mgr inż. Grzegorz Gołaszewski golaszewski@fis.agh.edu.pl	dr inż. Szymon Łukasik slukasik@agh.edu.pl

Wprowadzenie do biblioteki OpenCV

Ćwiczenie laboratoryjne nr 1
Analiza i przetwarzanie obrazów

Wprowadzenie

OpenCV to biblioteka funkcji wykorzystywanych podczas obróbki obrazu, oparta na otwartym kodzie i zapoczątkowana przez Intela. Biblioteka ta jest **wieloplatformowa**, można z niej korzystać w Mac OS X, Windows jak i Linux. Obecnie dostępna jest wersja 4.2.0. Biblioteka została stworzona w języku C, lecz istnieją nakładki umożliwiające korzystanie z niej również w językach C++, C#, Python, Java.

Instalacja

W przypadku korzystania z języka Python – można użyć Pythonowskiego menedżera pakietów **pip**.

Dostępne pakiety:

`opencv-python`: podstawowe moduły OpenCV

`opencv-contrib-python`: podstawowe moduły I moduły contrib (zalecane)

`opencv-python-headless`: to samo co `opencv-python` ale bez GUI

`opencv-contrib-python-headless`: to samo co `opencv-contrib-python` ale bez GUI

Wadą tego rozwiązania jest brak w wersji z repozytorium wszystkich algorytmów non-free (np. SIFT, SURF) – począwszy od wersji 3.4.2. Aby z tych technik korzystać trzeba bibliotekę skompilować samemu z flagą

```
CMAKE_ARGS="-DOPENCV_ENABLE_NONFREE=ON"
```

lub korzystać ze starszej wersji OpenCV:

```
pip install —user opencv-contrib-python==3.4.2.17
```

Instalacja w przypadku C++ - kompilacja ze źródeł :D

https://docs.opencv.org/4.2.0/d7/d9f/tutorial_linux_install.html

A pod Windowsem:

https://docs.opencv.org/4.2.0/d3/d52/tutorial_windows_install.html

Podstawy użytkowania

Na start: `import cv2`
Wczytanie obrazu: `image = cv2.imread("obraz.png")`
Sprawdzenie parametrów: `image.shape`
Kopia obrazu: `new_image = image.copy()`
Wyświetlenie: `cv2.imshow` (potem przydatne `cv2.waitKey(0)` żeby zaczekać na klawisz)
Stosując jupyter notebook `cv2.imshow` sprawia problemy więc można zastosować

```
from matplotlib import pyplot
pyplot.imshow(...)
```

Zmiana maksymalnych wymiarów obrazu w matplotlib, proporcje zostaną zachowane
`pyplot.rcParams['figure.figsize'] = [20, 10]`

Czym jest tak naprawdę `image`?

Podejrzenie koloru indywidualnego piksela:

```
(B, G, R) = image[70, 10]
```

Uwaga – pamiętaj o kolejności, OpenCV stosuje domyślnie format BGR

Konwersja systemu barw `cv2.cvtColor(image,cv2.COLOR_BGR2RGB)` również istnieją inne jak `COLOR_BGR2GRAY COLOR_RGB2GRAY`

Wycięcie fragmentu obrazu – jak w typowej tablicy tj. przez zakres np. `image[20:120, 80:220]`

Zmiana rozmiaru: `resized = cv2.resize(image, (new_x, new_y))`

Obracanie obrazu:

- Wyznaczenie środka na podstawie szerokości i wysokości
- Wyznaczenie macierzy obrotu z użyciem `cv2.getRotationMatrix2D`
- Obrócenie obrazu z wykorzystaniem macierzy obrotu i `cv2.warpAffine`.

	mgr inż. Grzegorz Gołaszewski golaszewski@fis.agh.edu.pl	dr inż. Szymon Łukasik slukasik@agh.edu.pl

Rysowanie kształtów po obrazie: `cv2.rectangle` i `cv2.circle`

Umieszczenie tekstu: `putText`

Zapisanie obrazu `cv2.imwrite`

Realizacja ćwiczenia

a) Proszę zainstalować bibliotekę OpenCV/upewnić się że jest zainstalowana

b) Proszę wczytać zamieszczony w folderze przedmiotu obraz `lab1.jpg`, sprawdzić jego parametry i wyświetlić

c) Podejrzyj zawartość indywidualnego piksela.

d) Wytnij z obrazu jedną z twarzy

e) Zmniejsz obraz o 50% - obliczając automatycznie `new_x, new_y` tak by zachować proporcję wymiarów obrazu

f) Obróć pierwotny obraz o 30 stopni zgodnie przeciwnie do ruchu wskazówek zegara. Spróbuj znaleźć rozwiązanie dla problemu wychodzenia obróconego obrazu poza okno pierwotnego obrazu

g) Oznacz jedną z twarzy na pierwotnym obrazie czerwonym prostokątem i podpisz poniżej imieniem i nazwiskiem

h) Zapisz obraz z punktu g)

i) Tematy dodatkowe:

   a. Proszę policzyć łączną liczbę kolorów użytych w obrazie

   b. Proszę zapisać obraz w skali szarości