

Wstęp do multimediów

Laboratorium – Podstawowe przetwarzanie obrazów

Celem ćwiczenia jest praktyczne zapoznanie się z podstawowymi (punktowymi i lokalnymi) operacjami przetwarzania obrazów oraz obserwacja i ocena wyników ich działania.

Zakres ćwiczenia obejmuje: korekcję kontrastu i jasności obrazu poprzez wyrównanie histogramu, zastosowanie filtrów liniowych i nieliniowych do usuwania szumu z obrazu, detekcji krawędzi w obrazie oraz wyostrażania obrazu.

Środowisko do wykonania zadań

Do realizacji zadań wykorzystywany jest język Python wraz z dodatkowymi pakietami wykorzystywanymi do analizy danych i przetwarzania obrazów: NumPy, Matplotlib i OpenCV.

Najwygodniejsze jest pobranie i zainstalowanie dystrybucji Pythona zawierającej już najpopularniejsze pakiety, np. Anaconda, która w wersji „Individual Edition” dostępna jest za darmo (<https://www.anaconda.com/products/individual>). Standardowo Anaconda zawiera wiele popularnych pakietów Pythona, ponadto zawiera wygodny program narzędziowy do przeglądania i instalowania nowych pakietów – conda. Po pobraniu i zainstalowaniu środowiska Anaconda konieczne może być doinstalowanie pakietu OpenCV, co można zrobić poleceniem: `conda install opencv` w konsoli Anaconda Prompt.

Interpeter języka Python dostępny jest z konsoli tekstowej (polecenie `python`; w przypadku Anacondy konieczne może być uruchomienie konsoli tekstowej poprzez ‘Anaconda Prompt’ z Menu Start). Skrypty języka Python można edytować dowolnym edytorem tekstu, ale wygodniej jest posługiwać się środowiskiem zintegrowanym, np. Spyder (dostępny standardowo w środowisku Anaconda) czy PyCharm. Można też wykorzystać środowisko Google Colab, dostępne z poziomu przeglądarki internetowej (<https://colab.research.google.com/>), przy czym wymagane jest posiadanie konta Google.

W ramach przygotowania do zajęć laboratoryjnych należy zapoznać się i uruchomić przykładowy skrypt, w którym zademonstrowane zostały różne operacje na obrazach, w szczególności te, które są konieczne do wykonania zadań. Skrypt dostępny jest w dwóch formatach: `.py` (do wykonania w lokalnym środowisku Pythona) oraz `.ipynb` (format Jupiter Notebook, wykorzystywany w środowisku Google Colab).

Zadania

Korzystając z przykładowego skryptu zapoznać się z wybranymi operacjami przetwarzania obrazów oraz wpływem parametrów na wyniki ich działania.

Następnie napisać skrypty realizujące wymienione poniżej zadania. W sprawozdaniu zamieścić skrypty, uzyskane obrazy, wyniki, oraz opisać wnioski.

UWAGA: Skrypty powinny być opatrzone odpowiednim komentarzem ułatwiającym analizę skryptu – w szczególności modyfikując skrypt przykładowy należy modyfikować również komentarz, aby był zgodny z realizowanymi operacjami!

W sprawozdaniu należy zamieścić wyniki uzyskane dla obrazu testowego, wybranego na podstawie numeru indeksu w następujący sposób:

```
numer_obrazu = numer_indeksu % liczba_obrazow
```

gdzie numer obrazu wskazuje pozycję pliku na posortowanej alfabetycznie liście nazw plików testowych (pozycje numerowane od 0, tak jak elementy list w Pythonie), a % oznacza operację modulo).

- Zrealizować operację filtracji barwnego obrazu cyfrowego.

Do realizacji zadania wykorzystać obrazy zaszumione (szumem gaussowskim oraz impulsowym).

Każdy z obrazów wejściowych poddać przetwarzaniu filtrem wygładzającym (Gausa) i filtrem medianowym. Każdy obraz wynikowy wyświetlić i obliczyć dla niego PSNR (w stosunku do obrazu oryginalnego, nie zaszumionego!), funkcja do obliczania PSNR dostępna jest w przykładowym skrypcie). Ocenąć działanie filtrów dla masek o rozmiarach: 3x3, 5x5, 7x7.

Zebrać w tabeli PSNR dla różnych rodzajów szumów, filtrów i rozmiarów maski.

Jaki wpływ na skuteczność filtracji i na zniekształcenie obrazu ma rozmiar maski filtru?

Czy ocena subiektywna uzyskanych obrazów wynikowych, jest zgodna z PSNR (lepszą jakość – większy PSNR)?

Opisać wnioski w sprawozdaniu.

- Zrealizować operację wyrównania histogramu dla obrazu barwnego i zapisać obraz wynikowy do pliku. UWAGA: operację wyrównania histogramu należy wykonać wyłącznie dla składowej odpowiadającej za jasność, w tym celu należy wejściowy obraz RGB skonwertować do innej przestrzeni (np. YCbCr/YUV), a po wyrównaniu histogramu dla właściwej składowej powrócić do pierwotnego formatu.

Porównać uzyskane obrazy i ich histogramy (w szczególności: histogram dla składowej, dla której wykonano operację wyrównywania histogramu).

Czy obraz po wyrównaniu histogramu jest subiektywnie *lepszego* jakości?

- Korzystając z filtru Laplace'a do wyznaczenia wysokoczęstotliwościowych składowych obrazu dokonać *wyostrzenia* obrazu:

`img_out = img_in + W*img_laplace.`

Jaki jest wpływ *wagi* składowej wysokoczęstotliwościowej na postać obrazu wynikowego?

Dla jakich wartości tej *wagi* uzyskuje się *dobry, przyjemny* dla oka wyniki?

Uwaga: należy pamiętać, że wyostrzanie obrazu powoduje również uwydatnienie szumu w obrazie, w niektórych przypadkach (niezbyt dobrej jakości obrazów oryginalnych) przydatne może być wstępne wygładzenie obrazu filtrem dolnoprzepustowym (np. filtrem Gaussa).

Wskazówka: wykorzystać funkcję `cv2.addWeighted()` do sumowania obrazów i równoczesnej konwersji wyniku do 8-bitowych liczb całkowitych z przedziału [0, 255].

Uwaga: ze względu na sposób wyliczania laplasjanu w bibliotece OpenCV w celu uzyskania *'dobrych'* wyników należy odjąć *'obraz wysokoczęstotliwościowy'*, czyli zastosować ujemne wartości *wagi* W .