

Politechnika Warszawska

Cyfrowe metody przetwarzania obrazu

Projekt II

Rozpoznawanie kart UNO

Wykonał:

Franciszek Żuchowski
305248

Prowadzący:

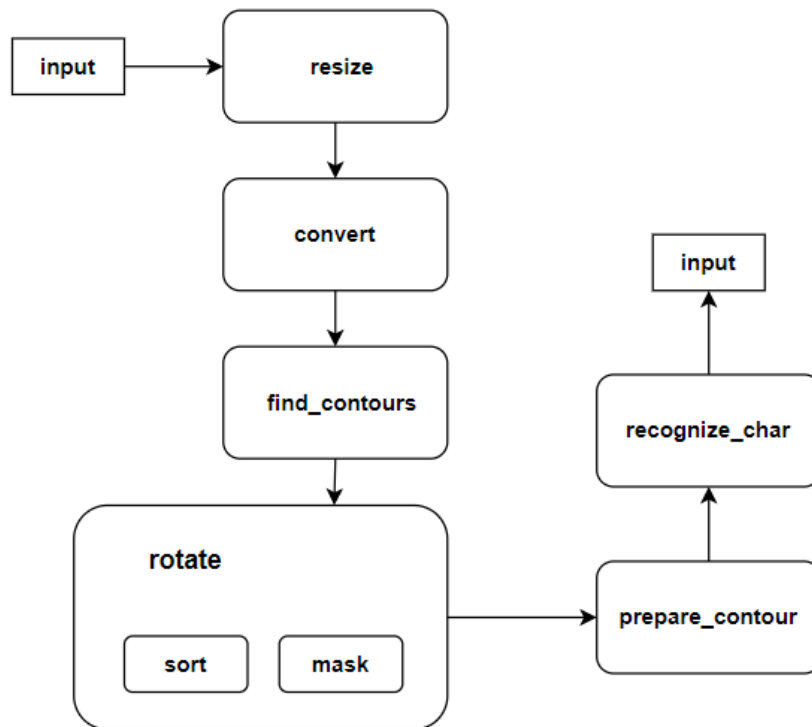
mgr inż. Filip Brzeski

2023

1. Opis projektu

Projekt polegał na rozpoznaniu rodzaju i koloru kart do gry UNO na zdjęciach. Kart było 16, po 4 na każde zdjęcie. Dodatkowo, algorytm przetwarzania obrazu musiał poradzić sobie z dodatkowymi filtrami nałożonymi na bazowe 4 zdjęcia, były to gradient, blur oraz pieprz i sól. Projekt został wykonany za pomocą języka python, bazując na bibliotece OpenCV. W tym przypadku do zidentyfikowania są karty o symbolach: 3, 5, 7, pauza oraz zmiana kierunku.

2. Implementacja kodu i zawarte w nim rozwiązania



Rysunek 1 Schemat funkcji wykorzystanych w programie

Zaimplementowany kod opierał się na 7 funkcjach gdzie kolejno:

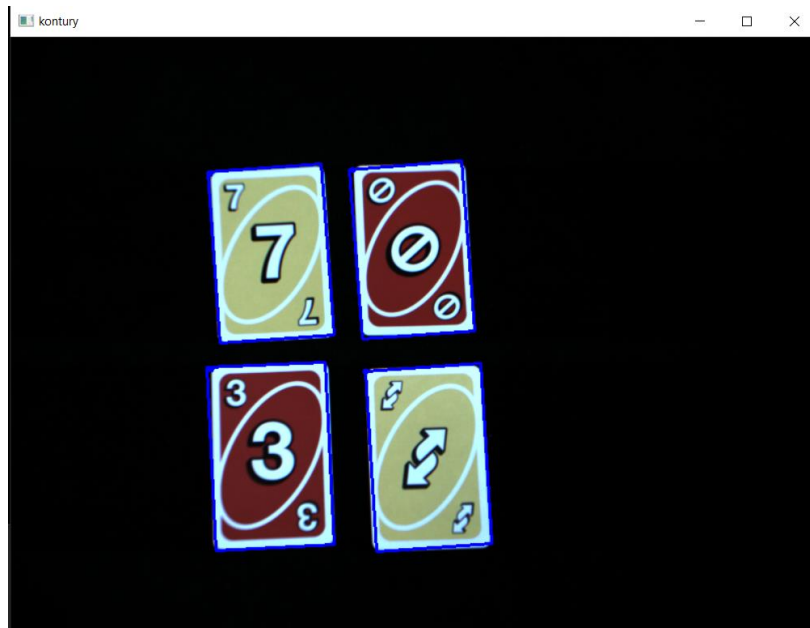
resize – obraz wejściowy jest przeskalowywany do rozmiarów pozwalających na wygodne wyświetlanie na ekranie,

convert – obróbka obrazu dla uniwersalizacji kodu dla każdego z filtrów jakie zostały nałożone na zdjęcia bazowe,

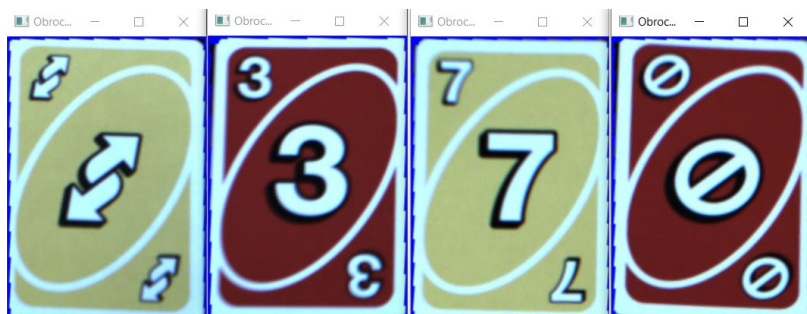


Rysunek 2 Karty po przekształceniu, pozwalające znaleźć główny kontur karty

find_contours – znalezienie konturów nadrzędnych tj. tych, które nie zawierają się w żadnych innych
rotate – funkcja obraca karty do pionu (z racji że na zdjęciu nie są ułożone prostopadle),



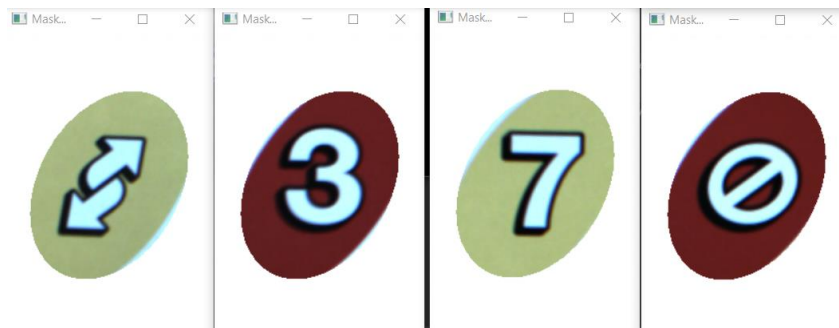
Rysunek 3 Znalezione kontury, dla których zostaną posortowane współrzędne krańcowe



Rysunek 4 Zdjęcie obróconych kart

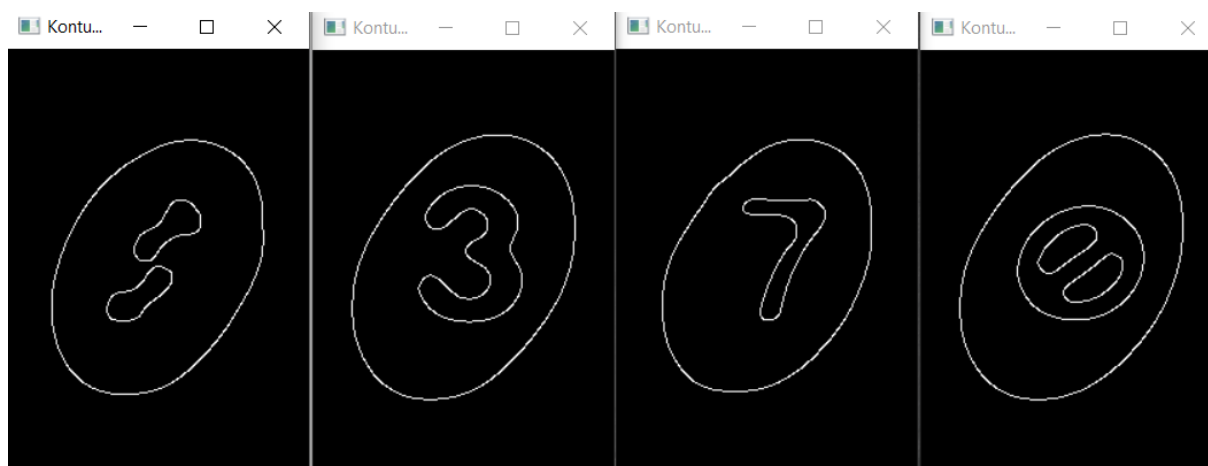
sort – funkcja sortująca współrzędne rogów kart do obrócenia, pozwala na jednakowy obrót niezależnie od tego jak leży karta,

mask – wycina ROI już z wyznaczonego obrazu karty, z którego będzie pobierany kontur do określenia jaka to karta



Rysunek 5 Nałożenie maski

prepare_contour – filtracja w celu uzyskania konturu do identyfikacji tak, aby cechy identyfikacji były uniwersalne dla każdej karty na każdym zdjęciu



Rysunek 6 Obrazy kart, przygotowanych do identyfikacji cech

recognize_contour – identyfikacja karty wraz z jej kolorem, poprzez określenie konturu do identyfikacji i określenia jego cech



Rysunek 7 Wyznaczone kontury do identyfikacji

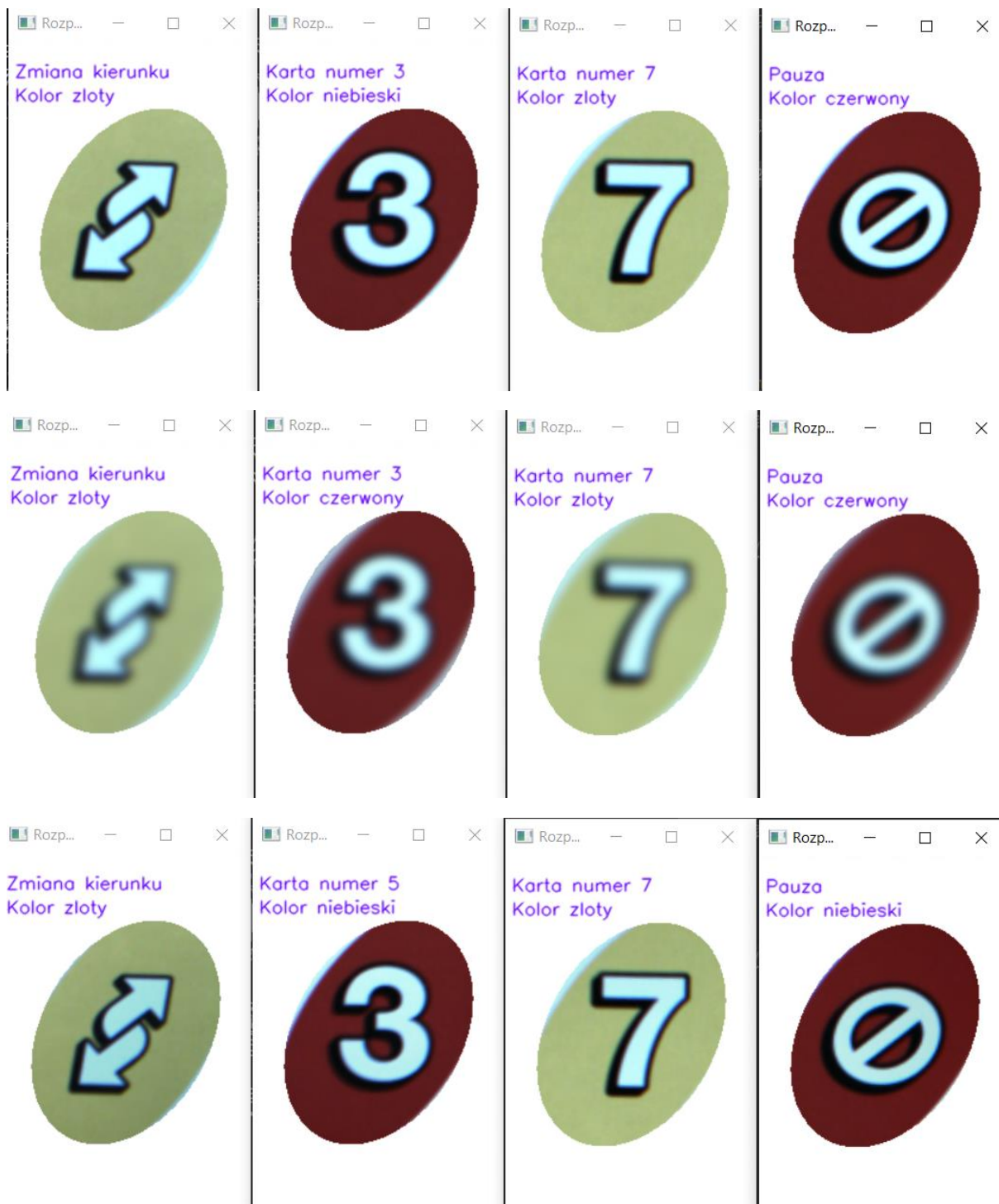
3. Otrzymane wyniki

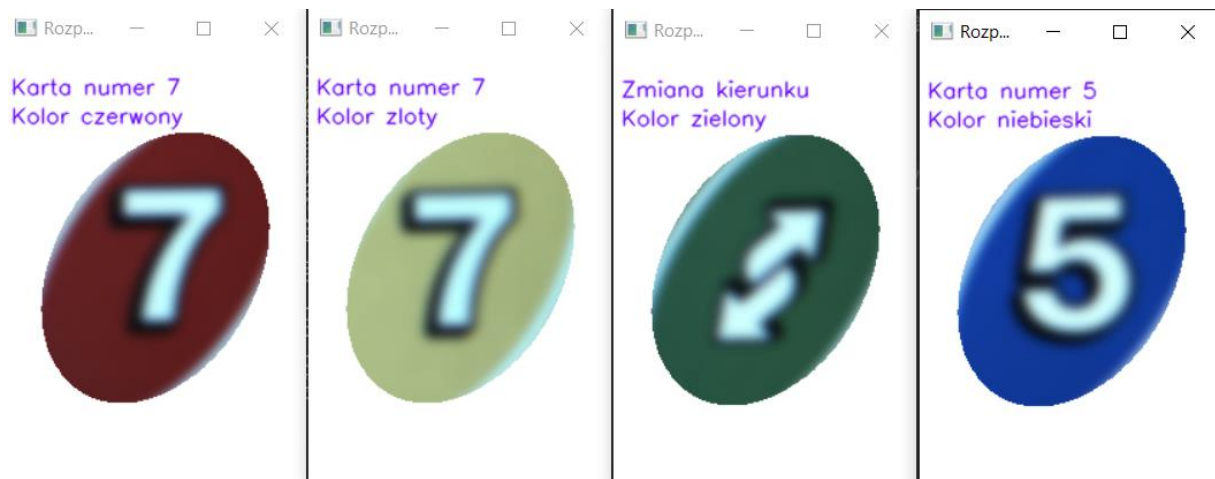
Poniżej prezentują się wyniki otrzymane dzięki algorytmowi. Dla dobranych cech oraz zakresów kolorów udało się rozpoznać wszystkie karty co do znaku, niestety co do koloru 2 nie zostały poprawnie zdefiniowane.













4. Wnioski

Wyzwania jakim należało zaradzić podczas pisania kodu były różnego rodzaju. Pierwszym i dość znaczącym była wstępna filtracja kart w taki sposób, aby identyfikacja konturu karty poprawnie je identyfikowała. Tutaj pomógł medianBlur w celu usunięcia pieprzu i soli z jednej serii zdjęć jednocześnie nie ingerując mocno w inne zdjęcia wejściowe. Problemem było również to, że karty na niektórych zdjęciach były bardzo blisko siebie, co groziło zlaniami się konturów. Przez co parametry thresholdingu musiały być odpowiednio dobrane oraz użyto erozji aby temu zapobiec.

