

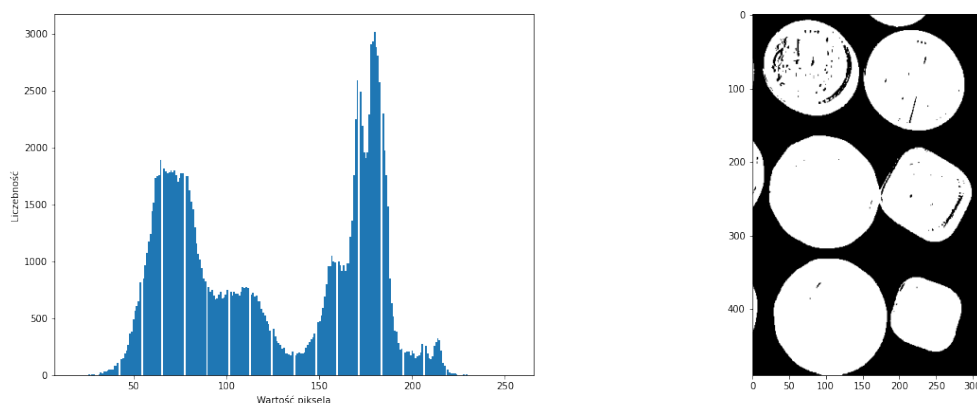
# Analiza i przetwarzanie obrazów i wideo - laboratorium 2

Michał Marciniak 244811

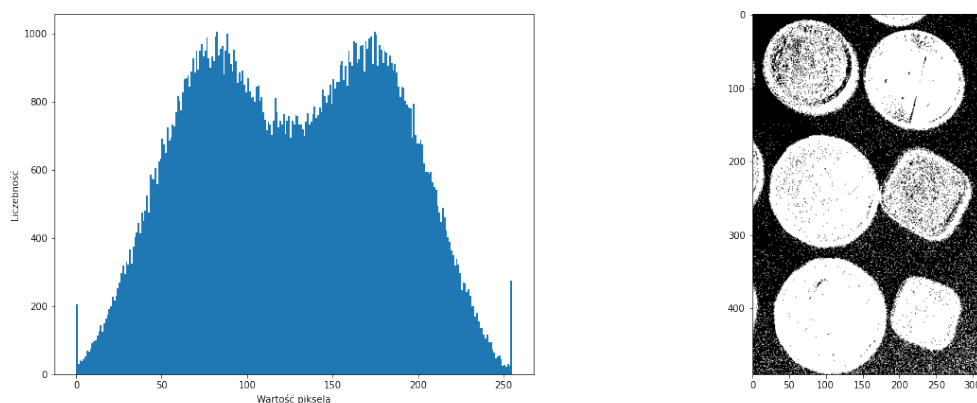
17 października 2021

## 1 Zadanie nr 1

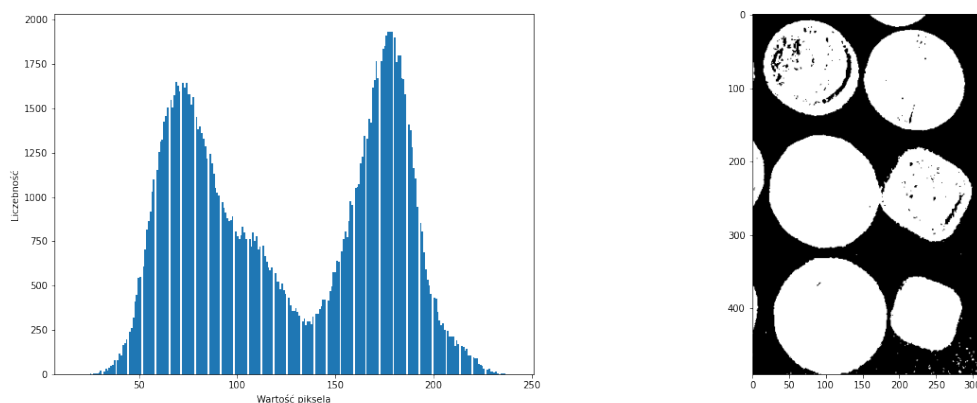
Zadanie polegało na binaryzacji dwóch czarno-białych obrazów za pomocą progowania. Aby znaleźć wartość progu wykonano histogramy obu obrazów. W przypadku pierwszego obrazu od razu widać 2 separowalne wartości modalne. Natomiast drugi obraz jest lekko zaszumiony. W celu jego odszumienia zastosowano filtr uśredniający o rozmiarze  $3 \times 3$  z wartościami filtru wynoszącymi  $1/9$ . Binarystacja pierwszego obrazu została przeprowadzona z wartością progu wynoszącą 135, a jej wynik przedstawiony jest na rys. 1. Na rys. 2 przedstawiono histogram obrazu przed filtrowaniem oraz wynik jego binaryzacji z progiem 135. Odpowiednie wyniki po zastosowaniu filtrowania przedstawiono na rys. 4. Jak widać zastosowanie filtru uśredniającego skutecznie poradziło sobie z zminimalizowaniem wpływu szumu na proces binaryzacji. Efekt odszumiania jest bardzo dobrze widoczny w postaci histogramów, ponieważ możemy zaobserwować 'przeniesienie' wartości odpowiadających za szum w stronę którejś z dwóch wartości modalnych.



Rysunek 1: Pierwszy obraz. Od lewej: histogram obrazu, efekt binaryzacji



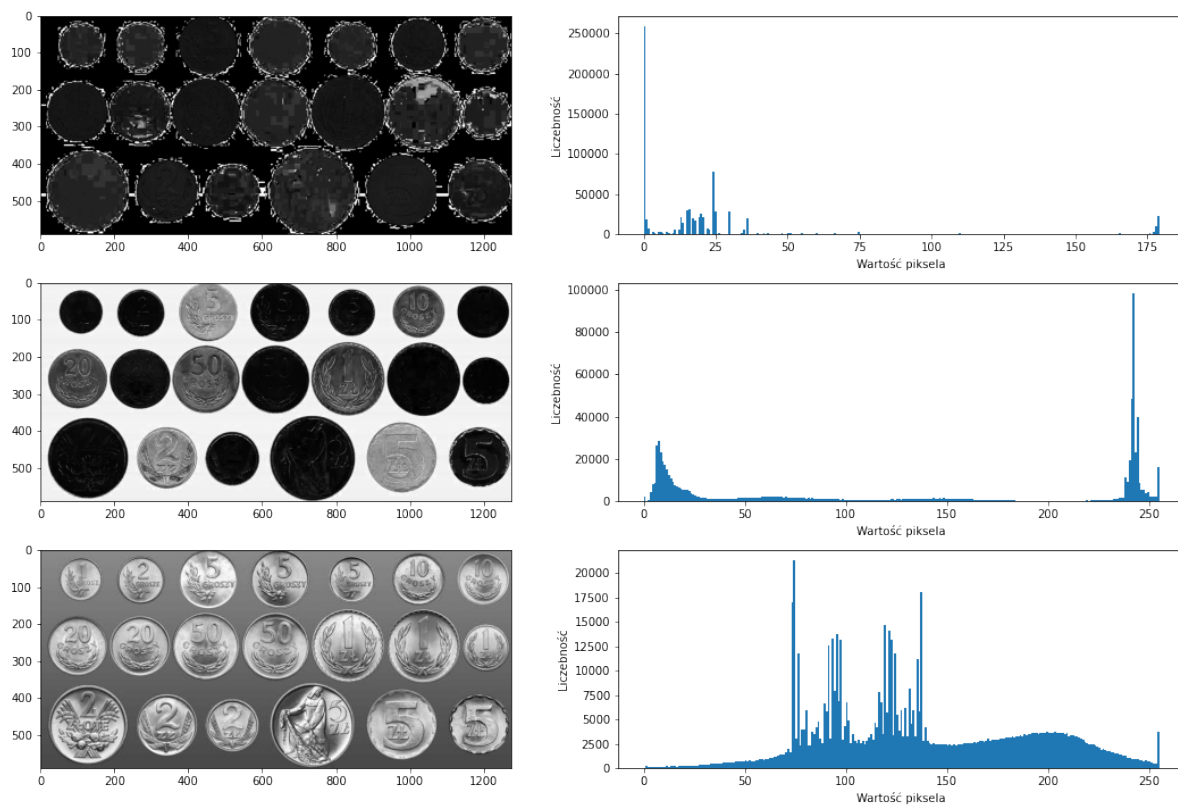
Rysunek 2: Drugi obraz. Od lewej: histogram obrazu, efekt binaryzacji



