

Tuan2\_BatoanDulieuMNIST.ipynb

File Edit View Insert Runtime Tools Help Saving failed since 4:52 PM

+ Code + Text

+ Code + Text

RAM Disk

- 21022008\_NguyenHuuTho

CHẮC CÓ KIỂM TRA

Nét chữ Kiểu chữ Góc chụp

Train : Huấn luyện Test : Kiểm thử Location

MNIST tập dữ liệu: Áo quần, chữ số => Tùy nhu cầu

[1] from google.colab import drive  
drive.mount('/content/gdrive')  
  
Mounted at /content/gdrive

[2] %cd /content/gdrive/MyDrive/AI\_1333/AI\_Tuan2/Tuan2\_BatoanDulieuMNIST  
  
/content/gdrive/MyDrive/AI\_1333/AI\_Tuan2/Tuan2\_BatoanDulieuMNIST

[3] !pwd  
  
/content/gdrive/MyDrive/AI\_1333/AI\_Tuan2/Tuan2\_BatoanDulieuMNIST

[ ] #pip install opencv-python keras tensorflow

[16] !ls

Automatic saving failed. This file was updated remotely or in another tab. Show diff Automatic saving failed. This file was updated remotely or in another tab. Show diff

Xây dựng model nhận diện, dữ liệu được xây dựng training model là mnist Đầu tiên load dataset và load các thư viện cần dùng:

```
[4] import cv2  
import numpy as np  
from skimage.feature import hog # Nền tảng AI, máy học: đưa sao học sau ra vậy, chưa học chưa ra được. Thuật toán: HOG, SVC (Machine learning), trích xuất đặc trưng ra một vector  
from sklearn.svm import LinearSVC  
from sklearn.metrics import accuracy_score # Độ chính xác, độ tin cậy  
from keras.datasets import mnist # Deep learning, huấn luyện mô hình, train module, thư viện mnist  
from google.colab.patches import cv2_imshow
```

Tiếp theo ta sẽ tính hog. Ta dùng orientations=9,pixels\_per\_cell=(14,14),cells\_per\_block=(1,1).

```
[5] (X_train,y_train),(X_test,y_test) = mnist.load_data()  
  
X_train_feature = []  
for i in range(len(X_train)):  
    feature = hog(X_train[i],orientations=9,pixels_per_cell=(14,14),cells_per_block=(1,1),block_norm="L2")  
    X_train_feature.append(feature)  
X_train_feature = np.array(X_train_feature, dtype = np.float32)  
  
#cho x_test  
X_test_feature = []  
for i in range(len(X_test)):  
    feature = hog(X_test[i],orientations=9,pixels_per_cell=(14,14),cells_per_block=(1,1),block_norm="L2")  
    X_test_feature.append(feature)  
X_test_feature = np.array(X_test_feature, dtype = np.float32)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff Automatic saving failed. This file was updated remotely or in another tab. Show diff

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>  
11490434/11490434 [-----] - 1s 0us/step

▼ Tiếp theo ta build model vào predict

- predict Tính toán độ chính xác mô hình

```
[6] model = LinearSVC(C=10)  
model.fit(X_train_feature,y_train)  
y_pre = model.predict(X_test_feature)  
DoChinhxacMohinh = accuracy_score(y_test,y_pre)  
print(DoChinhxacMohinh) # In ra độ chính xác mô hình (độ chính xác của mô hình vừa huấn luyện)
```

0.8988  
/usr/local/lib/python3.8/dist-packages/sklearn/svm/\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.  
warnings.warn(  
  
[27] DoChinhxacMohinh.shape  
  
( )

Predict trên ảnh

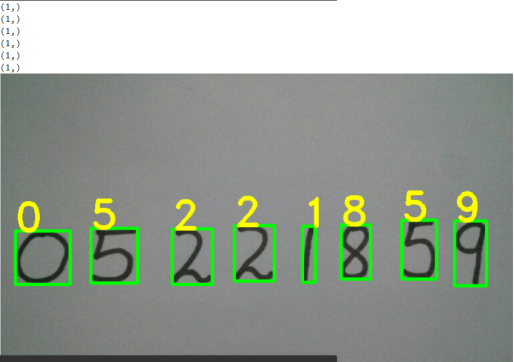
Automatic saving failed. This file was updated remotely or in another tab. Show diff Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
im_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
im_blur = cv2.GaussianBlur (im_gray, (5,5),0)  
im, thre= cv2.threshold (im_blur,90,255,cv2.THRESH_BINARY_INV)  
contours, hierarchy = cv2.findContours(thre, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) # Di tìm ra những nét của chữ số, canny, edge: phát hiện cạnh biên  
rects = [cv2.boundingRect(cnt) for cnt in contours] #
```

Sau đó predict digit của mỗi box.

```
[30] # Basic  
for i in contours:  
    (x,y,w,h) = cv2.boundingRect(i)  
    cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),3)  
    roi = thre[y:y+h,x:x+w]  
    roi = np.pad(roi,(20,20),'constant',constant_values=(0,0))  
    roi = cv2.resize(roi, (28, 28), interpolation=cv2.INTER_AREA)  
    roi = cv2.dilate(roi, (3, 3))  
    # Calculate the HOG features - Tính toán độ chính xác mô hình  
    roi_hog_fd = hog(roi, orientations=9, pixels_per_cell=(14, 14), cells_per_block=(1, 1), block_norm="L2")  
    nbr = model.predict(np.array([roi_hog_fd], np.float32)) # np.array([roi_hog_fd], np.float32) -> Hình đang xử lý  
    cv2.putText(image, str(int(nbr[0])), (x, y), cv2.FONT_HERSHEY_DUPLEX, 2, (0, 255, 255), 3) # putText : Đền chữ lên tâm hình, putText(ảnh vẽ đề lên, str(int(nbr[0])) - đề số 9 lên ảnh, (x, y) - tọa độ vẽ lên,  
  
cv2.imshow('image')  
cv2.imwrite("image_pand.jpg",image) # để lưu ảnh  
# cv2.waitKey()  
# cv2.destroyAllWindows()
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff



automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

True

▾ Bài tập nâng cao

- Anh chỉ hãy in ra độ chính xác của mô hình vừa huấn luyện? - Xong
- Anh chỉ chỉ hiển thị những chữ số có độ chính xác (accuracy) trên 98%?
  - Tìm ra độ chính xác từng số
  - bỏ vào hàm if

```
# Đánh giá mô hình
# test_loss, test_acc = model.evaluate(X_test, y_test)

# Anh chỉ chỉ hiển thị những chữ số có độ chính xác trên 98%?
for i in contours:
    (x,y,w,h) = cv2.boundingRect(1)
    cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),3)
    roi = three[j,yyh,x,x+w]
    roi = np.pad(roi,(20,20),'constant',constant_values=(0,0))

# Calculate the HOG features - Tính toán độ chính xác mô hình
roi_hog_fd = hog(roi, orientations=9, pixels_per_cell=(14, 14), cells_per_block=(1, 1), block_norm="L2")
nbr = model.predict(np.array([roi_hog_fd], np.float32)) # np.array([roi_hog_fd], np.float32) -> Hình đang xử lý
cv2.putText(image, str(int(nbr[0])), (x, y), cv2.FONT_HERSHEY_DUPLEX, 2, (0, 255, 255), 3) # putText : Đền chữ lên tâm hình, putText(ảnh vẽ đè lên, str(int(nbr[0])) - đè số 9 lên ảnh, (x, y) - tọa độ vẽ lên,

cv2.imshow(image)
cv2.imwrite("image_pand.jpg",image) # để lưu ảnh

[17] wget https://thorphan.github.io/assets/images/digit.jpg
```

automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

1s completed at 5:17 PM

x