

PRZETWARZANIE CYFROWE OBRAZÓW (POBR)

Projekt - rozpoznawanie loga mBanku

Napieralski Adam

4 czerwca 2020

1 Temat projektu

Dla indywidualnie wybranej klasy obrazów należało dobrać, zaimplementować i przetestować odpowiednie procedury wstępnego przetworzenia, segmentacji, wyznaczania cech oraz identyfikacji obrazów cyfrowych. Powstały w wyniku projektu program powinien poprawnie rozpoznawać wybrane obiekty dla reprezentatywnego zestawu obrazów wejściowych. Wybrana klasa obrazów to zdjęcia zawierające logo mBanku (Rys. 1).



Rys. 1: Logo mBanku

2 Wykorzystane narzędzia

Projekt od strony technicznej został wykonany w języku *C++*, posługując się ograniczonym zbiorem funkcji i obiektów z biblioteki *OpenCV* (bez funkcji przetwarzania obrazu, wyłącznie podstawowe wczytywanie, zapis, wyświetlanie i struktury danych). Do budowy i komplikacji projektu wykorzystano narzędzie *CMake*. Instrukcja budowy i uruchomienia projektu znajduje się w pliku *README.md* w plikach źródłowych.

3 Charakterystyka wybranego logo

Wybrane logo mBanku ma kształt prostokątny, złożone jest z:

- pionowych pasów koloru: czerwonego, żółtego, zielonego, czarnego, niebieskiego;
- umieszczonego na nich białego napisu *mBank*.

Jako najbardziej charakterystyczne cechy obrazu wybrane zostały trzy kolorowe pasy o największej powierzchni - czerwony, żółty i zielony. Ich bliskie siebie położenie oraz względnie niepowtarzalny kształt związany z nachodzeniem na nie białego napisu będą przydatne na etapie dalszego przetwarzania i ich wykrywania.

4 Akwizycja obrazów wejściowych

Jako zdjęcia wejściowe do tworzonego algorytmu przetwarzania i wykrywania logo wykorzystano obrazy:



(a) Obraz a



(b) Obraz b



(c) Obraz c



(d) Obraz d

Rys. 2: Wybrane obrazy wejściowe

- naturalne - nie wygenerowane syntetycznie,
- z niejednorodnym tłem.

Jako obrazów użyto zdjęć pochodzących przede wszystkim z *Google Street View*. Wybrano 4 obrazy (Rys. 8d) - na dwóch z nich znajduje się po jednym logo, na pozostałych znajdują się po 2 i po 3.

5 Zarys algorytmu przetwarzania

W zaproponowany algorytmie wykrywania logo mBanku można wyróżnić 4 podstawowe kroki przetwarzania obrazu:

1. Konwersja przestrzeni barw.
2. Segmentacja.
3. Wyznaczanie cech.
4. Identyfikacja.

Każdy z nich zostanie bliżej omówiony, przedstawione zostaną wybrane szczegóły implementacyjne oraz wnioski wyniesione na podstawie wykonanych prac.

6 Przetwarzanie wstępne

Do etapu przetwarzania wstępnego zaimplementowano algorytmy rozmycia i wykrywania krawędzi obrazu - przy pomocy filtru dolno- i górnoprzepustowego. Na etapie eksperymentalnym zaobserwowano jednak, że obrazy wejściowe w swojej domyślnej formie lepiej sprawdzają się w detekcji, przez co użycie filtrów stało się niekonieczne, przez co we właściwym procesie nie zostały wykorzystane. Efekty ich działań w celach poglądowych zaprezentowano jednak na Rys. 3.



(a) Filtr dolnoprzepustowy - rozmycie



(b) Filtr górnoprzepustowy - zaznaczenie krawędzi

Rys. 3: Efekty działania zaimplementowanych filtrów na obrazie d

7 Konwersja przestrzeni barw

Wejściowy obraz zapisany jest w formacie *RGB* (a właściwie *BGR* - z racji wykorzystania wczytywania z użyciem biblioteki *OpenCV*). W celu łatwiejszej selekcji interesujących pikseli na etapie progowania zdecydowano się na konwersję przestrzeni barw do przestrzeni *HSV*.

Zaimplementowany algorytm konwersji bazuje na odpowiadającej mu metodzie z *OpenCV*.

$$V = \max(R, G, B) \quad (1)$$

Kolejnym krokiem algorytmu, jest obliczenie nasycenia koloru *S*, korzystając ze wzoru 2.

$$S = \begin{cases} 0, & V = 0 \\ \min(R, G, B), & V \neq 0 \end{cases} \quad (2)$$

Ostatnim krokiem algorytmu jest obliczenie barwy *H* zgodnie z wzorem 3.

$$H = \begin{cases} \frac{(G-B)*60}{V-\min(R,G,B)} + 60, & V = R \\ \frac{(B-R)*60}{V-\min(R,G,B)} + 120, & V = G \\ \frac{(R-G)*60}{V-\min(R,G,B)} + 240, & V = B \end{cases} \quad (3)$$

Tak uzyskane wartości parametrów *S* i *V* zawierają się w przedziale [0, 255], natomiast parametr *H*, z racji przechowywania go jako 8-bitowej liczby, skalowany jest z typowego zakresu [0, 359] do [0, 179]. Operacja konwersji przestrzeni barw jest operacją punktową, przez co może być realizowana na każdym pikselu osobno, niezależnie od innych, sąsiednich. W programie za ten etap przetwarzania odpowiadają klasy: *BGR2HSVConverter* i *BGR2HSVPixelConverter*.

8 Segmentacja

Na etapie segmentacji skorzystano z dwóch jej różnych typów, mających w końcowym efekcie oznaczyć obszary obrazu jednorodne pod określonymi cechami.

8.1 Segmentacja punktowa - progowanie

W celu wyselekcjonowania pikseli należących do jednego z trzech wybranych obszarów zainteresowania należących do loga zastosowano metodę progowania na wcześniej przekonwertowanym do przestrzeni HSV obrazie. Drogą eksperymentalną wybrano progi dla każdej z trzech rozważanych barw - przedziały wartości parametrów *H*, *S*, *V* - które przedstawiono w Tabeli 1.

Przetwarzanie progowania zawarte zostało w plikach *Binarization*, a główną funkcją je realizującą jest *inRange* - analogiczna do funkcji o tej samej nazwie z biblioteki *OpenCV*. Z funkcji zwracany

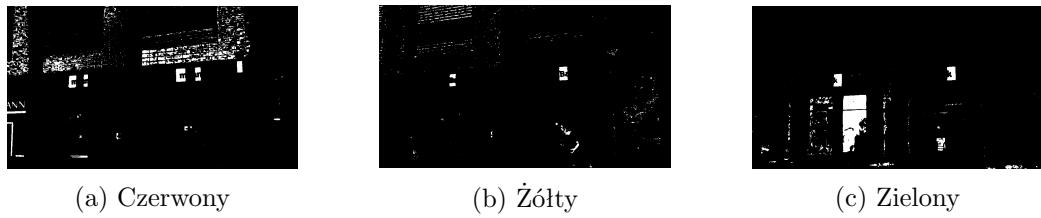
Kolor	H_{\min}	H_{\max}	S_{\min}	S_{\max}	V_{\min}	V_{\max}
czerwony 1	150	179	110	255	90	255
czerwony 2	0	3	110	255	90	255
żółty	9	22	60	255	80	255
zielony	25	87	70	255	80	255

Tabela 1: Tabela wybranych progów

jest zbinaryzowany obraz, gdzie piksele należące do wybranego przedziału oznaczone są na biało (wektorem wartości [255, 255, 255]), a pozostałe na czarno ([0, 0, 0]).

Ponieważ dla koloru czerwonego wartości progów dla parametru H przechodziły przez skrajne możliwe wartości (0 i 179), przeprowadzono osobne progowanie dla obu wariantów i tak uzyskane obrazy scalono w jeden wykorzystując operację logicznego *OR* na pikselach.

Wyniki działania progowania przedstawiono na Rys. 4.



Rys. 4: Binarne obrazy uzyskane po zastosowaniu progowania na obrazie d dla poszczególnych kolorów

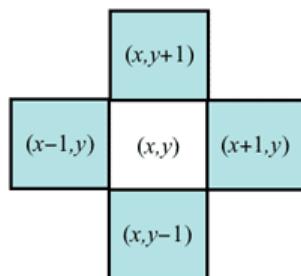
8.2 Segmentacja obszarowa - rozrost obszarów

Na tym etapie przetwarzania z jednostkowe piksele oznaczone na etapie progowania zostały pogrupowane w większe segmenty. Dokonano tego stosując implementując algorytm podążający za ogólną zasadą algorytmów typu *flood fill*.

Rozpoczynając grupowanie od jednego znalezionej zaznaczonego piksela, analizowano rekurencyjnie sąsiadów kolejnych pikseli. Wybranym wariantem sąsiedztwa pikseli został wariant 4-pikselowy (przedstawiony na Rys. 5).

9 Identyfikacja

W celu rozpoznania całego logo oparto się na identyfikacji cech wyznaczonych segmentów składowych: jednego czerwonego, żółtego i zielonego. By to osiągnąć skorzystano z parametrów, które w trakcie analizy dawały najbardziej charakterystyczne dla danych



Rys. 5: Wybrany wariant określający sąsiedztwo piksela

segmentów wyniki. Wybrano: niezmienniki momentowe $NM1$ i $NM7$ oraz współczynnik $W3$ - Malinowskiej, którego wzór wygląda następująco:

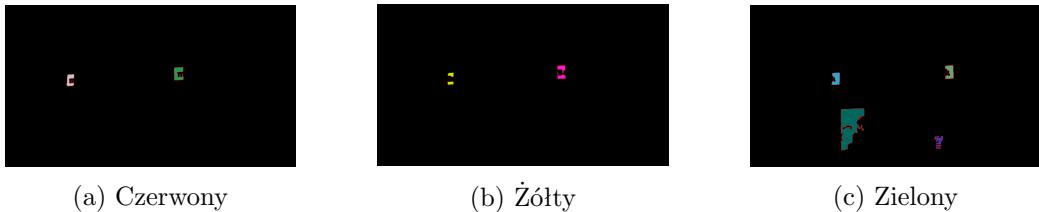
$$W3 = \frac{L}{2\sqrt{\pi}S} - 1 \quad (4)$$

gdzie L - obwód segmentu, S - jego pole.

Dla każdego z nich metodą eksperimentalną wyznaczono przedziały wartości - przedstawione w Tab.. Segment, którego wszystkie wartości współczynników będą należały do tych przedziałów zostanie zaklasyfikowany jako prawdopodobny segment składowy identyfikowanego logo (Rys. 6).

Segment	NM1	NM7	W3
czerwony	[0, 21, 0, 28]	[0, 011, 0, 014]	[0, 66, 1, 85]
żółty	[0, 18, 0, 38]	[0, 005, 0, 022]	[0, 75, 1, 75]
zielony	[0, 18, 0, 34]	[0, 007, 0, 12]	[0, 45, 2, 51]

Tabela 2: Przedziały wartości wybranych współczynników charakterystycznych dla segmentów składowych logo



Rys. 6: Zaznaczone zidentyfikowane segmenty na obrazie d dla poszczególnych kolorów

Ostatecznym krokiem to identyfikacji samego logo jest przeszukanie kolekcji wszystkich wykrytych 3 typów segmentów i określenie położenia logotypu na podstawie względnego położenia 3 typów segmentów. Wykonywane jest to przy pomocy metody *LogoRecognizer::findLogos*, a kryterium poprawnego zaklasyfikowania 3 segmentów jako składowe jednego logo jest wzajemna bliskość ich geometrycznych środków przy współczynniku 1,5, określającym mnożnik długości przekątnej prostokąta opisanego na segmencie, który następnie traktowany jest jako promień koła wyznaczającego obszar sąsiedztwa danego segmentu. Tak pogrupowane segmenty są następnie łączone w jeden, który reprezentuje faktyczne zidentyfikowane logo (Rys. 7).



Rys. 7: Oznaczone segmenty połączone w jeden określający faktyczne zidentyfikowane logo

10 Uzyskane wyniki

W ramach projektu dokonano analizy 4 zdjęć, z czego na dwóch z nich logo pojawiało się wielokrotnie. Na wszystkich udało się rozpoznać wszystkie logotypy mBanku.



(a) Obraz a



(b) Obraz b



(c) Obraz c



(d) Obraz d

Rys. 8: Analizowane obrazy z zaznaczonymi wykrytymi logotypami

Literatura

- [1] Image segmentation <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic3.htm>
- [2] prof. Przemysław Rokita. Wykłady z przedmiotu przetwarzanie obrazów cyfrowych