Wprowadzenie do grafiki maszynowej(LAB-4)

Paweł Idzikowski, nr.albumu: 141080

21 listopada 2018

Spis treści

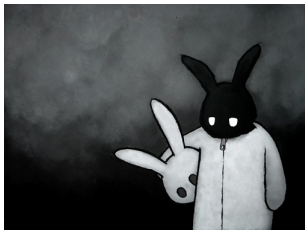
1 Informacje

Na wstępie chciałbym poinformować, że zdecydowałem się wykonać zadanie1 na dwóch fotografiach. Chciałem otrzymać lepszą perspektywę(to raz) i nie mogłem się zdecydować, którą fotografię wybrać(to dwa - raczej przeważający powód). Jak wpadłem(przy okazji) na zdjęcie psa w skafandrze pilota X-Wing to nie mogłem się powstrzymać żeby na nim nie przetestować tego co mamy sprawdzić i uwzględnić w sprawozdaniu. Przy zdjęciu komandosa - wiem, jest jeszcze jeden z tyłu jednak łatwiej mi posługiwać się liczbą pojedynczą więc będę o tym obrazku mówił jako 'komandos'. Zdjęcia, które użyłem wraz z linkami:

(a) komandos

(b) pies pilot X-Wing'a

(c) komiksowe



(d) rysunkowe/pastelowe

(e) bardzo dokładne, z gry

(f) mało wyraźne, z kamery sklepowej

2 Zadanie 1

Treść zadania:
Wczytaj dowolne zdjęcie/obraz (nie z testowych).
A) Stwórz kopię obrazu w wersji czarno-białej
B) Stwórz kopię obrazu z wymieszanymi kolorami
C) Stwórz kopię obrazu po korekcji gamma
D) W jednym oknie graficznym wyświetl cztery histogramy (oryginalnego obrazu oraz każdej kopii). Czy różnia się od siebie? Dlaczego?

Rozwiązanie:

```
// Zadanie 1 - obrobka obrazow
// Obrobka zostanie wykonana dla dwoch obrazow aby uzyskac wiecej informacji

// wczytaj obrazki
komandos=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'komandos.jpeg'));
pies=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'pies.jpg'));


// ***********************************************************
// A. obrazek w wersji czarno-bialej

// uzyskanie obrazu binarnego funkcja: im2bw
// 0.5 - stosunek bieli do czerni

// dla komandosa 0.2 biel-czern najlepiej odwzorowuje obraz
// zwrocmy uwage ze w przypadku komandosa wiecej czerni
// sprawia ze szczegoly staja sie nieczytelne chociaz i tak
// przy tej wartosci gdyby komus pokazac ten obrazek to watpie
// zeby stwierdzil ze jest tam wiecej niz jeden komandos
// bw=im2bw(komandos,0.2);

// dla psa 0.5 biel-czern najlepiej odwzorowuje obraz
// bw=im2bw(pies,0.5);
// imshow(bw);

// ***********************************************************
// obrazki w wersji mieszania kolorami
kolorowy_komandos=komandos;
kolorowy_pies=pies;

// przemnazanie macierzy dla komandosa w malych wartosciach
// daje efekt jakby doslownie niszczyl nam sie monitor...
// z kolei przemnozenie macierzy przez wysoka wartosc np. 15
// powoduje ze... nic sie nie da odczytac - kontury sa zniszczone.
// imshow(kolorowy_komandos*13);

// przemnazanie macierzy dla psa za pomoca malej liczby np. 3
// powiedzialbym ze naklada filtr w stylu retro ale niektore
// miejsca rowniez sprawiaja wrazenie jakby monitor sie niszczyl..
// z kolei mnozenie macierzy wysoka liczba np. 13 w przeciwienistwie
// do komandosa nie zaburza tak mocno konturow obrazka. W przypadku
// psa gdzie uzyte barwy sa jasniejsze(nie goruje czern, zielen itp.)
// przemnozenie macierzy nie zaburza tak mocno obrazku
// imshow(kolorowy_pies*13);

// zwrocmy jednak uwage na przemnozenie macierzy przez liczbe ujemna bo
// tu otrzymujemy rowniez ciekawy efekt - odwracamy kolory i mamy
// przeswietlenie
// imshow(kolorowy_pies*-1);
// imshow(kolorowy_komandos*-1);

// dodawanie zachowuje sie tak samo jak przemnazanie - zaburza kolory
// imshow(kolorowy_pies+10);
// imshow(kolorowy_komandos+10);

// ciekawa operacja jest dzielenie..
// w przypadku naszych bohaterow obrazkow otrzymuja oni efekt
// podobny do "Winiety" gdzie mamy takie przyciemnienie na
// krawedziach. Przy duzej wartosci np. 50
// imshow(kolorowy_pies/3);
// imshow(kolorowy_komandos/3);

// podniesienie wszystkich elementow macierzy do potegi np. 3 rowniez powoduje
// powstanie szumu. Dla wartosci 3 pies wytrzymuje test natomiast
```

```
// komandos poddaje sie i go nie widac nic a nic
// imshow(kolorowy_komandos.^3);


// ********************************************************
// KORYGOWANIE GAMMA OBRAZKOW
// aby sprawdzic dla drugiego obrazka zmien nazwy z pies na
// komandos
function y=ga(x)
    // korekcja gamma funkcja pierwiastek
    y=sqrt(x);
endfunction

im=rgb2gray(komandos);
IM=im2double(im);
IM2=ga(IM);
// imshow(IM2);


// ********************************************************
// Histogramy
// ****************************

// oryginal
// im = pies*-1;

// subplot(2,2,1); imhist(im(:,:,1), 40, 'red'); xtitle("Kanal R");
// subplot(2,2,2); imhist(im(:,:,2), 40, 'green'); xtitle("Kanal G");
// subplot(2,2,3); imhist(im(:,:,3), 40, 'blue'); xtitle("Kanal B");
// subplot(2,2,4); imshow(im); xtitle("Zrodlo:");

bw=im2bw(komandos,0.2);

// ****************************
// czarno-bialy
subplot(2,2,1); imhist(bw, 2, 'gray'); xtitle("Czarno-bialy");
subplot(2,2,2); imshow(bw); xtitle("Zrodlo:");
// ****************************
// po korekcji gamma
subplot(2,2,3); imhist(IM2(:,:,:),[], 1, 'gray'); xtitle("Po korekcji gamma");
subplot(2,2,4); imshow(IM2); xtitle("Zrodlo:");

// ****************************
// wymieszane kolory(wezmy tu przemnozenie przez -1)
// tu trzeba znowu rozlozyc RGB na 3

// im = pies*-1;

//subplot(2,2,1); imhist(im(:,:,1), 40, 'red'); xtitle("Kanal R");
//subplot(2,2,2); imhist(im(:,:,2), 40, 'green'); xtitle("Kanal G");
//subplot(2,2,3); imhist(im(:,:,3), 40, 'blue'); xtitle("Kanal B");
//subplot(2,2,4); imshow(im); xtitle("Zrodlo:");
```

2.1 Wnioski

2.1.1 Czarno-białe obrazy

(g) Komandos

(h) Pies

(i) Komandos po liftingu

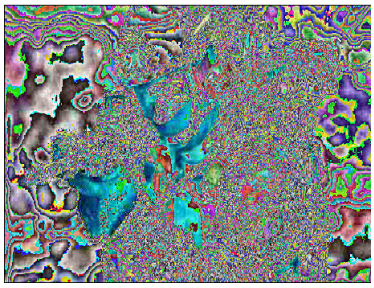
(j) Pies po liftingu

Pierwsza rzecz na jaką zwróciłem uwagę jest taka, że przerobiony obrazek z psem(j) jest o wiele bardziej dokładny niż ten z komandosem(i)(u psa w oryginalnym obrazku(h) dominują jasne barwy, stąd myślę tak dobre odwzorowanie fotografii). W przypadku komandosa, poziom czerń/biel musi być naprawdę mały(pokazane zdjęcie po liftingu ma wartość 0,2 - (i)) ponieważ czerń bardzo szybko przejmuje obraz. W obrazku z komandosem(g) dominują barwy ciemne (ciemna zieleń, czerń itd..). Przy większych wartościach w metodzie im2bw, komandos staje się coraz słabiej widoczny. Zwróćmy też uwagę na to, że w przypadku przeróbki zdjęcia z komandosami, komandos, który znajduje się na drugim planie jest niemożliwy do odczytania!

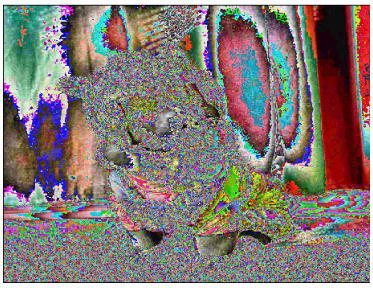
2.1.2 Operacje na obrazach - mieszanie kolorów

(k) Komandos - oryginał

(l) Pies - oryginał



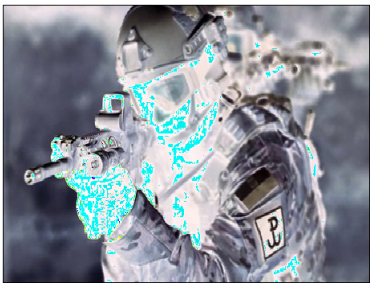
(m) Komandos po operacji



(n) Pies po operacji

Rysunek 1: Przemnożenie macierzy przez wartość 13

Mnożenie macierzy obrazów powoduje w głównej mierze powstawanie dziwnych plam, które wyglądają tak jakby monitor dogorywał... Szczególnie widoczny jest taki efekt gdy użyjemy małej liczby np. 3. Zobrazowałem jednak wartość 13 gdzie dostajemy efekt zbliżony do szumu. W tej potyczce pies - komandos, wygrał zdecydowanie pies. Na przerobionych zdjęciach komandos jest nie do odczytania natomiast kontury psa nie zostały tak mocno zaburzone i jesteśmy w stanie coś odczytać(n). Myślę, że to zasługa tego iż pies jest obrazkiem z jaśniejszymi barwami. Warto też w tym miejscu wspomnieć o potęgowaniu ponieważ ta operacja ma taki sam wpływ na obrazy. Przy użyciu potęgi np. 3 obraz psa jest do odczytania, natomiast obraz z komandosem... nie.



(a) Komandos po operacji

(b) Pies po operacji

Rysunek 2: Przemnożenie macierzy przez wartość -1

Mnożąc macierze przez wartość -1 dostaliśmy efekt odwrócenia kolorów. Uzyskaliśmy coś w stylu prześwietlenia. W przypadku psa możemy zaobserwować ciekawe zachowanie zmiany kolorów. Czerń ze skafandra na łapkach została przetworzona na kolor biały a ta z zasłony po prawej stronie, zmieniła kolor na mieszankę różu z czerwienią. Na obrazku z komandosem czerń o wiele lepiej została zaadaptowana. Większość miejsc otrzymała jednolity kolor jasno-niebieski. Takie zachowania mogą dawać sygnał, że tym razem to obrazek ciemniejszy(komandos) góruje nad jaśniejszym(psem) i lepiej znosi odwracanie kolorów.

(a) Komandos po operacji

(b) Pies po operacji

Rysunek 3: Dodanie do każdego elementu macierzy wartość 10

Dodawanie zachowuje się tak samo jak przemnażanie. Zaburza kolory na obrazkach jednak nie jest to tak 'inwazyjne' jak mnożenie. W dodawaniu możemy na obrazku z komandosem zaobserwować wcześniej wspominane plamki jakby monitor nam się psuł. Odnośnie przeróbki psa - powiedzmy, że wygląda trochę jak nieudana próba podmiany barwy skafandra na zielony.

(a) Komandos po operacji

(b) Pies po operacji

Rysunek 4: Podzielelnie każdego elementu macierzy przez wartość 3

Dzielenie jest bardzo ciekawą operacją! W przypadku naszych obrazków, po podzieleniu ich macierzy przez 3, otrzymujemy efekt podobny do popularnego filtru "Winieta"(tak przynajmniej się nazywa w edytorze pixlr), gdzie obraz zostaje przyciemniony w mniejszym bądź większym stopniu. Przy dużej wartości oczywiście ciężko cokolwiek odczytać, jednak jeżeli byśmy taki test wykonali, to z pewnością obrazek z psiakiem by go wytrzymał dłużej bo jest obrazkiem z jaśniejszymi barwami...

2.1.3 Korekcja gamma obrazków

(a) Komandos

(b) Pies

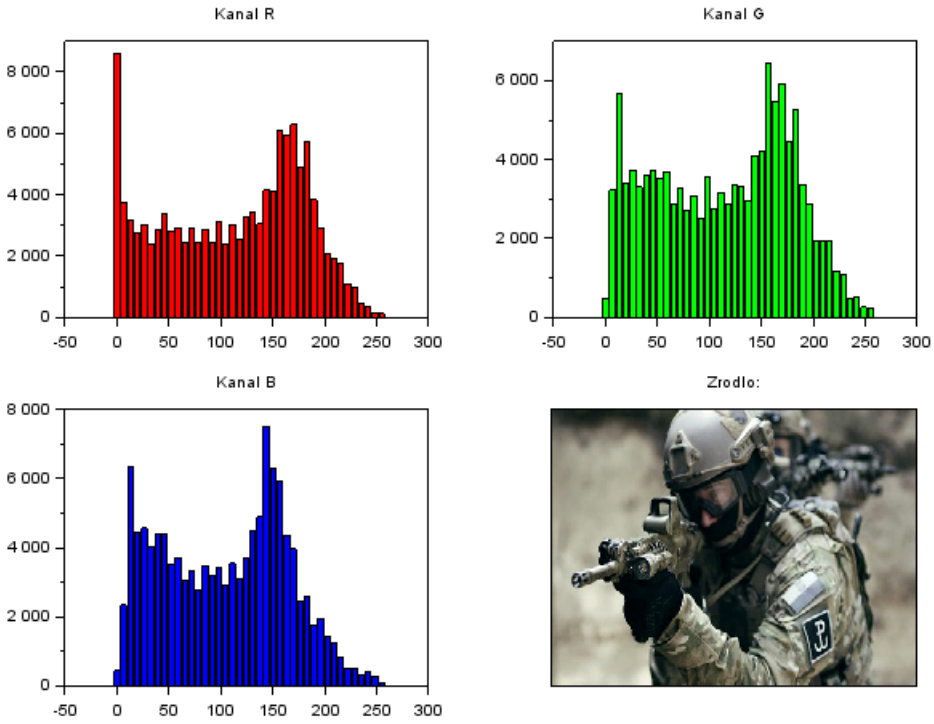
(c) Komandos

(d) Pies

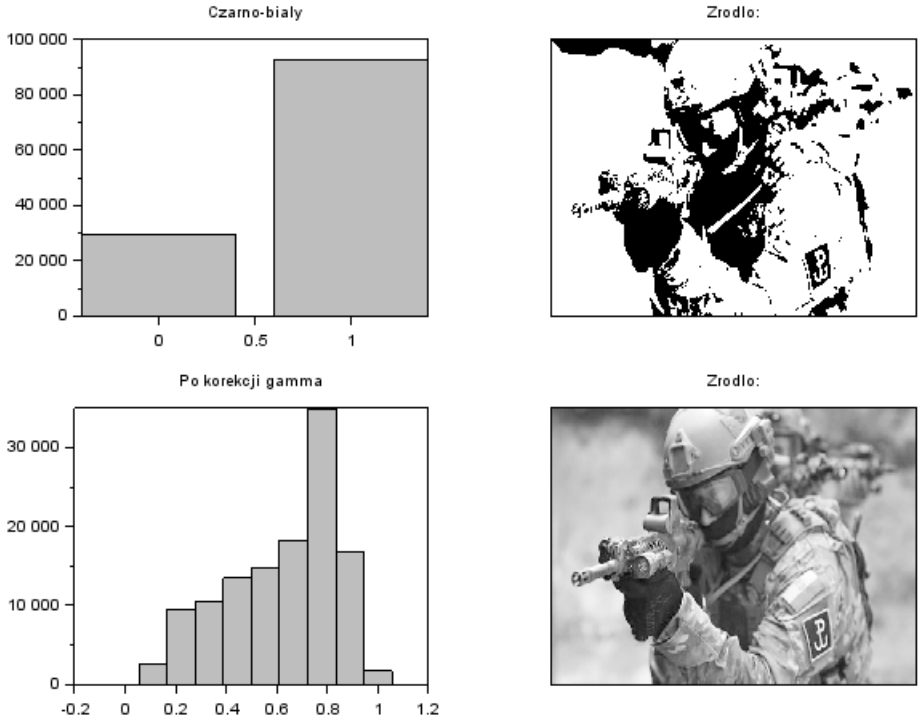
Korekcję musieliśmy wykonać na jednym kanale, stąd obrazki zostały skonwertowane do odcieni szarości. Korekcja gamma odbyła się za pomocą funkcji pierwiastek(gdyby zamienić sqrt na cos to można uzyskać ciekawy efekt). Na zdjęciach, które nie są 'wielką pikselozą' albo rozmyte(ang. blurred) trudno zauważyć żeby korekta usunęła jakiś szum. Tutaj widzimy zmianę bardziej w postaci poziomu jasności.

Jeżeli jednak przyjrzymy się na siłę w oryginał zdjęcia z psem(b) to po korekcji(d) rzuca się w oczy mniej dokładny dywan - nie widzimy tak dobrze kropek jak w przypadku oryginału.

2.1.4 Histogramy - Komandos

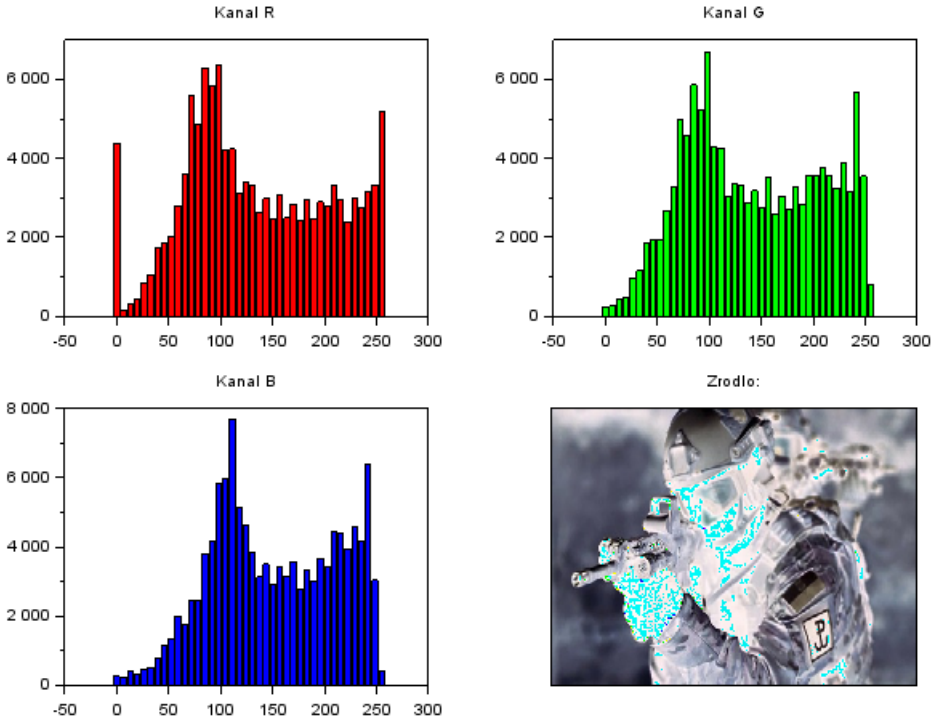


Nasz komandos jest wykonany w ciemnych barwach. Przeważają szczególnie kolory czerń/zieleń i ich odmiany co zgadza się z tym co prezentują histogramy. Jeżeli przyjrzymy się dokładniej na wykresy słupkowe to zobaczymy, że największa liczba pikseli ma albo wartość z zakresu 0-20 w przypadku wszystkich kanałów albo wartość z zakresu: (160-180, dla kanału R), (160-190, dla kanału G), (150-170, dla kanału B).



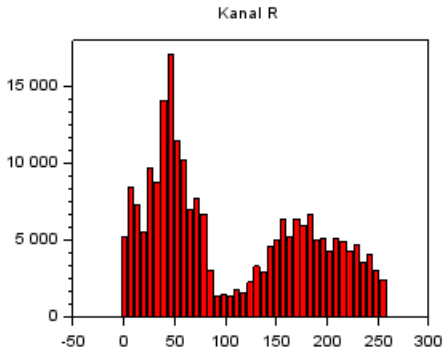
(pierwszy wiersz figury)Komandos poddany obróbce czarno-białej posiada tylko dwa kolory co rzeczywiście ukazuje histogram 'Czarno-biały'(jeżeli odnieść się do scilabowego helpa to: jeżeli obrazek jest typu bool(a rzeczywiście jest patrząc na okienko z wartościami) - wtedy domyślnie pokaże 2 słupki). Przy wykorzystaniu funkcji im2bw ustawiłem wartość konwersji obrazku z komandosami na 0.2. Przy większych wartościach trudno było doszukać się szczegółów na pierwszym planie. Prawie 30 tysięcy pikseli jest barwy czarnej natomiast ponad 90 tysięcy jest barwy białej. Możemy zatem powiedzieć, że zgadza się to z rozmiarami zdjęcia, które wynoszą $450x272 = 122400$. Jak zerkniemy sobie na źródłowe zdjęcie to rzeczywiście gołym okiem widać, że białego koloru jest zdecydowanie więcej. Zauważmy, że komandos, który był na drugim planie jest kompletnie niewidoczny.

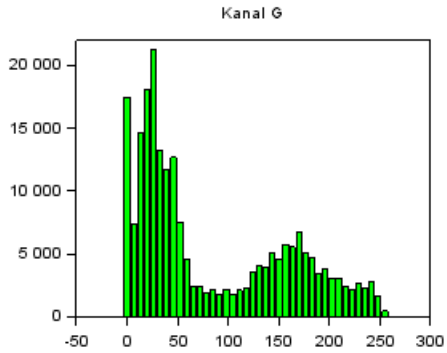
(wiersz drugi figury)W przypadku komandosa po korekcji gamma, histogram już nie jest taki 'ubogi'. Otrzymujemy informacje o tym jaka liczba pikseli(oś Y) posiada taki sam poziom koloru(oś X). Poziom koloru czarnego dla ponad 30 tysięcy pikseli wynosi od 0,70 do 0,81 co w przełożeniu na wartość barwy daje od 178,50 do 206,55 czyli odmiany koloru szarego! W tym przypadku trudno doszukiwać się poprawy wyraźistości skoro nie występuje 'pikseloza' na fotografii. W tym przypadku zadziałanie korekcji gamma było widoczne bardziej w formie rozjaśnienia obrazka.

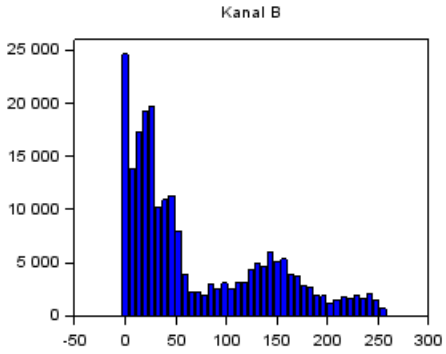


Odwracając kolory za pomocą typowego przemnażania macierzy przez wartość -1, otrzymujemy odwrócone histogramy(w porównaniu do histogramów zdjęcia oryginalnego). Patrząc na każdy kanał możemy powiedzieć, że większość pikseli w każdym kanale zachowuje się podobnie(za wyjątkiem tych ponad 4 tysięcy pikseli w kanale R, które oznaczają kolor czarny).

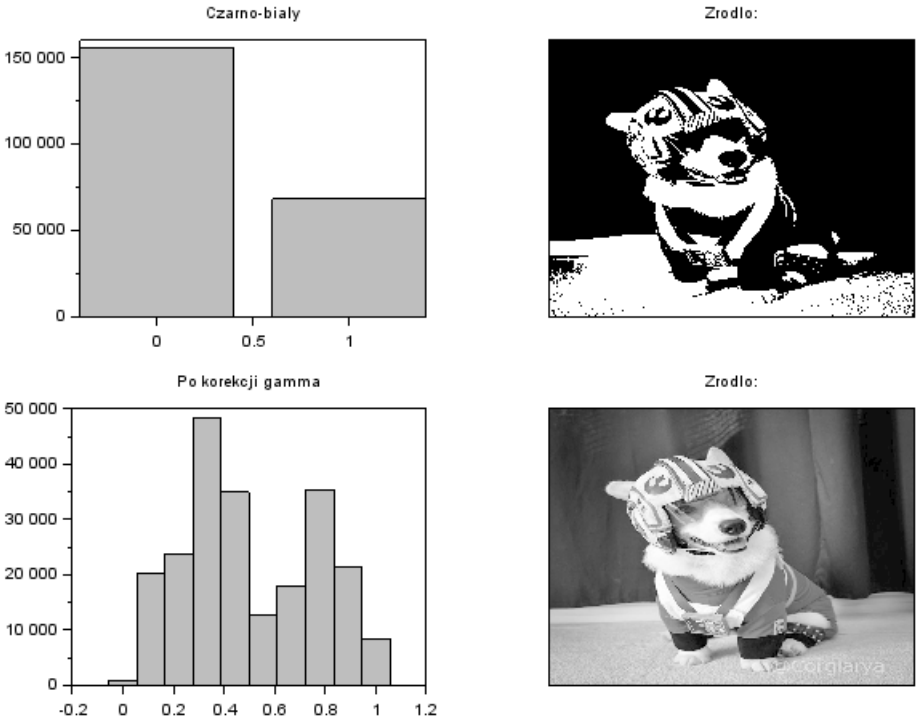
2.1.5 Histogramy - Pies





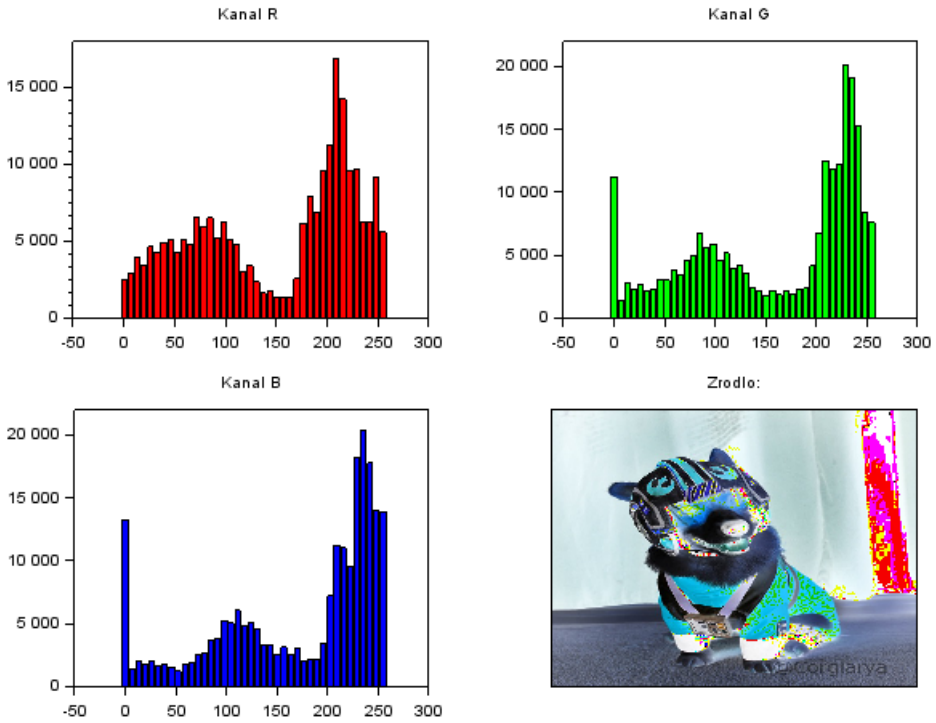


Histogramy myślę, że nie są tutaj zaskoczeniem. W oryginalnej wersji dominują kolory czerwień/pomarańcz/biel. Także tło jest koloru czerwonego w głównej mierze. O dziwo najwięcej pikseli jest w kanałach G i B(prawdopodobnie dlatego, że zostały wykorzystane do wymieszania barw w celu uzyskania np. pomarańczowego koloru).



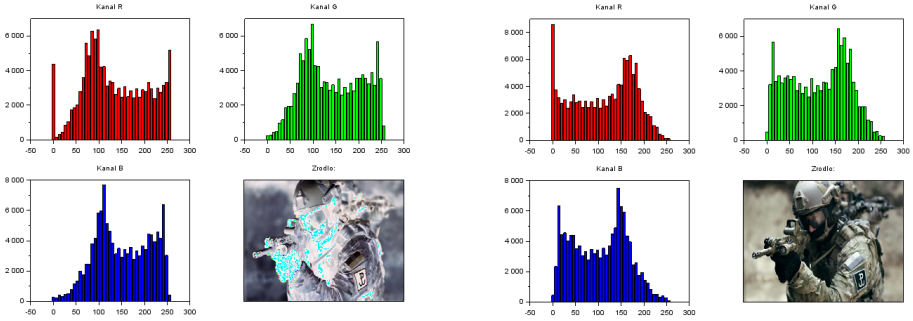
(pierwszy wiersz figury)W histogramie dotyczącym czarno-białego obrazka z psem mamy znowu zasadę - dwa słupki - albo czarny, albo biały kolor. Ustawiłem w funkcji im2bw wartość 'thresh' na 0.5 - czyli jeżeli wartość luminancji piksela jest większa niż 0.5 to piksel będzie biały. Pozostałe piksele będą koloru czarnego. Przeważają piksele koloru czarnego(jest ich ponad 150 tysięcy). Patrząc na źródłowy obrazek dla którego został wykonany histogram możemy stwierdzić, że rzeczywiście barwa czarna zdecydowanie dominuje nad białą.

(drugi wiersz figury)Histogram obrazka po korekcji gamma po raz kolejny ma rozrzucone wartości. Mamy tu różne poziomy odcieni, z czego dominującą jest odcień w punkcie(X)=0.3. Ma ją prawie 50000 pikseli. 0.3 odpowiada wartości 76,5. Korekcja gamma spowodowała(w porównaniu do oryginalnego zdjęcia), że dywan sprawia trochę wrażenie lekko 'rozmytego'.



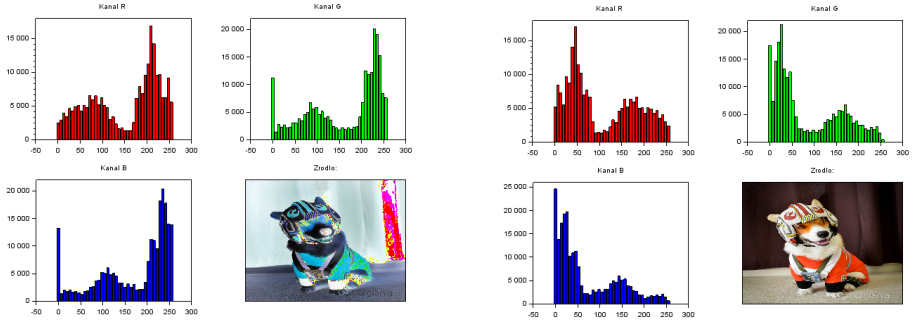
Zdjęcie z psem poddaliśmy takiemu samemu zabiegowi - przemnożeniu macierzy przez wartość -1. Porównując ze sobą histogramy obrazków: oryginalnego i odwróconego zauważymy, że rzeczywiście 'słupki' zostały odwrócone w osi X.

2.1.6 Ogólne podsumowanie histogramów



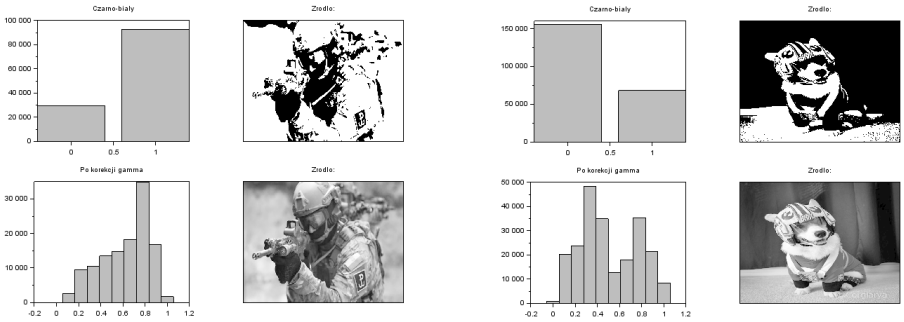
(e) Komandos z odwróconymi kolorami (f) Komandos oryginał

Rysunek 5: Komandos



(a) Pies z odwróconymi kolorami (b) Pies oryginał

Rysunek 6: Pies pilot



(a) Komandos (b) Pies pilot

Rysunek 7: Histogramy dla obrazków czarno-białych i tych po korekcji gamma

- histogramy odnośnie obrazków oryginalnych i tych z mieszanymi kolorami są oczywiście inne. Odwracając kolory uzyskaliśmy również odwrócenie kanałów co pokazują Rysunek 5 i Rysunek 6. Zatem zmiana kolorów obrazka rzeczywiście wpływa na kanały RGB.

- korekcja gamma zdjęcia komandosa wypadła o wiele lepiej niż psa. Korekcja gamma w przypadku zdjęć ciemniejszych(mających ciemne barwy) daje lepsze efekty.

- histogramy obrazków zmienionych na czarno-białe dają jasny przekaz. Wartość 1 oznacza kolor biały, wartość 0 oznacza kolor czarny. W obrazku z komandosem przeważa kolor biały a zdjęciu z psem kolor czarny, co potwierdzają histogramy. Konwersja na obrazek czarno-biały wypada lepiej w przypadku zdjęć jaśniejszych. Ciemniejsze zdjęcia tracą przy dużych wartościach 'thresh' na szczegółach jak np. na obrazku z komandosami nie widzimy operatora GROM, który jest w drugim planie..

- korekcja gamma w przypadku (b) sprawiła, że dywan jest mniej wyrazisty.

3 Zadanie 2

Treść zadania:
Sprawdź jak dziala wykrywanie krawedzi - wybierz do tego 3-4 zdjecia/obrazy.
Od czego twoim zdaniem zalezy skutecznosc wykrywania?

Rozwiązanie:

```
// Zadanie 2 - testowanie wykrywania krawedzi

// wczytaj obrazki
wyg1=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'wygladzanie1.jpg'));
wyg2=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'wygladzanie2.jpg'));
wyg3=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'wygladzanie3.jpg'));
wyg4=imread(fullpath("C:/Users/Idzik/Desktop/Grafika maszynowa/SPR04/cw_grafika/" + 'wygladzanie4.jpg'));

wyg1 = rgb2gray(wyg1);
wyg2 = rgb2gray(wyg2);
wyg3 = rgb2gray(wyg3);
wyg4 = rgb2gray(wyg4);

// ***************************************************
// test wykrywania krawedzi 1 (metoda sobel)

// ciezko.. (zle)
E = edge(wyg1, 'sobel', 0.4);
imshow(E);

// przy wartosci 0.4/0.5 dosc dobrze (ok)
// 0.5 nadaje czarne tlo
// 0.4 zostawia tlo biale
E = edge(wyg2, 'sobel', 0.5);

// na wartosci 0.55 idealnie krawedzie (wygrana)
E = edge(wyg3, 'sobel', 0.55);

// trudno cokolwiek wycisnac ze zdjecia.. (zle)
// wyglada jak kolonia bakterii czy cos..
E = edge(wyg4, 'sobel', 0.5);
// imshow(E);

// ***************************************************
// test wykrywania krawedzi 2 (metoda prewitt)

// 0.65 jest najbardziej OK - (wygrana)
E = edge(wyg1, 'prewitt', 0.65);

// 0.1 - idealnie (wygrana)
E = edge(wyg2, 'prewitt', 0.1);

// 0.1 - idealnie (wygrana)
E = edge(wyg3, 'prewitt', 0.1);

// 0.15 - OK, chociaz takie wykrywanie
// czlowieka dla komputera byloby utrudnione?
E = edge(wyg4, 'prewitt', 0.15);
// imshow(E);


// ***************************************************
// test wykrywania krawedzi 3 (metoda fftderiv)

// 0.65 - ok
E = edge(wyg1, 'fftderiv', 0.65);
// 0.1 - ok
E = edge(wyg2, 'fftderiv', 0.1);
// 0.1 - ok
E = edge(wyg3, 'fftderiv', 0.1);
// 0.1 - ok
E = edge(wyg4, 'fftderiv', 0.15);
// imshow(E);

// WNIOSEK
// metoda fftderiv daje takie same efekty jak prewitt
// dla naszych obrazkow


// ***************************************************
// test wykrywania krawedzi 4 (metoda canny)

// 0.4 - ok
E = edge(wyg1, 'canny', 0.4);
```

```
// 0.9 - ok, przy 0.1 dorzuca dziwne tlo
E = edge(wyg2, 'canny', 0.9);

// 1 - ok
E = edge(wyg3, 'canny', 1);

// 0.6 - ale slabo, moglaby budzic kontrowersje
E = edge(wyg4, 'canny', 0.6);
// imshow(E);
```

3.1 Wnioski

3.1.1 Oryginalne zdjecia

Każde zdjęcie podpiszę jednym słowem aby łatwiej było się do nich odnosić.

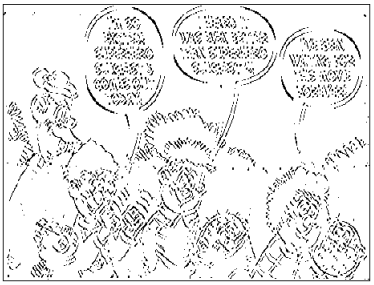
(a) Komiks

(b) Pastelowe

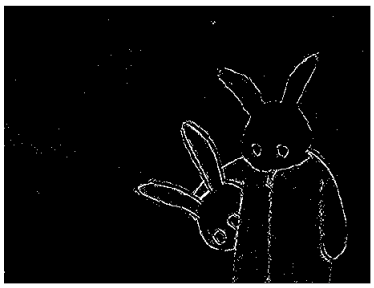
(c) Dokładne

(d) Niewyraźne

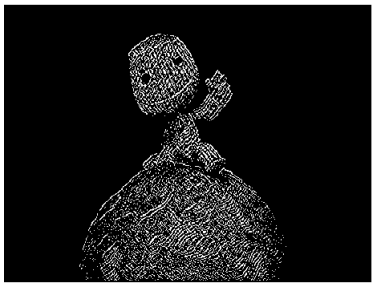
3.1.2 Test 1 - metoda sobel



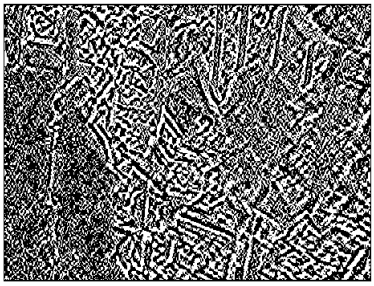
(e) Komiks



(f) Pastelowe

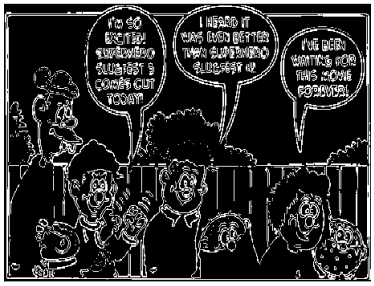


(g) Dokładne

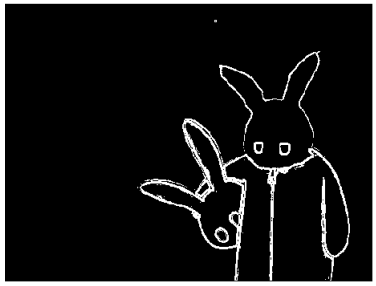


(h) Niewyraźne

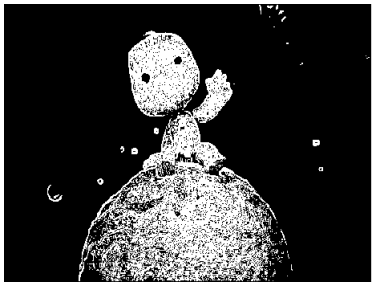
3.1.3 Test 2 - metoda prewitt



(i) Komiks



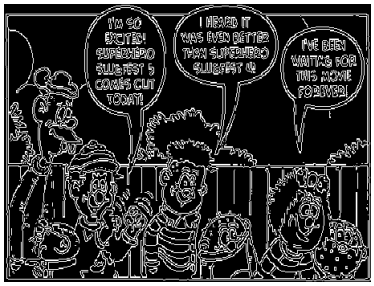
(j) Pastelowe



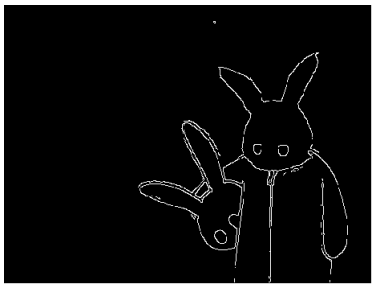
(k) Dokładne

(l) Niewyraźne

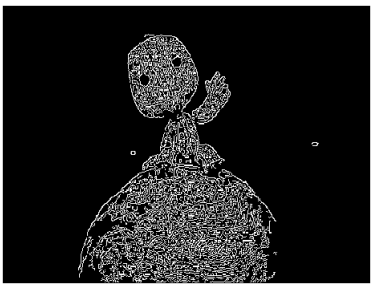
3.1.4 Test 3 - metoda canny



(m) Komiks



(n) Pastelowe



(o) Dokładne

(p) Niewyraźne

3.1.5 Podsumowanie

Obrazek/Metoda	Metoda sobel	Metoda prewitt	Metoda canny
Komiks	X	V	V
Pastelowe	V	V	V
Dokładne	V	V	V
Niewyraźne	X	+/-	+/-

Oznaczenia:
X „fatalne"
V „ok"
+/- „zależy"

Rysunek 8: Ocena wykrywania krawędzi według danej metody na danym zdjęciu

- Metoda sobel
  Wykrywanie krawędzi metodą sobel nie poradziło sobie z obrazkiem, który jest wycinkiem z komiksu i zdjęciem z monitoringu. O ile w przypadku komiksu otrzymujemy mało precyzyjny zarys to w przypadku zdjęcia niewyraźne(h) otrzymujemy.. bakterie pod mikroskopem.. Mimo wysiłku nie udało się wycisnąć lepszych krawędzi na tych dwóch zdjęciach. Sobel dobrze poradził sobie z odwzorowaniem postaci wykonanej pastelowo(f) i sackboy'a(szmacianki) z Little Big Planet(g).

- Metoda prewitt
  Prewitt poradził sobie na piątkę ze zdjęciami (i), (j), (k). Pozostaje kwestia zdjęcia(l). Tu sprawa jest skomplikowana. Niby udało się dosyć dobrze odwzorować elementy ze zdjęcia jednak wydaję mi się, że ciężko by było rozpoznać gdzie na takiej fotografii są ludzie(zaznaczeni czerwonymi strzałkami). Chciałbym tutaj zauważyć, że testowałem metodę fftderiv jednak efekty nie różniły się od tych, które oferuje metoda prewitt.

- Metoda canny
  Canny poradziła sobie podobnie jak Prewitt z pierwszymi trzema zdjęciami - (m), (n), (o). Dodatkowego komentarza wymaga zdjęcie z kamery sklepowej. Wydaje mi się, że Canny o wiele lepiej odwzorowała ludzkie postaci chociaż jest jedno ale. Miejsce, które zaznaczyłem znakiem zapytania mogłoby być interpretowane jako człowiek co powoduje, że raczej taka metoda nie zawitałaby w sprawdzaniu klatek video gdzie np. uciekł złodziej?

Patrząc na wyniki zaprezentowane w tabeli mogę stwierdzić, że metody Prewitt i Canny okazały się bardziej uniwersalne niż Sobel. W przypadku zdjęcia z kamery ze sklepu między tymi dwiema wyróżnionymi metodami - kwestia sporna. Nie chciałem skreślać tych dwóch metod, dlatego wprowadziłem oznaczenie +/-. Zależy od tego czy jest np. sens szukać ludzi na takiej klatce? Nie znam się to trudno mi się wypowiedzieć ale jeżeli odpowiedź brzmi nie to oczywiście takie wykrywanie krawędzi jest bezużyteczne i wtedy wniosek będzie taki, że żadna z wymienionych metod nie radzi sobie z wykrywaniem krawędzi z kamer sklepowych.