# Tablice

*Aleksander Kosma*
*22.01.2020*

## Opis rozwiązania

Działanie programu opierałem na odrzucaniu niepasujących elementów po odpowiednich obróbkach graficznych. Następnie pozostałe elementy oceniałem na podstawie 3 parametrów. Na koniec wybierałem najlepszy możliwy element.

Przedstawię teraz kolejne procesy pomagające w detekcji tablicy w kolejności zgodnej z algorytmem:

- wyliczenie minimalnych i maksymalnych potencjalnych rozmiarów tablicy. Przyjęte zostały stałe:
    - 9% szerokości * 9% wysokości oryginalnego obrazu jako minimalne pokrycie powierzchni przez tablice
    - 25% szerokości * 25% wysokości oryginalnego obrazu jako maksymalne pokrycie powierzchni przez tablice
- ustalenie rozmiaru rozmycia dla rozmycia typu ***wiener2*** poprzez podzielenie wysokości przez 100, co daje wartości w okolicach od 1 do 10 rosnąco z rozmiarem obrazu.
- rozmycie obrazu z wykorzystaniem ***wiener2*** a następnie ***medfilt2***
- binaryzacja z progiem 100. Domyślny próg mógłby obcinać strefy blisko tablicy.
- usunięcie wszystkich ‘wysp’ większych niż ustalona górna liczba pixeli
- usuniecie wszystkich ‘wysp’ mniejszych niż ustalona dolna liczba pixeli
- z wykorzystaniem metody ***regionprops*** uzyskanie informacji o każdej z pozostałych wysp
- dla każdej z wysp sprawdzenie proporcji prostokąta(***boundingBox***) przykrywającego wyspę
- wyliczenie pokrycia białymi pixelami na całym prostokącie
- odrzucenie wszystkich wysp, które nie mieszczą się w mało restrykcyjnych wymaganiach:
    - $3 > proporcje > 6.6$
    - $pokrycie > 0.2$

Co pozwala odrzucić wyspy z bardzo małą szansą na bycie tablicą.

- dla pozostałych wysp wykonanie operacji ***bwconvhull***, która tworzy najmniejszą otoczkę w około z każdej z nich.
- ponowne wykonanie operacji ***regionprops***
- ustalenie idealnych parametrów jako porządane:

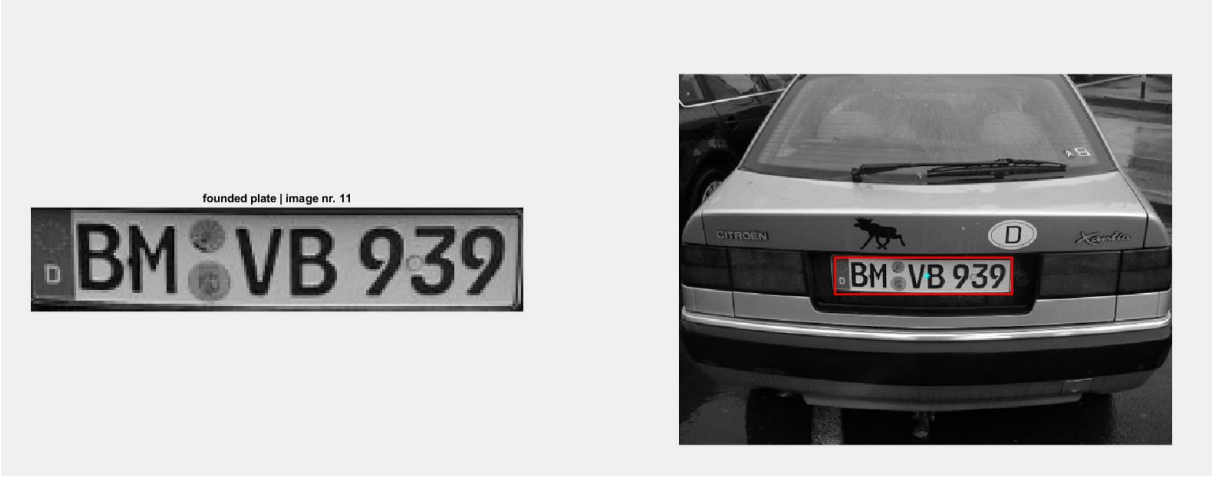
- najlepsze proporcje: *5.2*
- najlepsze zaokrąglenie: *2*
- najlepsze pokrycie: *0* (procentowa ilość czarnych pixeli w obejmującym prostokącie)

- dla każdej wyspy wyliczenie zaokrąglenia, różnicy bezwględnej między proporcjami wyspy, a najlepszymi proporcjami. To samo z zaokragleniem oraz wyliczenie pokrycia.
- zsumowanie wartości tworzących ostateczny ‘wynik’ wyspy, gdzie tym niższy wynik, tym lepszym kandydatem jest wyspa na tablice:

***wynik = pokrycie + różnicaZaokrągleń + różnicaProporcji***

- wybranie najlepszego kandydata z listy

Przykładowy resultat wraz z wykryciem:




Przykładowy moment wyliczania wniku dla znalezionego fragmentu:


Podane liczby oznaczają:

| **krągłość** | **proporcje** | **pokrycie zerami** | **końcowy wynik** |
|---|---|---|---|
| **2.3183** | 4.7679 | 0.17033 | **0.92073** |

**0.92073** = 0.17033 + abs(2.3183 – 2.0) + abs(4.7679 – 5.2)

# Ilościowa i jakościowa analiza wyników

Podstawową zebraną statystyką jest procent udanych detekcji tablic. Podzieliłem je na 3 kategorie:

- 1 - detekcja bez erozji i dylatacji dla wszystkich zdjęć z paczki
- 2 - detekcja z dodatkowym wykorzystaniem ***erozji*** i ***dylatacji*** na poziomie 1 pixela dla wszystkich zdjęć z paczki
- 3 - detekcja bez erozji i dylatacji, tylko na zdjęciach zgodnych z założeniami( bez tablic kwadratowych, pod dużym kątem, zabrudzonych, bardzo ciemnych, o małych rozmiarach tablic)

| Typ | Ilość próbek | Poprawna detekcja | Błędnia detekcja | Procentowa poprawna detekcja |
|---|---|---|---|---|
| **1** | 450 | 301 | 149 | **67%** |
| **2** | 450 | 269 | 181 | **60%** |
| **3** | 396 | 289 | 102 | **73%** |

Poniżej tabela z uśrednionymi czasami kalkulacji algorytmu dla jednego zdjęcia w zależności od rozmiarów. Czas liczony byl każdorazowo bezpośrednio przed wczytaniem zdjęcia, a kończył się przed decyzją użytkownika czy wykryto tablice. Na koniec wyliczona została średnia ze wszystkich pomiarów:

| Rozmiar zdjęcia | Ilość próbek | Średni czas dla jednego zdjęcia | Procentowa poprawna detekcja |
|---|---|---|---|
| **480x640** | 396 | 0.0911 | 73% |
| **384x512** | 396 | 0.811 | 71% |
| **288x384** | 396 | 0.0689 | 68% |
| **192x256** | 396 | 0.0621 | 61% |

Zmniejszająca sie poprawna detekcja prawdopodobnie wynika z zatracania szczegółu czarnej obramówki na tablicy rejestracyjnej, która oddziela tablice od reszty auta.

# Wnioski, propozycje ulepszeń

Otrzymane wyniki są zadowalające, biorąc pod uwagę różnorodność danych wejściowych. Zdjęcia różniły się w wielu aspektach takich jak odległość, oświetlenie, kąty nachylenia, czy rozmiary tablic.

Wnioski z czego wynikały niepowodzenia oraz spostrzerzenia:

- duże odblaski i połyski na lakierze powodowały ‘mieszanie się’ z tablicą tworząc nieprawdziwą wyspę, lub tworząc oddzielne, czasami przypominajace kształtem tablice.
- słabe oświetlenie czy wieczorowa pogoda powodowały przyciemnianie tablic rejestracyjnych, przez co czasami nawet na poziomie binaryzacji zostawały one odrzucane. Słabe oświetlenie powodowało ujednolicenie obrazu przez co ciężko było oddzielić charakterystyczne elementy od siebie.

- białe / srebrne lakiery aut również ułatwiały mieszanie się tablicy z fragmentami auta znajdującymi sie dookoła.
- wiele detali na drugim planie mogły czasami mieć podobne proporcje do tablic.
- odbicia białego nieba w tylnym oknie auta po binaryzacji przypominały kształtem tablice.
- brudne tablice często powodowały podzielenie tablicy na fragmenty przez co zostawały odrzucane na poziomie usuwania małych wysp.
- zaskakująco dobrze algorytm radził sobie ze zdjęciami pod większymi kątami niż założenie projektu.

Propozycje ulepszeń:

- jeśli program mógłby założyć fakty odnośnie środowiska gdzie jest robione zdjęcie ułatwiło by to odrzucenie potencjalnych złych kandydatów. Stały dystans od auta, ta sama pogoda, lub stały drugi plan mogłby by pomóc w odrzuceniu fragmentów ‘w ciemno’ z oryginalnego obrazu.
- próba wykorzystania detekcji krawędzi, operając się na fakcie, że tablice mają sporo pionowych lini. Możnaby to bylo wykorzystać na jednym z etapów odrzucania potencjalnych wysp nie spełniających np. Ilości/proporcji pionowych krawędzi do poziomych.
- wykorzystanie operacji manipulującymi jasnością/kontrastem/intensywnościa kolorów do dobrania takich by uwydatnić informacje o tablicy a przykryć inne.