

Komunikacja Człowiek-Komputer			
Rozpoznawanie i sumowanie wartości banknotów na zdjęciach			
Prowadzący: Magdalena Martyn	Autorzy: Mateusz Politycki 145263 Marcin Korcz 144446 Agata Szpotek 145341	Grupa dziekańska:	I7.2



1 Opis problemu:

Rozpoznawanie banknotów należy do problemów klasyfikacji obiektów, w którym danemu elementowi wykrytemu na obrazie należy przypisać pewną etykietę. W przypadku wybranego zagadnienia oczekiwanymi danymi wejściowymi jest zdjęcie, na którym po odpowiednim przetworzeniu zostaną zaznaczone banknoty, a także wypisane ich wartości. Dodatkowo pojawia się również informacja o zsumowanej kwocie odnalezionych przez algorytm pieniędzy.

2 Rozwiązanie problemu:

Aby rozpoznać wartości banknotów na zdjęciu zastosowaliśmy podejście obiektowe, w którym podstawą są dwie klasy Image oraz Banknote posiadające odpowiednie atrybuty i metody. Początkowo załadowane zdjęcie jest odszumiane, a następnie przetwarzane przez algorytm poszukiwania konturów, który skupia się na obiektach mogących być banknotem. Następnie na każdym z wykrytych elementów wycinany jest obszar będący jego lewym górnym rogiem, aby porównać go ze stworzonymi maskami każdej z możliwych wartości. Jeśli nie przypisano mu żadnej wartości banknot jest obracany o 180 stopni i procedura powtarza się ponownie.

Po przetworzeniu całego obrazu i wszystkich obiektów, wykryte banknoty są zaznaczane na zdjęciu oraz podpisywane odpowiadającymi im wartościami. Dodatkowo pojawia się informacja o sumie wszystkich pieniędzy.

3 Poszczególne etapy poszukiwania wartości banknotów na zdjęciu:

W celu szczegółowego prześledzenia działania programu zajmiemy się przykładowym obrazem przedstawionym na rysunku nr 1.



Rysunek 1: Dane wejściowe w postaci zdjęcia banknotów o wartościach 10zł, 20zł oraz 50zł.

3.1 Noise reduction

Początkowo wczytane zdjęcie poddawane jest przetworzeniu przez filtry redukujące poziom szumów i usuwające niepotrzebne zakłócenia.

3.1.1 Skala szarości:

Obraz wczytywany jest na samym początku w postaci RGB oraz w skali szarości, aby poprawić działanie i umożliwić zastosowanie na nim niektórych filtrów.



Rysunek 2: Zdjęcie w skali szarości.

3.1.2 Rozmycie Gaussowskie:

Rozmywa obraz stosując wybrany kernel.



Rysunek 3: Efekty działania rozmycia.

3.1.3 Thresholding:

Zamienia wartości kolorów na obrazie, w wyniku czego jest on czarno-biały.



Rysunek 4: Efekty działania thresholdingu.

3.1.4 Erozja i dylatacja:

Kilkukrotnie zastosowana erozja w celu pozbycia się cienkich obiektów będących niepotrzebnym szumem. Następnie kilkakrotnie zastosowana dylatacja, aby banknoty powróciły do swojej poprzedniej postaci.



Rysunek 5: Efekty działania erozji i dylatacji.

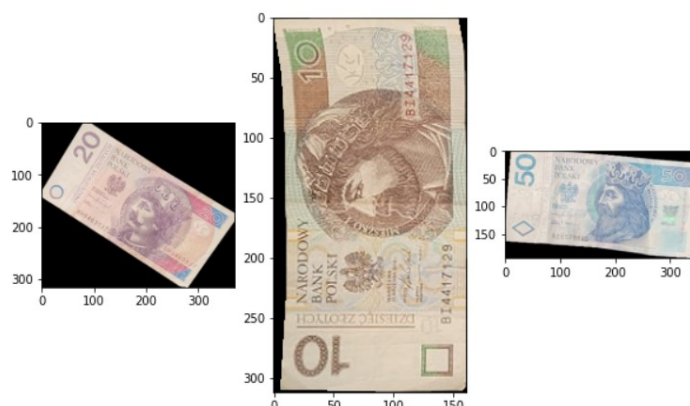
3.2 Poszukiwanie obiektów będących banknotami:

Pracując na wyniku poprzedniego przetworzenia poszukujemy przy pomocy funkcji FIND_CONTOURS wszystkich konturów znajdujących się na obrazie, a następnie wypełniamy ich środki na białą co redukuje liczbę zauważalnych szczegółów.



