Gliwice, 19-01-2022 r.

**Modelowanie matematyczne**

Dokumentacja projektu końcowego

Tytuł: **Wybrane algorytmy przetwarzania obrazów**

Autor: Dawid Bitner, Daniel Broczkowski, Damian Kwaśniok

Kierunek: Informatyka, studia 2 stopnia (sem.II)

***Cel zadania:***Celem zadania było zaprezentowanie wybranych algorytmów przetwarzania obrazów dostępnych w programie Mathematica.

***Opis:***  
  
W ramach projektu zostały zaprezentowane następujące algorytmy:

**1. Metoda odszumiania obrazu poprzez zastosowanie nielokalnego filtru średnich**

Nielokalny filtr średnich opiera się na średniej liczonej nie na lokalnym obszarze obrazu, lecz na średniej wszystkich pikseli na obrazie. Skutkuje to większą przejrzystością po filtrowaniu i mniejszą utratą szczegółów obrazu w porównaniu z filtrami lokalnymi.

Wartości pikseli są obliczane ze wzoru:

gdzie:

* – obszar obrazu
* i – piksele obrazu
* – funkcja ważenia określająca jak ściśle są powiązane punkty na obrazie, najczęściej używana jest funkcja ważenia Gaussa:

,gdzie:

* - paremetr filtrowania (najcześciej odchylenie standardowe)
* – lokalna średnia wartość pikseli obrazu w otoczeniu
* Całka jest liczona
* – współczynnik normalizacyjny wyznaczany z wzoru:

Efekt zastosowania tego filtru jest widoczny gołym okiem:



**2. Alorytm konwersji obrazu na grafikę wektorową**

Wektoryzacja obrazu, czyli konwersja grafika rastrowej na wektorową ma zastosowanie przy tworzeniu clip-artów, projektowaniu grafik i różnych produktów.

Przekonwertowane zdjęcie traci niektóre szczegóły, wygładza często powierzchnie obiektów.

Algorytm stusuje filtr przepływu krzywizny, który w swoim działaniu wykorzystuje równanie różniczkowe:

z krzywizną konturu: dla każdego kanału obrazu .

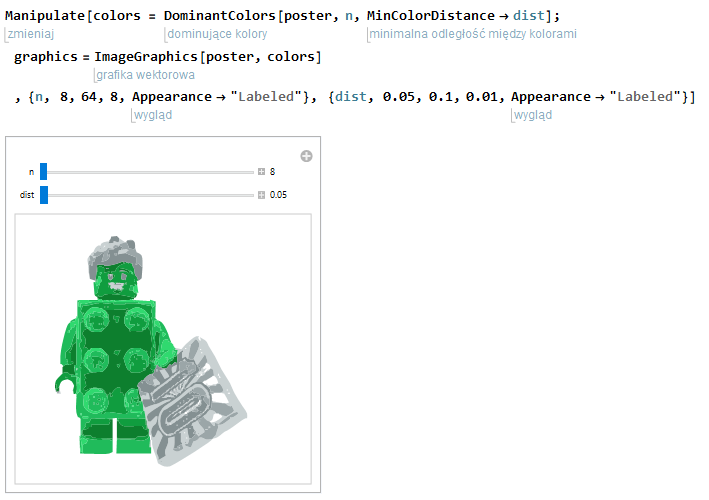
Filtr przepływu krzywizny wygładza obraz przy jednoczesnym zachowaniu krawędzi. Skutecznie rozkład krzywiznę wzłuż konturu, zaokrąglając w ten sposób rogi i zmniejszając euklidesową długość konturu.

Następnie z grafiki z pomocą metody DominantColors wyszukiwane jest ***n*** najbardziej dominujących kolorów z minimalną odległością kolorów = ***d***.

Algorytm ten przetestowano dla danych ***n = 32*** i odległości ***d = 0.05.***

Została również przygotowana procedura Manipulate sterująca parametrami n oraz d.   
  
