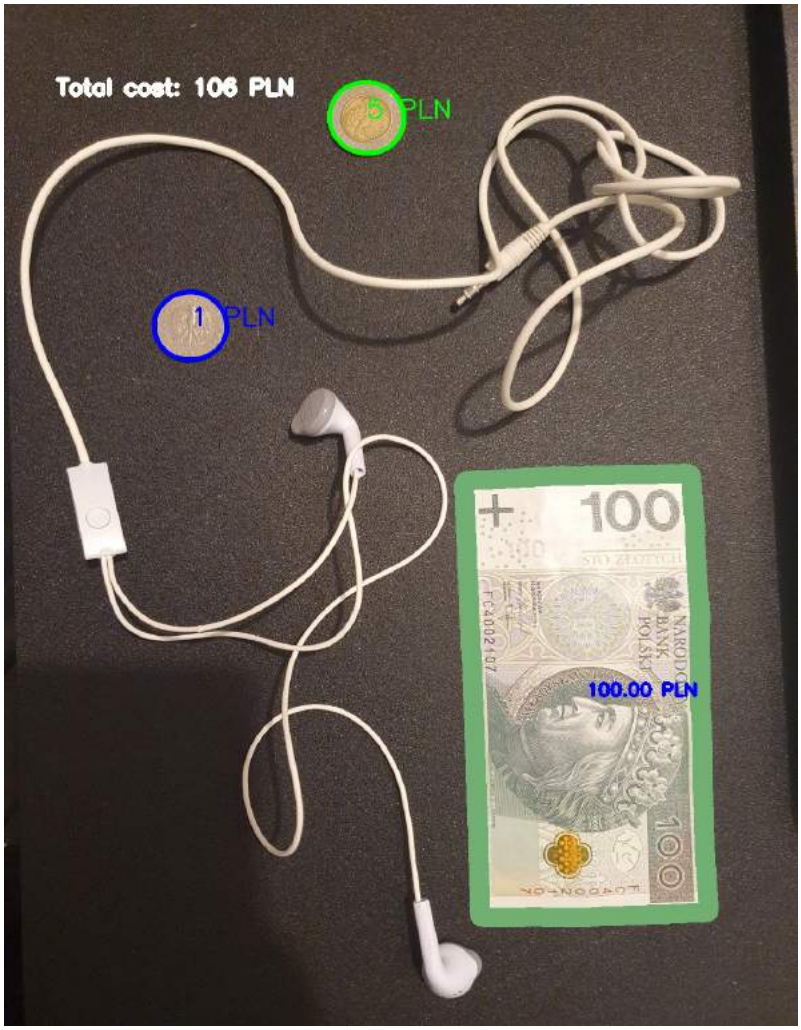
Sebastian Grabowski 145248
Tobiasz Gruszczyński 145333

Rozpoznawanie banknotów i monet na zdjęciach

Poznań 07.12.2021



Spis treści

1. Opis projektu

Głównym celem projektu było przygotowanie zarysu programu komputerowego, który będzie w stanie rozponać banknoty i monety stosowane w Polsce. Zmierzyliśmy się z problemem klasyfikacji obiektów, w którym danemu elementowi wykrytemu na obrazie, należy przypisać odpowiednią etykietę.

2. Ograniczenia

Przy przygotowaniu realizacji zabronione było wykorzystywanie w pełni gotowych rozwiązań ułatwiających wyszukiwanie monet i banknotów na obrazach. Jako ułatwienie przyjęliśmy ograniczenie analizowanych monet i banknotów do *1 PLN, 2 PLN, 5 PLN, 10 PLN, 20 PLN, 50 PLN, 100 PLN*. Dodatkowo banknoty są fotografowane stroną królem do góry.

3. Rozwiązanie problemu

Aby rozpoznać monety i banknoty na zdjęciu zaimplementowaliśmy odpowiednie metody. Zdjęcie po załadowaniu jest odszumiane i przetwarzane w celu odnalezienia konturów. Następnie wykryte obiekty są grupowane na monety i banknoty. Po przetworzeniu obrazu, wykryte i rozpoznane obiekty są zaznaczane na zdjęciu. Dodatkowo dodajemy napis, który pokazuje sumę pieniędzy.

3.1. Rozwiązanie monety

W przypadku rozpoznawania monet zdecydowaliśmy się na rozróżnianie ich po kolorze. Dla każdego wykrytego obiektu, który został zaklasyfikowany jako moneta, obliczany jest średni kolor. Dodatkowo wycinamy centrum, dla którego również obliczamy średni kolor.

3.2. Rozwiązanie banknoty

W przypadku rozpoznawania banknotów wycinany jest lewy górny róg banknotu w okolicy, w której spodziewamy się liczby. W przypadku wykrycia kontury porównujemy go z utworzonymi przez nas templatkami. Gdy kontur nie zostanie wykryty, banknot jest obracany o 180 stopni i proces się powtarza.

