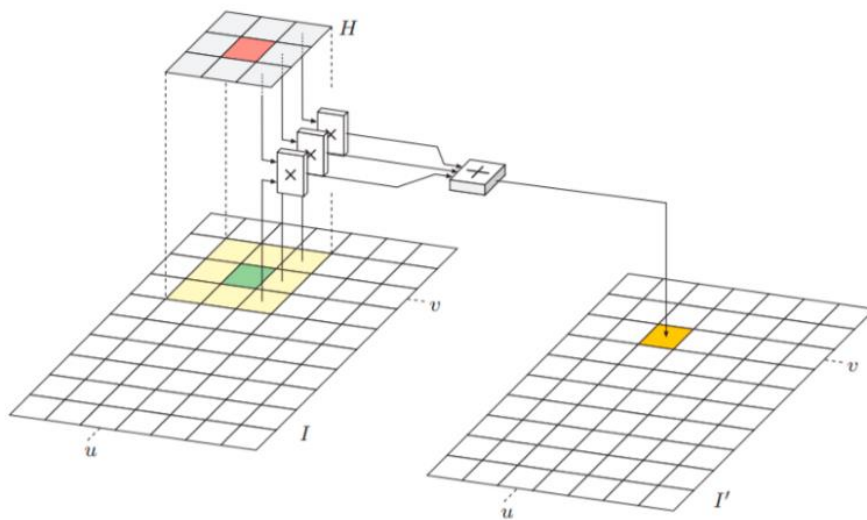


Lab7 Metody filtrowania obrazów

Ogólna zasada: oglądamy obraz przez okno określonego wymiaru (najczęściej kwadratowe o nieparzystej długości boku). Na podstawie 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.



Filtry Minimum i Maximum: Niech R oznacza otoczenie danego punktu (u, v) . Wtedy punktowi (u, v) dajemy wartość min lub max, odpowiednio.

$$I'(u, v) \leftarrow \min \{I(u+i, v+j) \mid (i, j) \in R\}$$

$$I'(u, v) \leftarrow \max \{I(u+i, v+j) \mid (i, j) \in R\}$$

Filtr Mediany: Niech R oznacza otoczenie danego punktu (u, v) . Wtedy punktowi (u, v) dajemy wartość mediany.

$$I'(u, v) \leftarrow \text{median} \{I(u+i, v+j) \mid (i, j) \in R\}$$

Na podobnej zasadzie możemy stworzyć Filtr średniej (Mean Filter) - punktowi (u,v) dajemy wartość średniej z otoczenia punktu.

Do wyznaczenia filtru opartego na sumie ważonej lub średniej 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. Ten sposób nazywa się metodą konwolucji (ang. **Convolution**) a tablica z wagami nazwana jest jądrem (ang. **Kernel**)

Teoretycznie:

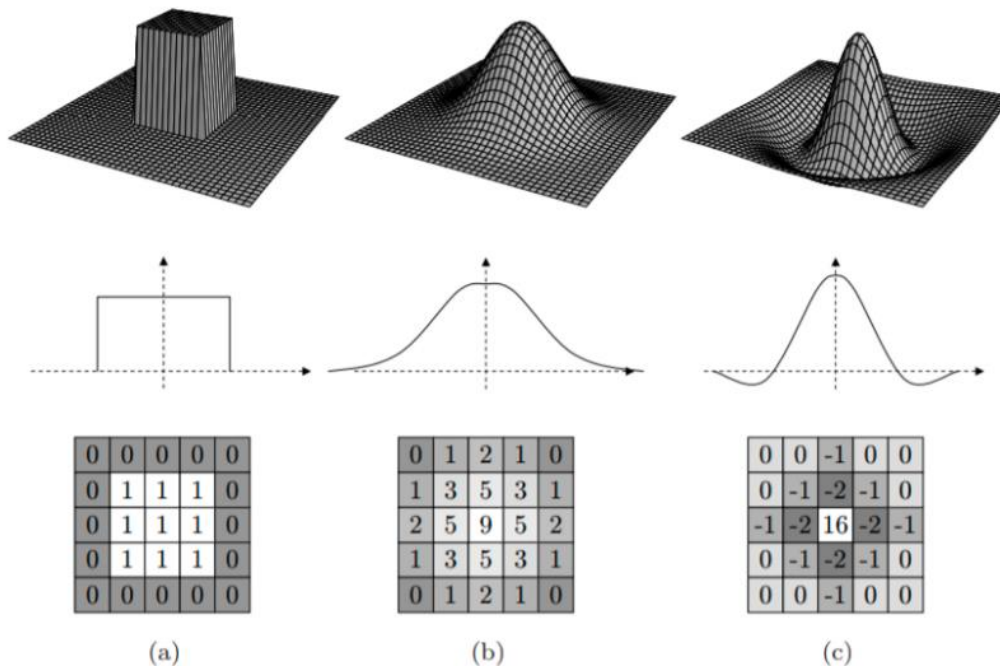
$$I'(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u-i, v-j) \cdot H(i, j)$$

W praktyce dla okna wymiaru 3 mamy:

$$I'(u, v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i, j)$$

Jak się zmieni wzór dla n = 5, 7, 9, 11 itd.?

Przykłady:



Moduł ImageFilter

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

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

Zadania Wybierz **obraz.jpg** w trybie RGB i wczytaj jako obraz

1. Napisz funkcję `filtruj(obraz, kernel, scale)`, która na podstawie podanej tablicy (lub listy) `t` dzieli wartości tej tablicy przez współczynnik `scale` a następnie wykonuje konwolucję.

Ważne: przypominam, że metoda `load` zmienia wartości pikseli w obrazie, więc dane trzeba pobierać z jednej kopii obrazu a zmiany zapisywać w drugiej.

2. Pobierz informacje o filtrze BLUR, wstaw je jako parametry funkcji `filtruj`. Otrzymany obraz nazwij `blur`
 - a. Zastosuj filtr BLUR do swojego obrazu. Otrzymany obraz nazwij `blur1`

- b. Na diagramie plt ([fig1.png](#)) umieść obrazy blur i blur1 oraz wynik ich porównania.
- 3. SOBEL, podobnie jak Emboss wyróżnia krawędzie. Przekonwertuj swój obraz na tryb 'L' (`obraz.convert('L')`) Pobierz informacje o filtrze EMBOSS a następnie zmień zawartość listy kernel.
 - a. SOBEL1: (-1, 0, 1, -2, 0, 2, -1, 0, 1). Zastosuj filtr
 - b. 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. Zapisz jakie widzisz różnice między powyższymi obrazami.
- 4. 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.
 - a. Wyszukać w dokumentacji Pillow jakie parametry stosuje się w przypadku filtrów 11.-18. i napisać w raporcie.
 - b. 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 się nazwa filtru i wartości zastosowanych parametrów) i obok każdego wynik porównania obrazu oryginalnego z obrazem przefiltrowanym.

Raport, plik z kodem oraz wszystkie obrazy zaznaczone na czerwono wstaw na Moodle.