Rysunek 6: Znalezione obiekty.

Następnie ponownie poszukujemy konturów, jednak tym razem dzięki mniejszej ilości informacji możemy skupić się na odfiltrowaniu elementów, które wymiarami nie mogą odpowiadać banknotom. Dzięki temu uzyskujemy listę obiektów, podejrzanych o bycie banknotami.



Rysunek 7: Obiekty podejrzane o bycie banknotami.

Dodatkowo niektóre z obiektów mogą być ustawione pod kątem i posiadać różne wymiary, co w kolejnym etapie znacząco utrudni działanie programu, dlatego każdy z nich jest dodatkowo skalowany i obracany.



Rysunek 8: Obrócone i odpowiednio przeskalowane obiekty.

3.3 Poszukiwanie interesującej strefy na banknotach:

Dzięki odnalezieniu obiektów podejrzanych o bycie banknotami możemy skupić się na interesującej nas strefie każdego z nich, w której spodziewamy się odnaleźć wartość, czyli lewym górnym rogu. Każdy poszczególny banknot poddawany jest kolejnym filtrom, które pomagają w znalezieniu konturów szczegółów, a następnie zadecydowaniu czy dana strefa jest tą, którą chcemy analizować.

3.3.1 Korekcja Gamma

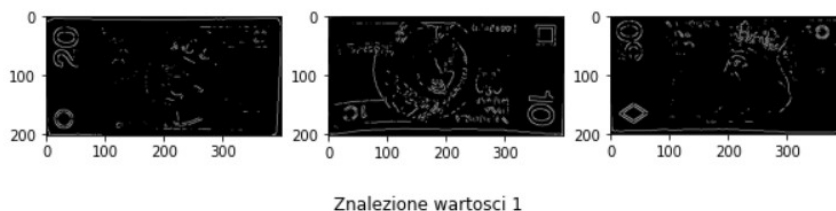
Przetworzenie luminacji na obrazie.



Rysunek 9: Efekty korekcji Gamma.

3.3.2 Canny

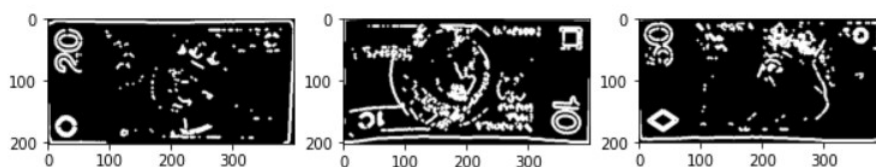
Detekcja krawędzi, w celu odnalezienia interesujących szczegółów.



Rysunek 10: Efekty filtru Cannego.

3.3.3 Dylatacja

Zastosowana w celu zwiększenia widoczności mniej wyraźnych elementów oraz połączenia się konturów wartości banknotu, dzięki czemu nie jest ona traktowana jako pojedyncze liczby, lecz jako całość.

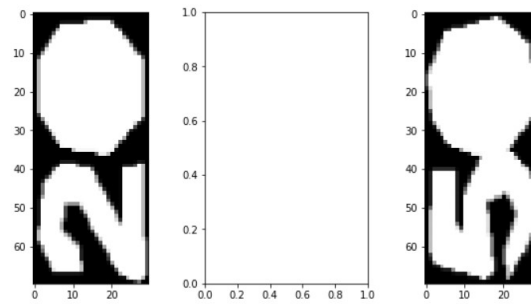


Rysunek 11: Efekty dylatacji.

