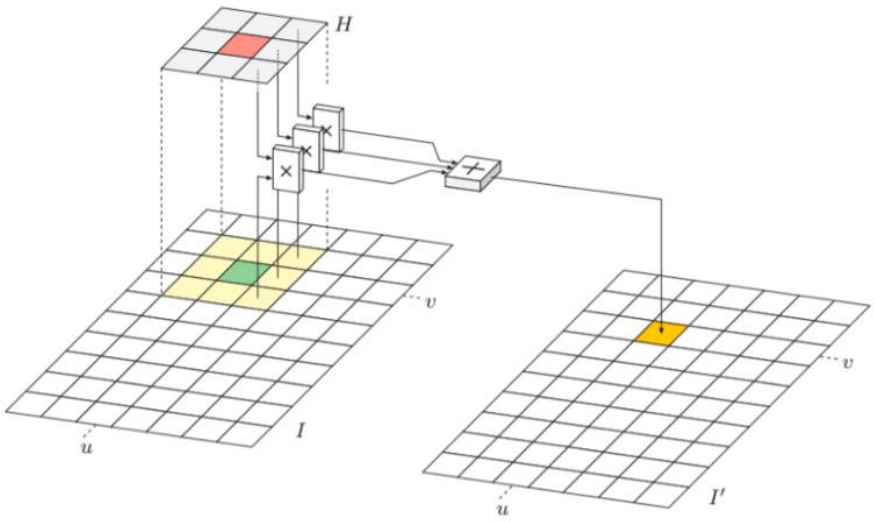
Lab6

1. Inne metody filtrowania obrazów, filtry konwolucyjne

Ogólna zasada: oglądamy obraz przez okno określonego wymiaru (kwadratowe o nieparzystej długości boku). Na postawie wartości pikseli widocznych przez okno (i ewentualnie innych podanych wartości) wyznaczamy wartość piksela środkowego.

Ważne! Całe okno musi mieścić się w obrazie, więc punkty na brzegu obrazu nie „dostaną" nowych wartości.



Kwadratowe otoczenie K punktu (m, n) o boku k można wyznaczyć jak na ćwiczeniach 7.

Filtry Minimum, Maximum, Mediany, Średniej: Niech K oznacza otoczenie danego punktu (m, n). Wtedy punktowi (m, n) dajemy wartość $p'(m, n)$, odpowiednio.

$$p'(m,n) \leftarrow \min\{p(i,j): (i,j) \in K\}$$
$$p'(m,n) \leftarrow \max\{p(i,j): (i,j) \in K\}$$
$$p'(m,n) \leftarrow \text{median}\{p(i,j): (i,j) \in K\}$$
$$p'(m,n) \leftarrow \text{mean}\{p(i,j): (i,j) \in K\}$$