4. Etapy poszukiwania konturów na zdjęciu

Wczytane zdjęcie jest przetwarzane przez filtry redukujące szumy i usuwające zakłucenia. Kolejne kroki algorytmu dla przykładowego obrazu przedstawionego na rysunku nr 1.

Rysunek 1. Dane wejściowe - banknot *10 PLN* oraz moneta *5 PLN*

4.1. Skala szarości

Obraz jest wczytywany w postaci RGB oraz skali szarości. Dzięki temu mogliśmy zastosować niektóre filtry i poprawić działanie programu.

Rysunek 2. Zdjęcie w skali szarości

4.2. Rozmycie Gaussa

Rozmywamy obraz przy użyciu wskazanego kernela.

Rysunek 3. Zdjęcie po nałożeniu rozmycia

4.3. Progowanie obrazu

Uzyskujemy obraz binarny dzięki czemu możemy oddzielić obiekty pierwszoplanowe od tła.

Rysunek 4. Zdjęcie po wykonaniu tresholdingu

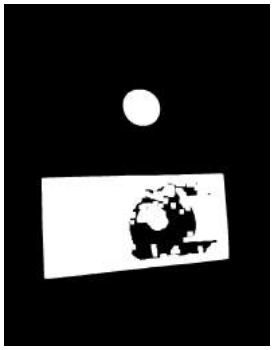
4.4. Erozja

Wykorzystanie erozji w celu pozbycia się obiektów będących niepotrzebnym szumem. Proces wykonany w kilku iteracjach.

Rysunek 5. Zdjęcie po wykonaniu erozji

4.5. Dylatacja

Wykorzystanie dylatacji, aby obiekty powróciły do poprzedniego stanu. Proces wykonany w kilku iteracjach.



Rysunek 6. Zdjęcie po wykonaniu dylatacji

4.6. Wycięte kontury

Odnalezione kontury są wycinane i następnie przetwarzane przez algorytmy rozpoznające znajdujące się wśród nich banknoty i monety.

Rysunek 7. Wycięte odnalezione kontury

5. Historia pracy nad algorytmem

Przygotowanie algorytmu do wykrywania obiektów na obrazie nie jest rzeczą trywialną. Wiele czynników może negatywnie oddziaływać na proces analizy. Są to między innymi różnice w naświetleniu obiektu, pora dnia, refleksy, jakość zdjęcia, urządznie robiące zdjęcie. W czasie implementacji algorytmu przekonaliśmy się o tym wiele razy.

Zarówno dla monet jak i banknotów początkowym pomysłem było wykorzystanie kolorów w celu rozróżniania obiektów. Pomysł ten w częsci 'monetowej' przetrwał do samego końca, jednakże w trakcie implementacji pojawiły sie problemy takie jak podobne układy kolorów monet *2 PLN* i *5 PLN*. Przy obliczaniu średniego koloru wycinka obie monety były praktycznie nierozróżnialne. W tym celu zdecydowaliśmy się na wycięcie dodatkowo centrum monety. Pozwoliło nam to na odróżnienie monety *2 PLN* od monety *5 PLN*

W przypadku Banknotów pomysł ten nie przetrwał. Początkowo chcieliśmy wykorzystać charakterystyczne kolory królów. Algorytm miał wycinać z banknotów centralne cześci zawierające wizerunek władców i za pomocą funkcji odnaleźć zależności miedzy kolorem, a wartością banknotu. Niestety nie udało się uzyskać żadnej funkcji definiujące te zależności. Próbowaliśmy zarówno funkcji wyliczających średnie wartości barw jak i średnie różnice miedzy nimi. Żadna z nich nie okazała się satysfakcjonująca, co poskutkowało całkowitym porzuceniem koncepcji. Przeszliśmy po tym do intuicyjnie trudniejszego pomysłu rozpoznawania wartości banknotów *10 PLN, 20 PLN, 50 PLN, 100 PLN* przez ich graficzną reprezentacje w lewym górnym rogu banknotu. Pomysł okazał się jednak łatwiejszy do implementacji i zdecydowanie skuteczniejszy. Algorytm wycina lewy górny róg banknotu z wartością liczbową, którą rozpoznaje. Robi to dla 'strony z królem'. Planowaliśmy robić to dla obu stron banknotu, jednka druga strona okazała się znacznie trudniejsza. Liczby na niej znajdują sie na niejednolitych tłach, które w procesie konturowania sprawiały naszemu algorytmowi trudności i okazały się barierą nie do przejścia.

6. Analiza algorytmu

6.1. Wyszukiwanie monet

Wyszukiwanie monet odbywa się poprzez wybranie odpowiednich konturów wyciętych we wcześniejszym etapie. W tym celu interujemy po wszystkich konturach.