Kolejnym większym problemem był poprawny obrót karty, nie działało wyznaczanie maksymalnych współrzędnych w każdym kierunku w zakresie pewnego boundingRect ponieważ tak jak wcześniej, karty były zbyt blisko siebie. Z pomocą minAreaRect wyznaczono współrzędne konturu karty. Należało je jeszcze odpowiednio posortować i tutaj przyjęto zasadę że prawy dolny róg jest najdalej od początku układu współrzędnych a lewy górny róg karty najbliżej. Problemy z funkcją warpAffine rozwiązano poprzez jej nieużywanie 😊

Na tym poziomie czasu spędzonego z programem można powiedzieć że to już prawie finisz. Kolejny problem pojawił się przy doborze zakresów dla poszczególnych kolorów i w przestrzeni RGB i w przestrzeni HSV, zdecydowano się na HSV, wyniki były lepsze. Tutaj można stwierdzić, że należałoby zastosować oddzielne zakresy dla różnych typów obrazu wejściowego (w zależności od filtru) lub uśrednić wartość z większego obszaru. Rolę grało to jak się robiło zdjęcia. Jeżeli na jedną kartę pada światło z otoczenia a na drugą nie ma to wpływ na późniejszą identyfikację koloru.

W przypadku symboli nie było problemu, oparto się na długości konturu, jego powierzchni oraz stosunkowi wysokość/szerokość. Możliwe że pojawiłby się problem przy większej różnorodności kart, jednak wtedy rozwiązanie mogłoby opierać się na większej ilości cech jak np. momenty.