Po zastosowanych filtrach, skupiamy się na konturach, które są interesującymi nas strefami. W tym celu szukamy najlepiej dopasowanych prostokątów, które wymiarami oraz położeniem zgadzają się z przewidywanymi wartościami (zarówno rozmiar jak i położenie w lewym górnym narożniku).



Rysunek 12: Banknoty z zaznaczonymi strefami I.

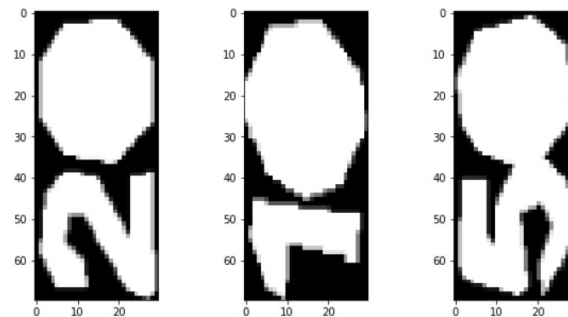


Rysunek 13: Znalezione strefy na banknotach I.

Jeśli któryś z banknotów nie ma wyznaczonej strefy do analizowania może to oznaczać, iż jest on źle obrócony, gdyż zastosowana poprzednio funkcja sprawia jedynie, że jego położenie jest w poziome. Dlatego w takim wypadku obracamy obiekt o 180 stopni i wykonujemy procedure poszukiwania interesującej nas strefy ponownie.⁴



Rysunek 14: Banknoty z zaznaczonymi strefami II.



Rysunek 15: Znalezione strefy na banknotach II.

3.4 Porównanie znalezionych stref z maskami:

Ostatnim krokiem prowadzącym do sklasyfikowania banknotów jest porównanie znalezionych stref ze wszystkimi maskami. Dzieje się to przy pomocy funkcji z biblioteki skimage - `mean_squared_error()`, która liczy średni kwadrat odchylenia między dwoma obrazami. Wartości zapisywane są w liście, a następnie wybierana ta maska, dla której MSE jest najmniejszy.



Rysunek 16: Maski stosowane w klasyfikacji banknotu.

3.5 Oznaczenie banknotów oraz wypisanie ich zsumowanej wartości:

Ostatnim etapem jest obrysowanie znalezionych banknotów, czyli elementów dla których udało się wykryć wartość. Dodatkowo sumowane są znalezione predykcje i wyświetlane jako informacja dodatkowa w konsoli.

```
Banknot o wartosci: 20 zł  
Banknot o wartosci: 10 zł  
Banknot o wartosci: 50 zł  
Wartość gotówki na zdjęciu: 80 zł
```

Rysunek 17: Output w konsoli.



Rysunek 18: Ostateczny wynik - początkowe zdjęcie z zaznaczonymi banknotami i ich wartościami.

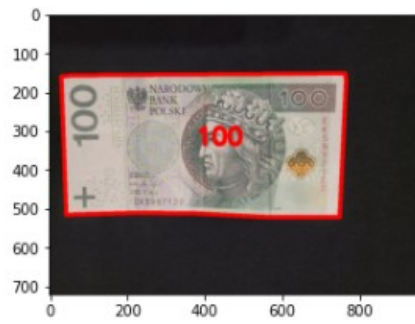
4 Otrzymane wyniki

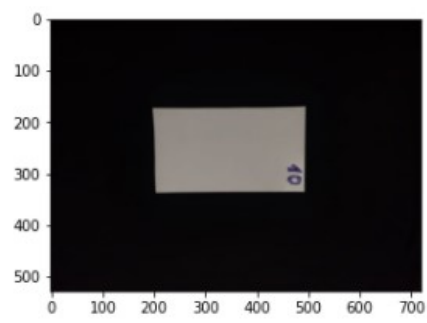
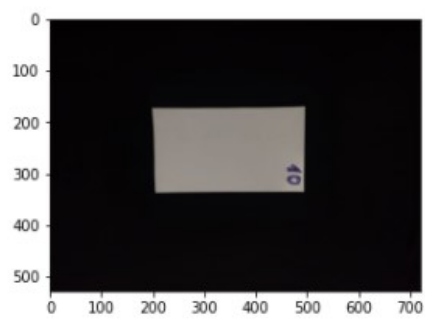
4.1 Przypadki łatwe:

Zdjęcia z pojedynczymi banknotami na jednolitym tle oraz z pojedynczymi banknotami wraz z dodatkowymi obiektami.

