

Przetwarzanie obrazów

System przetwarzania obrazów do
rozpoznawania logotypów

Dokumentacja projektu

Michał Dobrzański

Politechnika Warszawska,
Wydział Elektroniki i Technik Informacyjnych.

10 stycznia 2017

Spis treści

1	Szczegółowy opis merytoryczny zadania	2
1.1	Założenia projektu	2
2	Projekt rozwiązania	3
2.1	Obrazy wejściowe	4
3	Wstępne przetwarzanie	5
3.1	Redukcja rozmiarów	5
3.2	Filtracja dolnoprzepustowa	5
3.3	Rozbicie na kanały HS	6
3.3.1	Filtracja medianowa	8
3.3.2	Zamknięcie obrazu	9
4	Segmentacja	10
5	Analiza cech	11
5.1	Niezmienniki momentowe	11
6	Identyfikacja	12
7	Otrzymane wyniki	14
7.1	Wnioski	16

1 Szczegółowy opis merytoryczny zadania

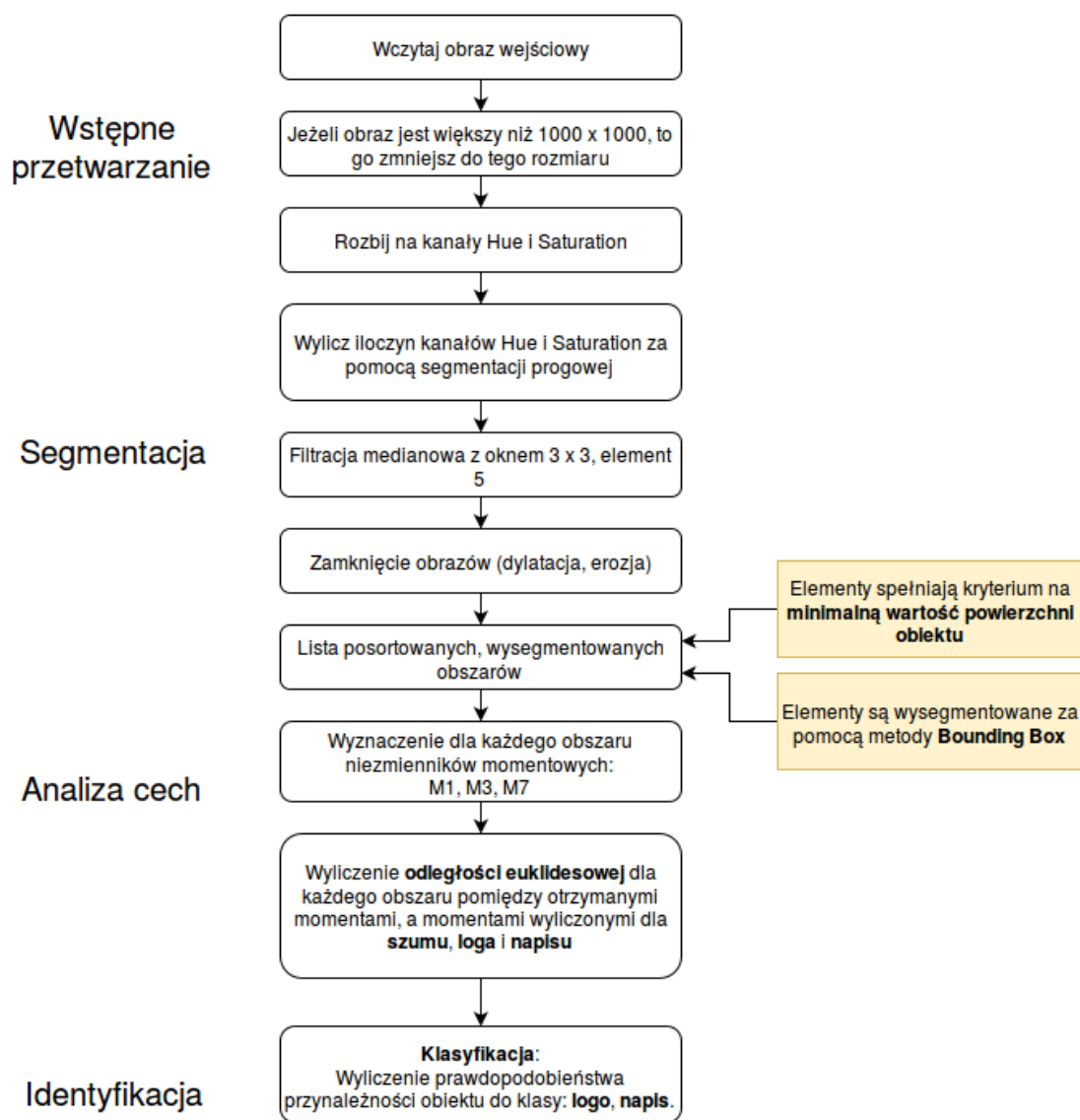
W ramach projektu zaimplementowano **system do rozpoznawania logotypu „ORLEN”** na zdjęciach cyfrowych. Obrazy wejściowe (zdjęcia) są zaszumione (tj. przedstawiają kolorowe tło, którym jest niebo, chmury, drzewa). Przynajmniej na jednym obrazie znajdują się dwie instancje logotypu.

