

Instytut Informatyki UP Kraków

Przetwarzanie obrazów cyfrowych :: projekt zaliczeniowy  
sprawozdanie

Autor: Maksymilian Górka

22 stycznia 2019

## 1. Cel i zakres projektu

Celem projektu jest zliczenie ilości monet znajdujących się na obrazie(zdjęciu), obliczeniu dla nich sumy nominalów. Dodatkowo obliczyć jaki procent obrazu zajmują obiekty, wyznaczyć środek ciężkości obiektów, obliczyć dla każdego obiektu współczynniki Fereta, Blaira-Blissa i Haralicka.

## 2. Wykonane operacje

Do uzyskania satysfakcjonującego rezultatu wykorzystałem kolejno:

- konwersję obrazu na odcienie szarości,
- filtr dolnoprzepustowego (blur),
- binaryzację li z automatycznym progowaniem,
- operację morfologicznego zamknięcia,
- filtr Canny'ego do wykrycia krawędzi oraz erozji do rozdzielenia obiektów,
- segmentację z indeksacją (funkcja własna).

W projekcie korzystałem głównie z bibliotek skimage, matplotlib, opencv oraz numpy. Podczas obliczania współczynników użyłem funkcji z biblioteki math oraz funkcji omawianych na zajęciach.

### 3. Dane wejściowe

W celu sprawdzenia poprawności algorytmu wykorzystałem 15 zdjęć, które zostały wykonane z tej samej odległości. Na niektórych zdjęciach obiekty są ze sobą złączone (stykają się). Oto dwa przykładowe obrazy:

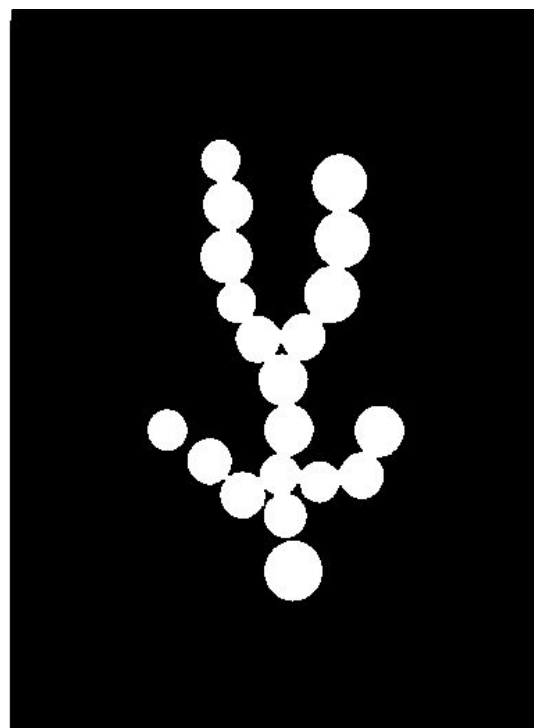
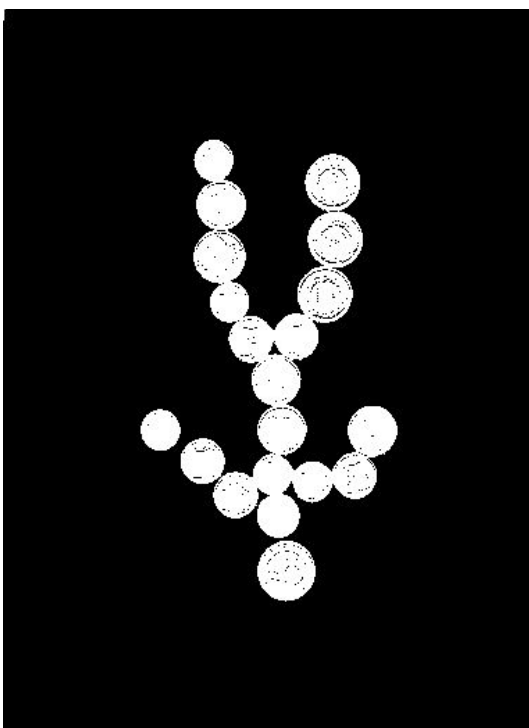


## 4. Opis działania

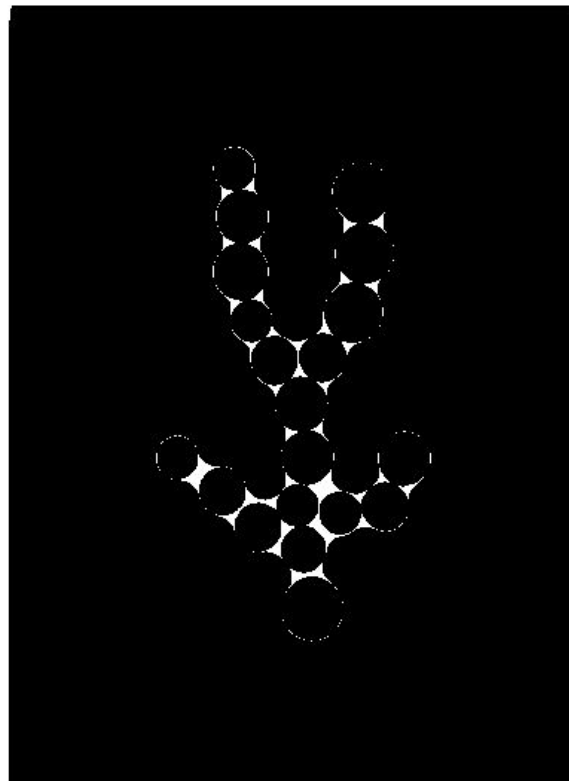
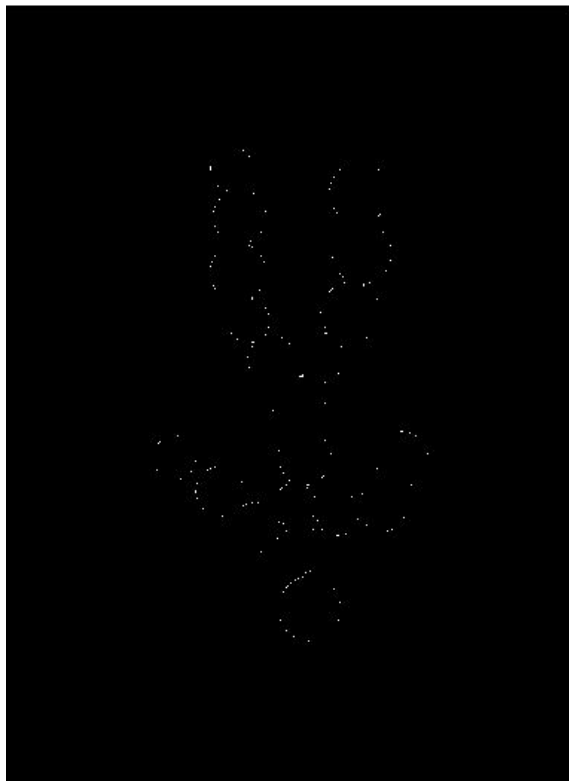
I. Wczytanie obrazu oraz konwersja do grayscale (z dodatkowym rozmyciem):



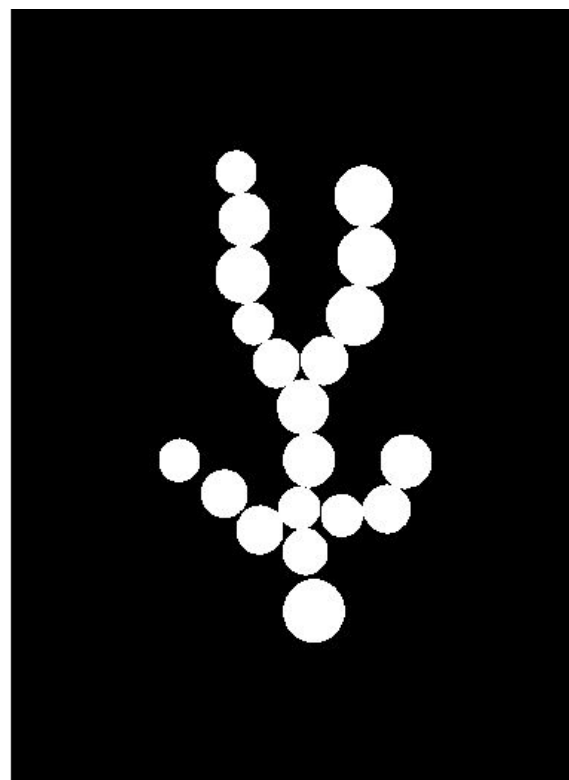
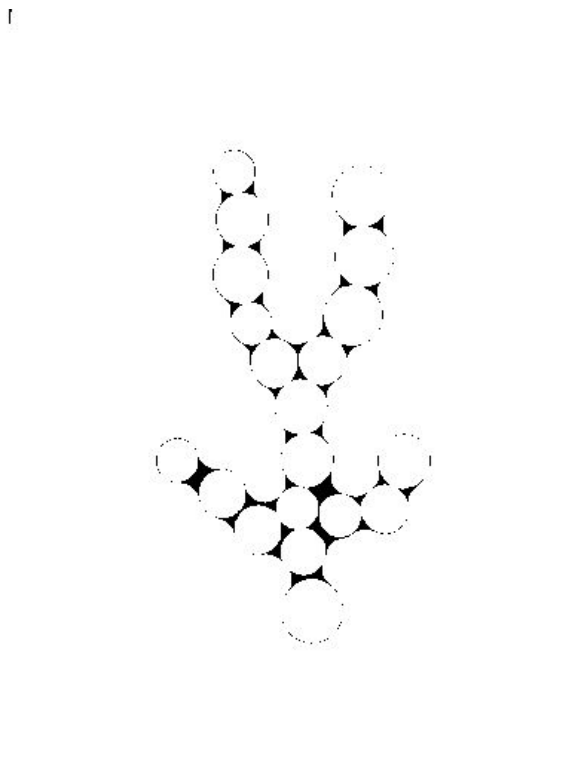
II. Binaryzacja obrazu oraz wykonanie operacji zamknięcia:



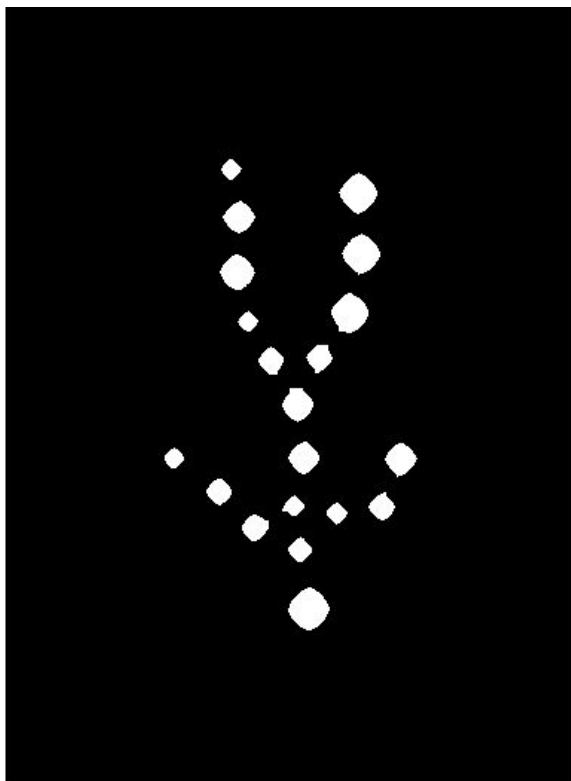
III. Użycie filtra Canny'ego oraz operacji zamknięcia:



IV. Negacja obrazu z wykrytymi krawędziami oraz połączenie z obrazem binarnym po erozji:



- V. Kolejna erozja obrazu aby rozdzielić resztę obiektów oraz porównanie z obrazem źródłowym:



- VI. Obraz po segmentacji z indeksacją oraz ostateczny wynik działania algorytmu:



## 5 Wyniki eksperymentu

Sprawność algorytmu liczącego obiekty wynosi 100% na wszystkich przedstawionych obrazach. Niestety algorytm liczący kwotę ze zdjęcia popełnia czasem błędy co przedstawiłem w poniższej tabelce.

Nr	Obiekty			Kwota		
	Zdjęcie	Algorytm	%	Zdjęcie	Algorytm	%
068	20	20	100%	13.52	13.52	100%
069	8	8	100%	4.60	4.60	100%
070	16	16	100%	10.10	10.10	100%
071	20	20	100%	15.50	15.50	100%
072	20	20	100%	15.50	15.50	100%
073	20	20	100%	15.50	15.50	100%
074	34	34	100%	15.77	15.77	100%
075	24	24	100%	1.77	1.64	92.7%
076	24	24	100%	11.37	11.37	100%
077	30	30	100%	14.97	15.03	99.6%
078	21	21	100%	7.07	7.06	99.9%
079	22	22	100%	12.07	12.07	100%
080	24	24	100%	13.27	13.27	100%
081	23	23	100%	11.27	11.27	100%
082	22	22	100%	8.75	8.75	100%

## 6 Wnioski

Analizując działanie algorytmu, mogę stwierdzić że dla dobrze przygotowanych zdjęć pod względem oświetlenia, kontrastu między obiektami a tłem oraz braku obiektów pokrywających się, algorytm policzy ilość obiektów bezbłędnie, jednak dla stykających się monet o podobnej wielkości algorytm potrafi wprowadzić mały błąd. Błędy powodowane są głównie przez erozję która dla okrągłej monety potrafi sprowadzić obiekty o małej różnicy wielkości (przykładowo dwugroszówka oraz dwudziestogroszówka lub dwudziestogroszówka i pięciogroszówka) po bardzo podobnych przedziałów (a czasem nakładających się przedziałów).

Przykładowe błędne rozwiązanie dla obrazu 075:

