

Rozpoznawanie Obrazów

Semestr VII, Informatyka

mgr inż. Marcin Skobel

2021

Laboratorium nr 5: Operacje sąsiedztwa i filtry liniowe.

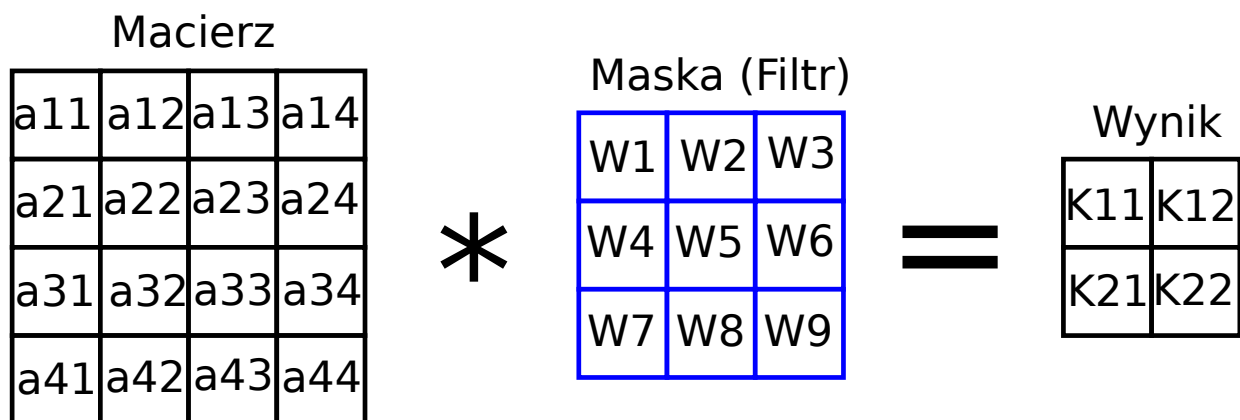
I. Zagadnienia teoretyczne

Wstęp

Oprócz poznanych na poprzednich zajęciach operacji punktowych istnieją również operacje sąsiedztwa, w których ostateczna wartość transformowanego piksela wynika także z lokalnych wartości sąsiednich pikseli. Pojęcie operatorów sąsiedztwa wiąże się z pojęciem splotu oraz filtrów obrazu. Zagadnienia te zostaną przedstawione w kolejnych punktach.

Operacja splotu

Zanim przejdziemy do filtracji obrazu warto najpierw przyjrzeć się operacji splotu. Dokładne zrozumienie operacji splotu jest niezbędne do kontynuacji tematu filtracji obrazów, a także może być przydatne w zrozumieniu działania splotowych sieci neuronowych. Do wykonania opera-



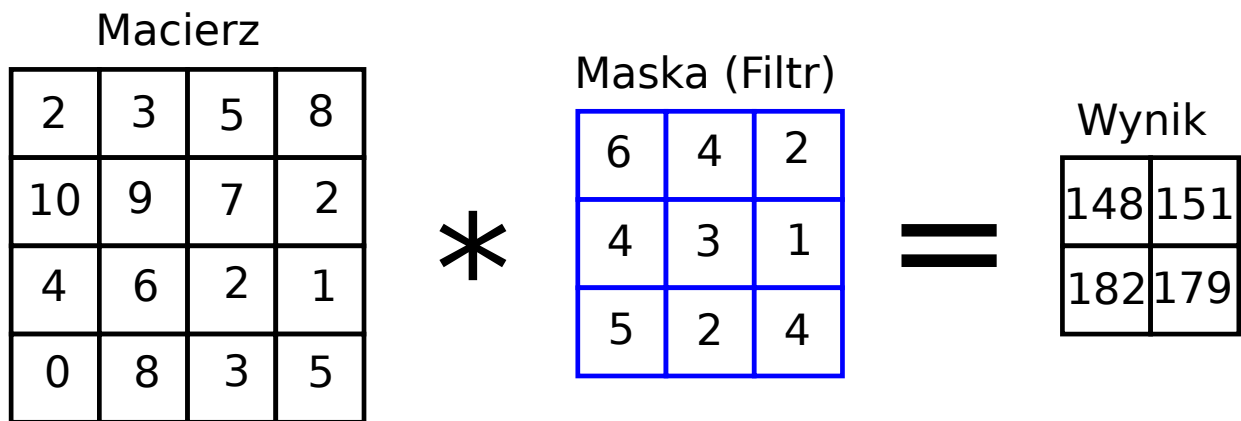
Rysunek 1: Przebieg operacji splotu

cji splotu niezbędna będzie macierz wejściowa czyli na przykład obraz, następnie filtr wraz z

wagami. Splot w literaturze często określa się symbolem *. Splot polega na mnożeniu poszczególnych pikseli obrazu przez odpowiadające im piksele na masce a następnie zsumowaniu tych iloczynów. Chcąc obliczyć przykładowo wartość K11 z obrazka 1 należy dokonać następującego obliczenia:

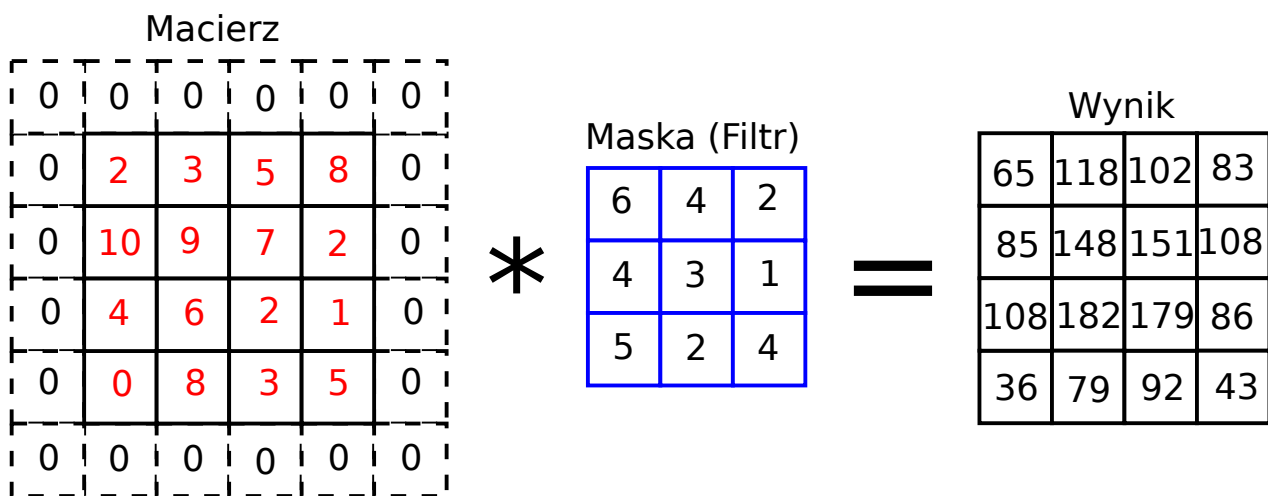
$$a_{11} \times W_1 + a_{12} \times W_2 + a_{13} \times W_3 + a_{21} \times W_4 + a_{22} \times W_5 + a_{23} \times W_6 + a_{31} \times W_7 + a_{32} \times W_8 + a_{33} \times W_9 \quad (1)$$

Ostatecznie otrzymamy macierz splotu o wymiarach 2×2 : Tutaj może się pojawić pytanie co



Rysunek 2: Wynik splotu

zrobić w przypadku gdy chcemy uzyskać macierz splotu o identycznych wymiarach jak macierz wejściowa. Istnieją dwa rozwiązania. Pierwsze rozwiązanie polega na powiększeniu obrazu wejściowego o odpowiednią liczbę zer wokół obrazu. Jeżeli maska splotu jest wymiaru 3×3 to dopisujemy jedną kolumnę z zerami po prawej i po lewej stronie macierzy a następnie po jednym wierszu na dole i jednym na górze. Drugie rozwiązanie polega na dopisaniu obwiedni z zer do macierzy wynikowej. Drugie rozwiązanie jest stosowane w splotowych sieciach neuronowych.