1.1 Założenia projektu

W projekcie został wybrany do analizy logotyp marki Orlen. Jako język programowania wybrano język *C++* wraz z Frameworkiem *OpenCV*. Do pisania projektu użytko *IDE CLion* oraz narzędzia do budowania projektów *CMake*. Postęp prac analizowany był systemem kontroli wersji *Git*.

System został napisany i skompilowany na środowisko *Linux*.

2 Projekt rozwiązania



Rysunek 1: Projekt implementowanego systemu

Na rysunku 1 przedstawiono zaimplementowany tor przetwarzania. Obrazy wejściowe znajdują się w osobnym katalogu.

Na potrzeby projektu utworzono kilka folderów zawierających przetwarzanie obrazu. Niektóre z nich przedstawiają dany etap toru przetwarzania.

- Folder **przechowujący logotypy**, które będą potrzebne przy budowaniu klasyfikatora.
- Folder przechowujący **obrazy traktowane jako szum** (nie zawierające logotypu). Zostaną użyte przy klasyfikacji.
- Folder przechowujący *obrazy pośrednie* otrzymane w wyniku wykonania się poszczególnych algorytmów toru przetwarzania.

2.1 Obrazy wejściowe



Rysunek 2: Przykładowy obraz wejściowy

Przykładowy obraz wejściowy przedstawiono na rysunku 7. Należy zadbać o odpowiednią jasność dostarczanego obrazu. Implementowany system **przyjmuje obrazy w różnych rozmiarach** dostosowując wykonanie algorytmów do rozmiaru danych.

3 Wstępne przetwarzanie

Poniżej omówiono poszczególne kroki zaimplementowanego toru przetwarzania.

3.1 Redukcja rozmiarów

System przyjmuje wejściowe obrazy zawierające 3 kanały kolorów - R, G, B . Jeżeli te obrazy mają rozmiar większy niż 1000×1000 pikseli, są one **redukowane** do tych wymiarów. Ma to na celu przyspieszenie wykonywania się obliczeń.

3.2 Filtracja dolnoprzepustowa

W ramach tego etapu doświadczalnie sprawdzono, czy przed konwersją składowych obrazu R, G, B na H, S i V opłaca się zastosować **filtrację dolnoprzepustową**. Motywacją do tego zabiegu był fakt, że tło w obrazach może zawierać szum wysokoczęstotliwościowy. Elementami tła świadczącymi o istnieniu tego szumu mogą być miejsca, w których wartość intensywności pikseli się gwałtownie zmienia, np. liście drzewa na niebie, różnie oświetlona roślinność.

