**Wyższa Szkoła Bankowa w Chorzowie**

**Mateusz Piórkowski 104268**

|  |
| --- |
|  |

**SPRAWOZDANIE Z ĆWICZENIA NR 3**

***Przetwarzanie Informacji Multimedialnej***

**ZADANIE 1**  
Operacje filtracji – Filtering. Przebadać wpływ parametrów filtru medianowego, Gaussowskiego, ruchomej średniej na efektywność filtracji szumu typu ‘Salt & Pepper’ oraz białego szumu Gaussowskiego. Pokazać działanie na 3 własnych obrazach w tabelce gdzie pokazany jest obraz przed i po filtracji. Opisać działanie jednego z wybranych trzech filtrów.

|  |  |
| --- | --- |
| **OBRAZEK 1 Salt and pepper** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

|  |  |
| --- | --- |
| **OBRAZEK 1 Gaussian white noise** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

|  |  |
| --- | --- |
| **OBRAZEK 2 Salt and pepper** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

|  |  |
| --- | --- |
| **OBRAZEK 2 Gaussian white noise** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

|  |  |
| --- | --- |
| **OBRAZEK 3 Salt and pepper** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

|  |  |
| --- | --- |
| **OBRAZEK 3  Gaussian white noise** | |
| ***Oryginalny*** | ***Filtr medianowy*** |
|  |  |
| ***Gaussowski*** | ***Ruchoma średnia*** |
|  |  |

**FILTR GAUSSOWSKI**

Filtr Gaussowski jest stosowany do usuwania szumów i wygładzania obrazów. Działa na zasadzie rozmywania obrazu, przez co niedoskonałości i szumy są zmiękczane. W OpenCV, filtr Gaussowski jest realizowany przez funkcję cv2.GaussianBlur(). Funkcja ta przyjmuje trzy argumenty: źródłowy obraz, rozmiar jądra filtru oraz standardowe odchylenie (sigma) filtru. Po wywołaniu funkcji filtru na obrazie, wynikowy obraz jest zwracany jako wynik działania funkcji.

**ZADANIE 2**Przebadać działanie algorytmu detekcji tablic rejestracyjnych na własnych pięciu obrazach. Przykład wykonania poniżej. Opisać działanie tego algorytmu (poszukać w dokumentacji programu).

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Algorytm detekcji tablic rejestracyjnych w OpenCV polega na kilku krokach:

1. Przekształcenie obrazu na odcienie szarości - w celu ujednolicenia kolorów i usunięcia informacji o kolorze.
2. Wydobycie cech z obrazu, np. za pomocą detekcji krawędzi lub transformacji Hougha.
3. Wykrycie kandydatów na tablice rejestracyjne poprzez wyszukiwanie kształtów zbliżonych do prostokątów.
4. Ocena i weryfikacja kandydatów na tablice rejestracyjne na podstawie cech, takich jak proporcje, wielkość, umiejscowienie i kształt liter i cyfr.
5. Wyodrębnienie i rozpoznanie liter i cyfr na wybranej tablicy rejestracyjnej.

W OpenCV detekcja tablic rejestracyjnych jest realizowana za pomocą kilku funkcji, takich jak cv2.cvtColor(), cv2.Canny(), cv2.findContours() i cv2.getStructuringElement(). Funkcje te są używane do przekształcenia obrazu, wydobywania cech, wykrywania kandydatów i oceny i weryfikacji tablic rejestracyjnych. Wynik działania algorytmu to zaznaczone na obrazie tablice rejestracyjne i wyodrębnione litery i cyfry.

**ZADANIE 3**Zbadać działanie algorytmu GrabCut na własnych wybranych 4 obrazach. Opisać działanie algorytmu.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

GrabCut to algorytm wykorzystywany do selektywnego usuwania tła z obrazu. Działa on na zasadzie określenia granicy pomiędzy tłem a obiektem interesującym. W OpenCV, algorytm GrabCut jest realizowany przez funkcję cv2.grabCut().

Kroki działania GrabCut to:

1. Określenie początkowej maski - użytkownik określa część obrazu, która jest tłem lub obiektem interesującym, za pomocą prostokątnego okna lub ręcznie zaznaczając fragmenty obrazu.
2. Wykonanie algorytmu - GrabCut korzysta z modelu Gaussa i modelu tła, aby określić prawdopodobieństwo, że dany piksel jest tłem lub obiektem interesującym.
3. Iteracyjne udoskonalanie maski - algorytm powtarza proces, aż do uzyskania właściwego wyniku lub osiągnięcia maksymalnej liczby iteracji.
4. Usuwanie tła i wyodrębnienie obiektu interesującego - na podstawie ostatecznej maski, tło jest usuwane, a obiekt interesujący jest wyodrębniany i zaznaczany na obrazie.