Przykładowe otrzymane wyniki:





**3. Metoda zmiany nasycenia barw z palety RGB na obrazach**

Wbudowana w Mathematica funkcja **ImageApply** pozwala na zastosowanie na obrazie różnorakich filtrów, które można również samemu zdefiniować.

Przykładowym filtrem może być operacja zerująca wartości nasycenia koloru czerwonego i zwiększeniu wartości nasycenia koloru niebieskiego. Opisuje to funkcja:

Efekt zastosowania takiego przekształcenia jest następujący:

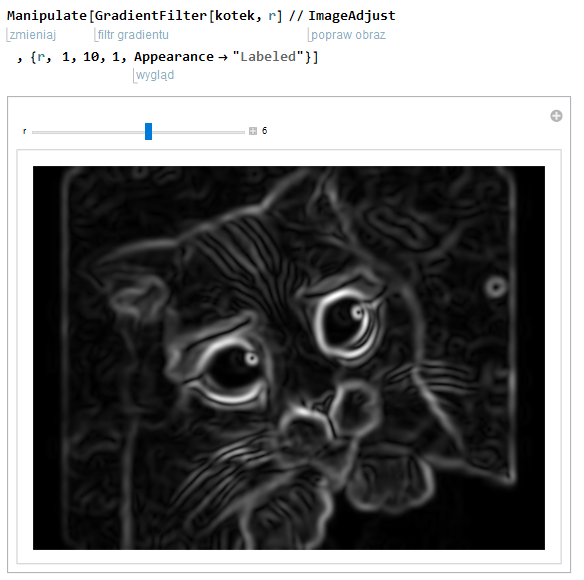


**4. Prezentacja domyślnych filtrów wbudowanych w Mathematica**

1. **Filtr gradientu**

Filtr gradientu służy do wykrywania krawędzie na obrazie. Wartości pikseli są obliczane jako norma euklidesowa gradientu g w pozycji piksela, aproksymowaną przy użyciu dyskretnych pochodnych Gaussa w każdym wymiarze:

Funkcja ta przyjmuje jeden argument r, który jest promieniem próbki. Została przygotowana instrukcja Manipulate sterująca wartością tego parametru.

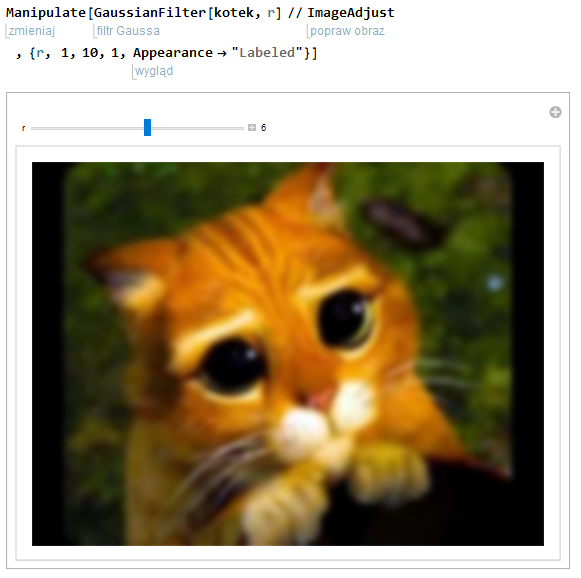


1. **Filtr Gaussa**

Filtr Gaussa służy to rozmywania obrazów. Przyjmuje jeden parametr r, który odpowiada za promień próbki. Wartości pikseli obliczane są ze wzoru:

,

gdzie odchylenie standardowe

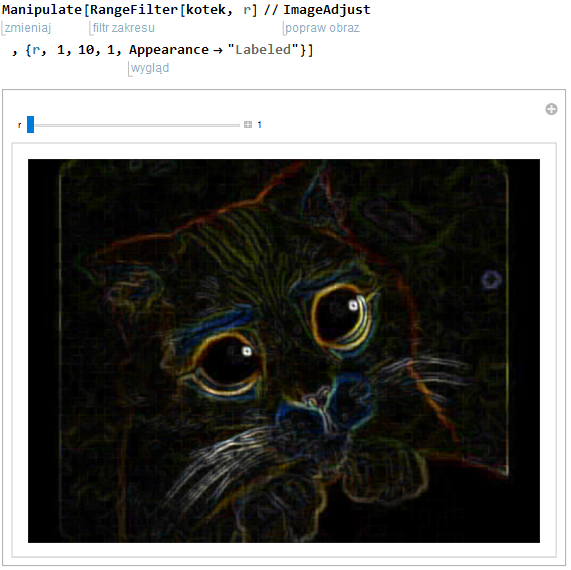


1. **Filtr zakresu**

Filtr zakresu służy do wykrywania lokalnych skoków w danych, gdzie wielkość lokalnego sąsiedztwa zależy od wartości promienia .

Wartości obliczane są jako różnica między maksymalną i minimalną wartością w danym sąsiedztwie.

W celu prezentacji została przygotowana instrukcja Manipulate sterująca parametrem :

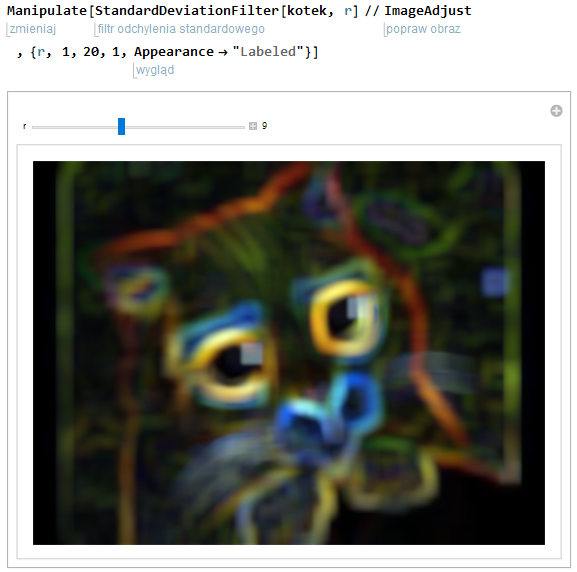


1. **Filtr odchylenia standardowego**

Filtr odchylenia standardowego służy do wykrywania lokalnego rozproszenia danych w danym otoczeniu wyznaczanym przez promień .

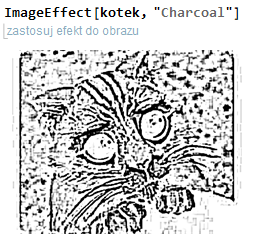
Wartość danego piksela jest obliczana jako odchylenie standardowe punktów w wyznaczonym sąsiedztwie.

W celu prezentacji została przygotowana instrukcja Manipulate sterująca parametrem :



**5. Prezentacja efektów graficznych dostępnych w Mathematica**

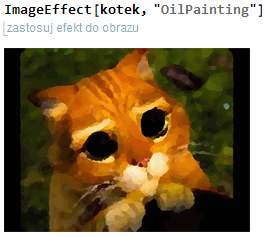
1. **Efekt „Charcoal” – szkic**



1. **Efekt „Embossing” – wytłoczenie**



1. **Efekt „OilPainting” – farba olejna**



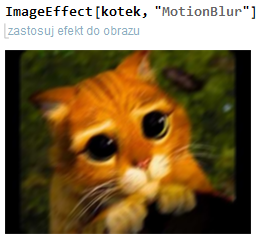
1. **Efekt „Posterization” – posteryzacja**