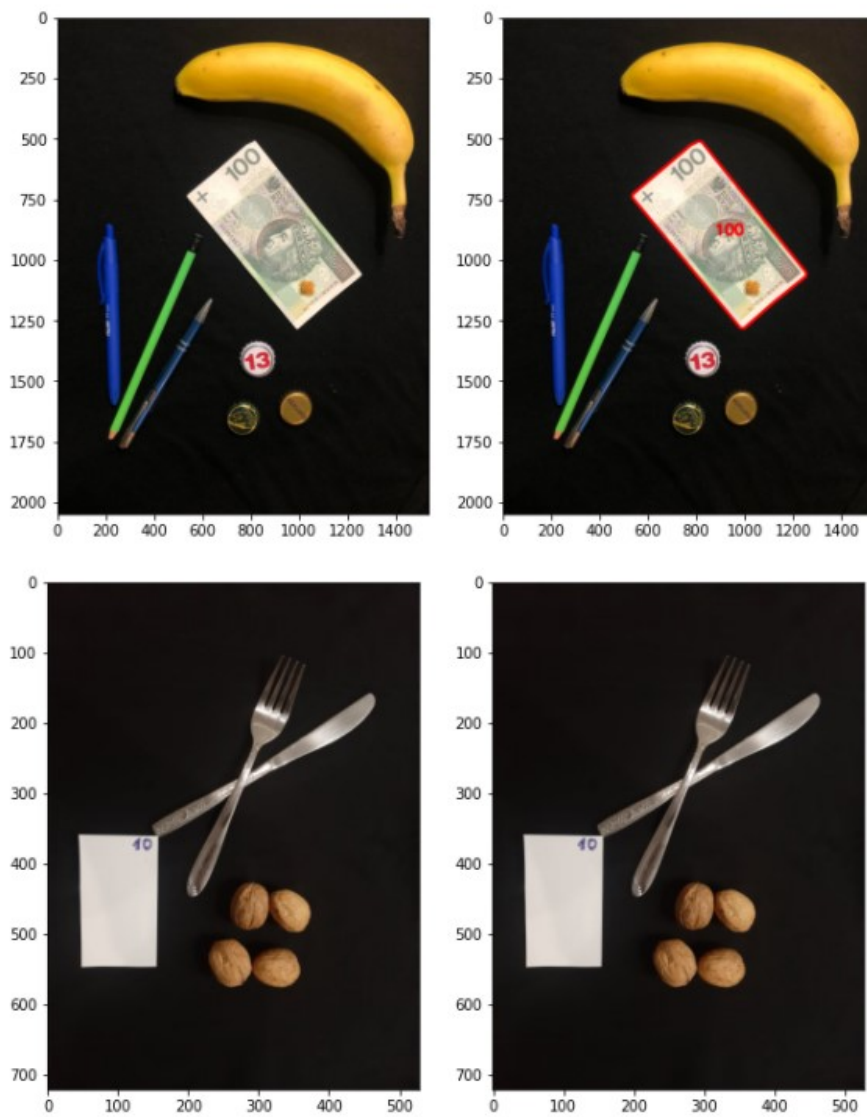
Skuteczność: 100% (8/8 wykrytych banknotów)









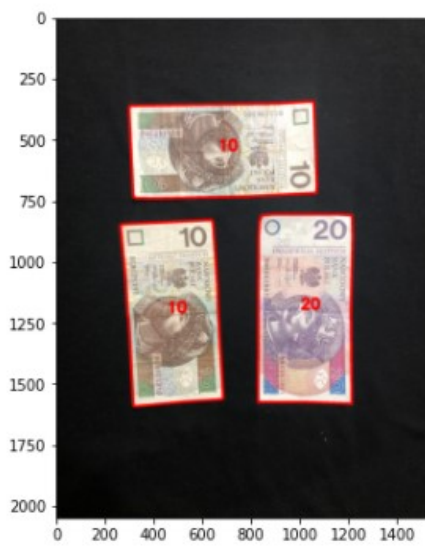


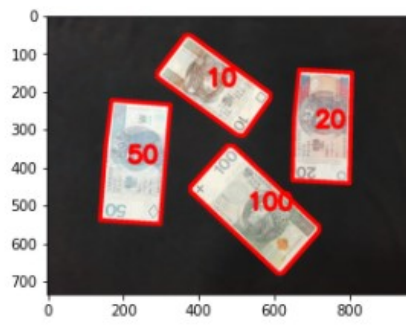
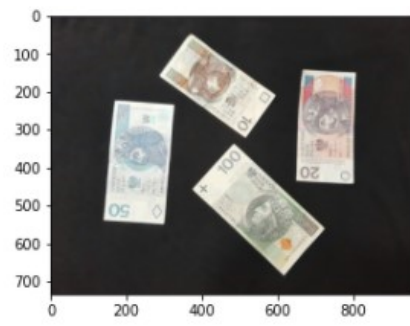
4.2 Przypadki średnie:

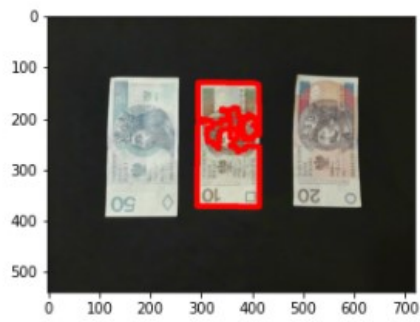
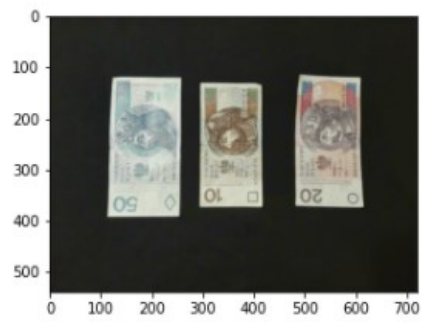
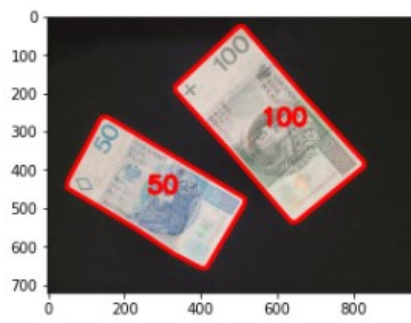
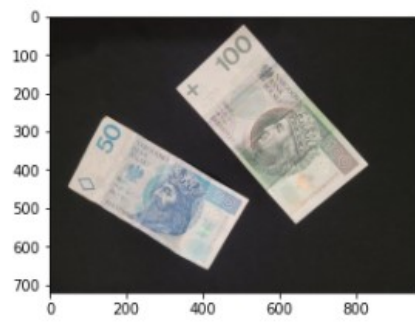
Zdjęcia z wieloma banknotami na jednolitym tle.

