

Przetwarzanie obrazów

System przetwarzania obrazów do
rozpoznawania logotypów

Dokumentacja projektu

Michał Dobrzański

Politechnika Warszawska,
Wydział Elektroniki i Technik Informacyjnych.

23 grudnia 2016

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
4	Segmentacja	8
4.1	Poprawa jakości segmentacji	9
4.1.1	Filtracja medianowa	9
4.1.2	Zamknięcie obrazu	10
5	Analiza cech	11
5.1	Niezmienniki momentowe	11
6	Identyfikacja	12
6.1	Stworzenie modelu	13
7	Otrzymane wyniki	13
7.1	Wnioski	14

1 Szczegółowy opis merytoryczny zadania

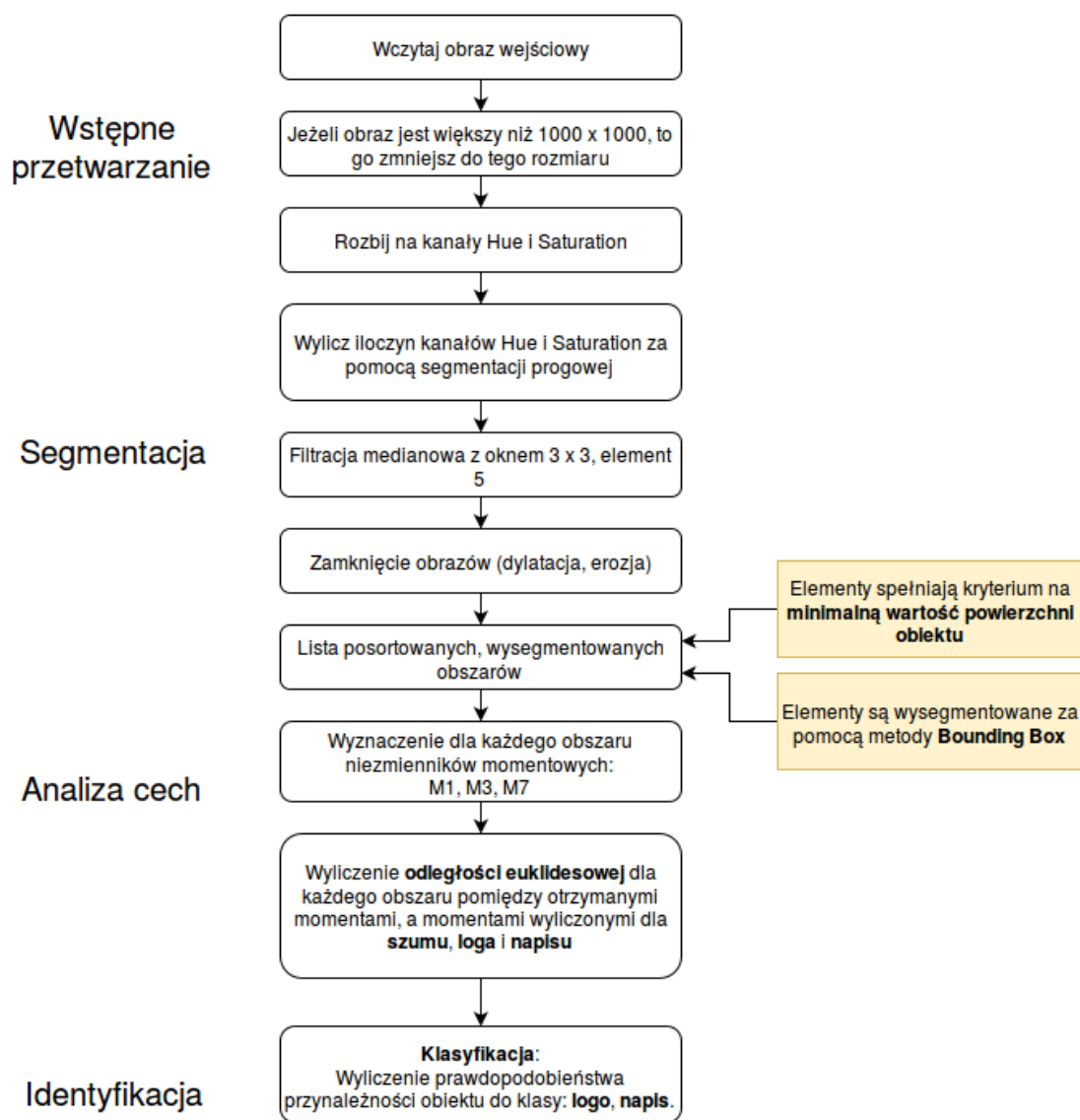
W ramach projektu zaimplementowano **system do rozpoznawania logotypu „ORLEN”** na zdjęciach cyfrowych. Obrazy wejściowe (zdjęcia) są zaszumione (tj. przedstawiają kolorowe tło, którym jest niebo, chmury, drzewa). Przynajmniej na jednym obrazie znajdują się dwie instancje logotypu.

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 9. 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.

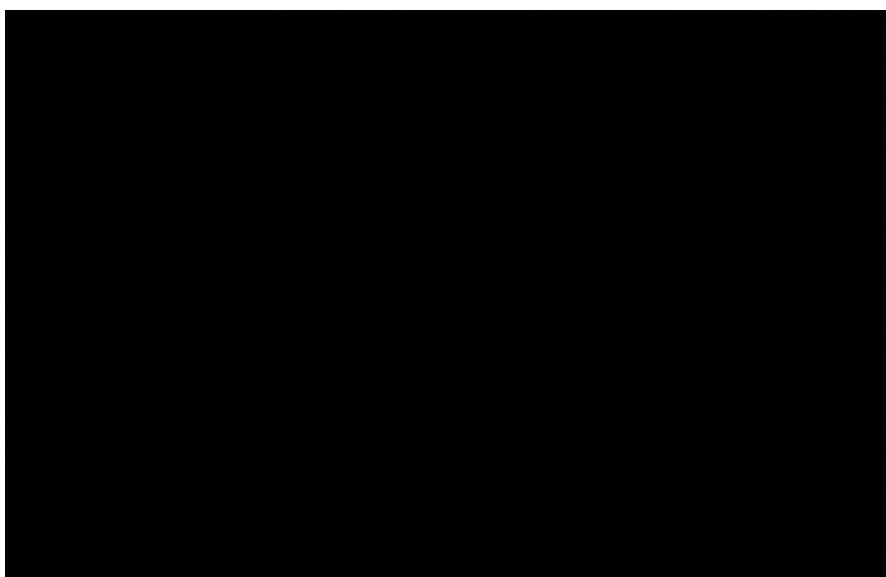
Zaproponowano rozwiązanie polegające na **segmentacji dwuprzędziałowej** dla dwóch kanałów $H(Hue)$ oraz $S(Saturation)$ 34. Doświadczalnie wykazano, że informacja niesiona przez składową $V(Value)$ jest bardzo niewielka i ma znikomy wpływ na dalszy tor przetwarzania. W związku z tym, metodą prób i błędów wybrano najbardziej rokujące zakresy (przedziały) wartości, jakie mogą przyjmować parametry H i S . Wartości przedstawiono w tabeli 2, a rezultat na obrazie 7.

	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.

4 Segmentacja

System stosuje wspomnianą wcześniej segmentację dwuprzędziałową z wartościami z zakresu określonego w tabeli 2. 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 6: Przykładowy pierwszy wysegmentowany obszar.



Rysunek 7: Przykładowy drugi wysegmentowany obszar.

4.1 Poprawa jakości segmentacji

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

4.1.1 Filtracja medianowa

W proponowanym rozwiązaniu metodą doświadczalną wybrano najlepsze parametry dla filtracji medianowej. Stosuje się okno przesuwne w wymiarach **3x3**. 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.

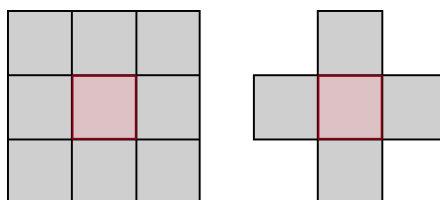


Rysunek 8: Wynik filtracji medianowej dla wybranych parametrów.

4.1.2 Zamknięcie obrazu

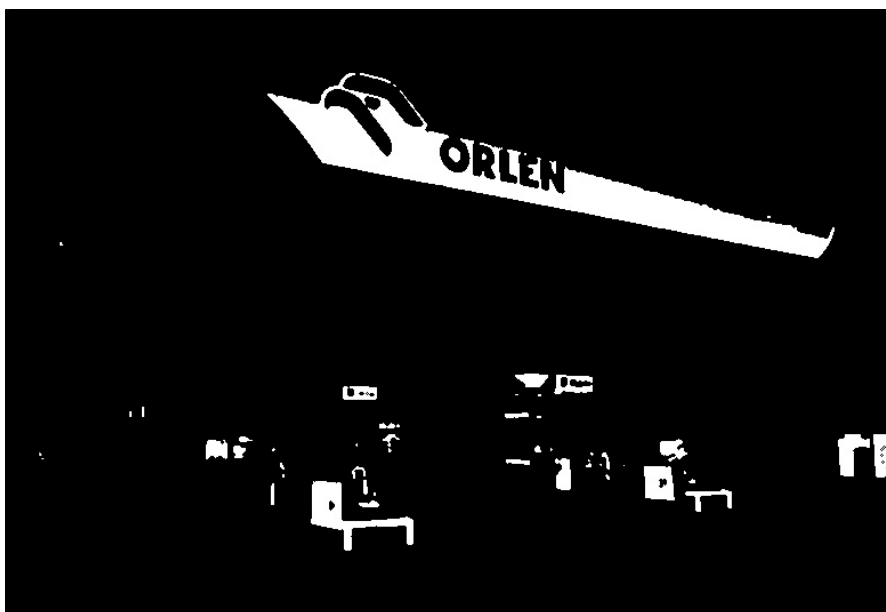
W następnym kroku badano wpływ operacji **erozji** i **dylatacji** na jakość wynikowego obrazu. Stosowano dwa rodzaje **elementu strukturalnego**:

- okno typu 3×3
- okno typu „plus”



Rysunek 9: 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 10: Wynik domknięcia obrazu.

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 - $M1$, $M3$, $M7$ porównywane one są z wartościami odpowiadającym niezmiennikom momentowym wyliczonym dla **loga** obrazu, **napisu** „ORLEN” oraz dla **szumu**. Wyliczana jest **odległość euklidesowa** w przestrzeni trójparametrowej pomiędzy wartościami niezmienników momentowych obiektu, a niezmienników momentowych wyliczonych dla loga, napisu i szumu.

Zaimplemetowany system zwraca odpowiedź, na ile jest pewien, że dla danego wysegmentowanego obiektu widzi w nim „logo” lub „napis”, czyli **prawdopodobieństwo klasyfikacji**. Przykładowe prawdopodobieństwo przynależności obiektu do klasy „logo” jest wyliczane zgodnie z poniższym wzorem.

$$p_{logo} = \frac{dist_{obiekt-logo}}{dist_{obiekt-logo} + dist_{obiekt-szum}} * 100\% \quad (5)$$

Gdzie $dist$ określa odległość euklidesową pomiędzy dwoma zestawami wartości parametrów określonych za pomocą niezmienników momentowych. W podobny sposób ocenia się przynależność obiektu do klasy „napis”.

6.1 Stworzenie modelu

Przed wykonaniem się toru wykrywania logotypu należało **skonstruować model klasyfikatora**. W tym celu wybrano dwa binarne obrazy reprezentujące oryginalny napis jak i logo dla logotypu. Samo logo poddano obróbce w programie graficznym *Pinta* w celu utworzenia pięciu reprezentatywnych obrazów.

Dla uzyskanych obrazów przedstawiających samo **logo** wyznaczono trzy niezmienniki momentowe - $M1$, $M3$, $M7$, a następnie wynik uśredniono. W ten sposób otrzymano **wartości parametrów w przestrzeni cech**. Dla **napisu** wyliczono niezmienniki tylko raz. W przypadku obrazów reprezentujących **szum** wyliczono niezmienniki dla wszystkich wysegmentowanych obiektów dla czterech obrazów przedstawiających inne stacje paliwowe. Wyniki uśredniono.

Otrzymane wartości wykorzystywane są przy klasyfikacji za pomocą odległości euklidesowej do tych klas.

7 Otrzymane wyniki

System rozpoznawania logotypu został uruchomiony dla czterech obrazów wejściowych. Analizowane były dwa największe pod względem powierzchni wysegmentowane obiekty. Dla nich dokonano oceny, czy znajduje się w nich *logo*, czy *napis* logotypu.

obraz	obiekt 1 - logo	obiekt 1 - napis	obiekt 2 - logo	obiekt 2 - napis
1.jpg	55.9538%	59.2755%	44.2359%	39.7206%
2.jpg	41.4008%	34.0128%	55.1634%	58.1096%
3.jpg	44.294%	39.835%	11.2504%	34.9365%
4.jpg	44.3518%	39.9284%	43.2932%	37.886%

Tablica 2: Otrzymane wyniki klasyfikacji

Na szczególną uwagę zasługuje obiekt pierwszy na pierwszym obrazu. Klasyfikator zwrócił **największe wartości prawdopodobieństwa**, co świadczy o dużej szansie na istnienie całego logotypu. Obiekt poddany tej klasyfikacji zaprezentowany na rysunku 11. W

kontekście reszty wyników ten wynik można uznać za zadowalający.



Rysunek 11: Obiekt poddany klasyfikacji.

W przypadku trzeciego obrazu widoczna jest wyraźna różnica pomiędzy przypisaniem obiektu do dwóch klas. Klasyfikator zwraca prawie trzykrotnie większe prawdopodobieństwo, że obiekt ten zawiera napis. Dla tego obiektu logo zostało ucięte i zniekształcone, co pogarsza jakość klasyfikacji 12.



Rysunek 12: Obiekt poddany klasyfikacji.

7.1 Wnioski

Utworzony klasyfikator wymaga poprawy jakości klasyfikacji. W tym celu należałoby zadbać o większą liczbę reprezentatywnych obiektów przedstawiających logo i napis logotypu oraz wytrenować model wyliczonymi wartościami niezmienników momentowych.

Istotne jest również określenie klasy „szum”. Zbyt mała liczba próbek obiektów powoduje mniej korzystne wyniki klasyfikacji.

W tworzonym systemie korzystne byłoby podejście statystyczne do przestrzeni cech, w której dokonuje się klasyfikacji. Określenie obszarów cech, wraz z odchyleniami standardowymi oraz inne miary odległości pozwoliłyby na skonstruowanie dokładniejszego modelu.