Wykrywanie kości do gry na zdjęciach

Witold Kupś

Politechnika Poznańska, Wydział Informatyki, 2017

# Wprowadzenie

Celem projektu jest wytworzenie oprogramowania, którego jedynym zadaniem jest wykrycie kości (dowolny kolor, dowolne tło) na zdjęciu (w dowolnej ilości) wykorzystując jedynie manipulacje na dostarczonych obrazach dowolnie je przetwarzając – krótko, bez zastosowań sztucznej inteligencji lub ingerencji zewnętrznej, jak `template matching`.

Strategia wykrywania jest prosta – początkowo poszukujemy samych kostek, następnie wyniki filtrujemy przez szereg porównań, po czym na odfiltrowanych kandydatach poszukujemy kropek. Warto jednak nadmienić, że w programie nie jest nigdzie zdefiniowane jak wygląda kostka – jest to po prostu obiekt wyróżniający się na tle otoczenia. Podobnie definiujemy kropkę, z tą różnicą, że tutaj przyjmujemy ogólne założenia co do jej rozmiarów w stosunku do kostki.

W programie posługujemy się abstrakcją obiektu, a konkretnie jego bounding boxem. Jedyne założenie co do niego jest takie, że ma on kształt prostokąta mniej lub bardziej przypominającego kwadrat (w przypadku poszukiwania kostek przyjmujemy, ze owy bounding box ma stosunek boków maksymalnie 2:1 niezależnie od krawędzi). W przypadku kropek na kostce dochodzi ograniczenie, że może być ich maksymalnie 6, jednak gdy żadna nie zostanie wykryta, takowa kostka nie jest w ogóle brana pod uwagę w wyniku.

Tworząc program próbowano utrzymać tendencje do ignorowania szumów występujących również na obrazie, jak tekst (pospolite w obrazach pobieranych z Internetu) czy odblaski światła. Zabiegi te często jednak mogą wpływać na poprawność programu (częściej na ignorowanie poprawnego wyniku, aniżeli jego zastosowanie).

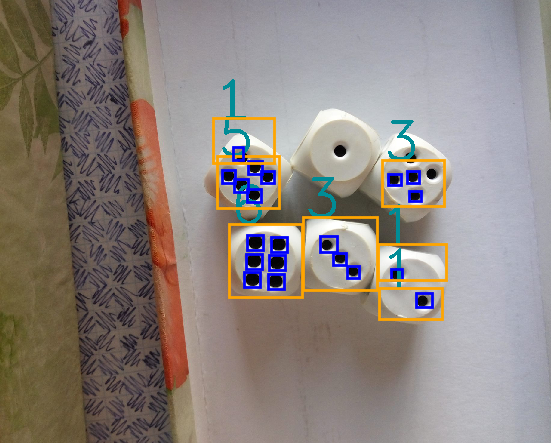
Niestety, powyższe funkcjonalności częściowo usztywniają możliwości aplikacji ze względu na:

- bliskość kości: w przypadku zbyt dużej bliskości kości mogą zostać zidentyfikowane jako jedna; gdy suma ich oczek jest większa niż 6, są one zawsze błędnie identyfikowane.

- gładkość krawędzi: gdy kości mają obłe krawędzi, może to powodować załamanie algorytmu wykrywającego kontury kości. Istnieje duże prawdopodobieństwo, że początkowo wykryje on ściany [ostre krawędzi], a następnie górną ścianę, która jednak ma zaokrąglone rogi. Kontur otaczający będzie jednak słabo wyraźny ze względu na mały kontrast pomiędzy górną a bocznymi ścianami. Wewnętrzy zaś z tego samego powodu (mały kontrast) zostanie wykryty, jednak możliwe jest, że jego kontury będą na tyle niewyraźne (lub częściowe), że zostaną przysłonięte przez kropki wewnątrz (lub w najlepszym razie zostanie wzięty pod uwagę mniejszy obszar poszukiwania).

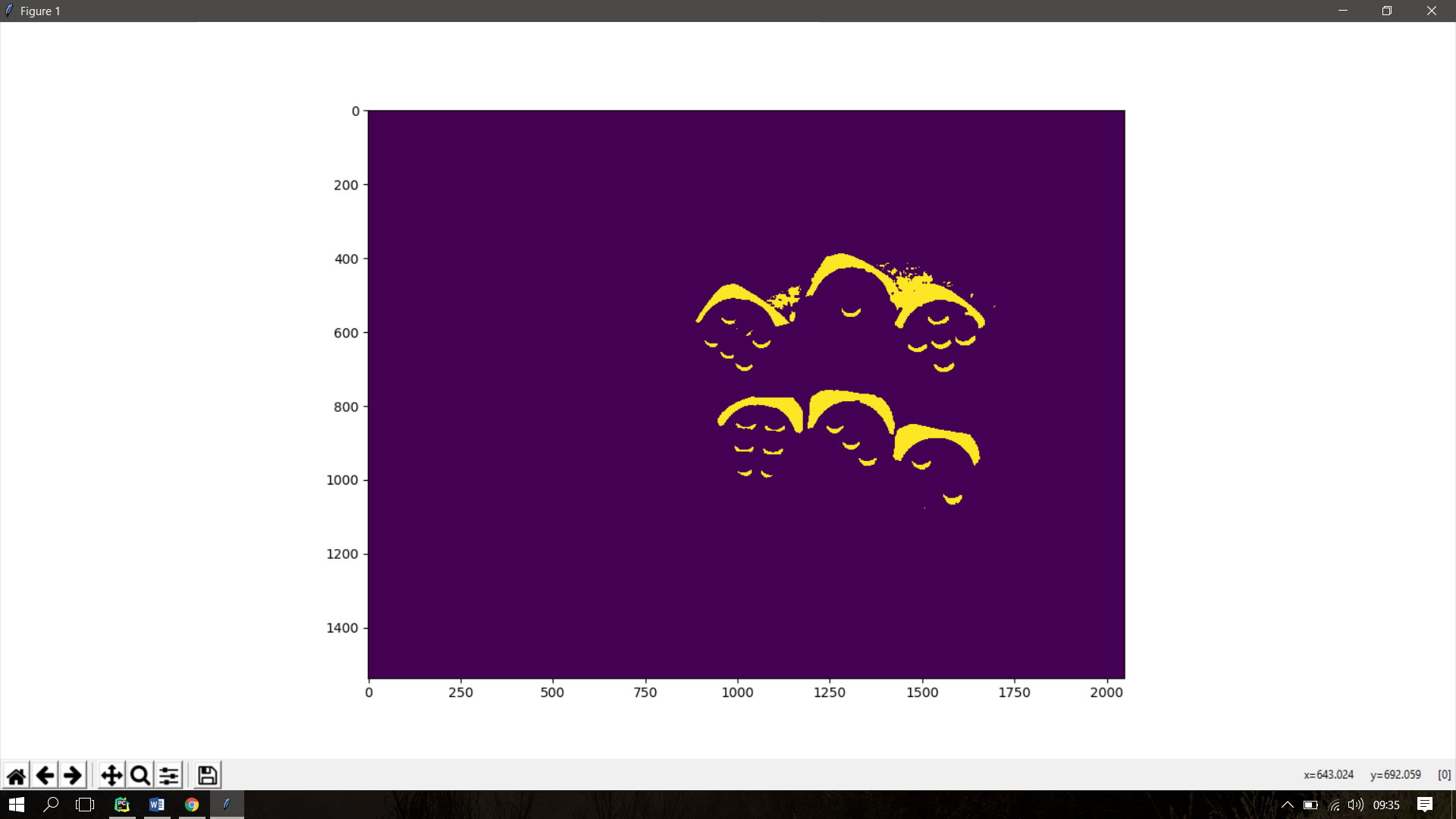
- nierównomierna wielkość: W przypadku, gdy na obrazie znajduje się wiele kości w perspektywie, lub różnej wielkości, ciężko stwierdzić które zostaną wykryte, a które uznane za szum. Algorytm bazuje bowiem na założeniu, że obiekty są podobnej wielkości (usuwane są wszystkie odstające w większym stopniu od zadanego percentyla).

Wszystkie te ograniczenia sprawiają, że program był pisany – niestety – metodą prowadzoną przypadkami testowymi. Starano się, by ich pula była jak najbardziej różnorodna, jednak oczywiście pokrycie wszystkich przypadków nie jest w pełni możliwe, stąd istnieje możliwość, że dla, jakby się mogło wydawać, najtrywialniejszych przypadków, program może zwrócić nieprawidłowy lub zniekształcony wynik. Na następnej stronie znajduje się wynik pozornie prostego przypadku, a także charakterystyka rezultatów.



Rys. 1: Wynik działania program.

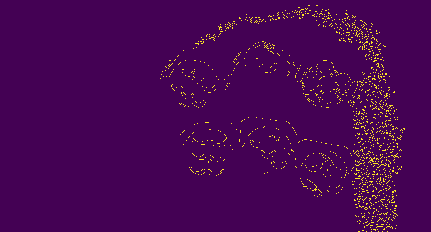
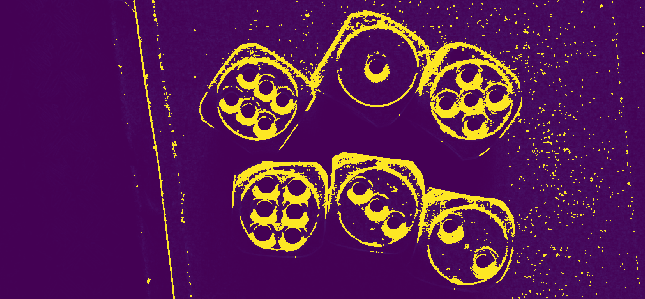
Rezultat może być zaskakujący ze względu na jego niedoskonałość, jednak gdy spojrzymy na wynik jednej z kilku operacji „wytrawiania” kości z rysunku, powód takiego zachowania wydaje się zrozumiały:



Rys. 2: Rezultat zwrócony przez jeden z wyodrębniaczy kości.

Cześć pozostałych przedstawiono poniżej:

Rys. 3: Część wyników pozostałych wyodrębniaczy.



Jak więc widzimy, rezultaty są z goła odmienne. Gdyby zawarta była sztucna inteligencja, prawdopodobnie udałoby sie z obrazka znajdującego się w prawym dolnym rogu wyodrębnić ilość oczek, lub nawet całą kość na podstawie zagęszczenia znalezionych obiektów. Widzimy jednak, że kółka są nieregularne (przypominają raczej elpse), oraz nieciągłe. Możnaby próbować łączyć owe linie, jednak prawdopodonie wtedy uzyskalibyśmy dużo połączonych kształtów, z czego dolnej kostce o 6 kropkach wynikiem byłoby 5. Aby temu zapobiec wymagane byłby zapewne heurystyki weryfikujące położenie kropek w zależności od wykrytej ich ilości, co mogłoby w tym wypadku dać prawidłowy rezultat, w ogólności jednak byłoby dużo bardziej złożone, oraz prawdopodobnie nieopłacalne ze względów czasowych. Same wyniki również mogłyby być niezadowalające oraz często zawodn (pochylenie obrazu, niedoskonałości).