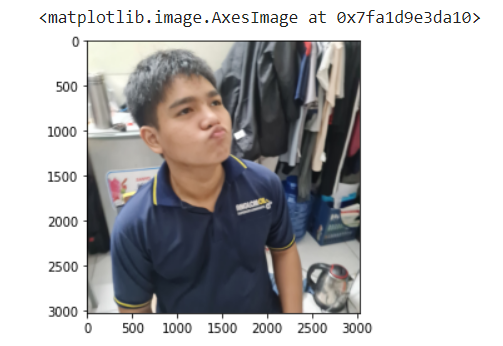
train\_path = '/content/drive/MyDrive/nhan\_dien\_khuon\_mat/train'

test\_path = '/content/drive/MyDrive/nhan\_dien\_khuon\_mat/test'

img = load\_img('/content/drive/MyDrive/nhan\_dien\_khuon\_mat/train/Hoa/Hoa (1).jpg')

rotated\_image = img.rotate(-90)

plt.imshow(rotated\_image)

****

cv2.imread('/content/drive/MyDrive/nhan\_dien\_khuon\_mat/train/Hoa/Hoa (1).jpg').shape

(3024, 3024, 3)

from tensorflow.keras.preprocessing.image import ImageDataGenerator

X\_train = ImageDataGenerator(rescale=1/255)

X\_test = ImageDataGenerator(rescale=1/255)

train\_dataset = X\_train.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/nhan\_dien\_khuon\_mat/train",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode="categorical",

    )

test\_dataset = X\_test.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/nhan\_dien\_khuon\_mat/test",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode= "categorical",

    )

Found 143 images belonging to 3 classes.

Found 53 images belonging to 3 classes.

train\_dataset.class\_indices

{'Hoa': 0, 'Huy': 1, 'Toan': 2}

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16, 3, padding='same', activation='relu', input\_shape=(150, 150, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(3, activation='softmax')

])

model.summary()

**Ảnh có chứa bàn

Mô tả được tạo tự động**

from tensorflow.keras.optimizers import SGD

#opt = SGD(lr = 0.01, momentum= 0.9)

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(train\_dataset,epochs=10,batch\_size=64,verbose=1,validation\_data= test\_dataset)

**Ảnh có chứa bàn

Mô tả được tạo tự động**

import plotly.graph\_objects as go

fig = go.Figure()

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['accuracy'],

                         mode='lines+markers',

                         name='Training accuracy'))

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['val\_accuracy'],

                         mode='lines+markers',

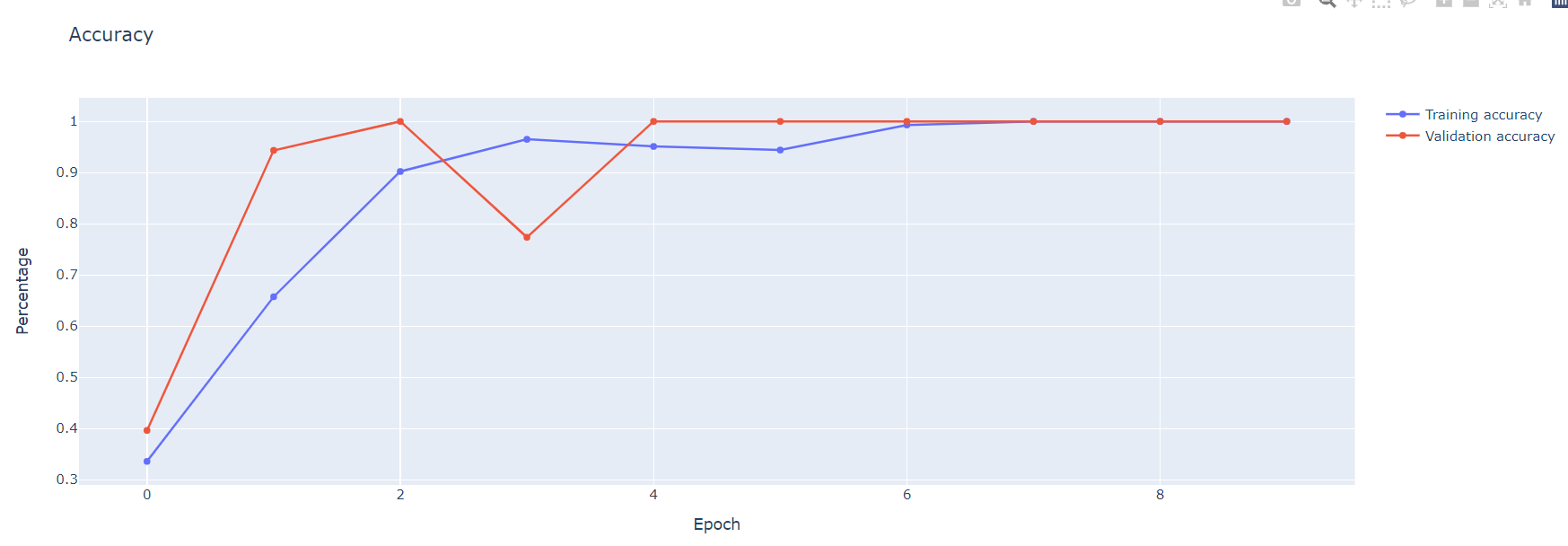
                         name='Validation accuracy'))