Pierwszym krokiem jest wyznaczenie ilości wierzchołków konturu. Na tej podstawie odrzucamy kontury, które mają mniej niż 4 wierzchołki.

Drugi krok to obliczenie pola figury i sprawdzenie czy nie jest ono zbyt małe, albo zbyt duże. Dodatkowo sprawdzamy w jakim stopniu obliczone pole pokrywa się z polem koła. W tym celu wykorzystaliśmy funkcję *minEnclosingCircle*, dzięki której otrzymujemy promień koła. Po obliczeniu pola powierzchni koła $\Pi * r^2$ o otrzymanym promieniu (pomniejszonym o 5%) sprawdzamy stosunek pola powierzchni figury do obliczonego pola koła. Na tej podstawie decydujemy czy dany kontur jest okręgiem, czyli w naszym przypadku monetą. Przyjęliśmy margines błędu i akcpetujemy kontur, gdy jego stosunek do pola powierzchni koła jest większy niż 0,77. Pozwala nam to odróżnić od monet inne obiekty takie jak banknoty.

6.2. Wyszukiwanie banknotów

Poszukiwanie konturów, które mogą być potencjalnymi prostokątami zawierającymi banknoty, wymaga sprawdzenia kilku warunków.

Podobnie jak w przypadku monet bierzemy pod uwage liczbę wierzchołków konturów. Odrzucamy kontury nieposiadające 4 wierzchołków, dodatkowo algo-

rytm oblicza pole konturów i sprawdza aby miało ono odpowiednią wielkość, dodatkowym ograniczem jest tutaj aby kontur był wypukły.

Następnym krokiem jest znormalizowanie potencjalnego banknotu. Banknot w zależności od zdjęcia może być różnej wielkości, a dodatkowo znajdować się w różnych pozycjach. Algorytm wycina czworokąt z potencjalnym banknotem, obraca go do postaci poziomej, a następnie skaluje go do rozmiaru *(400x200)*.

Rysunek 8. Wykryte banknoty po wycięciu

Rysunek 9. Banknoty po obróceniu

6.3. Sposób rozpoznawania monet

W celu rozpoznania monet iterujemy po zbiorze konturów, które we wcześniejszym etapie zostały zklasifikowane jako monety. Wykorzystana została właściwość badanych obiektów, jakim u nas jest różnica w barwach monet. Dzięki temu możemy określić, z którą z trzech możliwych monet mamy do czynienia.

Na początku dla wyciętego kontury obliczamy sumę wartości bezwzględnych różnic pomiędzy poszczególnymi częściami koloru tj. *r - g, r - b, g - b*. Suma ta jest dodawana do tablicy wartości cząstkowych, która na samym końcu zostaje uśredniona dzięki zastosowaniu *numpy.average*. Następnie analogiczny proces przechodzi centrum wycięte z naszego konturu.

Po zebraniu i przeanalizowaniu danych określono następujące zależności:

$$moneta = \begin{cases} 1 \text{ PLN} & \text{gdy średnia} \leqslant 108 \text{ i średnia centrum} < 150 \\ 2 \text{ PLN} & \text{gdy średnia} > 108 \text{ i średnia centrum} \leqslant 150 \\ 5 \text{ PLN} & \text{gdy średnia} > 108 \text{ i średnia centrum} > 150 \end{cases}$$

Podejście rozpoznawania monet po kolorze ma pewne wady. Jest to wyjątkowo wrażliwa implementacja, która może mieć problem z prześwietlonymi lub zacienionymi obiektami, z którymi człowiek bez problemu sobie poradzi.

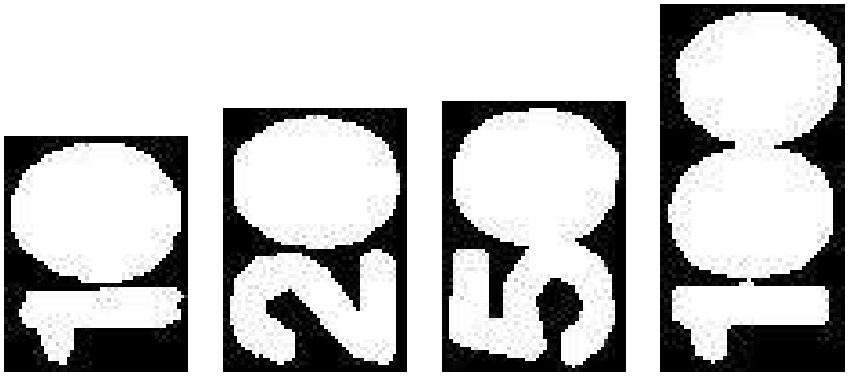
6.4. Sposób rozpoznawania banknotów

Mając juz znormalizowany obraz banknotu ponownie poszukujemy konturów, lecz tym razem konturów liczb znajdujących sie na banknocie w jego lewym

górnym rogu. W celu rozpoznania wartości banknotu korzystamy z wcześniej przygotowanych masek poszczególnych wartości.

Jeśli w wyznaczonej strefie lewego górnego rogu algorytm nie jest w stanie odnaleźć konturów, może to oznaczać, że potencjalny banknot jest źle obrócony, gdyż do tej pory doprowadziliśmy banknot jedynie do postaci poziomej. W takim przypadku obracamy badany obiekt o 180 stopni.

W przypadku kiedy w badanej strefie znajduję sie liczba, algorytm porównują ją z wcześniej wspomniannymi maskami. Dzieję się to przy pomocy funkcji z biblioteki skimage *mean_squared_error()* która liczy średni kwadrat odchylenia miedzy dwoma obrazami. Algorytm zapisuje wartości w licie, a następnie banknotowi przypisywana jest wartość o najmniejszym MSE.

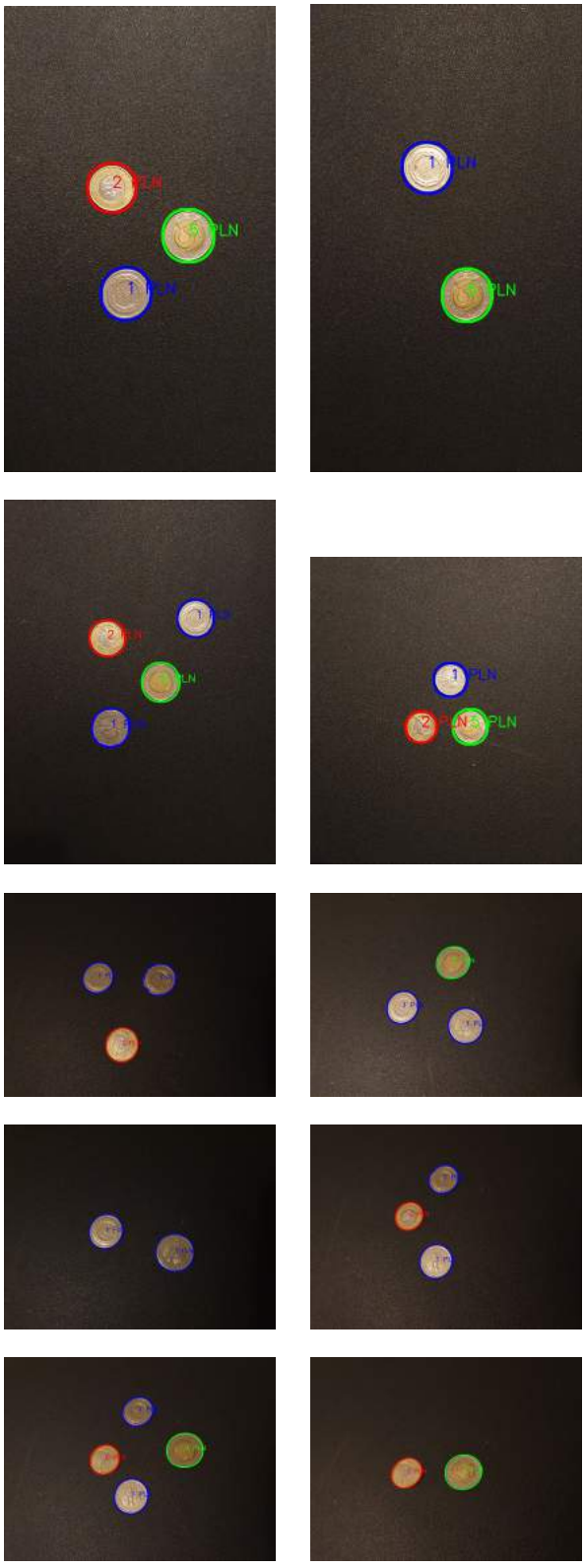


Rysunek 10. Wykorzystane zrobione przez nas templatki

7. Wyniki działania programu

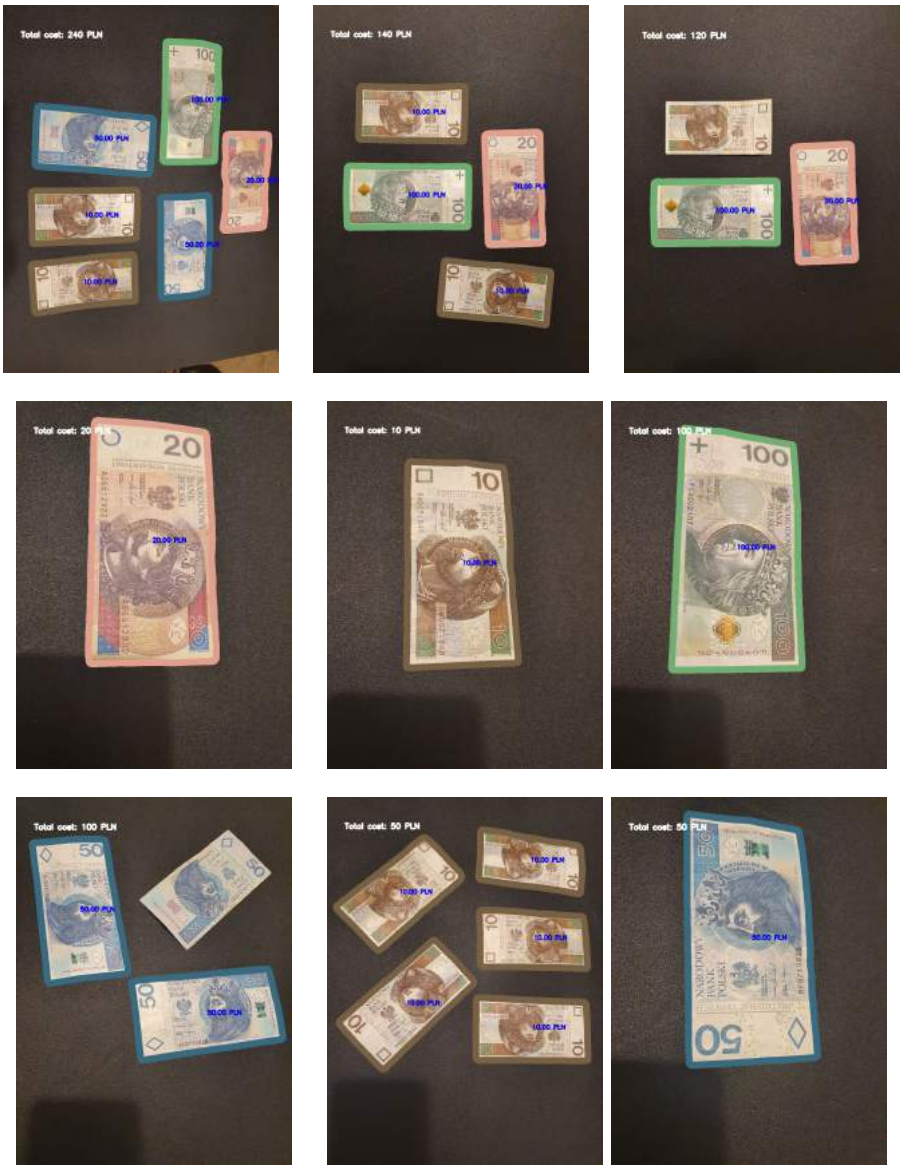
7.1. Przypadek I:

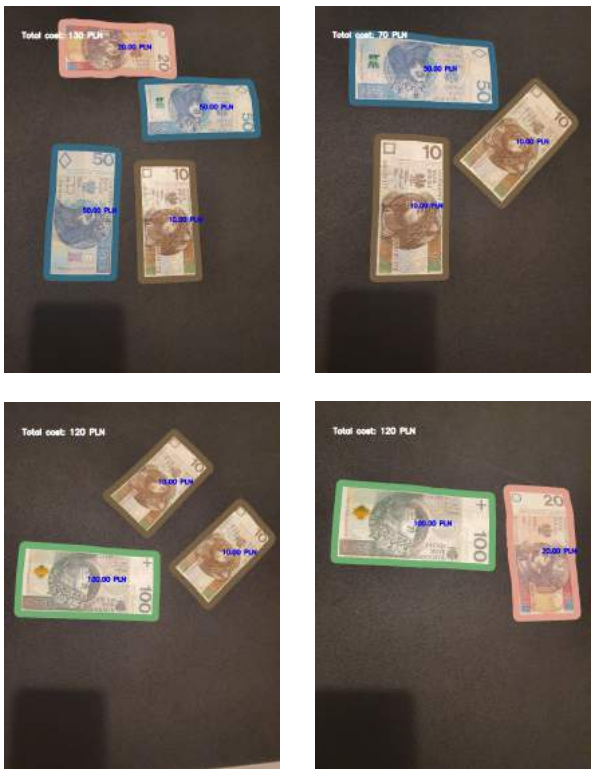
Na zdjęciach znajduje się od jednej do kilku monet.
Procent poprawnie rozpoznanych monet: 100%



7.2. Przypadek II:

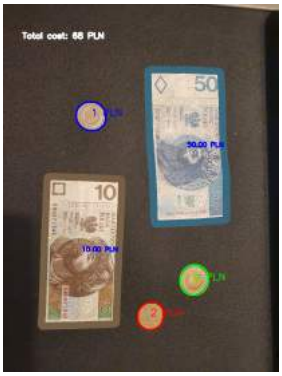
Na zdjęciach znajduje się od jednego do kilku banknotów.
Procent poprawnie rozpoznanych banknotów: 95%





