

1 Dokumentace

1.0.1 Kódy příkazů

- *0* : `obrazek = cv2.resize(obrazek, None, fx=par0, fy=par0, interpolation=interpolace)`
fx a *fy* násobí velikost původního obrázku a získávají tak velikost výsledného obrázku. Přísluší jednotlivým osám. Jedná se o desetinná čísla. [1]
 - *01*: `interpolace = cv2.INTER_AREA`
Vhodné pro zmenšování obrázku.
 - *02*: `interpolace = cv2.INTER_CUBIC`
Nejlepší pro zvětšení obrázku, ale pomalejší.
 - *03*: `interpolace = cv2.INTER_LINEAR`
Pro zvětšení obrázku. Rychlejší, ale ne nejlepší.
- *11*: `obrazek = cv2.cvtColor(obrazek, cv2.COLOR_BGR2GRAY)`
Převeď obrázek do grayscale.
- *21*: `obrazek = cv2.fastNlMeansDenoising(obrazek, par1, par2, par3)`
Vyžaduje grayscale obrázek. Desetinné číslo *par1* určuje sílu filtru. Celá lichá čísla *par2* a *par3* jsou parametry určující velikost okna. Výpočetní náročnost se zvětšuje pro vyšší hodnoty.
- *3*: Proveď rozmazání (angl. blurring) obrázku. [2]
 - *31*: `obrazek = cv2.blur(obrazek, (par4, par4))`
Použij aritmetický průměr pole, kde výška a šířka je rovna *par4*, což je celé číslo.
 - *32*: `obrazek = cv2.bilateralFilter(obrazek, par5, par6, par7)`
Je efektivní pro odstranění šumu a zároveň zachovává ostré hrany. *par5* je celé číslo

určující průměr okolí použitého ve filtraci. *par6* a *par7* jsou desetinné hodnoty odpovídající SigmaSpace a SigmaColor. Čím jsou větší, tím větší je síla efektu (viz dokumentace [3])

- 33: `obrazek = cv2.GaussianBlur(obrazek, (par8,par8), par9)`
par8 určuje velikost oblasti, se kterou se počítá, musí být kladné liché číslo. *par9* určuje standartní deviaci, pokud je nulový, tak se vypočítá z velikosti oblasti
- 34: `obrazek = cv2.medianBlur(obrazek, par10)`
Střed pole o výšce a šířce *par10* je určen mediánem hodnot v tomto poli. *par10* by měl být kladná liché číslo.
- 4 Vstupem by měl být grayscale obrázek. Výsledek je binarizovaný obrázek [4]
 - 41: `obrazek = cv2.adaptiveThreshold(obrazek, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, par11, par12)`
Celé číslo *par11* určuje velikost bloku. Spočítá se průměr a odečte se od něj konstanta *par12*.
 - 42: `obrazek = cv2.adaptiveThreshold(obrazek, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, par13, par14)`
Celé číslo *par13* určuje velikost bloku. Spočítá se průměr a odečte se od něj konstanta *par14*.
 - 43: `ret, obrazek = cv2.threshold(obrazek, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)`
Jedná se o binarizaci s globálním prahem, který je určen pomocí Otsuovi metody. Nevyžaduje žádný parametr. Funkce vrací 2 hodnoty a až druhá je upravený obrázek.

1.0.2 Formát souborů

Databáze obrázků

Každému obrázku připadá jeden řádek, který má mezerou oddělený název obrázku a text, který se na něm nachází.

Příkazy pro OpenCV

Název zdrojové složky, databáze obrázků, soubor s parametry a poté jednotlivé číselné kódy pro příkazy, jsou na jedné řádce odděleny mezerami. Chceme-li provést další operaci jednoduše ji stejným způsobem zapíšeme na následující řádku.

Soubor s parametry

Jednotlivé parametry jsou odděleny mezerou a všechny se nachází na jednom řádku.

Příkazy pro Tesseract

Na každém řádku je název složky, ze které má Tesseract brát obrázky, a mezerou oddělený název souboru s databází obrázků.

Výpočet úspěšnosti

Stejný formát předpokládá i algoritmus pro vyhodnocení úspěšnosti. V databázi se ale musí nacházet i text, který je na obrázku

Bibliografie

- [1] *Geometric Image Transformations*. URL: https://docs.opencv.org/master/da/d54/group__imgproc__transform.html#ga47a974309e9102f5f08231edc7e7529d (cit. 16.02.2020).
- [2] *Smoothing Images*. URL: https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html (cit. 16.02.2020).
- [3] *Image Filtering*. URL: https://docs.opencv.org/master/d4/d86/group__imgproc__filter.html#ga9d7064d478c95d60003cf839430737ed (cit. 16.02.2020).
- [4] *Image Thresholding*. URL: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html (cit. 16.02.2020).