Komunikacja człowiek-komputer	
Rozpoznawanie pieniędzy	
Dominik Doberski 132207	Artur Olejnik 132...

Sprawozdanie

OVERVIEW

Celem projektu było stworzenie programu wykrywającego pieniądze na zdjęciach.
Należało wykryć:

Monety o nominałach 5gr, 20gr, 2zł, 5zł.

Banknoty o nominałach 20zł, 50zł.

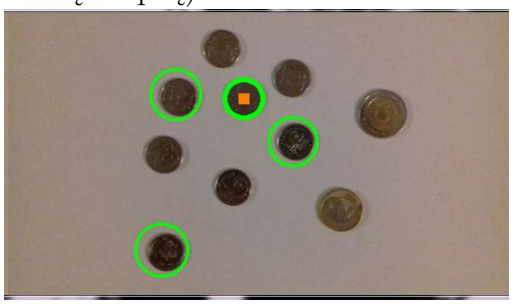
KROKI DZIAŁANIA

Pierwszym krokiem było wykrycie okręgów oraz prostokątów na zdjęciu.
Nasze podejście zakładało stworzenie kilku wyspecjalizowanych funkcji wykrywających monety i banknoty zależnie od rodzaju zdjęcia, a następnie zsumowanie ich wyników.
Odrzucaliśmy okręgi które są za duże lub większość ich obszaru zajmuje kolor tła.
Niektóre z funkcji wykrywały te same monety dlatego odrzucamy kontury które przynajmniej w połowie zawierają się w innych.

Hough

Funkcja wykorzystująca momenty Hougha okazała się nieprzydatna. W większości przypadków znajdowała okręgi w złych miejscach (najczęściej na banknotach).

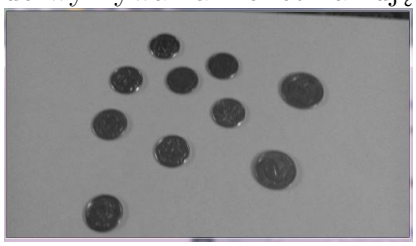
W nielicznych przypadkach algorytm był wstanie wykryć monetę poprawnie (moneta oznaczona pomarańczową kropką).



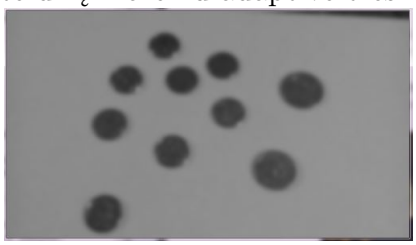
Algorytm okazał się tak nieskuteczny, że nie użyliśmy go w ostatecznym rozwiązaniu.

Adaptive thresholding

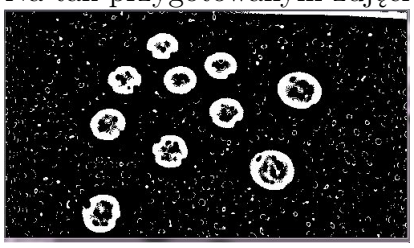
Algorytm adaptive thresholding okazał się przydatny w niektórych przypadkach, szczególnie dobrze nadając się do wykrywania monet na zdjęciach pod kątem. Najpierw konwertujemy zdjęcie na odcienie szarości.



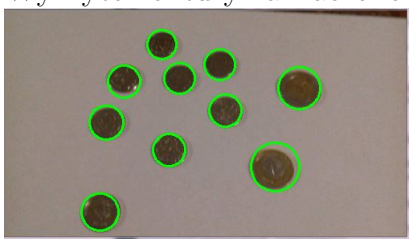
Następnie używamy funkcji GaussianBlur aby pozbyć się szczegółów na zdjęciu, monety stają się rozmyte dzięki czemu adaptive treshold nie wyrkywa wzorów na nich.



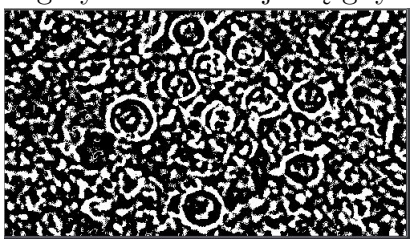
Na tak przygotowanym zdjęciu uruchamiamy algorytm adaptive Tresholding.