fig.update\_layout(title='Accuracy',

                  xaxis=dict(title='Epoch'),

                  yaxis=dict(title='Percentage'))

fig.show()

****

labels = {0: 'Hoa', 1: 'Huy', 2: 'Toan'}

from keras.preprocessing.image import load\_img, img\_to\_array

img = load\_img('/content/Hoa\_2.jpg', target\_size =(150,150))

rotated\_image = img.rotate(-90)

plt.imshow(rotated\_image)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

img.shape

val = model5.predict(img)

np.argmax(val,axis= 1)

print('Main ba: ', labels[np.argmax(val)])

Ảnh có chứa văn bản, người, ảnh chụp màn hình

Mô tả được tạo tự động

img = load\_img('/content/Huy\_19146060.jpg', target\_size =(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

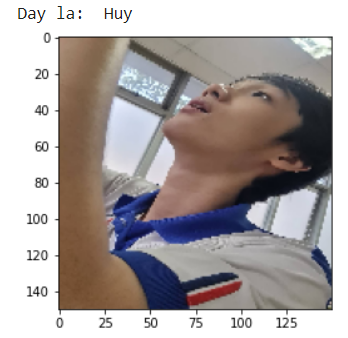
img = img/255

img.shape

val = model5.predict(img)

np.argmax(val,axis= 1)

print('Main ba: ', labels[np.argmax(val)])



img = load\_img('/content/drive/MyDrive/nhan\_dien\_khuon\_mat/train/Toan/Toan (30).JPG', target\_size =(150,150))

rotated\_image = img.rotate(-90)

plt.imshow(rotated\_image)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

img.shape

val = model5.predict(img)

np.argmax(val,axis= 1)

print('Day la ', labels[np.argmax(val)])

Ảnh có chứa văn bản, người, ảnh chụp màn hình

Mô tả được tạo tự động

**Bài 3: Nhận diện 10 món ăn VN**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.utils import to\_categorical

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from google.colab import drive

drive.mount('/content/drive')

train\_path = '/content/drive/MyDrive/VN\_food/train'

test\_path = '/content/drive/MyDrive/VN\_food/test'

img = load\_img('/content/drive/MyDrive/VN\_food/train/banhchung/banhchung (1).jpg')

rotated\_image = img.rotate(0)

plt.imshow(rotated\_image)



from tensorflow.keras.preprocessing.image import ImageDataGenerator

X\_train = ImageDataGenerator(rescale=1/255)

X\_test = ImageDataGenerator(rescale=1/255)

train\_dataset = X\_train.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/VN\_food/train",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode="categorical",

    )

test\_dataset = X\_test.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/VN\_food/test",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode= "categorical",

    )

Found 433 images belonging to 10 classes.