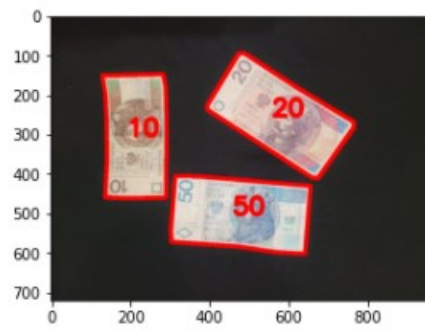
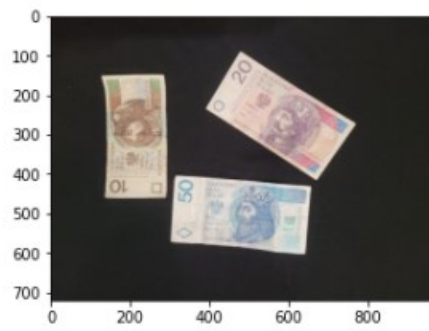
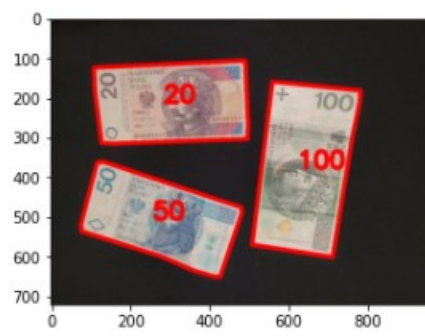
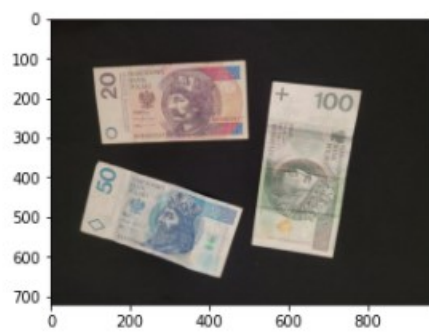
Skuteczność: 89% (33/37 wykrytych banknotów)