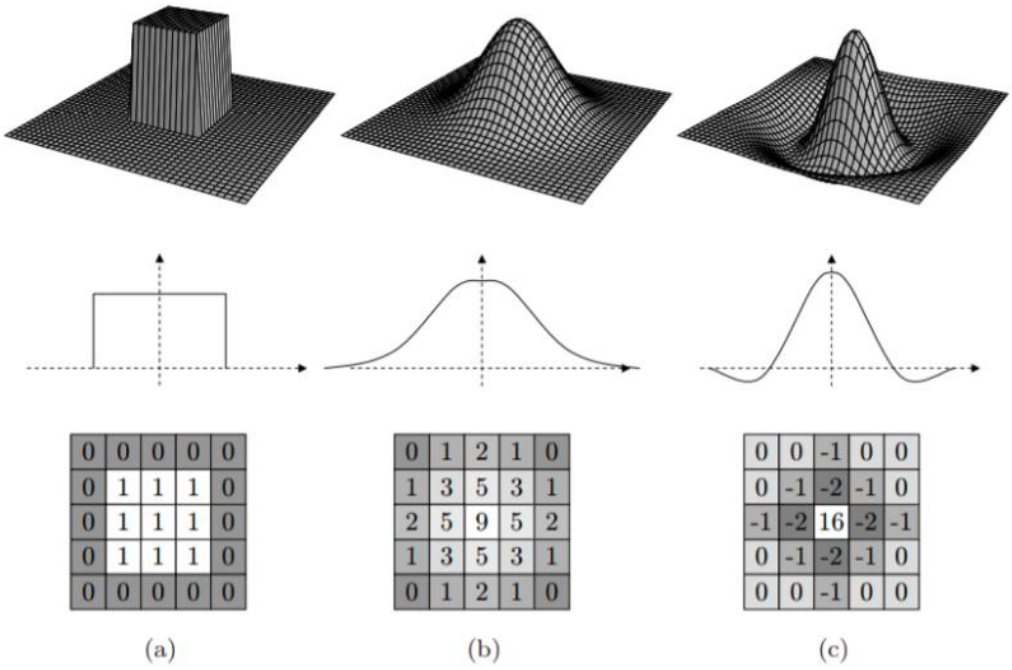
Filtry konwolucyjne

Do wyznaczenia filtru opartego na sumie ważonej, wagi przechowujemy w tablicy (lub innej strukturze danych) **H** o wymiarach takich jak okno i wykonujemy mnożenie odpowiednich wartości, sumujemy i ewentualnie dzielimy przez stałą (**scale**). Ten sposób nazywa się metodą konwolucji (ang. **Convolution**), a tablica z wagami nazwana jest jądrem (ang. **Kernel**)

Dla okna wymiaru k mamy:

$$p'(m,n) \leftarrow \sum_{a=0}^{k-1}\sum_{b=0}^{k-1} p(m-d+a, n-d+b)\cdot H(a,b), \text{ gdzie } d = int(k/2)$$

Przykłady:



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

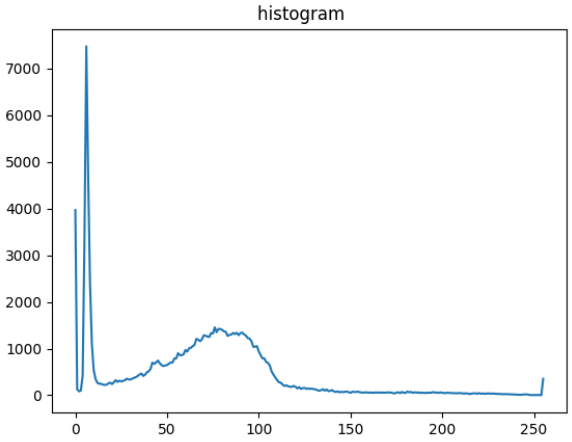
Moduł ImageFilter

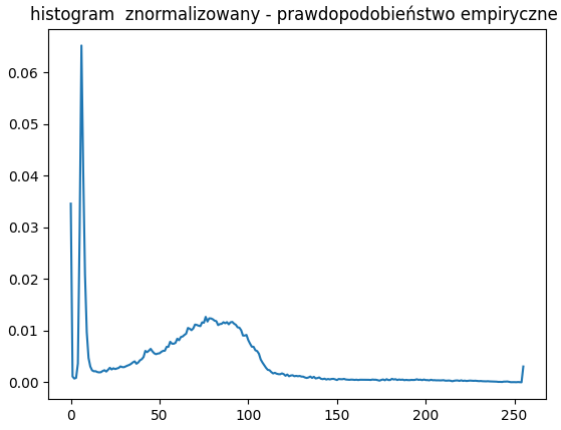
Filtry ze stałymi ustawieniami (1.-10) oraz z parametrami (11-18)