Wynik działania algorytmu GrabCut to obraz z wyodrębnionym obiektem interesującym i usuniętym tłem. Funkcja cv2.grabCut() zwraca obraz po usunięciu tła.

**ZADANIE 4**Zbadać działanie operatorów morfologicznych na własnych 4 obrazach. Przebadać wszystkie 6 operatorów. Opisać działanie wybranych dwóch.

|  |  |
| --- | --- |
| **OBRAZEK 1** | |
|  | |
| ***Dilatation*** | ***Erosion*** |
|  |  |
| ***Opening*** | ***Closing*** |
|  |  |
| ***Gradient*** | ***Tophat*** |
|  |  |
| **OBRAZEK 2** | |
|  | |
| ***Dilatation*** | ***Erosion*** |
|  |  |
| ***Opening*** | ***Closing*** |
|  |  |
| ***Gradient*** | ***Tophat*** |
|  |  |
| **OBRAZEK 3** | |
|  | |
| ***Dilatation*** | ***Erosion*** |
|  |  |
| ***Opening*** | ***Closing*** |
|  |  |
| ***Gradient*** | ***Tophat*** |
|  |  |
| **OBRAZEK 4** | |
|  | |
| ***Dilatation*** | ***Erosion*** |
|  |  |
| ***Opening*** | ***Closing*** |
|  |  |
| ***Gradient*** | ***Tophat*** |
|  |  |

**Filtrowanie dylatacji** to operacja morfologiczna w obrazach, która polega na rozszerzaniu obszarów jednolitego koloru lub tekstury w obrazie. Jest to jeden z dwóch podstawowych operatorów morfologicznych w OpenCV, obok erozji.

W OpenCV, filtrowanie dylatacji jest realizowane przez funkcję cv2.dilate().

Kroki działania filtrowania dylatacji to:

1. Wybór elementu strukturalnego (jądra) - jest to mały kształt, który będzie powielany na obrazie w celu jego rozszerzenia.
2. Dylatacja - jądro jest powielane w miejscach, w których wartość piksela jest równa wartości piksela w jądrze. W ten sposób obszary jednolitego koloru lub tekstury są rozszerzane.
3. Zwrócenie wyniku - wynik filtrowania dylatacji to obraz, w którym obszary jednolitego koloru lub tekstury są rozszerzone.

Filtrowanie dylatacji jest często używane w obrazach do łączenia i rozszerzania kształtów, co może pomóc w ich wykrywaniu i analizie.

**Filtrowanie Top Hat** to operacja morfologiczna w obrazach, która polega na wykonaniu różnicy pomiędzy oryginalnym obrazem a jego obrazem po filtrowaniu dylatacji. Jest to jedna z operacji morfologicznych, które są często używane w analizie obrazów do wykrywania i analizy obiektów o jednolitej teksturze.

W OpenCV, filtrowanie Top Hat jest realizowane przez funkcję cv2.morphologyEx() z flagą cv2.MORPH\_TOPHAT.

Kroki działania filtrowania Top Hat to:

1. Dylatacja obrazu - oryginalny obraz jest poddawany operacji dylatacji, aby rozszerzyć obszary jednolitego koloru lub tekstury.
2. Wykonanie różnicy - różnica pomiędzy oryginalnym obrazem a jego obrazem po dylatacji jest obliczana.
3. Zwrócenie wyniku - wynik filtrowania Top Hat to obraz, w którym obszary jednolitego koloru lub tekstury są wyodrębnione i wyraźnie widoczne.

Filtrowanie Top Hat jest często używane w obrazach do wykrywania i analizy obiektów o jednolitej teksturze, takich jak komórki, naczynia krwionośne lub inne obiekty o podobnej teksturze.