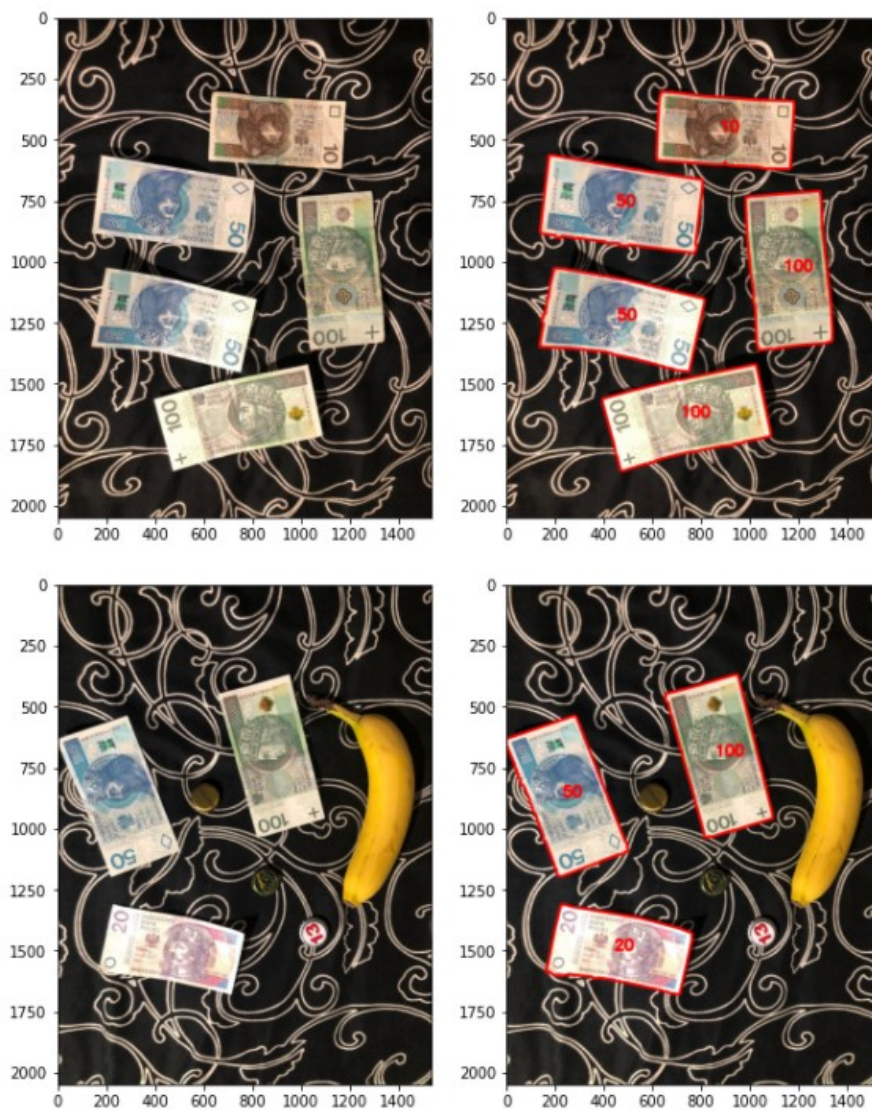




4.3 Przypadki trudne:

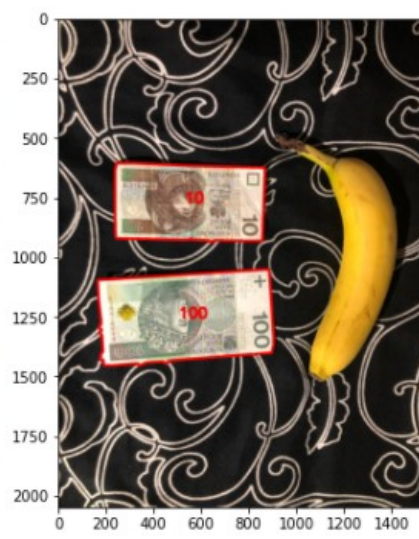
Zdjęcia z wieloma banknotami na niejednorodnym tle oraz z dodatkowymi obiektami.

Skuteczność: 96% (27/28 wykrytych banknotów)









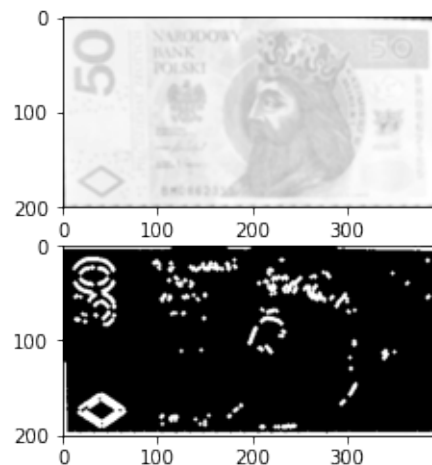


5 Podsumowanie wyników

5.1 Niedziałające przypadki:

5.1.1 Przypadek I:

Na przedstawionym przykładzie widzimy, że filtry przygotowujące do użycia filtru canny'ego potrafią zbyt rozmyć obraz przez co sam filtr niepoprawnie wykryje krawędzie co uniemożliwi rozpoznanie jego wartości gdyż nasza implementacja nie uzna tego obszaru jako poszukiwanego. Rozwiązaniem tego problemu jest lepsze dopasowanie filtrów, lecz wtedy inne przypadki naszej implementacji przestaną działać. Z tego powodu najlepszym rozwiązaniem byłoby przepuszczanie każdego nierozpoznanego obiektu przez drugi filtr, który rozwiązuje nasz problem np. korekcja gamma.

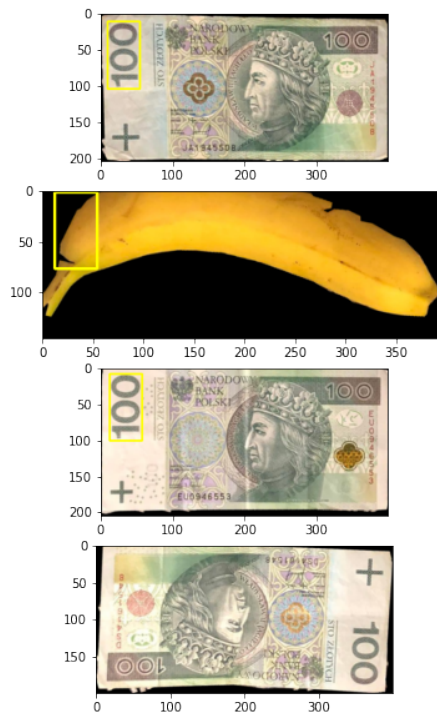


Rysunek 19: Niedziałający przypadek I.

