

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Automatyka i Robotyka (AIR)  
SPECJALNOŚĆ: Technologie Informacyjne w Automatyce (ART)

**PRACA DYPLOMOWA  
INŻYNIERSKA**

Rozpoznawanie znaków tekstowych drogowych w  
systemie osób niewidomych

The text road signs recognition in the system of  
blind people

AUTOR:  
Karolina Raczyńska

PROWADZĄCY PRACĘ:  
dr inż. Łukasz Jeleń

OCENA PRACY:

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Cel pracy . . . . .	2
1.2	Zakres pracy . . . . .	2
<b>2</b>	<b>Przetwarzanie obrazu</b>	<b>3</b>
2.1	Przetwarzanie do skali szarości . . . . .	4
2.2	Binaryzacja . . . . .	5
2.2.1	Metoda Otsu . . . . .	5
2.3	Wykrywanie krawędzi . . . . .	7
2.3.1	Detektor Canny . . . . .	8
<b>3</b>	<b>Architektura systemu</b>	<b>10</b>
<b>4</b>	<b>Implementacja</b>	<b>11</b>
<b>5</b>	<b>Działanie aplikacji i testy</b>	<b>12</b>
5.1	Wymagania wstępne . . . . .	12
5.2	Widok główny . . . . .	12
5.3	Widok docelowy . . . . .	13
5.4	Widok testowy . . . . .	13

# Rozdział 1

## Wstęp

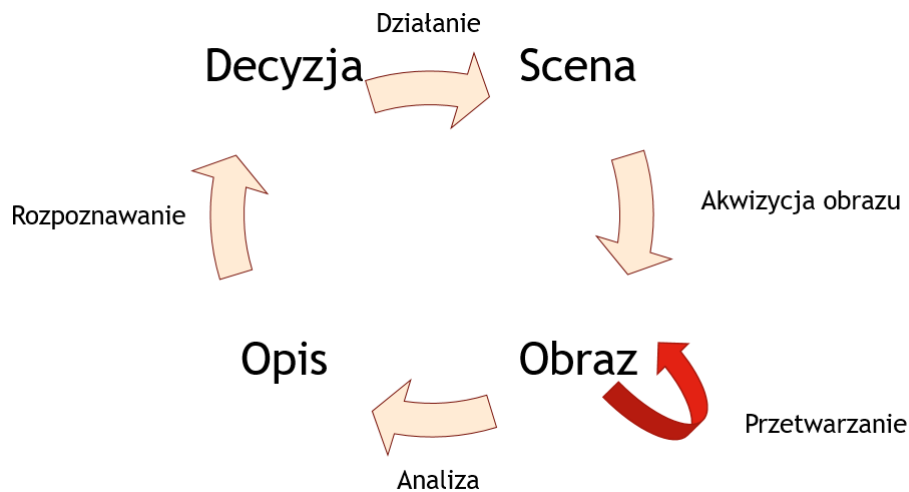
### 1.1 Cel pracy

### 1.2 Zakres pracy

# Rozdział 2

## Przetwarzanie obrazu

Obraz to między innymi źródło informacji - nie tylko wzrokowej. Aby uzyskać dane, które są w nim zapisane, należy go odpowiednio przetworzyć. W dziedzinie nauki, jaką jest przetwarzanie obrazów, wyróżniamy kilka etapów działania. Uproszczony schemat został przedstawiony na rysunku poniżej. Poniżej zostały omówione poszczególne etapy.



Rysunek 2.1: Schemat cyklu przetwarzania obrazu.

1. Aktywizacja obrazu – zamiana energii świetlnej od każdej punktu sceny na sygnał elektryczny, czyli zbieranie danych o każdym pikselu. Scena jaką obserwujemy jest funkcją ciągłą, przy przetwarzaniu jej na obraz cyfrowy, pewna część danych zostaje stracona. Typowo korzysta się z dwóch sposobów pozyskiwania sygnały dyskretnego z sygnału ciągłego i jest to: kwantyzacja oraz próbkowanie. Kwantyzacja jest to pobieranie danych w funkcji wartości funkcji – w zależności od niej przyporządkowuje się próbkę do odpowiedniego przedziału, czyli kwantu, natomiast próbkowanie polega na zbieraniu danych w funkcji czasu.
2. Przetwarzanie obrazu – zmniejszenie obrazu, eliminacja zakłóceń, filtracja wstępna, której celem jest wyeksponowanie na obrazie ważnych cech, takich jak krawędzie lub duże jednokolorowe obiekty. Do tego celu stosuje się np. transformację obrazu kolorowego na skalę odcieni szarości, tudzież na bitmapę składającą się jedynie z dwóch wartości.

3. Analiza obrazu – wydobywanie wcześniej wyeksponowanych cech, w celu ich późniejszego rozpoznania, już nie jako kształtu ale jako konkretnej informacji. Wynikiem tego etapu nie jest już obraz, a dane w postaci liczbowej lub statystycznej.
4. Decyzja – rozpoznawanie obrazu, zanalizowanie cech, pozyskanie informacji, a przede wszystkim klasyfikacja[1].

Wszystkie te etapy dzieją się w systemie wizyjnym do którego można zaliczyć np. komputer, telefon, aparat fotograficzny, a do jego podstawowych funkcji należy:

- Przyjęcie obrazu,
- zapisanie obrazu,
- właściwa obróbka obrazu,
- wyświetlenie obrazu[2];

## 2.1 Przetwarzanie do skali szarości

RGB, jest modelem przestrzeni barw, w którym każdy piksel opisują trzy wartości: intensywność koloru czerwonego, intensywność koloru zielonego oraz intensywność koloru niebieskiego. Obraz w odcieniach szarości to obraz w którym procentowy udział każdej z tych trzech instancji jest równy. Aby uzyskać taki wynik należy przeprowadzić pewne modyfikacje na wartościach pikseli. Najbardziej znane są trzy algorytmy:

- Lekkość (ang. Lightness) – metoda ta polega na tym, że z trzech wartości: intensywności koloru czerwonego, zielonego oraz niebieskiego w pikselu, zostaje wybrana wartość największa oraz najmniejsza, a wynikiem transformacji jest średnia arytmetyczna tych dwóch wartości.

$$p = \frac{\max(R, G, B) + \min(R, G, B)}{3} \quad (2.1)$$

- Średnia (ang. Average) – metoda ta, najprostsza, dzieli sumę wartości intensywności trzech kolorów przez trzy.

$$p = \frac{R + G + B}{3} \quad (2.2)$$

- Jasność (ang. Luminance) – metoda również opierająca się na średniej, ale w tym wypadku jest to średnia ważona. Ludzki wzrok jest najbardziej czuły na kolor zielony, właśnie dlatego intensywność zieleni ma największą wagę. Wyjściowy piksel przyjmuje wartość:

$$p = \frac{0.21R + 0.72G + 0.07B}{3} [3] \quad (2.3)$$

## 2.2 Binarizacja

Binarizacja to operacja punktowa. Wynikiem tej operacji jest obraz binarny, czyli taki, w którym każdy z pikseli przyjmuje tylko jedną z dwóch wartości. Ideą tego zabiegu jest wyodrębnienie obiektu od tła przez nadanie pikselom tych dwóch instancją różnych wartości. Wejściowym obrazem jest zazwyczaj obraz przetworzony do skali szarości, którego piksele przyjmują wartości od 0 do 255. Najczęściej stosowaną metodą klasyfikacji piksela do jednej z dwóch klas jest progowanie – sprawdzenie czy wartość piksela przekracza zadany próg czy też nie i w zależności od wyniku tego zdania logicznego – przypisanie odpowiedniej wartości. Ważnym zagadnieniem tego sposobu jest celne dobranie progu. Najbardziej intuicyjnym rozwiązaniem byłoby wybranie środkowej wartości z przedziału skali wartości zadanego obrazu. Jest to rozwiązanie globalne, sprawdzające się w przypadku gdy tło jest wyraźnie jaśniejsze lub ciemniejsze od pozostałych obiektów sceny w każdej z jej współrzędnych, a liczba pikseli tła jest w przybliżeniu równa liczbie pikseli reszty elementów. Dlatego szuka się optymalnych wartości progu i wykorzystuje się do tego takie metody jak opisana poniżej metoda Otsu.

### 2.2.1 Metoda Otsu

Metoda Otsu należy do rodziny metod optymalnych względem pewnej funkcji kryterialnej. W tym przypadku ową funkcją kryterialną jest wariancja wewnątrzklasowa oraz międzyklasowa. Metoda ta jest bardzo prosta, w uogólnieniu używa momentów zerowego oraz pierwszego stopnia histogramu obrazu. Pierwszym etapem poszukiwania wartości progowej metodą Otsu jest normalizacja histogramu obrazu w skali szarości. Zakłada się, że piksele przyjmują wartości z  $L$  poziomów, a do każdego  $i$ -tego poziomu jest przydzielana ni liczba pikseli. Przyjmując, że  $N$  to liczba wszystkich pikseli obrazu, rozkład prawdopodobieństwa tego histogramu przedstawiony jest następująco:

$$p_i = \frac{n_i}{N} \quad p_i \geq 0, \sum_{i=1}^L p_i = 1 \quad (2.4)$$

Następnym krokiem jest dychotomia pikseli, czyli podział wszystkich elementów na dwie klasy względem pewnego progu  $t$ . Wtenczas piksele, które przyjmują wartości mniejsze od zadanej wartości progowej  $t$  zostaną sklasyfikowane do klasy  $K_0$ , a te większe do klasy  $K_1$ . Wtedy prawdopodobieństwo tego, że badany piksel zostanie przypisany do danej klasy jest określone wzorami:

$$\omega_0 = Pr(K_0) = \sum_{i=1}^t p_i = \omega(t) \quad (2.5)$$

$$\omega_1 = Pr(K_1) = \sum_{i=t+1}^L p_i = 1 - \omega(t) \quad (2.6)$$

Średnia wartość piksela przyjmowana w klasie określona została w poniższych wzorach:

$$\mu_0 = \sum_{i=1}^t Pr(i|K_0) = \sum_{i=1}^t \frac{ip_i}{\omega_0} = \frac{\mu(t)}{\omega(t)} \quad (2.7)$$

$$\mu_1 = \sum_{i=t+1}^L iPr(i|K_1) = \sum_{i=t+1}^L \frac{ip_i}{\omega_1} = \frac{\mu_T - \mu(t)}{1 - \omega(t)} \quad (2.8)$$

Gdzie:

$$\omega(t) = \sum_{i=1}^t p_i \quad (2.9)$$

$$\mu(t) = \sum_{i=1}^t i p_i \quad (2.10)$$

Powyższe wartości są skumulowanymi momentami zerowego oraz pierwszego stopnia obrazu dla wartości progowej równej  $t$ . A średnia z wartości pikseli na całym obrazie może zostać zapisane wzorem:

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i \quad (2.11)$$

Niezależnie od wyboru wartości parametru  $t$ , prawdziwe są równania:

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T, \quad \omega_0 + \omega_1 = 1. \quad (2.12)$$

Wariancje obu klas można przedstawić następująco:

$$\sigma_0^2 = \sum_{i=1}^t (i - \mu_0)^2 Pr(i|K_0) = \sum_{i=1}^t (i - \mu_0)^2 \frac{p_i}{\omega_0} \quad (2.13)$$

$$\sigma_1^2 = \sum_{i=t+1}^L (i - \mu_1)^2 Pr(i|K_1) = \sum_{i=t+1}^L (i - \mu_1)^2 \frac{p_i}{\omega_1} \quad (2.14)$$

W kolejnym etapie należy sprawdzić czy wartość zmiennej  $k$  jest wartością optymalną względem funkcji kryterialnej. Funkcją kryterialną, jak wcześniej wspomniano jest:

- Wariancja wewnątrz klasowa, dąży do minimalizacji,

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2, \quad \sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_2)^2 \quad (2.15)$$

- Wariancja między klasowa, dąży do maksymalizacji.

$$\kappa = \frac{\sigma_T^2}{\sigma_W^2}, \quad \sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \quad (2.16)$$

Co intuicyjnie zgadza się z pożądanym efektem działania algorytmu. Chce się aby wartości pikseli w klasach były do siebie jak najbardziej zbliżone, a między klasami występowała jak największa różnica. Kryterium jakie stosuje się do optymalizacji wartości  $k$  jest łączna wariancja poziomów  $\eta$ . Parametr jest optymalny, jeżeli  $\eta$  przyjmuje wartość jak największą, tym samym wariancja wewnątrz klasowa. Wartość ta będzie zerowa jedynie w przypadku gdy wszystkie piksele obrazu zostaną zakwalifikowane tylko do jednej z klas, jednak jest to sprzeczne z celem działania algorytmu, więc w tym kontekście maksimum zawsze istnieje.

$$\eta = \frac{\sigma_B^2}{\sigma_T^2}, \quad (2.17)$$

Zaletą tej metody jest jej prostota – korzysta się jedynie z momentów zerowego i pierwszego rzędu, a także stabilność – optymalny próg jest wybrany automatycznie, nie bazuje na zróżnicowaniach (takich jak lokalne zbiory pikseli o niskich lub wysokich wartościach), skupia się na globalnych właściwościach obrazu, histogramie. Ważną cechą jest też uniwersalność metody, dzięki, której można ją wykorzystywać w różnych dziedzinach analizy obrazu. Może służyć do binaryzacji, jak wykorzystano w projekcie, wykorzystywana jest też przy progowaniu[4].

## 2.3 Wykrywanie krawędzi

Przetwarzanie obrazów opiera się na wydobywaniu z obrazu takich cech, które są istotne przy późniejszej analizie i identyfikacji obiektów na nim zawartych. Jedną z takich cech jest krawędź, czyli znaczna lokalna zmiana w intensywności obrazu, związana z brakiem ciągłości w intensywności obrazu lub w pierwszej pochodnej intensywności obrazu. Występują dwa rodzaje takiej zmiany ciągłości:

1. Nieciągłość skokowa - intensywność obrazu zmienia się z jednej wartości w drugą i przez jakiś czas tą wartość utrzymuje, przypomina odpowiedź skokową, na obrazie może być to moment styku dwóch obiektów;
2. Nieciągłość liniowa - intensywność obrazu zmienia się z jednej wartości w drugą lecz w krótkim okresie czasu do niej powraca, przypomina odpowiedź impulsową, na obrazie może to być przerwanie w obiekcie;

W praktyce jednak tego typu krawędzie zdarzają się dość rzadko, krawędzie nie są aż tak wyraziste i skok przypomina bardziej rampę, a linia - dach[5]. Wykrywanie krawędzi to przede wszystkim rozpoznawanie znacznych zmian w obrazie, jest to mocno związane z wyznaczaniem maksimum pierwszej pochodnej. Gradient jest miarą zmiany, wektorem, który wskazuje kierunki i szybkość wzrostu wartości, to dwuwymiarowy odpowiednik pierwszej pochodnej. W obrazie zmianą jest różnica w intensywności koloru. To właśnie on jest bazą większości algorytmów wykrywających krawędzie. Gradient obrazu dany jest wzorem:

$$\nabla f = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \begin{pmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{pmatrix} \quad (2.18)$$

$g_x$ - gradient w kierunku x,  
 $g_y$ - gradient w kierunku y

Kierunek zmian można wyliczyć z poniższego wzoru:

$$\theta = \tan^{-1} \begin{pmatrix} g_x \\ g_y \end{pmatrix} \quad (2.19)$$

Etapy wykrywania krawędzi:

### 1. Filtracja

Aby ulepszyć działanie detektora krawędzi należy pozbyć się szumów, które utrudniają wyliczanie gradientów. Należy jednak zachować umiar, zbyt duże wygładzanie obrazu może spowodować osłabienie krawędzi. Często w tym celu stosuje się rozmycie Gaussowskie, które zostało omówione szerzej w podrozdziale 2.3.1.



## 2. Wzmocnienie

W polach obrazu gdzie umiejscowione są największe zmiany intensywności koloru podkreśla się te piksele, które mają na tą zmianę największy wpływ. Używa się w tym celu gradientu.

## 3. Wykrycie

W związku z licznymi zanieczyszczeniami i szumami obecnymi na obrazie zdarza się, że piksele o niezerowym gradiencie nie należą do krawędzi. Aby pominąć piksele nieistotne, wybiera się próg wedle którego ocenia się czy piksel należy do krawędzi czy też nie.

## 4. Zlokalizowanie

Etap ten składa się na większość algorytmów, polega na określeniu rozdzielczości, położenia pikseli, które składają się na krawędź. Jest to zdecydowanie przydatne w momencie potrzeby wyodrębnienia z obrazu konkretnego obiektu[6].

### 2.3.1 Detektor Canny

Detektor ten jest najczęściej używanym detektorem w świecie przetwarzania obrazów. Należy do rodziny detektorów Gaussowskich, specjalizujących się w wykrywaniu krawędzi skokowych. Jest to algorytm optymalny względem kilku kryteriów:

- Poprawna detekcja - dąży się do minimalizacji klasyfikacji nieistniejących krawędzi jako krawędzi istniejących. Są większe szanse na pominięcie krawędzi niż na wykrycie krawędzi błędnej.
- Poprawna lokalizacja - dąży się do tego, aby wykryte krawędzie znajdowały się jak najbliżej istniejących krawędzi.
- Jednoznaczna odpowiedź - dąży się do minimalizacji lokalnych maksimów wokół wykrytej krawędzi, idea jest taka, aby dla każdego istniejącego punktu krawędzi zwracany był tylko jeden piksel.

Sposób działania algorytmu można opisać w kilku krokach:

Krok 1 Obraz wejściowy należy poddać operacji rozmycia Gaussa, w celu pozbycia się szumów. Polega to na nałożeniu na obraz maski, która jest dyskretną aproksymacją funkcji Gaussa opisaną wzorem poniżej:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.20)$$

Matematycznie, nałożenie maski na obraz to wyliczenie spłotu dyskretnych wartości funkcji Gaussa i pikseli obrazu[7].

Rozmycie Gaussa jest zależne od wartości gradientu.

Krok 2 Następnie wyliczona zostaje wartość oraz kierunek gradientu.

$$gradient = \sqrt{G_x^2 + G_y^2} \quad (2.21)$$

$$\theta = \tan^{-1} \frac{G_y}{G_x} \quad (2.22)$$

- Krok 3 Wykrywa się punkty należące do krawędzi i tłumi te wartości, które nie są maksimumami lokalnymi, aby pozbyć się niewyraźnych krawędzi.
- Krok 4 W ostatnim etapie punkty krawędzi zostają połączone i wybiera się histerezę - przedział od najniższej wartości progu do najwyższej wartości progu w jakim ma się znaleźć krawędź. Najwyższy próg  $k_{max}$  dotyczy mocnych krawędzi, natomiast najniższy próg  $k_{min}$  dotyczy krawędzi słabych.

## Rozdział 3

### Architektura systemu

# Rozdział 4

## Implementacja

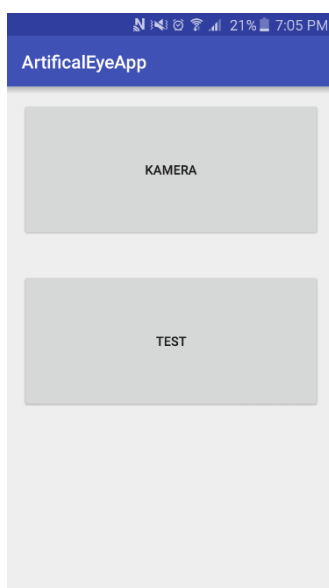
# Rozdział 5

## Działanie aplikacji i testy

### 5.1 Wymagania wstępne

Aplikacja działa na smartfonach z oprogramowaniem Android o wersji SDK nie starszej niż 17. Wersją najbardziej oczekiwaną jest wersja 23. Smartfon musi posiadać aparat oraz ponad 13Mb wolnej pamięci – zostanie na niej zapisany folder – tessdata, z plikiem zawierającym dane potrzebne bibliotece Tess4j do przetwarzania obrazu w języku polskim. Język telefonu powinien być ustawiony na ten w jakim oczekiwane jest czytanie komunikatów przez aplikację.

### 5.2 Widok główny



Rysunek 5.1: Główny widok aplikacji.

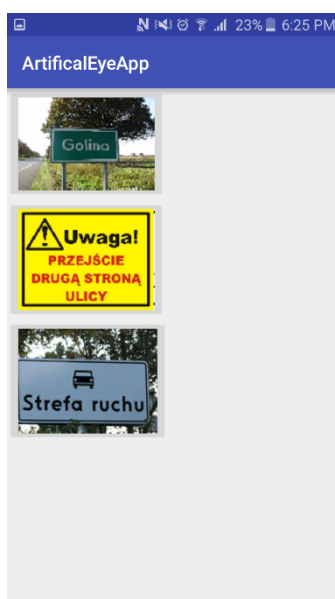
Jest to intuicyjny widok z dwoma dużymi przyciskami. Przy uruchomieniu aplikacji, informuje ona użytkownika komunikatem głosowym o położeniu przycisków i ich przeznaczeniu: „Kliknij na górze ekranu aby przejść do kamery – Kliknij na dole ekranu aby przejść do testów”. Aplikacja, poza widokiem głównym, posiada dwa widoki dodatkowe: użytkownika oraz testowy. Można się do nich dostać po wciśnięciu w odpowiedni przycisk.

Przycisk „Kamera” przekierowuje użytkownika do widoku z kamery smartfona, natomiast przycisk „Test” przekierowuje do widoku testowego.

## 5.3 Widok docelowy

Widok docelowy przenosi użytkownika do kamery zainstalowanej na urządzeniu na którym działa aplikacja. W tym momencie użytkownik ma możliwość kliknięcia w ekran i zrobienia fotografii. W zależności od rodzaju telefonu, kamera może poprosić o potwierdzenie przesłania zdjęcia do dalszej przeróbki, o czym aplikacja poinformuje komunikatem głosowym, z dokładnym umiejscowieniem odpowiedniego przycisku na ekranie, zdjęcie też może zostać od razu przesłane.

## 5.4 Widok testowy

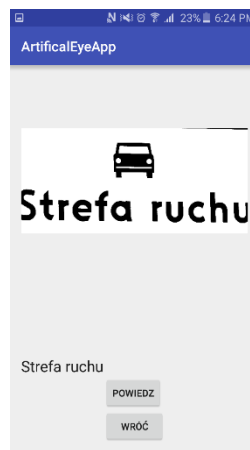


Rysunek 5.2: Testowy widok aplikacji.

Osoba testująca, klikając na wybrane zdjęcie sprawia, że zostaje ono poddane przetwarzaniu. W efekcie końcowym otrzymuje się rozpoznany znak, wycięty ze zdjęcia, jako obraz binarny – zaprezentowany przez dwa odcienie – czarny oraz biały, kolory odwrócone, odpowiednio tak aby tekst był czarny, a tło białe, czyli obraz jaki na wejście otrzymuje algorytm rozpoznający tekst. Pod obrazem zostaje wyświetlony tekst, który otrzymano w wyniku działania biblioteki Tess4j. Zostaje on przeczytany przez aplikację, użytkownik może odtwarzać go dowolną liczbę razy, klikając na przycisk „Powiedz”. Poniżej, na zdjęciach numer 5.3, 5.4 oraz 5.5, zostały przedstawione wyniki działania aplikacji w trybie testowym.



(a) Obraz przetwarzany.

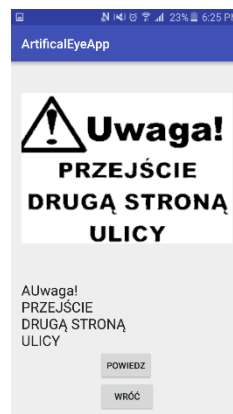


(b) Obraz przetworzony.

Rysunek 5.3: Wynik działania aplikacji.



(a) Obraz przetwarzany.



(b) Obraz przetworzony.

Rysunek 5.4: Wynik działania aplikacji.



(a) Obraz przetwarzany.



(b) Obraz przetworzony.

Rysunek 5.5: Wynik działania aplikacji.

# Bibliografia

- [1] R. Tadeusiewicz, P. Korohoda: *Komputerowa analiza i przetwarzanie obrazów*, wyd. Fundacji Postępu Telekomunikacji, ('3)
- [2] jw. ('5)
- [3] <http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>
- [4] <http://web-ext.u-aizu.ac.jp/course/bmclass/documents/otsu1979.pdf>
- [5] R. Jain, R. Kasturi, B. G. Schunck, *Machine Vision* McGraw-Hill, Inc., 1995 ('140)
- [6] jw. ('145-146)
- [7] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>