Found 159 images belonging to 10 classes.

train\_dataset.class\_indices

{'banhchung': 0,

'banhcuon': 1,

'banhmi': 2,

'banhxeo': 3,

'bunbo': 4,

'buncha': 5,

'bundaumamtom': 6,

'chagio': 7,

'comtam': 8,

'goicuon': 9}

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16,3, padding='same', activation='relu', input\_shape=(150, 150, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128, 3, padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(10, activation='softmax')

])

model.summary()

**Ảnh có chứa bàn

Mô tả được tạo tự động**

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(train\_dataset,epochs=30,batch\_size=64,verbose=1,validation\_data= test\_dataset)

Ảnh có chứa bàn

Mô tả được tạo tự động

import plotly.graph\_objects as go

fig = go.Figure()

model.save('vn\_food\_training.h5')

from keras.models import load\_model

model5 = load\_model('vn\_food\_training.h5')

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['accuracy'],

                         mode='lines+markers',

                         name='Training accuracy'))

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['val\_accuracy'],

                         mode='lines+markers',

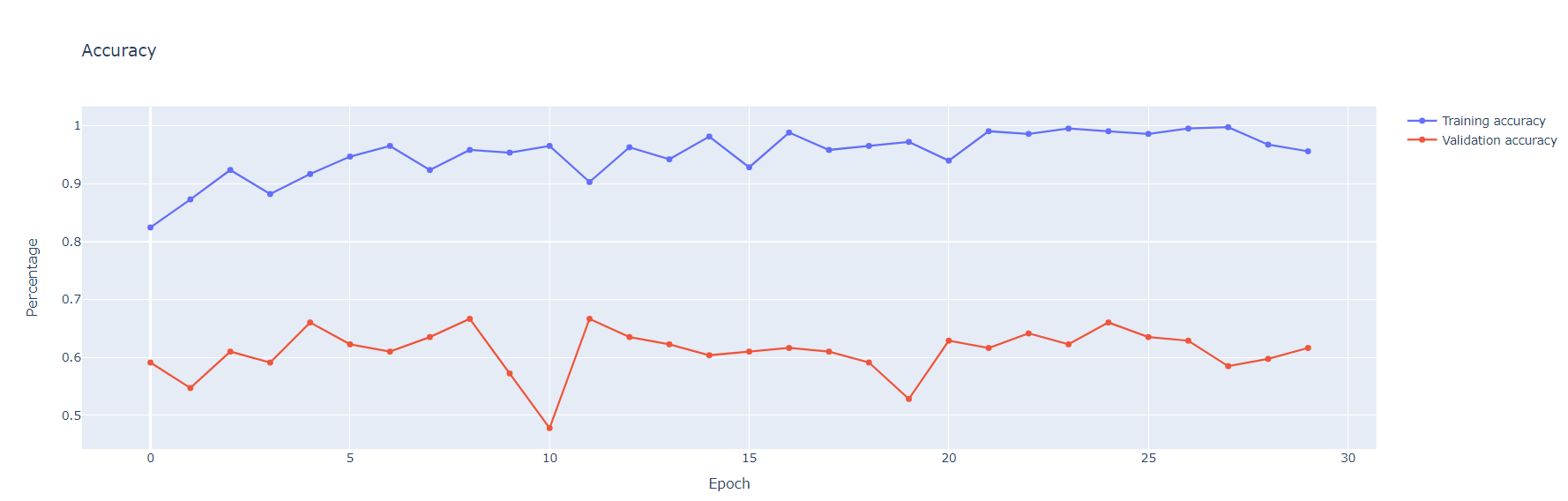
                         name='Validation accuracy'))

fig.update\_layout(title='Accuracy',

                  xaxis=dict(title='Epoch'),

                  yaxis=dict(title='Percentage'))

fig.show()



labels = {0: 'banhchung',

          1: 'banhcuon',

          2:'banhmi',

          3:'banhxeo',

          4:'bunbo',

          5:'buncha',

          6:'bundaumamtom',

          7:'chagio',

          8:'comtam',

          9:'goicuon'}

from keras.preprocessing.image import load\_img, img\_to\_array

img = load\_img('/content/banhmi (33).jpg', target\_size =(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

img.shape

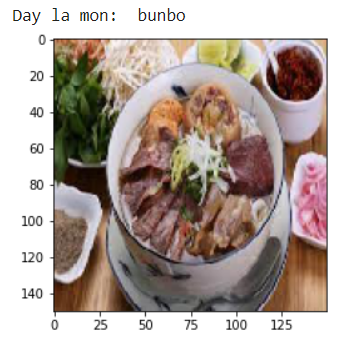
val = model5.predict(img)

np.argmax(val,axis= 1)

print('Day la mon: ', labels[np.argmax(val)])

**Ảnh có chứa văn bản, thực phẩm, thực phẩm ăn nhẹ, bánh sandwich

Mô tả được tạo tự độngẢnh có chứa văn bản, hạt, rau

Mô tả được tạo tự động**

**Bài 4: Nhận diện 10 loại trái cây nhiệt đới**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from google.colab import drive

drive.mount('/content/drive')

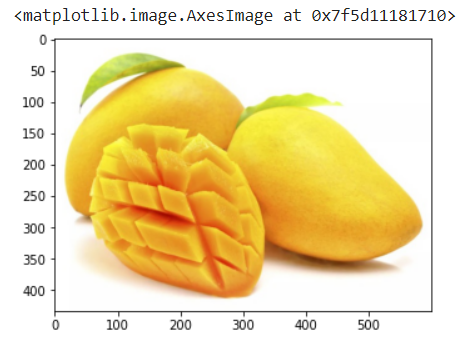
train\_path = '/content/drive/MyDrive/Tropical\_fruit/train'

test\_path = '/content/drive/MyDrive/Tropical\_fruit/test'

img = load\_img('/content/drive/MyDrive/Tropical\_fruit/train/mango/xoai (1).jpg')

rotated\_image = img.rotate(0)

plt.imshow(rotated\_image)



X\_train = ImageDataGenerator(rescale=1/255)

X\_test = ImageDataGenerator(rescale=1/255)

train\_dataset = X\_train.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Tropical\_fruit/train",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode="categorical",

    )

test\_dataset = X\_test.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Tropical\_fruit/test",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode= "categorical",

    )

Found 454 images belonging to 10 classes.

Found 193 images belonging to 10 classes.

train\_dataset.class\_indices

{'apple': 0,

'avovado': 1,

'banana': 2,

'dragonfruit': 3,

'kiwi': 4,

'mango': 5,

'orange': 6,

'passionfruit': 7,

'pinaple': 8,

'watermelon': 9}

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16,(3,3), padding='same', activation='relu', input\_shape=(150, 150, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(10, activation='softmax')

])

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(train\_dataset,epochs=30,batch\_size=64,verbose=1,validation\_data= test\_dataset)

Ảnh có chứa bàn

Mô tả được tạo tự động

model.save('10\_Tropical\_fruit\_training.h5')

from keras.models import load\_model

model5 = load\_model('10\_Tropical\_fruit\_training.h5')

import plotly.graph\_objects as go

fig = go.Figure()

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['accuracy'],

                         mode='lines+markers',

                         name='Training accuracy'))

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['val\_accuracy'],

                         mode='lines+markers',

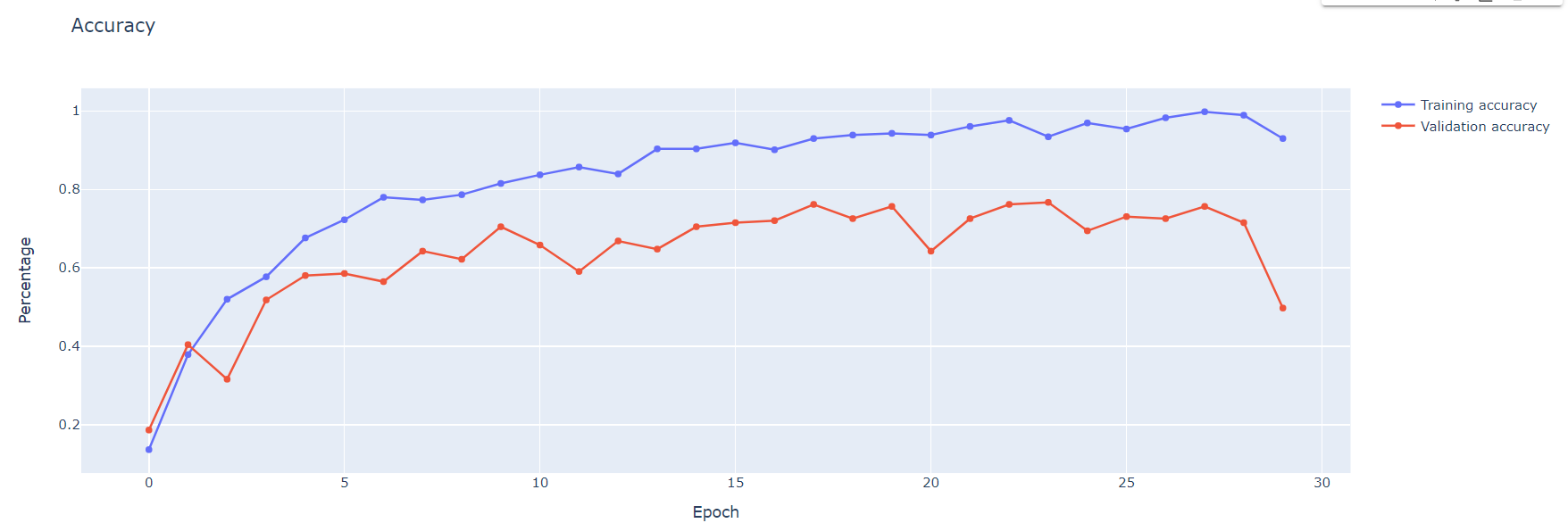
                         name='Validation accuracy'))

fig.update\_layout(title='Accuracy',

                  xaxis=dict(title='Epoch'),

                  yaxis=dict(title='Percentage'))

fig.show()



labels = {0: 'apple',

          1: 'avovado',

          2: 'banana',

          3: 'dragonfruit',

          4: 'kiwi',

          5: 'mango',

          6: 'orange',

          7: 'passionfruit',

          8: 'pinaple',

          9: 'watermelon'}

from keras.preprocessing.image import load\_img, img\_to\_array

img = load\_img('/content/drive/MyDrive/Tropical\_fruit/train/ ', target\_size =(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

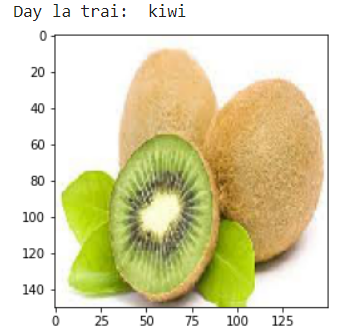
img.shape

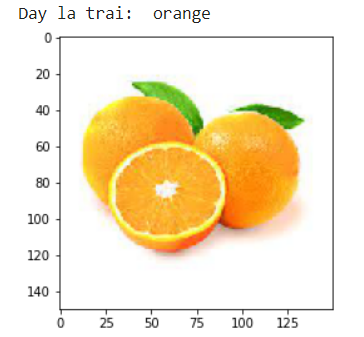
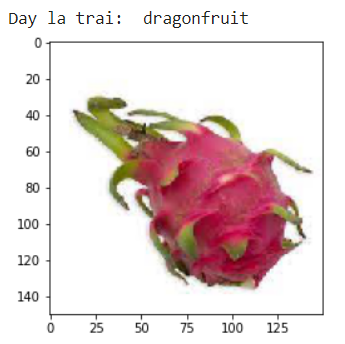
val = model5.predict(img)

np.argmax(val,axis= 1)

print('Day la trai: ', labels[np.argmax(val)])

**Ảnh có chứa văn bản, hoa quả, quả dứa

Mô tả được tạo tự động**

****

**Bài 5: Nhận diện tiền VN**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from google.colab import drive

drive.mount('/content/drive')

train\_path = '/content/drive/MyDrive/Nhan\_dien\_tien/train'

test\_path = '/content/drive/MyDrive/Nhan\_dien\_tien/test'

img = load\_img('/content/drive/MyDrive/Nhan\_dien\_tien/train/1k/1k (14).jpg')

rotated\_image = img.rotate(0)

plt.imshow(rotated\_image)



X\_train = ImageDataGenerator(rescale=1/255)

X\_test = ImageDataGenerator(rescale=1/255)

train\_dataset = X\_train.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Nhan\_dien\_tien/train",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode="categorical",

    )

test\_dataset = X\_test.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Nhan\_dien\_tien/test",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode= "categorical",

    )

Found 452 images belonging to 9 classes.

Found 238 images belonging to 9 classes.

train\_dataset.class\_indices

{'100k': 0,

'10k': 1,

'1k': 2,

'200k': 3,

'20k': 4,

'2k': 5,

'500k': 6,

'50k': 7,

'5k': 8}

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16,(3,3), padding='same', activation='relu', input\_shape=(150, 150, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(9, activation='softmax')

])

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(train\_dataset,epochs=30,batch\_size=64,verbose=1,validation\_data= test\_dataset)

Ảnh có chứa văn bản, bàn

Mô tả được tạo tự động

model.save('money\_training.h5')

from keras.models import load\_model

model5 = load\_model('money\_training.h5')

import plotly.graph\_objects as go

fig = go.Figure()

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['accuracy'],

                         mode='lines+markers',

                         name='Training accuracy'))

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['val\_accuracy'],

                         mode='lines+markers',

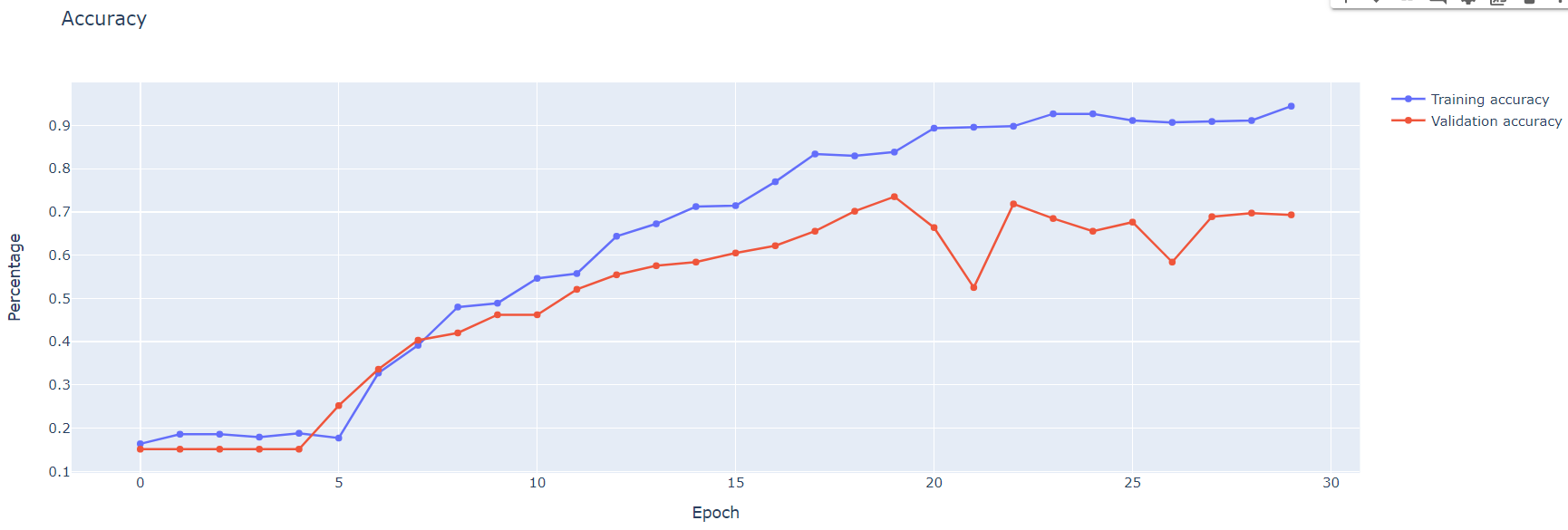
                         name='Validation accuracy'))

fig.update\_layout(title='Accuracy',

                  xaxis=dict(title='Epoch'),

                  yaxis=dict(title='Percentage'))

fig.show()



labels = {0: '100k',

          1: '10k',

          2: '1k',

          3: '200k',

          4: '20k',

          5: '2k',

          6: '500k',

          7: '50k',

          8: '5k'}

from keras.preprocessing.image import load\_img, img\_to\_array

img = load\_img('/content/drive/MyDrive/Nhan\_dien\_tien/test/.jpg', target\_size =(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