7.3. Przypadek III:

Na zdjęciach znajdują się monety i banknoty.
Procent poprawnie rozpoznanych monet: 86%
Procent poprawnie rozpoznanych banknotów: 93%



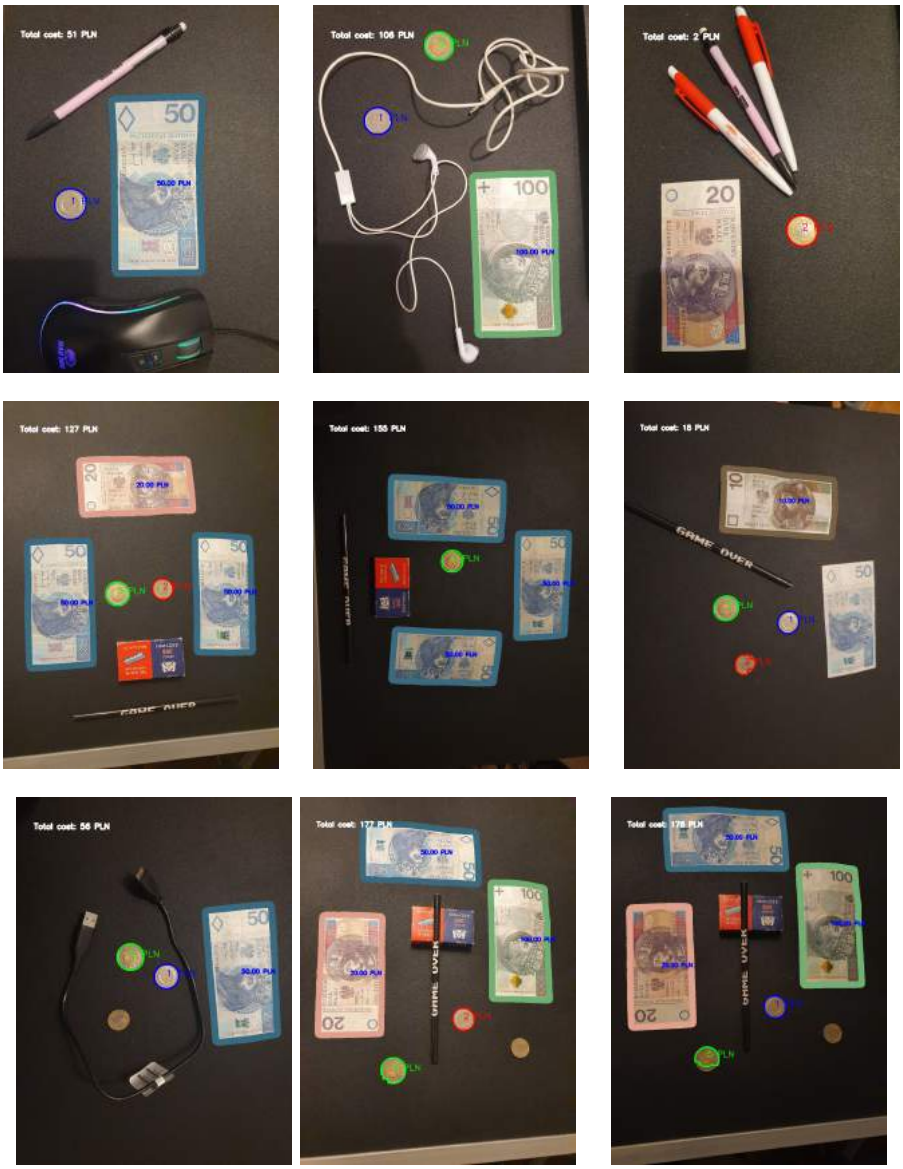


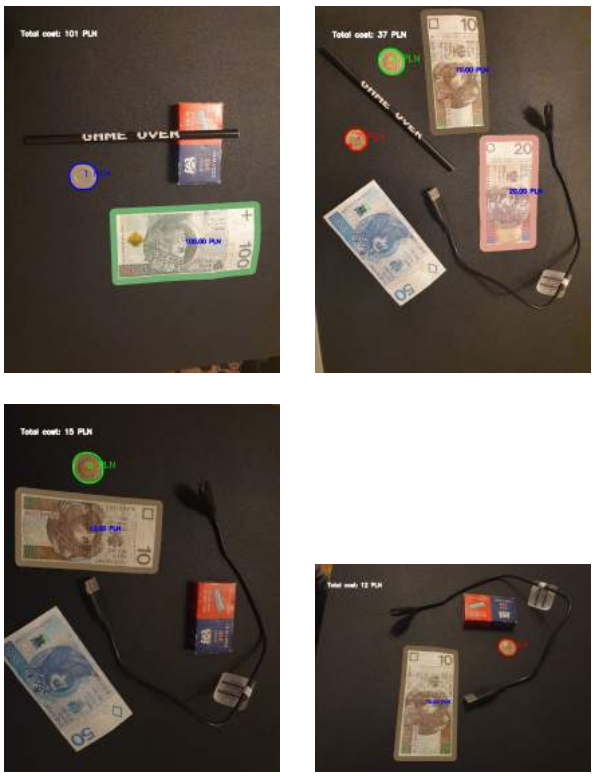





7.4. Przypadek IV:

Na zdjęciach znajdują się monety i baknoty wraz z innymi przedmiotami.
Procent poprawnie rozpoznanych monet: 83%
Procent poprawnie rozpoznanych banknotów: 87%





8. Podsumowanie wyników

8.1. Macierz pomyłek, czyli ocena jakości

Liczbowe wskaźniki jakości na podstawie otrzymanych przez nas wyników. Rozważamy fakt prawidłowego lub nieprawidłowego sklasyfikowania oraz wykrycia banknotów, monet oraz innych obiektów.
TP - True Positive (przewidywanie pozytywne i zaoberwowano klasę pozytywną)
TN - True Negative (przewidywanie negatywne i zaoberwowano klasę negatywną)
FP - False Positive (przewidywanie pozytywne i zaoberwowano klasę negatywną)
FN - False Negative (przewidywanie negatywne i zaoberwowano klasę pozytywną)

		Rzeczywistość	
		P	**N**
Przewidywanie	**P**	*TP*	*FP*
	N	*FN*	*TN*

8.1.1. Przypadek I:

Na zdjęciach znajduje się:
— 16 monet 1zł, 8 monet 2zł, 8 monet 5zł
— 0 innych obiektów (banknoty oraz inne przedmioty)

		Rzeczywistość			
		1 PLN	2 PLN	5 PLN	Inne obiekty
Przewidywanie	1 PLN	16	0	0	0
	2 PLN	0	8	0	0
	5 PLN	0	0	8	0
	Inne obiekty	0	0	0	0

Rysunek 11. Macierz pomyłek dla samych monet

8.1.2. Przypadek II:

— 16 banknotów 10zł, 6 banknotów 20zł, 9 banknotów 50zł, 6 banknotów 100zł
— 0 innych obiektów (monety oraz inne przedmioty)

		Rzeczywistość				
		10 PLN	20 PLN	50 PLN	100 PLN	Inne obiekty
Przewidywanie	10 PLN	15	0	0	0	0
	20 PLN	0	6	0	0	0
	50 PLN	0	0	8	0	0
	100 PLN	0	0	0	6	0
	Inne obiekty	1	0	1	0	0

Rysunek 12. Macierz pomyłek dla samych banknotów

8.1.3. Przypadek III:

— 10 monet 1zł, 8 monet 2zł, 11 monet 5zł
— 8 banknotów 10zł, 9 banknotów 20zł, 6 banknotów 50zł, 6 banknotów 100zł
— 0 innych obiektów (dla monet wliczamy banknoty i odwrotnie)

		Rzeczywistość			
		1 PLN	2 PLN	5 PLN	Inne obiekty
Przewidywanie	1 PLN	8	0	1	0
	2 PLN	1	7	0	0
	5 PLN	1	1	10	0
	Inne obiekty	0	0	0	29

		Rzeczywistość				
		10 PLN	20 PLN	50 PLN	100 PLN	Inne obiekty
Przewidywanie	10 PLN	7	0	0	0	0
	20 PLN	0	9	0	0	0
	50 PLN	0	0	5	0	0
	100 PLN	0	0	0	6	0
	Inne obiekty	1	0	1	0	29

Rysunek 13. Macierz pomyłek dla monet i banknotów

8.1.4. Przypadek IV:

— 7 monet 1zł, 8 monet 2zł, 9 monet 5zł
— 4 banknotów 10zł, 5 banknotów 20zł, 12 banknotów 50zł, 4 banknotów 100zł
— 28 innych obiektów (dla monet wliczamy banknoty i odwrotnie)

		Rzeczywistość			
		1 PLN	**2 PLN**	**5 PLN**	**Inne obiekty**
Przewidywanie	**1 PLN**	*6*	*0*	*1*	*0*
	2 PLN	*1*	*5*	*0*	*0*
	5 PLN	*0*	*0*	*9*	*0*
	Inne obiekty	*0*	*3*	*0*	*52*

		Rzeczywistość				
		10 PLN	**20 PLN**	**50 PLN**	**100 PLN**	**Inne obiekty**
Przewidywanie	**10 PLN**	*4*	*0*	*0*	*0*	*0*
	20 PLN	*0*	*4*	*0*	*0*	*0*
	50 PLN	*0*	*0*	*9*	*0*	*0*
	100 PLN	*0*	*0*	*0*	*4*	*0*
	Inne obiekty	*0*	*1*	*3*	*0*	*53*

Rysunek 14. Macierz pomyłek dla monet i banknotów wraz z innymi przedmiotami

8.2. Niedziałające przypadki

8.2.1. Zbyt ciemne zdjęcie

W przypadku zbyt ciemnego zdjęcia algorytm potrafi zawieść. Pojawiają sie problemy ze złym rozpoznaniem konturów monet jak i banknotów. Przez co banknot w ogóle potrafi nie być rozpoznany i pominięty w dalszej części algorytmu, a moneta bywa pominięta, lub tak jak na poniższym zdjęciu wykryta częściowo. Dodatkowo nieodpowiednie oświetlenie potrafi negatywnie wpłynąć na przypisanie wartości do poszczególnych monet.

Rysunek 15. Błędy wynikające z przyciemnionego zdjęcia

8.2.2. Fragment obiektu nie znajduje się na zdjęciu

W przypadku kiedy obiekt wychodzi poza obszar zdjęcia nawet niewielką częścią nie jest on brany pod dalszą uwagę przez algorytm. W takim przypadku obszar jego kontur łączony jest z obszarem całego zdjęcia, przez co jest zbyt duży. Dodatkowo sprawia, to że liczba jego wierzchołków jest inna niż 4, a tyle wymaga nasz algorytm do dalszej pracy nad banknotem.

Rysunek 16. Błąd spowodowany wyjściem obiektu poza obszar zdjęcia

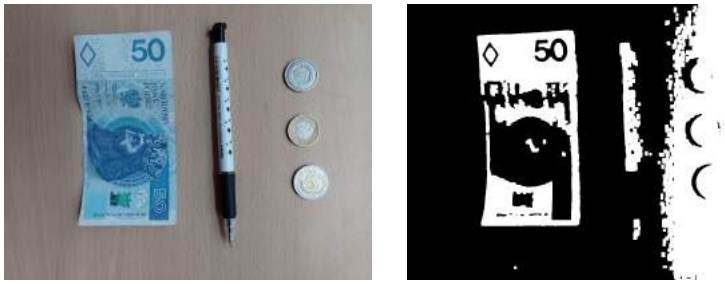
8.2.3. Nierozpoznanie liczby w lewym górnym rogu

Zdarzają się przypadki w których pomimo prawidłowo zaznaczonego banknotu algorytm zowodzi. Nie przyznaje mu żadnej wartości. Jest to spowodowane tym, że kontur liczby nie znajduje sie w rozpatrywanym obszarze lewego górnego rogu.

Rysunek 17. Błąd przy rozpoznaniu liczby po obróceniu banknotu

8.2.4. Odnajdywanie konturów na niejednolitym tle

W sytuacji gdy mamy do czynienia z niejednolitym tłem, nasz algorytm ma problemy z wykryciem konturów obiektów.



Rysunek 18. Efekt po obróbce obrazu i błędny efekt końcowy

8.3. Wnioski

Podsumowując otrzymane przez nas rezulataty możemy stwierdzić, że stanowią one podstawe do dalszej pracy i optymalizacji napisanego przez nas programu. Wyniki algorytmu są satysfakcjonujące, można jednka zauważyć jego niedoskonałości wymagające poprawy. Najważniejszą rzeczą jest sposób wykrywania potencjalnych monet oraz banknotów. Algorytm jest wrażliwy na światło i cienie. Niejednolite tło również znacząco utrudnia rozpoznawanie konturów.

Algorytm nie wyszukuje kół i elips, wprowadzenie tej zmiany mogło by usunąć przypadki w których algorytm znalazł jedynie fragment monety. Samo rozpoznawanie monet również ma miejsce do optymalizacji, patrzac na to jak bardzo oświetlenie i rodzaj światła ma na nie wpływ.

W przypadku banknotów na pewno warto zająć się drugą stroną, stroną 'bez królów'. Samo rozpoznawanie wartości banknotu za pomocą mask powinno być zoptymalizowane, a szczególnie przypadki w których banknot jest źle obrócony. W wielu przypadkach algorytm używa współczynników wybranych za pomocą prób i błedów. Z pewnościa użycie sztucznej inteligencji mogłoby zooptymalizować tę część algorytmu.

9. Bibliografia

Przygotując projekt, bazowaliśmy na wiedzy dostępnej w Internecie. Szczególnie przydatne okazały się:

— OpenCV documentation
— Skimage documentation
— Stack Overflow
— YouTube
— TutorialKart
— LearnOpenCV