Rysunek 3: Rozwiązanie problemu brzegowego

Filtr jednorodny i trójkątny - wygładzanie obrazu

Filtr jednorodny występuje w postaci macierzy z jednakowymi wartościami wag w przypadku macierzy 3×3 :

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

Biorąc pod uwagę, że po zastosowaniu takiego filtra należy potem dokonać normalizacji obrazu, można też zastosować wersję filtra podzielonego przez sumę elementów macierzy filtrującej ostatecznie otrzymując:

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (3)$$

Filtr ten służy do wygładzania obrazów w tym usuwania szumu oraz zakłóceń na obrazie. Wielkość filtra może być większa niż 3×3 , ale wraz z powiększaniem rozmiaru filtra tracimy jeszcze bardziej ostrość obrazu. Filtry jednorodne to nie jedyna grupa filtrów przeznaczonych do wygładzania obrazu. Drugi pokrewny rodzaj filtrów są to filtry trójkątne. Przykładowy filtr trójkątny może przybrać następującą formę:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

A także po podzieleniu przez sumę elementów:

$$\begin{bmatrix} \frac{1}{15} & \frac{2}{15} & \frac{1}{15} \\ \frac{2}{15} & \frac{1}{5} & \frac{2}{15} \\ \frac{1}{15} & \frac{2}{15} & \frac{1}{15} \end{bmatrix} \quad (5)$$

Zastosowanie tego typu filtrów generuje efekty uboczne w postaci zmniejszenia ostrości rozmazania krawędzi i innych szczegółów obrazu.

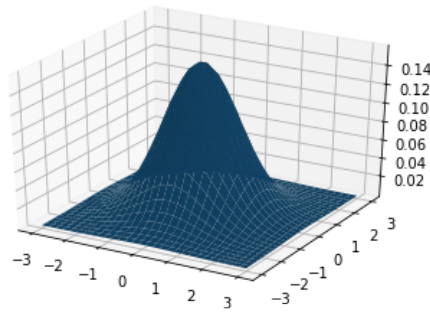
Filtr Gaussa

Do rozmycia obrazu często stosowany jest filtr Gaussa. Macierz filtracyjna otrzymuje wartości na podstawie funkcji Gaussa dla rozkładu normalnego:

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (6)$$

Oznacza to, że kiedy tworzymy filtr zgodnie z rozkładem Gaussa, centralny piksel ma największą wagę a sąsiednie piksele uzyskują mniejszą wagę podczas wykonywania splotu. Piksel, który ma zostać zmodyfikowany, będzie miał największą wagę w filtrze, a waga zmniejszy się dla pikseli, które są daleko. Przykładowo dla filtra o rozmiarze 3×3 i wartości $\sigma = 1$ filtr przyjmie wartości:

$$\begin{bmatrix} 0.060 & 0.098 & 0.060 \\ 0.098 & 0.162 & 0.098 \\ 0.060 & 0.098 & 0.060 \end{bmatrix} \quad (7)$$



Rysunek 4: Rozkład Gaussa

Co oznacza że suma wartości poszczególnych elementów zgodnie z rozkładem Gaussa powinna się sumować do wartości 0.78. Często stosowany jest również zapis filtru Gaussa w postaci potęgowej:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (8)$$

Wersja filtra o wartości 1σ dla rozmiaru 5×5 przyjmuje następującą postać:

$$\begin{bmatrix} 0.002 & 0.013 & 0.022 & 0.013 & 0.002 \\ 0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\ 0.022 & 0.098 & 0.162 & 0.098 & 0.022 \\ 0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\ 0.002 & 0.013 & 0.022 & 0.013 & 0.002 \end{bmatrix} \quad (9)$$

gdzie suma wszystkich elementów macierzy wynosi 0.98.

Filtr kierunkowy

W zależności od potrzeb istnieją również metody rozmywania obrazu w określonym kierunku. Przykładowo rozmycie w kierunku horyzontalnym można wykonać poprzez użycie poniższego filtra:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

Chcąc rozmyć obraz w kierunku wertykalnym użyjemy takiej macierzy:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (11)$$

Można też przetestować rozmycie po diagonalnej bądź po drugim ukosie:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ lub } \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

Wyostrenie obrazów

Proste filtry mogą być stosowane nie tylko do rozmazywania obrazów, ale także mogą pomóc wyostrzyć obraz. Przykładowo filtry o rozmiarze 3×3 :

$$F1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{lub} \quad F2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (13)$$

Wyostrając obraz nie trzeba poprzestawać na jednym splocie. Kolejny sploc jeszcze bardziej wyostrza obraz. Niestety efektem ubocznym wyostrzania jest powstawanie artefaktów na obrazie.

II. Przykład praktyczny

Wstępna konfiguracja

Tym razem bez większych zmian:

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from google.colab import drive
drive.mount('/content/drive')
```

Wczytanie pliku

Następnie należy wczytać plik testowy i skonwertować do go odcieni szarości. Znów użyjemy gotowej funkcji do konwersji obrazu do odcieni szarości i skonwertujemy do tablicy numpy.

```
path = testImagePath = 'drive/My Drive/Colab Notebooks/Images/Test.tif'
img = Image.open(path)
imgGray = img.convert('LA')
npImageArray = np.asarray(imgGray, dtype="uint8")
```

Filtr jednorodny

```
IA = npImageArray[:, :, 0]
Kernel = np.array([[1/9, 1/9, 1/9], [1/9, 1/9, 1/9], [1/9, 1/9, 1/9]])
Size = Kernel.shape + tuple(np.subtract(IA.shape, Kernel.shape) + 1)
subM = np.lib.stride_tricks.as_strided(IA, shape = Size, strides = IA.strides * 2)
m = np.einsum('ij,ijkl->kl', Kernel, subM)
imgAveraging = Image.fromarray(np.uint8(m))
```

W pierwszej linijce kodu pobieramy wartości tablicy do której został zapisany obraz wejściowy. W drugiej linijce podajemy interesujący nas filtr, czyli musi mieć odpowiedni rozmiar i wartości.

W trzeciej linijce przygotowujemy wektor zawierający 4 wartości. Pierwsze dwie to rozmiar filtra a kolejne dwie to rozmiar obrazu wyjściowego po splocie czyli pomniejszony o dwie kolumny i dwa wiersze. Czwarta linijka służy do utworzenia widoku na tablicę z podanym dokładnym krokiem i kształtem tablicy. Kolejna linijka kodu używa funkcji `einsum` do wykonania operacji splotu. Funkcja `einsum` może w efektywny sposób zastępować różne kombinacje sum i mnożenia elementów tablic. Dopisek w postaci `'ij,ijkl->kl'` pozwala na wykonanie operacji splotu. Ostatnia linijka kodu to po prostu zamiana otrzymanej tablicy na obraz cyfrowy.

Filtr Gaussa

```
IA = npImageArray[:, :, 0]
Kernel = np.array([[0.06, 0.098, 0.06], [0.098, 0.162, 0.098], [0.06, 0.098, 0.06]])
Size = Kernel.shape + tuple(np.subtract(IA.shape, Kernel.shape) + 1)
subM = np.lib.stride_tricks.as_strided(IA, shape = Size, strides = IA.strides * 2)
m = np.einsum('ij,ijkl->kl', Kernel, subM)
imgGauss = Image.fromarray(np.uint8(m))
```

Filtracja Gaussa została wykonana analogicznie do poprzedniego przykładu. Zmieniona została tablica filtracji.

Wyostrezanie obrazu