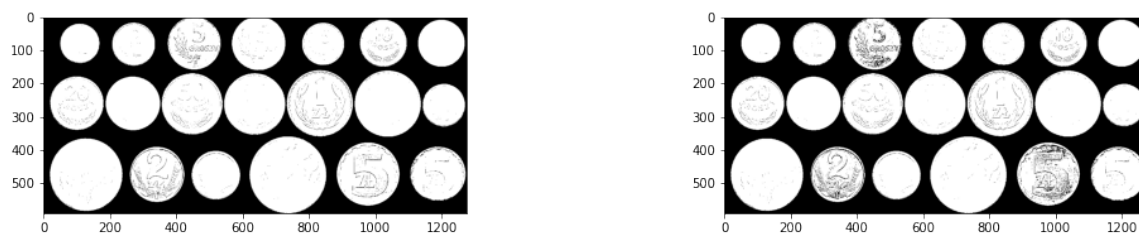
Rysunek 3: Drugi obraz. Od lewej: histogram obrazu, efekt binaryzacji

## 2 Zadanie nr 2

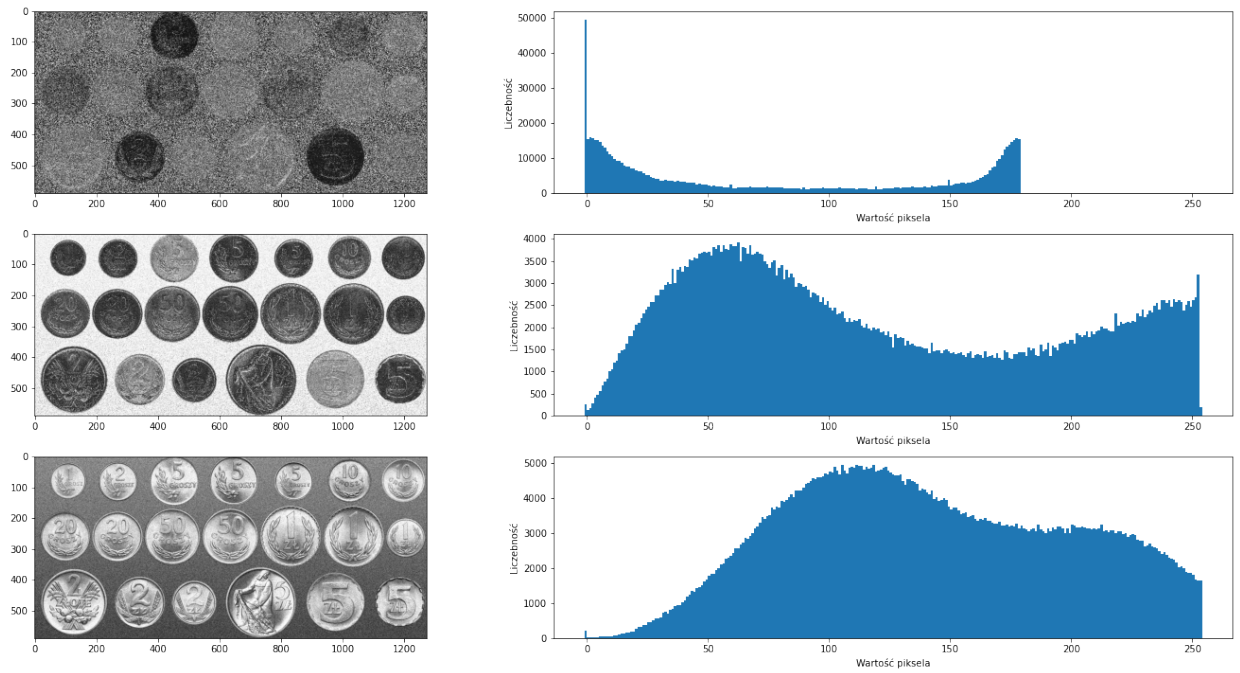
To zadanie polegało na tym samym co zadanie 1, tj. przeprowadzenie binaryzacji obrazów przedstawiających kolekcję monet. Jednakże w tym przypadku mieliśmy do czynienia z kolorowymi obrazami. W celu znalezienia wartości progu, dokonano konwersji w przestrzeni barw z RGB na HSV. Następnie wyznaczono histogram każdego z kanałów, a jako wzorec dla wyznaczenia wartości progu wybrano kanał S (Saturation). Poniższe wykresy obrazują dlaczego nadawał się on najlepiej. Wynika to z możliwości wyszczególnienia dwóch wartości modalnych w jego histogramie. W przypadku pierwszego obrazu jako wartość progu wybrano 200, a wynik operacji binaryzacji przedstawiono na obrazie 5. Przeprowadzono również binaryzację dla wartości progu równej 170. Analizując histogram kanału S z rys. 4 można założyć, że wartość 170 powinna być wystarczająca. Jednakże przybliżając na zakres 150-230 można dostrzec, że wartość 200 zapewnia lepszą separowalność, co widoczne jest również w wynikach binaryzacji. W przypadku drugiego obrazu mieliśmy od czynienia z szumem. Histogramy poszczególnych kanałów HSV zostały zaprezentowane na rys. 6. W celu odsumienia zastosowano filtr Gaussowski o rozmiarach 7x7 oraz 3x3. Histogramy zostały zaprezentowane na rys. 7. Widać, że stosowanie większego rozmiaru filtra skutkuje w powstaniu większej ilości wartości modalnych (tj. 'pagórków'). Na podstawie przeprowadzonych operacji zdecydowano się wykorzystać wartość progową o wartości 180. Wyniki binaryzacji zostały zaprezentowane na rys. 8. Zastosowanie większego filtra pozwoliło zmniejszyć ilość widocznych krawędzi wewnątrz monet, co jest pożądanym działaniem w procesie binaryzacji, ponieważ umożliwia łatwiejsze wykrycie krawędzi całego obiektu.



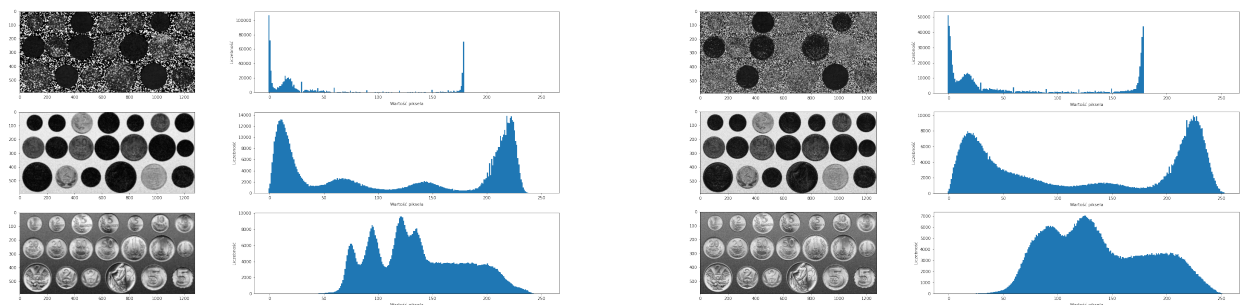
Rysunek 4: Pierwszy obraz. Lewa kolumna: składowe HSV, prawa kolumna: histogramy poszczególnych składowych



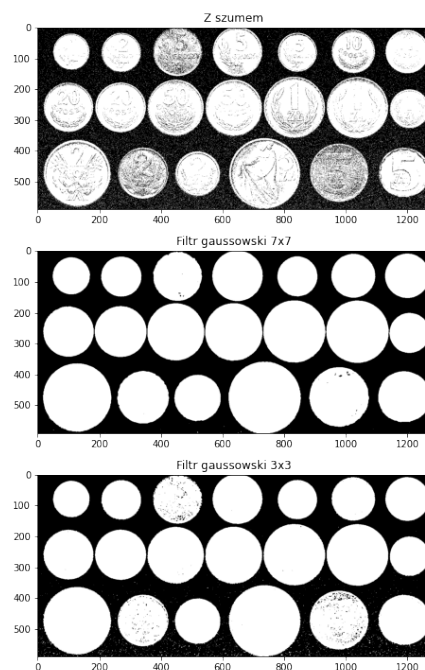
Rysunek 5: Wyniki binaryzacji pierwszego obrazu dla wartości progu 200 i 170.



Rysunek 6: Drugi obraz. Lewa kolumna: składowe HSV, prawa kolumna: histogramy poszczególnych składowych



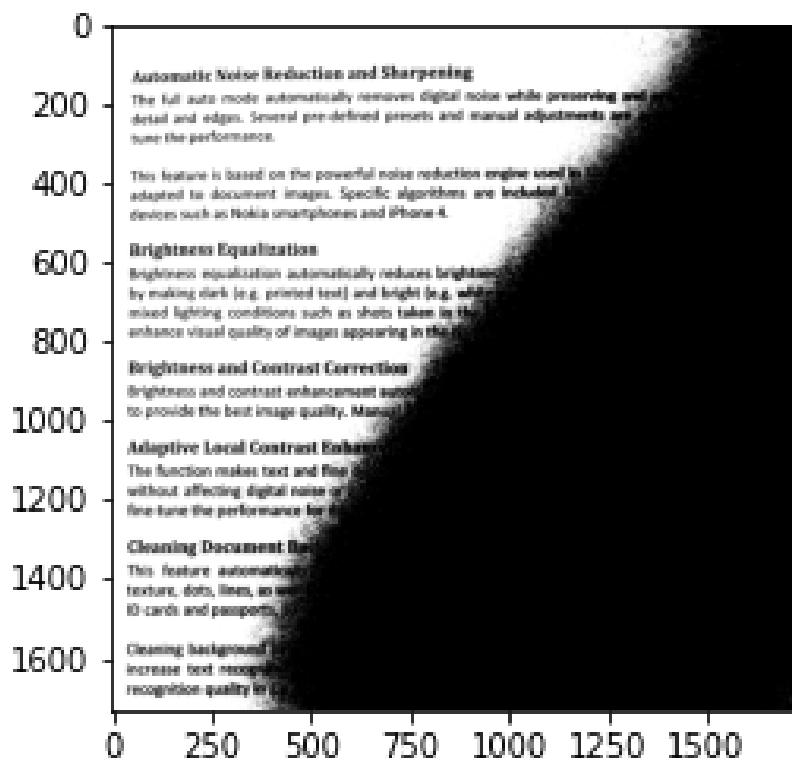
Rysunek 7: Drugi obraz. Histogramy po zastosowaniu filtra Gaussowskiego 7x7 oraz 3x3.



Rysunek 8: Wyniki binaryzacji dla obrazu zaszumionego

### 3 Zadanie nr 3

To zadanie polegało na wyodrębnieniu liter od bardzo nierówno oświetlonego tła. W tym celu wykorzystano algorytm Otsu. Najpierw porównano jego wersję działającą globalnie na całym obrazie, a następnie wersję działającą lokalnie na fragmentach obrazu. Wynik działania globalnej wersji przedstawiono na rys. 9. Jak można zauważyć, nie poradził on sobie najlepiej w tym przypadku. Nie był on w stanie rozdzielić liter oraz artefaktów pochodzących z padającego cienia. Zdecydowanie lepszym pomysłem jest zastosowanie wersji lokalnej. Polega ona na tym, że algorytm działa na fragmentach obrazu otrzymywanych za pomocą okna przesuwanego. Zastosowano okno nienachodzące na siebie. Wyniki tej operacji przedstawiono na rys. 10. Można dostrzec, że mały rozmiar okna powoduje dobre wysegmentowanie całych zdań, ale jednocześnie powoduje powstanie dużych artefaktów wizualnych w miejscu tła. Wraz ze zwiększaniem rozmiaru okna minimalizujemy liczbę artefaktów tła (staje się ono coraz bardziej jednorodne), natomiast zwiększa się szansa, że w obszarze zdania pojawiają się błędy wynikające z uwzględnienia cienia. Gdy zastosujemy zbyt duże okno nasila się wspomniany efekt. Kolejnym etapem badań było wykorzystanie informacji na temat wariancji w obrębie danego fragmentu obrazu. Algorytm Otsu był stosowany dopiero po przekroczeniu określonej wartości wariancji, w myśl zasady, że dla tła (czy innych obszarów jednorodnych) powinna ona być mała. W przypadku gdy wartość wariancji była mniejsza niż próg to wszystkie piksele w danym oknie były zastępowane wartościami 255. Na rys. 11 przedstawiono wyniki badań. Jak widać wybór wartości progu ma duże znaczenie i jest zależny od charakteru przetwarzanego obrazu. Zbyt mała wartość progu spowoduje, że zostaną pominięte obszary które w rzeczywistości stanowią tło. Natomiast zastosowanie zbyt dużej wartości spowoduje usunięcie części zdań.



Rysunek 9: Wynik działania algorytmu Otsu globalnie na całym obrazie.



Rysunek 10: Wynik działania algorytmu Otsu lokalnie - okno nienachodzące.



Rysunek 11: Wynik działania algorytmu Otsu lokalnie dla różnych progów wariancji.