

Fejlesztői dokumentáció

Alphabet_recogn.py

Feladat

Kézzel írt karakter felismerése homogén háttér előtt. Létező modell alapján.

Függőségek: OpenCV, Keras, NumPy, sys, getopt

Környezet

Python 3.9.7 futtatására alkalmas operációs rendszer.
Fejlesztői környezet VS Code.

Forráskód

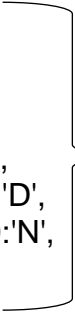
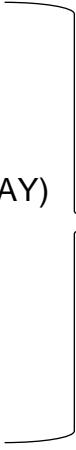
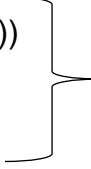
A paraméterben kapott elérési utat megjegyeztetjük az inputfile változóba. Ezek után betöltjük az előre elkészített modellünket, létrehozunk hozzá a szótárat, mivel amikor megjósoljuk a képen található betű/szám osztályát, azt ez alapján tudjuk behatárolni, hogy mit jelenítsen meg. Ezek után OpenCV segítségével megnyitjuk az inputfile-ba mentett elérési útról a képet, ezt egy imgcopy nevű változóban előfeldolgozzuk a modellünk számára, majd ezen a képen végrehajtjuk a jóslást. Az eredetileg beolvasott képünket átméretezzük és OpenCV segítségével ráírjuk a megjósolt eredményünket, majd megjelenítjük.

```
import cv2
import numpy as np
from keras.models import load_model
import sys,getopt
```

A szükséges könyvtárak
importálása

```
argv=sys.argv[1:]
inputfile=""
try:
    opts, args = getopt.getopt(argv,"hi:",["infile="])
except getopt.GetoptError:
    print('Alphabet_recogn.py -i <inputfile>')
    sys.exit(2)
for opt, arg in opts:
    if opt == '-h':
        print('Alphabet_recogn.py -i <inputfile>')
        sys.exit()
    elif opt in ("-i", "--infile"):
        inputfile = arg
```

Command
Prompt/Powershell-ről
indított program
argumentum ellenőrzése

<pre> model = load_model('modelname.h5') model.summary() alph_dict = {0:'0',1:'1',2:'2',3:'3',4:'4',5:'5',6:'6',7:'7',8:'8',9:'9', 10:'a',11:'b',12:'c',13:'d',14:'e',15:'f',16:'g',17:'h',18:'i',19:'j', 20:'k',21:'l',22:'m',23:'n',24:'o',25:'p',26:'q',27:'r',28:'s',29:'t', 30:'u',31:'v',32:'w',33:'x',34:'y',35:'z',36:'A',37:'B',38:'C',39:'D', 40:'E',41:'F',42:'G',43:'H',44:'I',45:'J',46:'K',47:'L',48:'M',49:'N', 50:'O',51:'P',52:'Q',53:'R',54:'S',55:'T', 56:'U',57:'V',58:'W',59:'X',60:'Y',61:'Z'} </pre>		<p>Modell betöltése, és a hozzá tartozó szótár létrehozása</p>
<pre> img= cv2.imread(inputfile) img_copy =img.copy() img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) img = cv2.resize(img, (400,400)) img_copy = cv2.GaussianBlur(img_copy, (3,3), 0) img_gray = cv2.cvtColor(img_copy, cv2.COLOR_BGR2GRAY) _, img_thresh = cv2.threshold(img_gray, 100, 255, cv2.THRESH_BINARY_INV) img_final = cv2.resize(img_thresh, (24,24)) img_final =np.reshape(img_final, (1,24,24,1)) img_pred = alph_dict[np.argmax(model.predict(img_final))] tipp=np.argmax(model.predict(img_final)) print(tipp) print('Number of arguments: {}'.format(len(sys.argv))) print('Argument(s) passed: {}'.format(str(sys.argv))) cv2.putText(img, "Tipp: " + img_pred, (20,370), cv2.FONT_HERSHEY_TRIPLEX, 1.3, color = (255,0,0)) cv2.imshow('Handwritten character recognition', img) cv2.waitKey(0) cv2.destroyAllWindows() </pre>		<p>Kép betöltése argumentumból, előfeldolgozás és tipp meghatározása</p>
<pre> cv2.putText(img, "Tipp: " + img_pred, (20,370), cv2.FONT_HERSHEY_TRIPLEX, 1.3, color = (255,0,0)) cv2.imshow('Handwritten character recognition', img) cv2.waitKey(0) cv2.destroyAllWindows() </pre>		<p>A megjósolt tippünk ráhelyezése a képre, ablak bezárása gombnyomásra</p>

Fejlesztési lehetőségek

- Az argumentumoknál több fájl fogadása.
- Több betűről készült kép fogadása, azok helyzetének detektálása és ezután a felismerésük.
- Modell megadásának lehetősége argumentumban.

seq_mod.py

Feladat

Modell létrehozása kézzel írt betűk és számok felismerésére

Függőségek: TensorFlow, Keras

Forráskód

Beolvassuk az előre meghatározott helyről az adatokat, melyeket két részre bontunk, egyikkel betanítjuk a modellt a másikkal, pedig validáljuk a pontosságát. Ezután létrehozuk a modellt. Ehhez a Keras Sequential modellt használtam. Ez egy egyszerű modell típus egy bemenettel és egy kimenettel, rétegek egymásutániságával. Ezek után összeállítjuk a modellt és betanítjuk a meglevő adatainkkal, majd elmentjük azt későbbi felhasználásra, emellett kiírjuk a képernyőre a modell pontosságát a veszteségek mértékét, valamint a modell összegzését.

```
import keras
from keras import layers
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
```

A szükséges könyvtárak
importálása

```
IMG_WIDTH=24
IMG_HEIGHT=24
DATA_DIR="path to the dataset directory"
NUM_CLASSES = 62
```

Főbb konstansok
kiemelése

```
trainset=tf.keras.utils.image_dataset_from_directory(
    DATA_DIR, labels="inferred",
    label_mode="categorical", class_names=None,
    color_mode="grayscale", batch_size=64,
    image_size=(IMG_WIDTH, IMG_HEIGHT),
    shuffle=True, seed=3,
    validation_split=0.2, subset="training",
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False
)
valset=tf.keras.utils.image_dataset_from_directory(
    DATA_DIR, labels="inferred", label_mode="categorical",
    class_names=None, color_mode="grayscale",
    batch_size=64,
    image_size=(IMG_WIDTH, IMG_HEIGHT),
    shuffle=True, seed=3,
    validation_split=0.2, subset="validation",
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False
)
```

Adatok beolvasása és
felbontása a rendezett
könyvtárstruktúrából

