

image_transforms

December 30, 2019

1 Kép transzformációk tanítás támogatására

Mivel nem mindig könnyen megvalósítható a megfelelő mennyiségű tanító állomány (kép) beszerzése, ezért érdemes olyan képtranszformációkban gondolkodni, amely lehetővé teszi további tanító képek létrehozását. Ezt szokás adathalmaz kibővítésnek is nevezni (data augmentation). Fontos megjegyezni, hogy ezen technikával akár gyorsítani is lehet a konvergencia sebességét, de figyelni kell a kibővítés mértékére.

Két egyszerű transzformációt definiálunk: - eltolás - forgatás

Eltolás esetén meg lehet adni a vízszintes és a függőleges eltolás mértékét, míg forgatás esetén a forgatási szöveget. Ezen transzformációk lineáris kombinációjával és megfelelő paraméterezésével nagyon egyszerűen lehet nagy mennyiségű kiterjesztő állományt generálni.

Ezen felül érdemes definiálni pár normalizáló függvényt. Ezek legegyszerűbb fajtái: - adott arányú skálázás - adott maximális méretre skálázás

```
[1]: ## @package imagertransform
# A modul alap képtranszformációkat tartalmaz.
#
# Két egyszerű transzformációt definiálunk:
# - eltolás
# - forgatás.

# A szükséges importok
# A transzformáció az opencv könyvtár megfelelő metódusaira épít
import cv2
import os
import glob
import numpy as np

## Parametrizálható eltolás
#
# A paraméterként kapott képet vízszintesen és függőlegesen eltolja az adott
# paraméter szerint.
def shift_image(img, horizontal_shift, vertical_shift):
    rows, cols, colors = img.shape
    M = np.float32([[1, 0, horizontal_shift], [0, 1, vertical_shift]])
    dst = cv2.warpAffine(img, M, (cols, rows))
```

```

    return dst

## Parametrizálható elforgatás
#
# A paraméterként kapott képett elforgatja az adott paraméter szerint.
def rotate_image(img, angle):
    rows,cols,colors = img.shape

    # cols-1 and rows-1 are the coordinate limits.
    M = cv2.getRotationMatrix2D(((cols-1)/2.0,(rows-1)/2.0),angle,1)
    dst = cv2.warpAffine(img,M,(cols,rows))

    return dst

## Parametrizálható átméretezés arányos skálázással
#
# A paraméterként kapott képett arányosan átskálázza az adott paraméter_
↪szerint.
def resize_image_scale(img, scale):
    rows, cols, colors = img.shape
    dst = cv2.resize(img,(scale*rows, scale*cols), interpolation = cv2.
    ↪INTER_AREA)

    return dst

## Parametrizálható aránytartó átméretezés maximális méret megadásával
#
# A paraméterként kapott képett aránytartóan átskálázza a megadott maximális_
↪méretre.
def resize_image_maxdim(img, maxdim):
    rows, cols, colors = img.shape
    aspect_ratio = cols/rows
    if (rows < cols):
        dims = (maxdim, round(maxdim / aspect_ratio))
    else:
        dims = (round(maxdim * aspect_ratio), maxdim)
    dst = cv2.resize(img, dims, interpolation = cv2.INTER_AREA)

    return dst

```

```

[3]: # A képeket tartalmazó könyvtár megadása
img_dir = "face-recognition-opencv/dataset/012.Tamas"
data_path = os.path.join(img_dir, '*')
files = glob.glob(data_path)
data = []
for f1 in files:

```

```

print("Filepath:\t{}".format(f1))
img = cv2.imread(f1)
print(f1.split('/'))
# A könyvtár komponens mindig az utolsó előtti komponens
directory = '/'.join(f1.split('/')[:-1])
print("Directory: ", directory)
# A filenév a struktúra függvénye, itt a könnyebség kedvéért be van
→drótozva [3]
file_name = f1.split('/')[3].split('.')[0]
print("Filename: ", file_name)
extension = f1.split('/')[3].split('.')[1]
print("Extension: ", extension)
# Ha újra ráfuttatjuk egy már feldolgozott könyvtárra, akkor lánc
→transzformációt kaphatunk
# ezért a saját hozzáfűzött nevet keressük és ha megtaláljuk, nem hajtjuk
→vége a transzformációt.
if (file_name.find("H_Shifted_") != -1):
    break
if (file_name.find("V_Shifted_") != -1):
    break
if (file_name.find("Rotated_") != -1):
    break

# Mivel adott esetben egy könyvtár sok képet is tartalmazhat, a biztoonság
→kevéért megszakíthatóvá
# tesszük a feldolgozást (q lenyomásával ki lehet lépni)
if cv2.waitKey(0) & 0xFF == ord('q'):
    break

# A képet leskálázzuk, hogy a legnagyobb mérete maximum 500 pixel legyen
new_img = resize_image_maxdim(img, 500)

# A képből készítünk vízszintesen és függőlegesen adott léptékkal eltol
→változatokat
# Az eltolást bedrótoltuk, 20-100-as tartományban 20 pixelenként léptetünk
for shift in range(20,100,20):
    #cv2.imshow('new_img', img)
    #if cv2.waitKey(0) & 0xFF == ord('q'):
    #    break
    h_shifted = shift_image(new_img, shift, 0)
    v_shifted = shift_image(new_img, 0, shift)
    #cv2.imshow('img', new_img)
    #if cv2.waitKey(0) & 0xFF == ord('q'):
    #    break
    cv2.imwrite(directory + "/" + "H_Shifted_" + str(shift) + "_" +
→file_name + "." + extension, h_shifted)

```

```

        cv2.imwrite(directory + "/" + "V_Shifted_" + str(shift) + "_" +
↪file_name + "." + extension, v_shifted)

# A képből készítünk adott szöggel elforgatott változatokat
# Az elforgatást bedrótoztuk, 45-360-as tartományban 45 fokként forgatunk
↪
    for rotate in range(45,360,45):
        #cv2.imshow('new_img', img)
        #if cv2.waitKey(0) & 0xFF == ord('q'):
        #    break
        rotated = rotate_image(new_img, rotate)
        cv2.imwrite(directory + "/" + "Rotated_" + str(rotate) + "_" +
↪file_name + "." + extension, rotated)

        #cv2.destroyWindow('img')
        #print(directory + "/" + "Shifted_" + file_name + "." + extension)
        #cv2.imwrite(directory + "/" + "Shifted_" + file_name + "." + extension,
↪img)

```

```

Filepath:      face-recognition-opencv/dataset/012.Tamas/20190603_162158.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190603_162158.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas
Filename:     20190603_162158
Extension:    jpg
Filepath:      face-recognition-opencv/dataset/012.Tamas/20190813_121446.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190813_121446.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas
Filename:     20190813_121446
Extension:    jpg
Filepath:      face-recognition-opencv/dataset/012.Tamas/20190712_211054.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190712_211054.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas
Filename:     20190712_211054
Extension:    jpg
Filepath:      face-recognition-opencv/dataset/012.Tamas/20190617_155906.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190617_155906.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas
Filename:     20190617_155906
Extension:    jpg
Filepath:      face-recognition-opencv/dataset/012.Tamas/20190526_170758.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190526_170758.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas
Filename:     20190526_170758
Extension:    jpg
Filepath:      face-recognition-opencv/dataset/012.Tamas/20190422_122055.jpg
['face-recognition-opencv', 'dataset', '012.Tamas', '20190422_122055.jpg']
Directory:    face-recognition-opencv/dataset/012.Tamas

```

Filename: 20190422_122055

Extension: jpg

[]: