Widzenie Komputerowe

Laboratorium nr 1.

Podstawowe operacje na obrazach

Autor: Magdalena Mazur Milecka

Gdańsk, 2019

WSTĘP

Zadania należy wykonać w Pythonie (wersja 3.6.4) korzystając z darmowej biblioteki OpenCV (wersja 3.4.0).

1. Wczytywanie plików

a) obrazy

Załaduj dowolny obraz przy pomocy kodu:

```
import cv2

img=cv2.imread('tekst1.jpg')

cv2.imshow(‘okno’,img)
```

Obsłuż zamknięcie okna przyciskiem escape:

```
k=cv2.waitKey(0)

if k==27:

    cv2.destroyAllWindows();
```

Zadanie 1. - 1 pkt

Wyświetl obraz przy użyciu pakietu matplotlib:

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.imshow.html

pamiętaj o odwróceniu kanałów BGR wczytanych biblioteką OpenCV.

Wyświetlić należy obraz wczytany oraz przekonwertowany obok siebie używając `pyplot.figure` (dodawanie kolejnych obrazów poleceniem `fig.add_subplot`):

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.figure.html#examples-using-matplotlib-pyplot-figure

b) wideo

Przechwytywanie obrazu z kamery:

```
import cv2

cap = cv2.VideoCapture(0)

while(True):
```

```
    ret, frame = cap.read()
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Zadanie 2. - 2 pkt.

Przechwyć obraz z kamery, zamień go na skalę szarości (metoda `cvtColor`

z pakietu OpenCV) i zapisz do pliku używając metody VideoWriter (w razie problemu z kodekami użyć "XVID").

Wskazówki: Skorzystaj z metod:

```
plik = cv2.VideoWriter()
plik.write(frame)                    #zapis do pliku
cv.cvtColor()                 # Zmiana przestrzeni kolorów
```

https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor – dokumentacja cvtColor()

2. Rysowanie na obrazie

Zadanie 3. - 2 pkt.

Zaznacz na obrazie prostokątem obszar ROI przedstawiający np. twarz, dodaj do obrazu podpis.

Pomocna dokumentacja:

https://docs.opencv.org/trunk/dc/da5/tutorial_py_drawing_functions.html

3. Progowanie

Zadanie 4. - 3 pkt.

Na 4 różnych plikach (oko.jpg, tekst1.jpg, tekst2.jpg, tekst3.jpg) zastosować progowania:

a) ze stałą wartością progu,
b) adaptacyjne,

c) Otsu.

Wybrać najlepszą metodę dla każdego obrazu wraz z parametrami.

Do okna z obrazem dodać suwaki ustawiające wartości potrzebne w metodach.

Przykładowy kod zastosowania suwaków:

```
import numpy as np
import cv2 as cv
def nothing(x):
    pass
# Create a black image, a window
img = np.zeros((300,512,3), np.uint8)
cv.namedWindow('image')
# create trackbars for color change
cv.createTrackbar('R','image',0,255,nothing)
cv.createTrackbar('G','image',0,255,nothing)
cv.createTrackbar('B','image',0,255,nothing)
while(1):
    cv.imshow('image',img)
    k = cv.waitKey(1) & 0xFF
    if k == 27:
        break
    # get current positions of four trackbars
    r = cv.getTrackbarPos('R','image')
    g = cv.getTrackbarPos('G','image')
    b = cv.getTrackbarPos('B','image')
    img[:] = [b,g,r]
cv.destroyAllWindows()
```

Zadanie 5. - 2 pkt.

Zastosuj na obrazie z kamery wykrywanie krawędzi poprzez detektor Canny oraz dowolne detektory obliczające pierwsza i drugą pochodną. Zaproponuj metodę poprawy jakości obrazu krawędzi dla drugiej pochodnej. Wyniki zapisz w plikach filmowych.