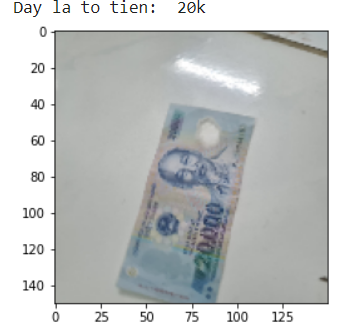
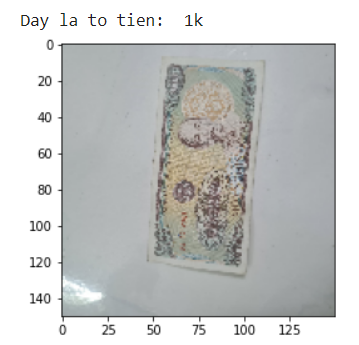
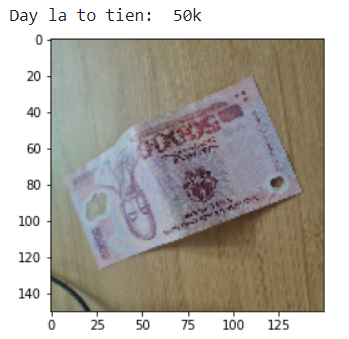
img = img/255

img.shape

val = model5.predict(img)

np.argmax(val,axis= 1)

print('Day la to tien: ', labels[np.argmax(val)])



**Bài 6: Nhận diện chỉ tay**

import cv2

import os

import numpy as np

import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense,Activation,Dropout

from keras.preprocessing.image import load\_img

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.layers.pooling import MaxPooling2D

from keras.backend import conv2d

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.layers.core.flatten import Flatten

from google.colab import drive

drive.mount('/content/drive')

train\_path = '/content/drive/MyDrive/Nhan\_dien\_van\_tay/train'

test\_path = '/content/drive/MyDrive/Nhan\_dien\_van\_tay/test'

img = load\_img('/content/drive/MyDrive/Nhan\_dan\_chi\_tay/train/giauco/3 (10).jpg')

rotated\_image = img.rotate(0)

plt.imshow(rotated\_image)

Ảnh có chứa văn bản

Mô tả được tạo tự động

X\_train = ImageDataGenerator(rescale=1/255)

X\_test = ImageDataGenerator(rescale=1/255)

train\_dataset = X\_train.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Nhan\_dan\_chi\_tay/train",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode="categorical",

    )

test\_dataset = X\_test.flow\_from\_directory(

    directory=r"/content/drive/MyDrive/Nhan\_dan\_chi\_tay/test",

    target\_size=(150, 150),

    batch\_size=3,

    class\_mode= "categorical",

    )

Found 100 images belonging to 4 classes.

Found 44 images belonging to 4 classes.

train\_dataset.class\_indices

{'co\_tinh\_cam\_va\_cong\_viec\_suat\_sac': 0,

'duong\_tinh\_duyen\_trac\_tro': 1,

'giauco': 2,

'tainang\_thongminh': 3}

from keras.layers.core.flatten import Flatten

model = Sequential([

    Conv2D(16,(3,3), padding='same', activation='relu', input\_shape=(150, 150, 3)),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(32,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Conv2D(64,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Conv2D(128,(3,3), padding='same', activation='relu'),

    MaxPooling2D(),

    Dropout(0.2),

    Flatten(),

    Dense(512, activation='relu'),

    Dense(4, activation='softmax')

])

model.compile(optimizer='adam',loss ='categorical\_crossentropy',metrics =['accuracy'])

history=model.fit(train\_dataset,epochs=30,batch\_size=64,verbose=1,validation\_data= test\_dataset)

Ảnh có chứa văn bản, tài liệu, ảnh chụp màn hình

Mô tả được tạo tự động

model.save('palm\_reading.h5')

from keras.models import load\_model

model5 = load\_model('palm\_reading.h5')

import plotly.graph\_objects as go

fig = go.Figure()