1. **Efekt „Solarization” – solaryzacja**

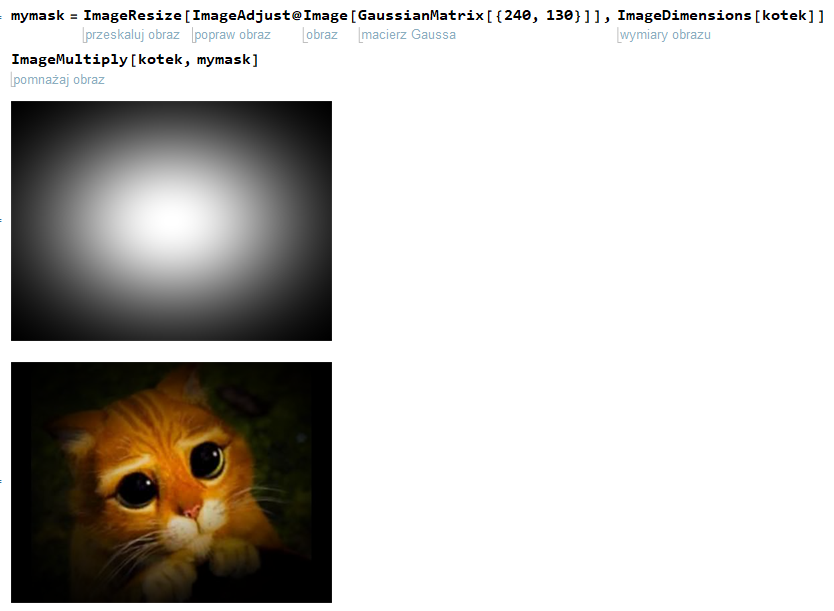


1. **Efekt „MotionBlur” – rozmycie ruchu**



**6. Zastosowanie masek w przetwarzaniu grafik w Mathematica**

Funkcjonalność przetwarzania obrazu w Mathematice umożliwia stosowanie masek, na przykład w celu tworzenia efektów winietowych. W tym celu należy wygenerować maskę. Można wybrać na przykład proste cieniowanie w rogach korzystając z maski Gaussa. Następnie wygenerowaną maskę należy przemnożyć przez wartości pikseli obrazu.



Można również zastosować kolorowe maski:



**7. Progowanie (binaryzacja) obrazu**

Algorytm binaryzacji polega na konwersji obrazu do skali szarości, a następnie tworzeniu obrazu, w którym każdy piksel ma wartość 0 lub 1. Przypisywanie wartości binarnych może odbywać się za pomocą analizy różnych danych statystycznych takich jak na przykład entropia, średnia czy mediana.

W celu prezentacji algorytmu została przygotowana instrukcja Manipulate sterująca wartością progową oraz metodą statystyczną użytą do wyznaczenia wartości binarnych:



**8. Metody poprawiania niedoświetlonych zdjęć**

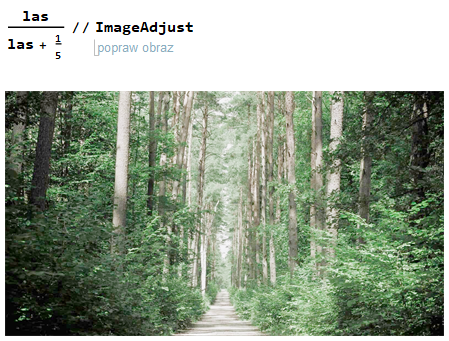
1. **Przemnożenie wartości pikseli**

Najprostszym sposobem na poprawę niedoświetlonej grafiki jest przemnożenie wartości pikseli.



1. **Nieliniowe przekształcenie pikseli**

Przekształcając wartości pikseli nieliniowo można korygować skalę jasności obrazu



1. **Zastosowanie operatora logarytmicznego**



1. **Zastosowanie operatora logarytmicznego dla każdego kanału**



1. **Zastosowanie średniej różnych poziomów jasności**



***Literatura:***

- <https://reference.wolfram.com/language/example/ImageDenoising>  
- <https://www.wolfram.com/language/12/new-in-image-processing/convert-images-to-vector-graphics.html?product=language>  
- <https://reference.wolfram.com/language/guide/ImageFilteringAndNeighborhoodProcessing.html>  
- <https://reference.wolfram.com/language/ref/ImageEffect.html>  
- <https://reference.wolfram.com/language/ref/Binarize.html>  
- <https://www.wolfram.com/language/12/new-in-image-processing/arithmetic-for-fast-algorithm-prototyping.html?product=language>

***Załącznik:***

* Plik z programem (MM\_Bitner\_Broczkowski\_Kwasniok\_projekt.nb)