**ZADANIE 5**Przebadać działanie filtrowania „Thresholding”. Przebadać działanie wszystkich trzech typów na 5 wybranych obrazach. Opisać działanie jednego wybranego algorytmu.

|  |  |
| --- | --- |
| **ORYGINAŁ** | ***Fixed Threshold*** |
|  |  |
| ***Otsu*** | ***Adaptive*** |
|  |  |

|  |  |
| --- | --- |
| **ORYGINAŁ** | ***Fixed Threshold*** |
|  |  |
| ***Otsu*** | ***Adaptive*** |
|  |  |

|  |  |
| --- | --- |
| **ORYGINAŁ** | ***Fixed Threshold*** |
|  |  |
| ***Otsu*** | ***Adaptive*** |
|  |  |

|  |  |
| --- | --- |
| **ORYGINAŁ** | ***Fixed Threshold*** |
|  |  |
| ***Otsu*** | ***Adaptive*** |
|  |  |

|  |  |
| --- | --- |
| **ORYGINAŁ** | ***Fixed Threshold*** |
|  |  |
| ***Otsu*** | ***Adaptive*** |
|  |  |

**Algorytm Adaptive** to algorytm, który polega na dostosowywaniu parametrów filtrowania obrazów do specyficznych cech obrazu. W przeciwieństwie do tradycyjnych metod filtrowania, gdzie jedne parametry są stosowane dla całego obrazu, algorytm Adaptive dostosowuje parametry filtrowania do różnych obszarów obrazu.

W OpenCV, algorytm Adaptive jest realizowany przez funkcję cv2.adaptiveThreshold(). Funkcja ta pozwala na dostosowanie progu binaryzacji obrazu do specyficznych cech obrazu.

Kroki działania algorytmu Adaptive to:

1. Podział obrazu na małe obszary - obraz jest dzielony na małe obszary, zwane oknami.
2. Obliczanie średniej wartości piksela dla każdego okna - średnia wartość piksela dla każdego okna jest obliczana.
3. Ustawianie progu binaryzacji dla każdego okna - dla każdego okna jest ustawiany indywidualny prog binaryzacji, bazujący na średniej wartości piksela dla danego okna.
4. Binaryzacja obrazu - każdy piksel w obrazie jest porównywany z indywidualnym progiem binaryzacji dla jego okna, a następnie jest zaznaczany jako czarny lub biały.
5. Zwrócenie wyniku - wynikiem algorytmu Adaptive jest obraz po binaryzacji, w którym każde okno ma indywidualnie dostosowany prog binaryzacji.

Algorytm Adaptive jest często używany w obrazach do wykrywania i analizy obiektów o zróżnicowanej jasności i kontraście, takich jak tekst na zdjęciu czy obiekty na tle niejednolitego tła.

**ZADANIE 6**Zbadać działanie algorytmu detekcji twarzy na wybranych własnych 5 obrazach. Opisać działanie algorytmu (poszukać w dokumentacji). Przykład poniżej.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Detekcja twarzy (Face Detection)** to proces identyfikowania i wyznaczania położenia twarzy na obrazie. W programie OpenCV demonstrator, detekcja twarzy jest realizowana za pomocą różnych algorytmów, takich jak Haar Cascades, HOG (Histogram of Oriented Gradients) i Deep Learning.

Haar Cascades to klasyfikator oparty na drzewie decyzyjnym, który jest w stanie wykryć twarze na obrazie, wykorzystując cechy takie jak kształt oczu, nos, ust i inne.

HOG (Histogram of Oriented Gradients) to algorytm oparty na cechach, który wykorzystuje histogramy gradientów orientacji do wykrywania twarzy.

Deep Learning to metoda uczenia maszynowego, w której sieć neuronowa jest uczona na dużych zbiorach danych twarzy, aby rozpoznawać twarze na nowych obrazach.

Kroki działania detekcji twarzy w programie OpenCV demonstrator to:

1. Przetwarzanie obrazu wejściowego - obraz jest skalowany, konwertowany na odcienie szarości i przetwarzany w celu uzyskania lepszej jakości obrazu.
2. Wykrywanie twarzy - algorytm detekcji twarzy jest uruchamiany na obrazie wejściowym i wyznacza położenie twarzy na obrazie.
3. Rysowanie prostokątów wokół wykrytych twarzy - prostokąty są rysowane wokół wykrytych twarzy, aby je zaznaczyć na obrazie.
4. Zwrócenie wyniku - wynikiem algorytmu detekcji twarzy jest obraz, na którym twarze są zaznaczone prostokątami.

Detekcja twarzy jest często używana w aplikacjach takich jak analiza emocji, identyfikacja twarzy i inne aplikacje związane z obrazami i wideo.