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['accuracy'],

                         mode='lines+markers',

                         name='Training accuracy'))

fig.add\_trace(go.Scatter(x=history.epoch,

                         y=history.history['val\_accuracy'],

                         mode='lines+markers',

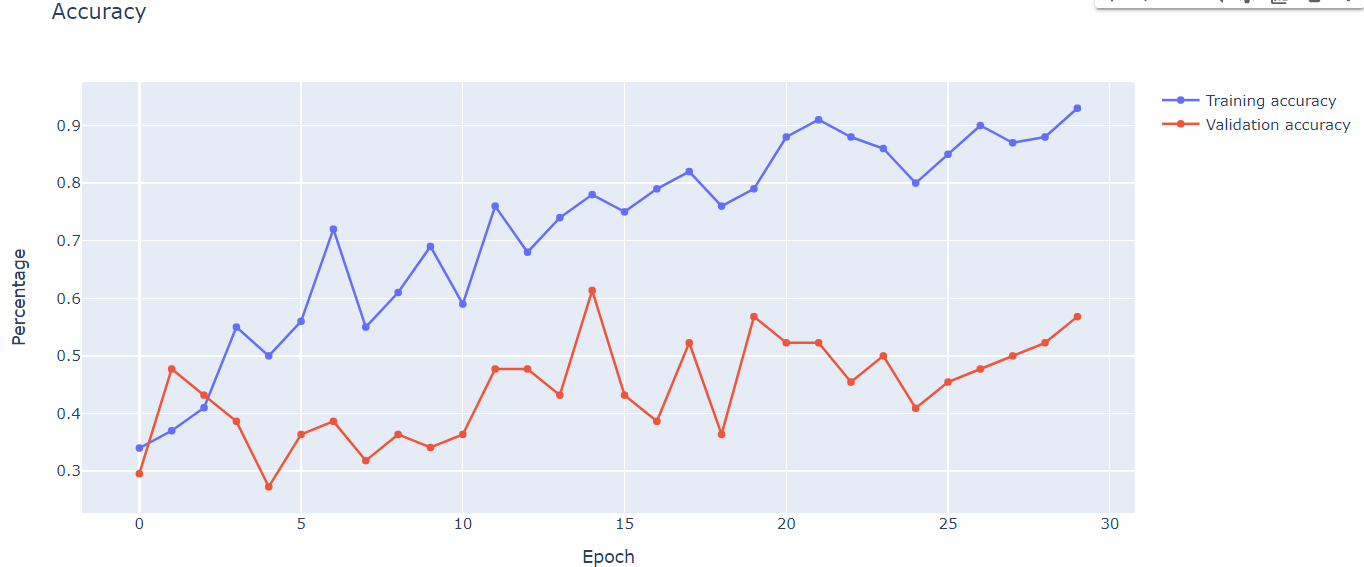
                         name='Validation accuracy'))

fig.update\_layout(title='Accuracy',

                  xaxis=dict(title='Epoch'),

                  yaxis=dict(title='Percentage'))

fig.show()



labels = {0: 'co\_tinh\_cam\_va\_cong\_viec\_suat\_sac',

          1: 'duong\_tinh\_duyen\_trac\_tro',

          2: 'giauco',

          3: 'tainang\_thongminh'}

from keras.preprocessing.image import load\_img, img\_to\_array

img = load\_img('/content/tay\_Huy.jpg', target\_size =(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

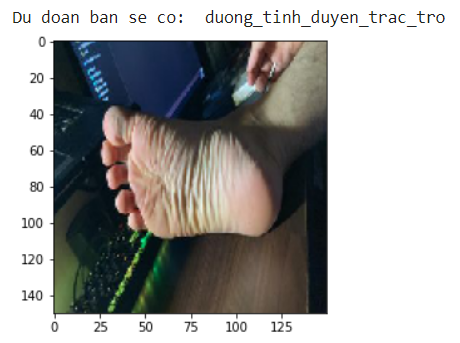
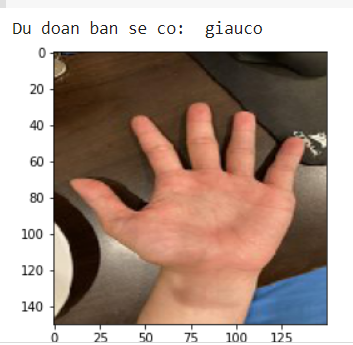
img.shape

val = model5.predict(img)

np.argmax(val,axis= 1)

print('Du doan ban se co: ', labels[np.argmax(val)])

Huy hand predict:



Hoa hand predict:

