

Laboratorium CPO

Temat: Morfologia obrazów

Cel: Zastosowanie metod morfologii matematycznej do przetwarzania obrazów

1. Erozja i dylatacja

Na poniższym obrazie sprawdź wynik działania operacji erozji i dylatacji dla różnych kształtów elementu strukturalnego. Wykorzystaj funkcje `cv2.erode()` i `cv2.dilate()` z biblioteki `opencv`.



2. Otwarcie i zamknięcie

Oczyść poniższy obraz z zakłóceń. Wykorzystaj funkcję `cv2.morphologyEx()`.



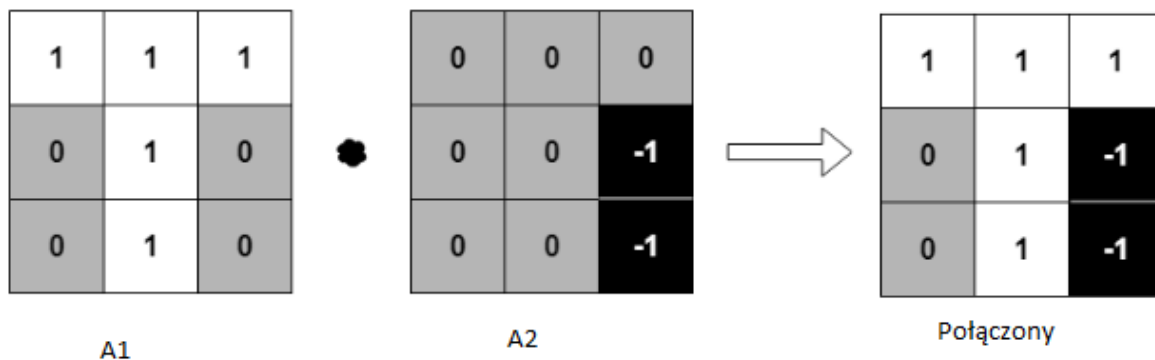
3. Transformacja Hit-or-Miss

Funkcja `cv2.morphologyEx()` wykonuje zarówno proste jak i zaawansowane transformacje morfologiczne obrazu.

W ćwiczeniu użyjemy transformacji typu Hit-or-Miss. Zasadniczo jest ona używana do wykrywania kształtów lub znajdowania określonych wzorów na obrazie. Kształt lub wzór do dopasowania musi być dostarczony przez element strukturalny.

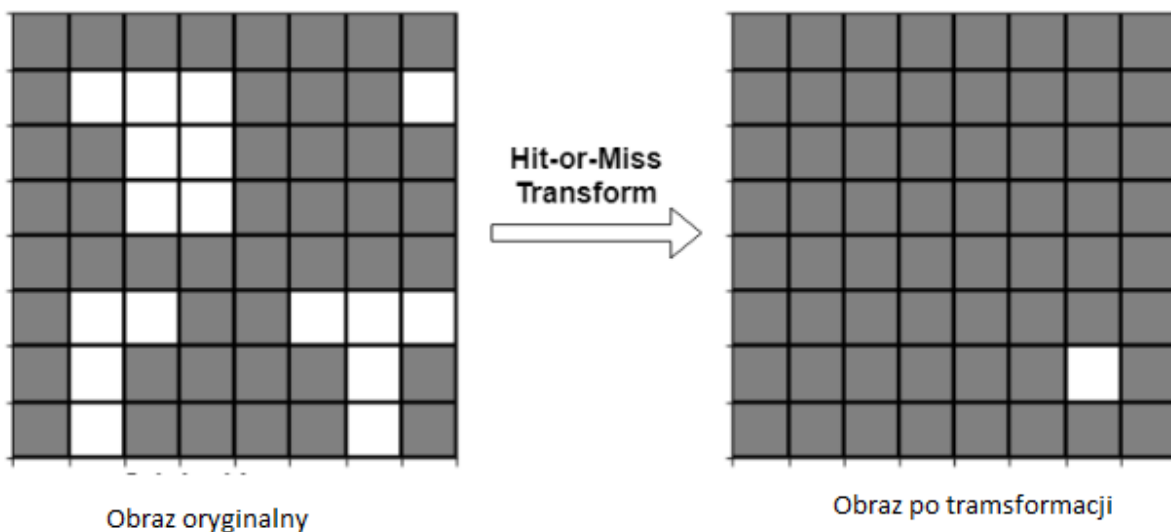
W zadaniu tym używane są dwa elementy strukturalne A1 i A2. Zadajemy pytanie, czy A1 pasuje do obiektu (trafia), a jednocześnie czy A2 nie pasuje (nie trafia), tzn. pasuje do tła? Innymi słowy, interesują nas tylko te piksele, których sąsiedztwo dokładnie pasuje do A1, a jednocześnie nie pasuje do A2. Do tego zadania wykorzystuje się operację erozji.

W ćwiczeniu zakładamy, że chcemy znaleźć w obrazie wzór w kształcie litery T zdefiniowany przez SE A1 i nie zawiera wzorca zdefiniowanego przez SE A2 w jego prawym sąsiedztwie. 0 oznacza, że nie zależy nam na tych pozycjach. Poniższy rysunek pokazuje jak połączyć oba SE w jeden.



Teraz wystarczy porównać podstawowe wartości pikseli z połączonym SE. Jeżeli pasuje dokładnie, piksel pod początkiem SE jest ustawiony na 1, w przeciwnym razie na 0.

Weźmy przykład i założmy, że chcemy znaleźć powyższy połączony SE na obrazku pokazanym poniżej. Przeanalizuj, dlaczego tylko kształt z prawego dolnego rogu pasuje do wzoru.



Wykonaj poniższy kod i sprawdź poprawność analizy.

```
# %%
import cv2
import numpy as np

# Obraz wejściowy
input_image = np.array((
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 255, 255, 255, 0, 0, 0, 255],
    [0, 0, 255, 255, 0, 0, 0, 0],
    [0, 0, 255, 255, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 255, 255, 0, 0, 255, 255, 255],
    [0, 255, 0, 0, 0, 0, 255, 0],
    [0, 255, 0, 0, 0, 0, 255, 0]), dtype="uint8")

# Element strukturalny
kernel = np.array((
    [1, 1, 1],
    [0, 1, -1],
    [0, 1, -1]), dtype="int")

# Zastosuj transformację hit-or-miss
output_image = cv2.morphologyEx(input_image, cv2.MORPH_HITMISS, kernel)

print(input_image)
print("\n\n")
print(output_image)
```

W ten sposób można znaleźć dowolne wzorce na obrazie wejściowym

Zmień element strukturalny w taki sposób, aby wyszukiwał w obrazie kontur lewego górnego narożnika.