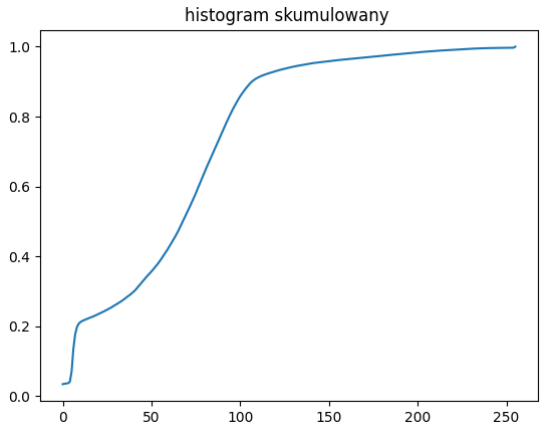
1. BLUR
2. DETAIL
3. EDGE_ENHANCE
4. EDGE_ENHANCE_MORE
5. FIND_EDGES
6. SHARPEN
7. SMOOTH
8. SMOOTH_MORE
9. CONTOUR
10. EMBOSS
11. BoxBlur
12. GaussianBlur
13. UnsharpMask
14. Kernel
15. RankFilter
16. MedianFilter
17. MinFilter
18. MaxFilter

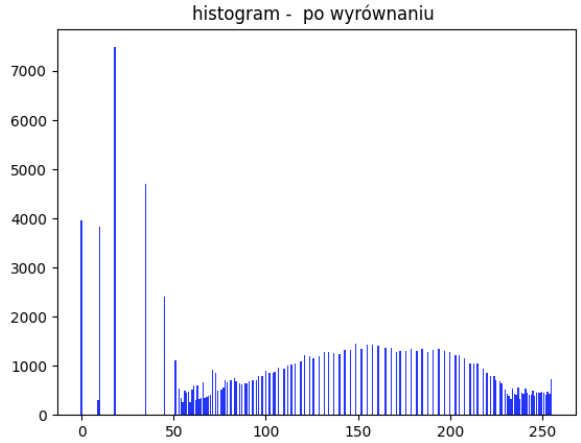
2. Filtrowanie przez wyrównanie histogramu

Histogram w trybie ‘L’ : każdej wartości z zakresu od 0 do 255 przyporządkowana jest liczba pikseli o tej wartości.









Wyrównanie histogramu (dla obrazów w trybie ‘L’):

1. Pobranie histogramu obrazu – lista hist długości 256
2. Normalizacja, tzn. każdy element histogramu dzielimy przez liczbę wszystkich pikseli w obrazie – lista hist_norm długości 256
3. Kumulacja, tzn. tworzymy z histogramu znormalizowanego histogram skumulowany – lista hist_kumul długości 256 – taka, że hist_kumul[i] jest sumą wszystkich elementów hist_norm o indeksach mniejszych równych i
4. Filtr obrazu przez wyrównanie histogramu, tzn. wartość p każdego piksela obrazu zamieniamy na int(255*hist_kumul[p])

3. Resize, rotate, transform

Filters comparison table

Filter	Downscaling quality	Upscaling quality	Performance
NEAREST			★★★★★
BOX	★		★★★★
BILINEAR	★	★	★★★
HAMMING	★★		★★★
BICUBIC	★★★	★★★	★★
LANCZOS	★★★★	★★★★	★