Wykryte kontury zaznaczone okręgami



Algorytm nie nadaje się gdy tło jest niejednolite (efekt działania przedstawiony na zdjęciu).



Bright coins

Aby znaleźć monety szczególnie mocno naświetlone w pierwszej kolejności zmniejszamy jasność i zwiększamy kontrast zdjęcia.

Zamieniamy model kolorów z RGB na HSV aby otrzymać dostęp do zmiennej value która pozwala łatwo odseparować jasne elementy od ciemnych.

Na tak przygotowanym zdjęciu uruchamiamy thresholding i rysujemy okręgi w miejscu wykrytych konturów.

Algorytm mimo, że przygotowany był z myślą o najjaśniejszych monetach skutecznie wykrywa większość monet jakie mamy na zdjęciach.

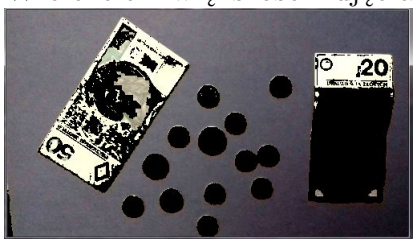
Dla zbyt ciemnych zdjęć zmniejszona jasność uniemożliwia znalezienie monet.

Silver coins

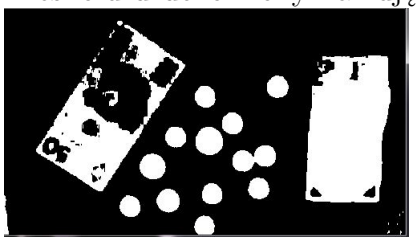
Kolejny algorytm filtruje srebrne monety na zdjęciu. W modelu RGB odrzucamy kolory które mają wartości RGB oddalone od siebie o więcej niż 40, zostawiając w ten sposób tylko odcienie szarości. Usuwamy także wartości poniżej 5 (czarny) oraz powyżej 250(biały).

Okazało się jednak, że srebrne monety często na zdjęciach mają kolor bardziej kremowy który nie spełnia naszych założeń, natomiast szaroniebieski kolor na banknocie 50zł nie był usuwany. Niektóre tła z których korzystaliśmy były ciemne przez co nasz algorytm często ich również nie usuwał.

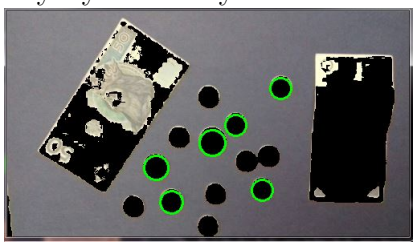
W efekcie z większości zdjęć algorytm usuwał jedynie monety co pozwoliło łatwo je wykrywać.



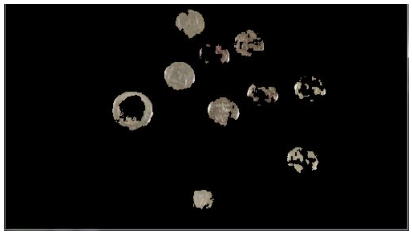
Treshold uruchomiony na zdjęciu z usuniętymi monetami.



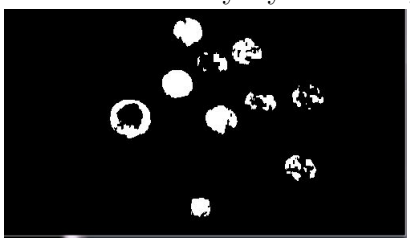
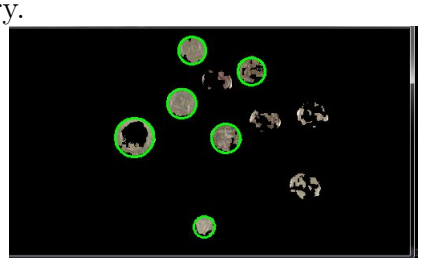
Wykryte monety.



Udało nam się znaleźć jednak zdjęcie z którego usuwane jest tło, a monety po części zostają.

Treshold oraz wykryte kontury.

Usuwanie dublikatów

Dla każdego konturu sprawdzamy czy w liście konturów które rysujemy istnieje taki którego przynajmniej połowa punktów jest wewnątrz danego konturu. Jeżeli tak, zakładamy, że jest to powtórzony kontur i odrzucamy go. W przeciwnym wypadku dodajemy kontur do listy konturów do narysowania. Do sprawdzenia, czy punkt jest wewnątrz konturu używamy wbudowanej w opencv funkcji pointPolygonTest.

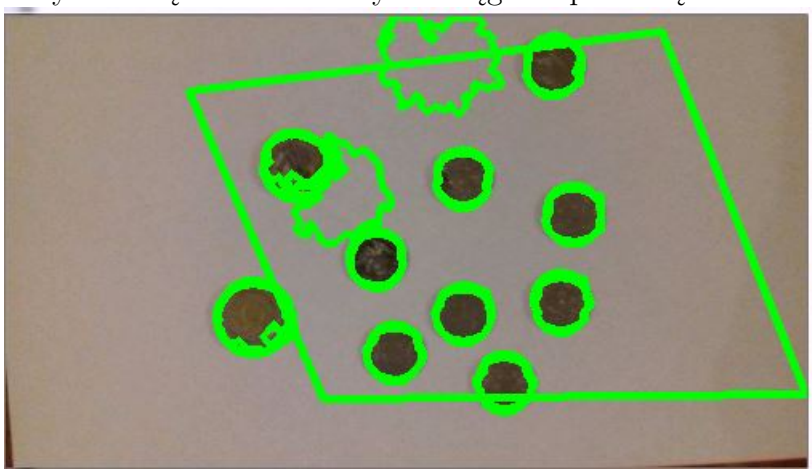
Zdjęcie z narysowanymi wszystkimi konturami wykrytymi przez nasze algorytmy.

Zdjęcie z odfiltrowanymi konturami (linie na monetach i banknotach są cieńsze - narysowane tylko raz)

WNIOSKI

Początkowo próbowaliśmy stworzyć jedną funkcję która wykrywa wszystkie monety na zdjęciu. Takie podejście okazało się błędne, odpowiednio dobrane parametry sprawiały, że funkcja działała tylko dla konkretnego zdjęcia, a dla innych zwraca błędny wynik.

Przykład błędnie znalezionych okręgów i prostokąta.



Dobrym rozwiązaniem okazało się stworzenie wielu funkcji i wybranie sumy znalezionych konturów co jest zaprezentowane w poprzednim punkcie sprawozdania.

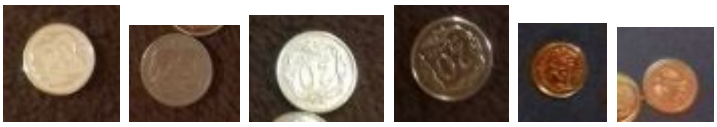
Aby odfiltrować błędnie znalezione kontury odrzucamy te, które są zbyt duże w stosunku do całego zdjęcia.

Odfiltrowane za duże okręgi.

Planujemy odfiltrowywać znalezione kontury, jeżeli większość ich pola ma kolor tła. Ta funkcjonalność nie jest jednak jeszcze zaimplementowana dlatego nie możemy poprzeć jej obrazem.

Wśród posiadanych zdjęć trafiliśmy też na takie, które nie są dobrze rozpoznawane przez żadną z naszych metod, przez co dużo monet pozostaje nierozpoznanych.

Zauważyliśmy, że na różnych zdjęciach takie same monety różnią się od siebie kolorem. Dodatkowo, niemal na każdym są one innego koloru niż w rzeczywistości co widać na przykładzie opisanej wcześniej funkcji, która miała filtrować kolory inne niż srebrny. Może spowodować to duże problemy z rozpoznaniem monet.

Monety 20 groszowe na różnych zdjęciach. 

Co gorsze monety mogą różnić się kolorem nawet w obrębie pojedynczego zdjęcia z powodu padającego na nie światła.

Próbowaliśmy edytować zdjęcie poprzez zmianę kanału "hue" oraz "saturation" w modelu HSV, jednak różnice nadal są wyraźne.