5.1.2 Przypadek II

Nieprawidłowa klasyfikacja pola poszukiwań żądanych wartości. Jak przedstawiono na powyższym zdjęciu algorytm zaklasyfikował banana jako wartość poszukiwaną po czym przypisuje mu wartość banknotu o nominale 20 złotych. Nadanie wartości dla banana wynika z braku implementacji odpowiednich zabezpieczeń w metodzie process_AoIs, gdyż zakładaliśmy że tylko pożądane wartości będą się kwalifikować. Postanowiliśmy ten błąd zostawić gdyż jest to świetny przykład akademicki (i dlatego, że przy okazji nas bardzo rozbawił). Oczywiście błąd wynika gdyż program tworzy polygon który spełnia warunki stref naszego pożądanego (znajdują się w prawym górnym rogu oraz jest odpowiednich wymiarów [W poszukiwaniu obszarów naszego zainteresowania nie sprawdzamy kształtu obiektu, lecz wcześniej wspomniane wymiary]).

Banknoty z zaznaczonymi wartościami 1



Rysunek 20: Niedziałający przypadek II.

5.2 Wnioski:

Podsumowując powyższe przykłady wyniki otrzymane są zadowalające i stanowią podstawę dla dalszego rozwoju i optymalizacji naszego algorytmu. Powinniśmy zwrócić szczególną uwagę na ulepszenie metody `find_aoi`. Można to uczynić poprzez sprawdzania banknotów wielokrotnie różnymi metodami. Należy poprawić również przypadki w których obiekt został uznany za banknot mimo że nim nie jest. Rozwiązania również należy szukać w poprawieniu funkcji `find_aoi`, ale być może nałożenie ograniczeń w metodzie `find_value` okazało by się wystarczające. Wspomniane ograniczenia mogą korzystać z wartości tworzonych podczas działania metody. Zbyt niski wynik mógłby dyskwalifikować dany obiekt z znajdowania jego wartości.

5.3 Miara oceny jakości:

Podstawowe liczbowe wskaźniki jakości na podstawie wyników działania programu. Pod uwagę bierzemy sam fakt prawidłowego bądź nieprawidłowego odnalezienia oraz sklasyfikowania banknotu, a także innych obiektów.

TP – True Positive – liczba obserwacji poprawnie zaklasyfikowanych do klasy pozytywnej.

TN – True Negative – liczba obserwacji poprawnie zaklasyfikowanych do klasy negatywnej.

FP – False Positive – liczba obserwacji zaklasyfikowanych do klasy pozytywnej podczas, gdy w rzeczywistości pochodzą z klasy negatywnej.

FN – False Negative – liczba obserwacji zaklasyfikowanych do klasy negatywnej podczas, gdy w rzeczywistości pochodzą z klasy pozytywnej.

Prawdziwa /Predykcja	Klasa Pozytywna	Klasa Negatywna
Klasa Pozytywna	TP	FN
Klasa Negatywna	FP	TF

5.3.1 Przypadki łatwe:

Liczba banknotów: 8

Liczba innych obiektów: 37

Prawdziwa /Predykcja	Klasa Pozytywna	Klasa Negatywna
Klasa Pozytywna	8	0
Klasa Negatywna	0	37

5.3.2 Przypadki średnie:

Liczba banknotów: 37

Liczba innych obiektów: 0

Prawdziwa /Predykcja	Klasa Pozytywna	Klasa Negatywna
Klasa Pozytywna	33	4
Klasa Negatywna	0	0

5.3.3 Przypadki trudne:

Liczba banknotów: 28

Liczba innych obiektów: 14

Prawdziwa /Predykcja	Klasa Pozytywna	Klasa Negatywna
Klasa Pozytywna	27	1
Klasa Negatywna	2	12