```
IA = npImageArray[:, :, 0]
Kernel = np.array([[ -1, -1, -1], [-1, 9, -1], [-1, -1, -1]])
Size = Kernel.shape + tuple(np.subtract(IA.shape, Kernel.shape) + 1)
subM = np.lib.stride_tricks.as_strided(IA, shape = Size, strides = IA.strides * 2)
m = np.einsum('ij,ijkl->kl', Kernel, subM)
imgSharpening = Image.fromarray(np.uint8(m))
```

Wyostrezanie obrazu również w podobny sposób zostało wykonane.

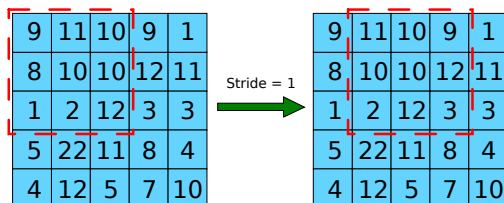
Wyświetlenie obrazów

```
fig = plt.figure()
fig.set_size_inches(20, 10)
ax1 = fig.add_subplot(2, 2, 1)
ax1.imshow(imgGray)
ax1.set_title("Input")
ax1 = fig.add_subplot(2, 2, 2)
ax1.imshow(imgAveraging, cmap="gray")
ax1.set_title("Averaging operator")
ax1 = fig.add_subplot(2, 2, 3)
ax1.imshow(imgGauss, cmap="gray")
ax1.set_title("Gaussian")
ax1 = fig.add_subplot(2, 2, 4)
ax1.imshow(imgSharpening, cmap="gray")
ax1.set_title("Sharpening")
```

W ostatnim okienku wykonano wyświetlanie wszystkich przetworzonych w ćwiczeniu obrazów.

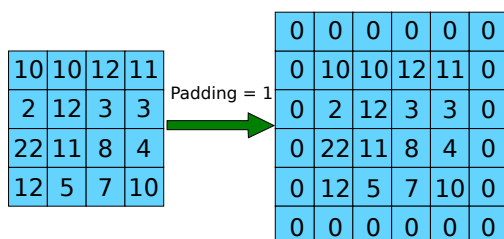
III. Uwagi

Istnieją gotowe funkcje pozwalające na przeprowadzenie filtracji. Funkcje te można odnaleźć m. in. w bibliotekach Pillow, OpenCV. Filtrowanie obrazów oraz operacja splotu nieodzownie wiąże się z dwoma ważnymi pojęciami. Pierwsze pojęcie to przesunięcie (*ang. stride*), które oznacza wielkość kroku przemieszczenia macierzy filtrującej po obrazie. Pojedynczy skok najczęściej przyjmuje wartość 1. Drugie pojęcie określa się jako dopełnienie (*ang. padding*) i oznacza



Rysunek 5: Przesunięcie

ono obszar o jaki poszerzono obraz wyjściowy po filtracji w celu uzyskania wymiarów identycznych jak obraz wejściowy.



Rysunek 6: Dopełnienie

Plik z przykładami można pobrać ze strony: <http://staff.uz.zgora.pl/mskobel/lab5.py>

IV. Lista zadań

1. Na podstawie przykładowych zadań wykonaj filtrację w kierunku horyzontalnym wertykalnym i jednym skośnym następnie wyświetl obrazy wspólnie z obrazem wejściowym i porównaj wizualnie wyniki.
2. Użyj wzoru 6 odnoszącego się do rozkładu normalnego w celu wygenerowania macierzy filtru Gaussa na podstawie zmiennych w postaci parametru σ oraz rozmiaru filtra. Następnie użyj zaimplementowany generator do wykonania filtracji Gaussa. Można wszystko obudować w funkcję.
3. Na podstawie wzoru: $IM = I * F * F$ oraz przykładów praktycznych wykonaj operację podwójnego wyostżenia obrazu. Symbol $*$ oznacza operację splotu.
4. Obraz po operacji splotu ma mniejsze wymiary niż obraz wejściowy. Skorzystaj z wycinania obrazów w celu pobrania z obrazu wejściowego brakującej obwiedni obrazu wynikowego. Następnie dodaj obwiednię do obrazu wynikowego. Sprawdź czy po przekształceniu obrazy mają identyczne wymiary.