Zaimplementowano dwa rodzaje filtrów dolnoprzepustowych stosując okno **przesuwne** i operację **splotu z obrazem wejściowym**:

- filtr dolnoprzepustowy uśredniający o masce o wymiarach 3×3 - pełna macierz

$$\begin{matrix} 1 & 1 & 1 \end{matrix}$$

współczynników o wartości 1: $\begin{matrix} 1 & 1 & 1 \end{matrix}$

$$\begin{matrix} 1 & 1 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 2 & 1 \end{matrix}$$

- filtr gaussowski o wymiarach 3×3 : $\begin{matrix} 2 & 4 & 2 \end{matrix}$

$$\begin{matrix} 1 & 2 & 1 \end{matrix}$$

Zastosowanie filtracji dolnoprzepustowej na tym etapie okazało się **nieskuteczne**, gdyż pogarsza jakość wysegmentowanych obszarów - napis logotypu staje się mniej czytelny, co pogarsza wynik klasyfikacji.

3.3 Rozbicie na kanały HS

Logotyp marki Orlen charakteryzuje się dwoma dominującymi barwami: czerwoną i białą. Na niego składa się biały napis i logo na czerwonym tle.

W projektowanym systemie założono **wysegmentowanie obszarów na podstawie intensywności czerwonej barwy**. W tym celu postanowiono przejść z przestrzeni barw *RGB* na *HSV*. Zaimplementowano algorytm konwersji wartości składowych pikseli *RGB* na odpowiadające wartości w przestrzeni *HSV*. Przejście na drugą reprezentację jest motywowane faktem, iż składowa $H(Hue)$ w wierniejszy sposób odwzorowuje informację o odcieniu barwy i umożliwia bardziej dokładną późniejszą segmentację obrazu.

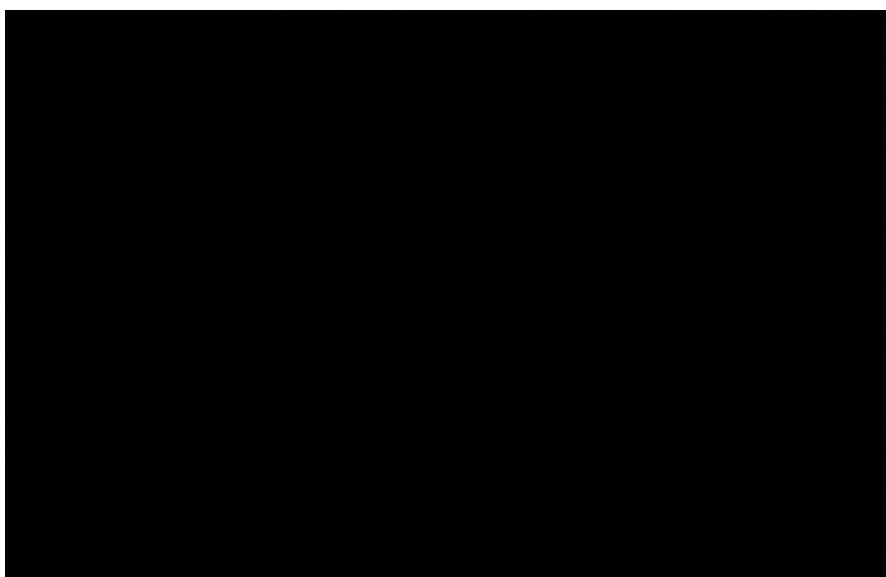
Zaproponowano rozwiązanie polegające na **segmentacji dwuprzędziałowej** dla dwóch kanałów $H(Hue)$ oraz $S(Saturation)$ 34. Doświadczalnie wykazano, że informacja niesiona przez składową $V(Value)$ jest bardzo niewielka i ma znikomy wpływ na dalszy tor przetwarzania. W związku z tym, metodą prób i błędów wybrano najbardziej rokujące zakresy (przedziały) wartości, jakie mogą przyjmować parametry H i S . Wartości przedstawiono w tabeli 1, a rezultat na obrazie 10.