Zadania

1. Wczytaj swój obraz w trybie RGB.
    a. Zastosuj filtr BLUR do swojego obrazu.
    b. Pobierz informacje o filtrze BLUR, wstaw je jako parametry filtru kernel. Zastosuj do obrazu.
    c. Na diagramie plt (fig1.png) umieść obraz wejściowy, obrazy otrzymane w pkt. a. i b. oraz wynik ich porównania.
2. SOBEL, podobnie jak Emboss, wyróżnia krawędzie. Przekonwertuj swój obraz na tryb 'L' (obraz.convert('L')). Na tym obrazie:
    a. Zastosuj filtr EMBOSS
    b. Pobierz informacje o filtrze EMBOSS a następnie zmień argumenty tego filtru tak, żeby zastosować dwa poniższe filtry.
        i. SOBEL1: (-1, 0, 1, -2, 0, 2, -1, 0, 1). Zastosuj filtr
        ii. SOBEL2: (-1, -2, -1, 0, 0, 0, 1, 2, 1). Zastosuj filtr
    c. Na diagramie plt (fig2.png) umieść obraz otrzymany po konwersji na L oraz obrazy z punktów a. i b. Napisz jakie widzisz różnice między powyższymi obrazami.
3. Na diagramie plt (fig3.png) umieść obrazy powstałe z obrazu po zastosowaniu filtrów 2,4,6,8 (w kolumnie, nad każdym obrazem w tytule powinna pojawić się nazwa filtru) i obok każdego wynik porównania obrazu oryginalnego z obrazem przefiltrowanym.
4. Wyszukaj w dokumentacji Pillow jakie parametry stosuje się w przypadku filtrów 11.-18. Na diagramie plt (fig4.png) umieść obrazy powstałe z obrazu po zastosowaniu filtrów 12,13,16,17,18 z własnymi wartościami parametrów (w kolumnie, nad każdym obrazem w tytule powinna pojawić
5. Wczytaj obraz: zeby.png. Sprawdź tryb i przekonwertuj do trybu 'L'. Zastosuj do obrazu ImageOps.equalize. Otrzymany obraz zapisz jako equalized.png
6. Zastosuj do obrazu zeby.png filtry DETAIL, SHARPEN i CONTOUR.
    a. Przedstaw obraz wejściowy, obrazy po zastosowaniu filtrów i obraz equalized.png na jednym diagramie plt (nad każdym z

obrazów umieść tytuł z nazwą zastosowanego filtru) i zapisz jako filtry.png. Który z tych filtrów działa najlepiej według ciebie?

   b. Przedstaw histogramy obrazów z pkt.6a. na jednym diagramie plt (nad każdym umieść tytuł z nazwą zastosowanego filtru) i zapisz jako histogramy.png. Napisz jakie widzisz różnice w stosunku do histogramu obrazu wejściowego.

7. Zastosuj metode resize do obrazu wybranego w zad.1 Utwórz 6 obrazów przyjmując skalę dla szerokości s_w = 0.15, skalę dla wysokości s_h = 0.27 oraz kolejno metody resamplingu 'NEAREST','LANCZOS','BILINEAR','BICUBIC','BOX','HAMMING' Przedstaw na jednym diagramie plt (fig5.png) obrazy po przeskalowaniu i ich różnice w stosunku do NEAREST. Pobierz statystyki różnic i skomentuj.
8. Wybierz z zad 7 jeden z obrazów a następnie stosując resize tą samą metodą wróć do rozmiaru obrazu wejściowego. Omów różnice między obrazem otrzymanym w ten sposób a obrazem wejściowym. Przedstaw te obrazy na jednym diagramie plt (fig6.png)
9. Obróć obraz
   a. o 60 stopni w lewo dobierając argumenty metody rotate tak, żeby widoczny był cały obraz, a nadmiarowy fragment był w kolorze czerwonym
   b. o 60 stopni w lewo dobierając argumenty metody rotate tak, żeby rozmiar obrazu się nie zmienił, a nadmiarowy fragment był w kolorze czerwonym
   c. o 300 stopni w prawo dobierając argumenty metody rotate tak, żeby widoczny był cały obraz, a nadmiarowy fragment był w kolorze zielonym
   d. o 300 stopni w prawo dobierając argumenty metody rotate tak, żeby rozmiar obrazu się nie zmienił, a nadmiarowy fragment był w kolorze zielonym
   e. Przedstaw otrzymane obrazy na jednym diagramie plt (fig6.png)
10. Czy przekształcenia Image.TRANSPOSE i Image.TRANSVERSE można otrzymać wykonując obroty i Image.FLIP_LEFT_RIGHT? Jeśli tak napisz, jak to zrobić.

Raport, plik z kodem oraz obrazy zaznaczone na zielono wstawić na Moodle