Deskryptor obrazu

Przetwarzanie i Rozpoznawanie Obrazów

Marek Rydlewski 117214
Mateusz Sarbinowski 117246
Politechnika Poznańska, Wydział Informatyki
Poznań

21 maja 2017

I. Wprowadzenie

Celem zadania było stworzenie deskryptora obrazu i efektywnej metody jego porównywania. Aby ułatwić testowanie i wyciąganie wniosków wycięliśmy fragmenty obrazów z zadanej bazy danych i zastosowaliśmy na nich różne przekształcenia takie jak obrót, rozmycie czy przesunięcie - w ten sposób otrzymaliśmy zbiór testowy liczący 700 obrazków - 350 przykładów pozytywnych i 350 przykładów negatywnych. Do zbioru przykładów negatywnych wzieliśmy dwa różne punkty z jednego obrazka, poddaliśmy je również (losowo) przekształceniom.

II. Opis metody

Metodą prób i błędów zdecydowaliśmy się na użycie kilku mniejszych deskryptorów - każdy z nich dokonuje oceny niezależnie, a następnie liczona jest średnia ważona z otrzymanych rezultatów.

- Deskryptor no. 1 - deskryptor oparty o filtr histogramowy, wyliczający histogram w otoczeniu punktu (dla 10 binów, liczbę tę wybraliśmy doświadczalnie - agreguje podobne wartości, jednocześnie gwarantując generowanie wystarczająco zróżnicowanych histogramów). Za porównanie dla tego deskryptora użyliśmy różnicy jedności i wartości bezwzględnej miary korelacji między dwoma histogramami.
- Deskryptor no. 2 - deskryptor opary o pomiar średniej jasności i jej odchylenia standardowego. Wynikiem pracy deskryptora jest suma odchylenia i średniej jasności. Dla odchylenia eksperymentalnie został dobrany mnożnik 12 a dla średniej 4.
- Deskryptor no. 3 - deskryptor oparty o punkty charakterystyczne - dla każdego punktu sprawdzamy jego otoczenie, jeśli oba piksele po przeciwnej stronie są większe (mniejsze) od piksela w środku, to do oceny piksela dodajemy 1. Ocena piksela wacha się pomiędzy 0 a 4, gdzie 4 oznacza punkt charakterystyczny (wszystkie punkty do około mają większą(mniejszą) wartość). a 0 oznacza brak widocznej regularności. Tworzymy wektor ocen zliczających liczność poszczególnych ocen punktów w ramce (w naszym przypadku 62x62, problem estymacji wartości na krawędziach) a następnie normalizujemy tak aby suma elementów wektora była równa 1, to jest nasz

deskryptor. Porównujemy deskryptory obliczając odległość euklidesową pomiędzy dwoma pięciowymiarowymi punktami, otrzymamy wartość w przedziale <0,1>.

- Deskryptor no. 4 - deskryptor oparty o metodę BRIEF - deskryptor działający na tej samej zasadzie jak klasyczny BRIEF. Z otoczenia punktu (którego dostajemy jako argument) losujemy nowy punkt i porównujemy je (czy nowy jest większy), robimy to n razy (w naszym przypadku 128), i z wyników tworzymy wektor. Następnie porównujemy te wektory licząc odległość Hamminga. Punkty losowane są dla całej ramki (64x64), przyjeliśmy rozkład normalny N(0, S**2/25) zgodnie z zaleceniami autorów [1], rozmiar deskryptora to 128 (zwiększenie rozmiaru nie dało znaczącej poprawy, tesotwaliśmy dla 256, 512 i 1024), oraz sigmę równą 1. Deskryptor został zaimplementowany w całości przez nas, ale wspomagaliśmy się implementacją skimage.feature.BRIEF, w celu lepszego zrozumienia działania oraz większej wydajności.

Dla każdego deskryptora określiliśmy dla jakiego progu decyzji (jeśli odległość mniejsza od x to obiekty są podobne) występuje jaka trafność klasyfikacji zarówno dla przypadków gdy porównujemy pasujące punkty jak i zupełnie różne. Dokładnie takiego samego pomiaru dokonaliśmy dla połączonych deskryptorów w sumę ważoną o parametrach: deskryptor pierwszy oraz drugi posiadają wagę 2, drugi i trzeci wagę 1. W ten sposób słabość każdego deskryptora w określonych przypadkach jest kompensowana przez inne deskryptory sprawiające się danych okolicznościach lepiej.

III. **Sprawność i podsumowanie**

Jako miarę jakości algorytmu przyjęliśmy pole pod krzywą ROC dla której nasz algorytm osiągnął wynik 0.8. Warto zauważyć iż taki wynik osiągnęliśmy dla konfiguracji wag, która wcale nie przyznaje najwyższych wag deskryptorom, które samodzielnie osiągają najlepszy wynik.

Bibliografia

[1] BRIEF: Binary Robust Independent Elementary Features *Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua*