Komunikacja Człowiek Komputer			
Kierunek *Informatyka*	**Specjalność** –	**Rok studiów** *III*	**Symbol grupy lab.** *I5.1*
Temat Laboratorium *Rozpoznawanie klocków domina*			**Numer lab.** *5*
Skład grupy ćwiczeniowej oraz numery indeksów *Radosław Dudek(144653)*			
Uwagi			**Ocena**

Cel

Zadanie polega na stworzeniu oprogramowania zdolnego rozpoznawać całkowitą liczbę oczek na klockach domina.

Blob Detection

Wprowadzenie

Blob Detection to najprościej mówiąc algorytm wyszukujący określone elementy na zdjęciu wyróżniające się wspólnymi cechami (np. kolor/kształt).

Ogólne działanie algorytmu [1]

1. Przekonwertuj obraz na binarny (biało-czarny)
2. Wyodrębnij połączone komponenty z każdego obrazu binarnego przez findContours i oblicz ich środki.
3. Grupuj centra z kilku obrazów binarnych według ich współrzędnych.Bliskie centra tworzą jedną grupę, która odpowiada jednemu obiektowi BLOB, który jest kontrolowany przez parametr minDistBetweenBlobs.
4. Z grup oszacuj końcowe centra bąbelków i ich promienie oraz zwracaj jako lokalizacje i rozmiary punktów kluczowych.

Zbiór domina

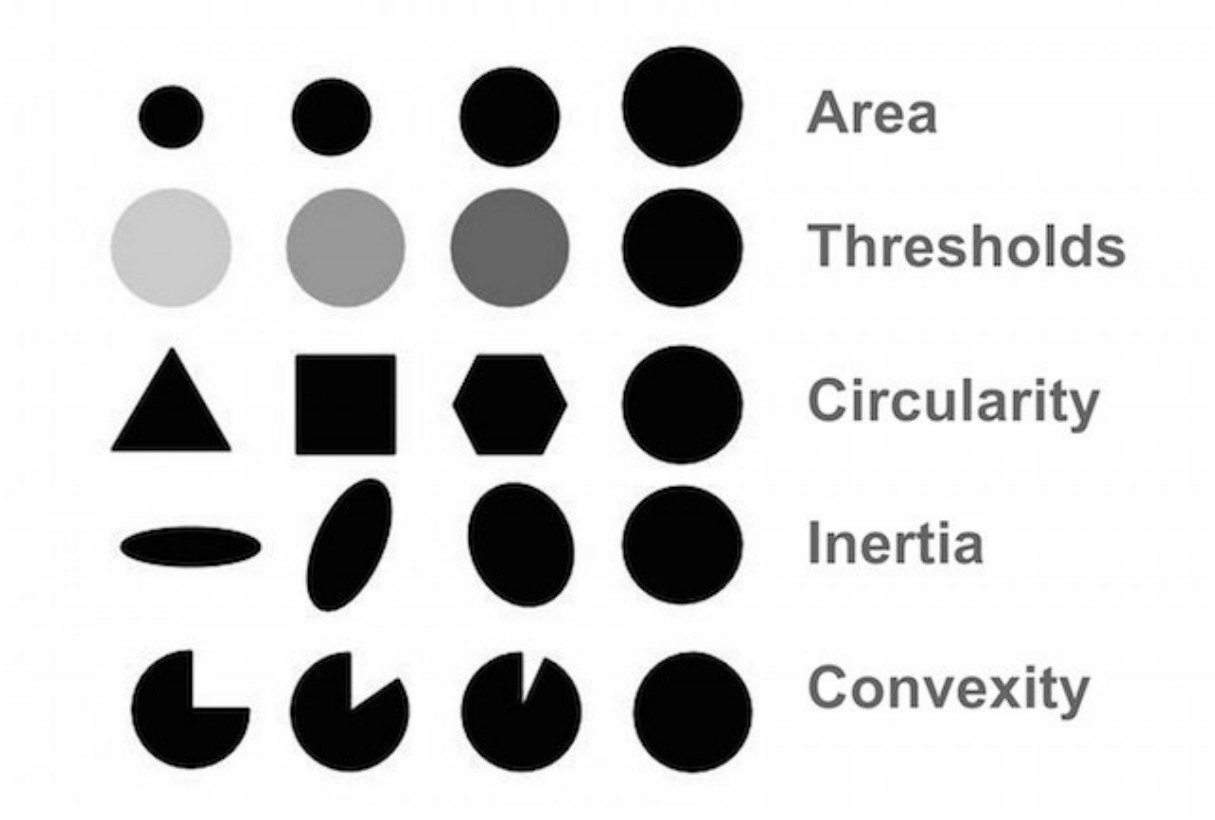
Z niewiadomych względów jedyne klocki domina były dostępne tylko w różowo-żółtym kolorze, co okazało się znacznym utrudnieniem projektu.