	Wartość minimalna	Wartość maksymalna
Hue	0.55	0.73
Saturation	0.49	1.0

Tablica 1: Wybrane wartości parametrów *Hue* i *Saturation*.



Rysunek 3: Kanał Hue.



Rysunek 4: Kanał Saturation.



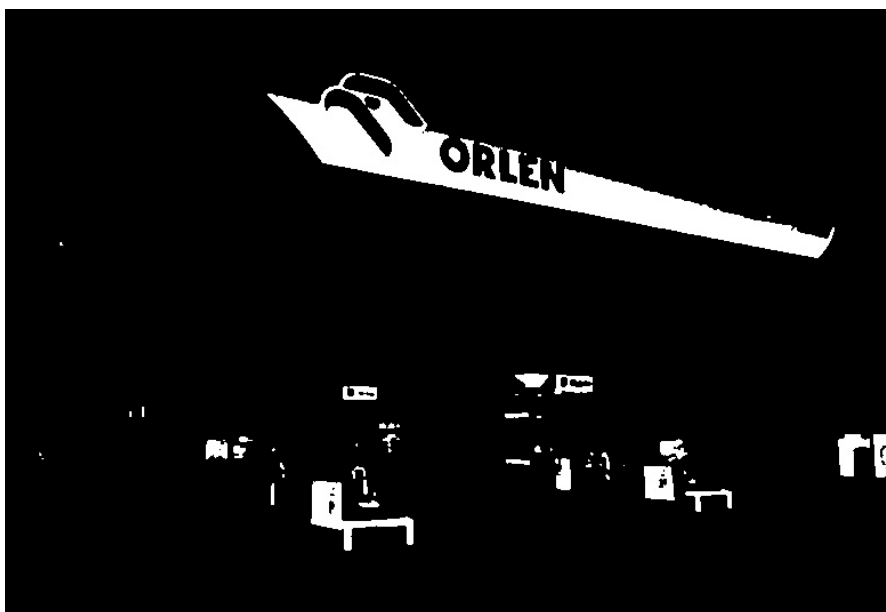
Rysunek 5: Iloczyn kanałów Hue i Saturation po progowaniu dla zadanych wartości.

3.3.1 Filtracja medianowa

Przed wykonaniem algorytmu *flood-fill* służącego do wyznaczenia obiektów zastosowano dwa kroki mające na celu poprawę jakości obrazów po wynikowej segmentacji progowej dla kanałów Hue i Saturation.

W proponowanym rozwiązaniu metodą doświadczalną wybrano najlepsze parametry dla filtracji medianowej. Stosuje się okno przesuwne w wymiarach 3×3 . Filtracja medianowa bierze **piąty element** z posortowanej listy wartości pikseli dla danego okna.

Zaimplementowany algorytm umożliwia regulację wielkości okna przesuwnego, wybór elementu oraz **dynamicznie zmienia okno przesuwne** dla pikseli brzegowych. W przypadku doboru okna 3×3 i napotkaniu pikseli brzegowych okno może przyjąć rozmiary na przykład 2×3 . Jeżeli został zadany 8 element do wyboru, to dla tych skrajnych sytuacji wybierany jest $8 - 3 = 5$ element z racji trzech brakujących pikseli w oknie.

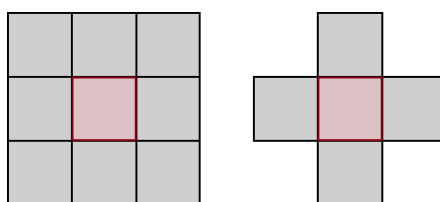


Rysunek 6: Wynik filtracji medianowej dla wybranych parametrów.

3.3.2 Zamknięcie obrazu

W następnym kroku badano wpływ operacji **erozji** i **dylatacji** na jakość wynikowego obrazu. Stosowano dwa rodzaje **elementu strukturalnego**:

- okno typu 3×3
- okno typu „plus”



Rysunek 7: Zaimplementowane rodzaje elementów strukturalnych

Wykazano, że operacją dającą najlepsze rezultaty jest operacja zamknięcia (**domknięcia**), czyli nałożenie operacji erozji na obraz poddany wcześniej operacji dylatacji stosując element strukturalny typu „plus”.



Rysunek 8: Wynik domknięcia obrazu.

4 Segmentacja

System stosuje wspomnianą wcześniej segmentację dwuprzędziałową z wartościami z zakresu określonego w tabeli 1. Wyliczona reprezentacja obrazu w przestrzeni HS potrzebna jest do segmentowania obszarów obrazu RGB.

Algorytm segmentacji działa **punktowo** sprawdzając piksel po pikselu, czy spełnia on kryteria dotyczące wartości *Hue* i *Saturation*. Jeżeli dany piksel spełnia te warunki, identyfikowany jest jako „obiekt”. W programie przyjmuje on arbitralną wartość 255 (kolor biały). Jeżeli nie spełni, jest uważany jako „tło”, czyli przyjmuje wartość 0 (kolor czarny).

Otrzymany jest wynikowy **obraz binary**. Następnie algorytm wykonuje dla tego obszaru algorytm „zalanania” (ang. *flood-fill*), który prowadzi do wyznaczenia rozłącznych obszarów reprezentujących obiekty. Podczas jego działania dla każdego obiektu wyznaczane są wartości minimalne i maksymalne współrzędnych. Dzięki temu możliwe jest stworzenie obrysu obiektu (ang. *bounding-box*), który posłuży do dalszej analizy.

Działanie tego algorytmu warunkowane jest parametrem określającym **minimalną powierzchnię obiektu** określaną w literaturze jako wartość *S*. Jeżeli wysegmentowany obszar będzie miał powierzchnię mniejszą od tej wartości progowej, nie jest on brany pod

uwagę w dalszym etapie przetwarzania. W ten sposób odrzuca się małe rejony, które są najczęściej zbiorem kilkunastu pikseli, dla których praktycznie niemożliwym jest określenie obiektu, jaki reprezentują.

Działanie algorytmu kończy się wynikowym **zbiorem wysegmentowanych obszarów** (bounding-boxes).



Rysunek 9: Przykładowy pierwszy wysegmentowany obszar.



Rysunek 10: Przykładowy drugi wysegmentowany obszar.

5 Analiza cech

W ramach analizy cech operowano na liście wyznaczonych obiektów ograniczonych za pomocą metody *bounding-box*. Lista obiektów jest posortowana malejąco.

Zaprojektowany system wyznacza cechy dwóch największych wysegmentowanych obiektów. Istnieje możliwość zadania większej liczby elementów do analizy.

5.1 Niezmienniki momentowe

Doświadczalnie wykazano, że najbardziej rokującymi współczynnikami określającymi cechy danego obiektu będą **niezmienniki momentowe** $M1$, $M3$, $M7$. Poniżej przedstawiono

równania na ich wyliczenie.

$$M_1 = (M_{20} + M_{02})/m_{00}^2 \quad (1)$$

$$M_3 = ((M_{30} - 3 * M_{12})^2 + (3 * M_{21} - M_{03})^2)/m_{00}^5 \quad (2)$$

$$M_7 = (M_{20}M_{02} - M_{11}^2)/m_{00}^4 \quad (3)$$

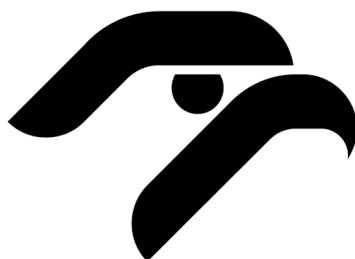
Do wyznaczenia niezmienników momentowych konieczne jest wcześniejsze wyliczenie **momentów centralnych** dla wysegmentowanych obiektów, które z kolei są wyliczane za pomocą **momentów zwykłych**. Aproksymacja niezbędnych do wyliczenia momentów zwykłych jest określona poniższym wzorem dla obrazu o wymiarach n wierszy na m kolumn.

$$m_{pq} = \sum_{i=1}^n \sum_{j=1}^m i^p j^q x_{ij} \quad (4)$$

6 Identyfikacja

Identyfikacja obszarów odbywa się za pomocą klasyfikacji otrzymanych wartości niezmienników momentowych. Mając wyznaczone trzy wartości momentów dla każdego obszaru - M_1 , M_3 , M_7 porównywane one są z wartościami odpowiadającym niezmiennikom momentowym wyliczonym dla **loga** obrazu, **napisu** „ORLEN”. Wyliczana jest **odległość euklidesowa** w przestrzeni trójparametrowej pomiędzy wartościami niezmienników momentowych obiektu, a niezmienników momentowych wyliczonych dla loga, napisu.

Przygotowano kilka przykładowych obrazów reprezentujących logo i napis.

*Rysunek 11: Przykładowe logo**Rysunek 12: Przykładowe logo 2**Rysunek 13: Przykładowe logo 3**Rysunek 14: Przykładowy napis*

Zaimplementowany system zwraca **odpowiedź w postaci zaznaczonych na zdjęciu obszarów**, które uznaje za logo. Taki obszar jest traktowany jako logotyp, jeżeli wykryto na nim logo lub napis. Przykładowa przynależność obiektu do klasy „logo” jest wyliczana zgodnie z poniższym wzorem:

$$p_{logo} = ||dist_{obiekt} - dist_{logo}|| \quad (5)$$

Jest wyliczana **odległość euklidesowa** pomiędzy wartościami momentów dla analizowanego obszaru, a dla logo i napisu.

Dla wyznaczonych segmentów zaimplementowano **algorytm analizujący coraz to mniejsze obszary segmentu** dla których wyznaczał wartości momentów za pomocą przesuwanego **okna analizy**. Dla całego segmentu tworzone jest kilkaset pomniejszych okien analizy. Wybierana jest **globalna minimalna odległość euklidesowa** pomiędzy obiektem, a napisem i logo. Następnie, ta odległość jest progowana.

Segment był **klasyfikowany jako logo**, gdy wartość odległości euklidesowej dla momentów pomiędzy segmentem, a logiem była **mniejsza od 0.007**.

Segment był **klasyfikowany jako napis**, gdy wartość odległości euklidesowej dla momentów pomiędzy segmentem, a napisem była **mniejsza od 0.2**.

7 Otrzymane wyniki

System rozpoznawania logotypu został uruchomiony dla czterech obrazów wejściowych. Analizowane były trzy największe pod względem powierzchni wysegmentowane obiekty. Wynika to z założeń treści zadania. Dla nich dokonano oceny, czy znajduje się w nich *logo*, czy *napis* logotypu.



Rysunek 15: Obiekt poddany klasyfikacji.



Rysunek 16: Obiekt poddany klasyfikacji.



Rysunek 17: Wyniki wykrywania logotypu.



Rysunek 18: Wyniki wykrywania logotypu.



Rysunek 19: Wyniki wykrywania logotypu.



Rysunek 20: Wyniki wykrywania logotypu.

7.1 Wnioski

Zaimplementowany system zaznacza **miejsca wystąpienia loga czarnym prostokątem**. Miejsca wystąpienia napisu są oznaczane niebieskim prostokątem.

Dla pierwszego obrazu system prawidłowo znajduje lokalizację loga Orlen jak i napisu.

Dla drugiego obrazu, gdzie są dwie instance loga, system detekuje faktyczne dwa miejsca wystąpienia loga. Jednak nie był w stanie wykryć większego napisu.

Dla trzeciego obrazu system znajduje loga w miejscach, gdzie ich nie ma, lecz są one bardzo blisko faktycznego wystąpienia logotypu.

Dla czwartego obrazu znaleziono logo jak i napisy w niewłaściwych miejscach. Jest to spowodowane niedoskonałością niezmienników momentowych wyliczonych dla napisów na dystrybutorze - wyszły one wyjątkowo zbliżone w stosunku do niezmienników wyliczonych dla loga i dla napisu.