Rysunek 1: Kostki domino

Opis problemu tła

Specyfika klocków wymusza wyodrębnianie kolorów z zakresu żółty-pomarańczowy, które w sztucznych warunkach oświetlenia wystepują najczęściej, co może znacznie zakłócić wynik działania programu. Powyższy problem można bardzo wyraźnie zaobserwować na poniższym zdjęciu - wszystkie elementy zaznaczone na biało zostały wykryte jako pewna pochodna koloru żółtego. W celu rozwiązania tego problemu wykorzystanie zostane specjalne tło oraz zostaną odpowiednio dobrane parametry detektora.

Rysunek 2: Biały - wykryty kolor żółty



Blob - wykorzystane parametry

Blob to w dosłownym tłumaczeniu *kleks*, bliżej nie określony kształt który w idealnych warunkach dla mojego projektu przyjmie kształt koła. W celu wykrywania kształtów przypominających koła zastosujemy próg dolny *Inertia = 0.4*. Jednak ze względu na zmienne warunki oświetlenia, zmienne położenia aparatu, czy też po prostu nieregularną powierzchnię może dojść do sytuacji w której część koła zostanie wycięta. Zostało to rozwiązane parametrem *Convexity = 0.5*. Ostatni parametr który warunkuje rozpoznawanie oczek domina jest *Area = 800*. Jest to minimalny rozmiar Blob'a, dzięki czemu jestem w stanie w znaczny sposób odfiltrować błędne wyniki z tła.

Eksperyment


Rysunek 3: Białe i czarne tło


Rysunek 4: Brudne i żółte/brązowe tło

W ramach eksperymentu przetestuję swój program na 3 różnych tłach: białym, czarnym oraz w filtrowanym kolorze (żółto-pomarańczowym). Dodatkowo zotanie przeprowadzony test 10 zdjęć na całkowicie losowych tłach wraz z innymi przedmiotami (*seria brudna*). Testowane były zdjęcia zawierające od jednej do maksymalnie czterech kostek. Kostki domina były całkowicie losowo ustawione i zawierały kombinacje kostek zawierających od 0 do 7 oczek. Warto zauważyć, że program skupia się na detekcji i opisywaniu samych oczek domina, przez co domino bez żadnych oczek nie powinno zostać rozpoznane. Łącznie zotały przeprowadzone 4 testy po 10 zdjęć na każdym kolorze tła. Każde zdjęcie było oceniane według następującego systemu:

- Jeżeli wszystkie kostki zostały prawidłowo rozpoznane zdjęcie zdobywało 1 punkt.
- W przypadku określenia części kostek, wynik wynosił *prawidłowe/wszystkie*
- W przypadku nie podania prawidłowego wyniku dla każdej kostki, wynik wynosił 0.
- Jeśli program znajdzie wynik na tle, wynik wynosi 0.

Rodzaje pomiarów			
Białe Tło	Czarne Tło	Żółte Tło	Brudne Tło
80.00 %	100.00%	86.66 %	73.33 %

Tablica 1: Wyniki pomiarów

Przykładowe wynik zdjęć

Wnioski

Całkowita skuteczność działania programu niezależnie od tła wyniosła: **84.99 %**. Dodatkowo zaskakującym wynik jest większa skuteczności działania programu na żółtym tle niż na tle białym. Może to być spowodowane robieniem zdjęć w sztucznym tle, przez co kolor tła mógł odbiegać w stronę brązowego. Wysoka skuteczność programu pokazuje, że odpowiednio dobrane parametry algorytmu Blob Detection wystarczają do skutecznego rozpoznawania *kleksów*.

Uwagi

- Wszystkie zdjęcia były robione w odległości około 15-20cm.
- Aparat: Iphone 8
- Godzina robienia zdjęc: 17:30 grudzień (po zmierzchu)
- Zdjęcia były robione całkowicie z góry (telefon równoległy do podłoża)

Literatura

[1] OpenCV - Blob Detection