

PROJEKT SKŁADA SIĘ Z KILKU NIŻEJ OPISANYCH CZĘŚCI. OPRÓCZ PONIŻSZYCH POLECEŃ OPISZ KRÓTKO KAŻDĄ Z METOD.

21.05.2019 Dodano B.2

DO ODDANIA: Raport w postaci pliku .pdf oraz pliki (preferowany: Python), które powinny się uruchamiać w lab. 417

UWAGA: Poniżej znajduje się opis co powinno znaleźć się w projekcie. Twój projekt powinien te wszystkie elementy zawierać, ale nie musi się tylko i wyłącznie do nich sprowadzać. W szczególności nie są tam wymienione wszystkie rzeczy krok po kroku, opis jest (zamierzenie) luźny.

Część A: Transformata Fouriera

Spróbuj usunąć regularne “zniekształcenia” z następującego obrazka

http://www.math.uni.wroc.pl/~lorek/image_analysis/images/fourier/man.jpg

Wykonaj transformatę Fouriera. Na tej transformacie “usuń” punkty, które mogą odpowiadać tym zniekształceniom. Możesz to zrobić tworząc osobno “ręcznie” obrazek z zaznaczonymi obszarami.

Wykonaj też całą procedurę na przynajmniej jednym innym obrazku.

Część B: Segmentacja semantyczna

B.1 Na stronie

http://www.math.uni.wroc.pl/~lorek/image_analysis/images/segmentation_WBC_part/

umieszczonych jest 5 zdjęć komórek, jest to fragment dostępnej bazy “WBC Image Dataset” dostępnej pod adresem:

https://github.com/zxaoyou/segmentation_WBC

Zadanie polega na segmentacji semantycznej, musimy przypisać każdy piksel do 1 z trzech klas. Do każdego obrazka RGB zapisanego w formacie .bmp dostępny jest obraz .png, w którym 3 kolorami zaznaczone są wszystkie klasy (są to tzw. *ground truth*).

- A) Spróbuj zastosować kmeans z $k = 3$ na chmurze punktów RGB, sprawdź jak wynik ma się do *ground truth*
- B) Podobnie jak wyżej, ale zamień obraz RGB na inną przestrzeń barw (np. YCbCr)
- C) Modelujemy następujące podejście: użytkownik może zaznaczyć ręcznie kilka punktów z każdej z klas, a następnie powinny zostać sklasyfikowane wszystkie pixele. Dla ułatwienia wybierzmy ręcznie (na jednym lub kilku obrazkach) po pięć obszarów (kwadraty 5x5 pixeli). Idea jest taka, by na tych zaznaczeniach się “nauczyć”, a następnie klasyfikować pozostałe pixele. Można zastosować różne podejścia, np.
 - * Punkty z zaznaczeń “wrzucić” do klasyfikatora (jako $x=(R,G,B)$, klasa=0,1 lub 2) i potem sklasyfikować pozostałe punkty
 - * Można np. w każdym zaznaczeniu policzyć wariancję kolorów - np. osobno po każdym kanale - i do klasyfikatora jako wejście dawać $x=(R,G,B, \text{varR}, \text{varG}, \text{varB})$
 - * inne....

Jakość swojej klasyfikacji można najprościej mierzyć licząc liczbę dobrze sklasyfikowanych pixeli (dzieląc przez liczbę wszystkich pixeli). Można też rozważyć wersję z dwoma klasami (tło=czarne na .png, reszta=obiekt), wówczas warto się zapoznać z miarami F1-score, recall, precision, zob. np. https://en.wikipedia.org/wiki/F1_score

B.2 (DODANE 21.05.2019)

Wykonaj to samo na obrazkach zawartych w pliku `iCoseg.part.tar.gz` umieszczonym na stronie: http://www.math.uni.wroc.pl/~lorek/image_analysis/images/

Tam jest 5 zestawów obrazków – dla każdego trzeba zrobić osobne “reczne” zaznaczenie obiektu. (są tam tylko dwie klasy: obiekt, tło)

Obrazy te są częścią zbioru danych: CMU-CORNELL ICASEG DATASET dostępnego na stronie:

<http://chenlab.ece.cornell.edu/projects/touch-coseg/>

Część C*: “Odzyskiwanie” obrazka poprzez Monte Carlo Markov Chains

UWAGA: ta część jest dodatkowa (na ocenę bdb “wystarczy” zrobić bardzo dobrze pozostałe części i poprzedni projekt).

Może tutaj wystąpić problem techniczny spowodowany tym, że obrazy są względnie duże. Jeśli to jest problemem, możesz całość zrobić na dużo mniejszych obrazach (wówczas musisz pomniejszyć “dobre” obrazy i samemu je “zepsuć” w sposób opisany poniżej)

Na stronie

http://www.math.uni.wroc.pl/~lorek/image_analysis/images/mcmc/
znajdują się obrazki binarne (o wartościach 0 lub 255): **mouse_binary.png** i jego dwie “zepsute wersje” oraz obrazek **LenaBinary.jpg** i jego dwie “zepsute wersje”.

Na obrazkach z “025” w nazwie każdy pixel został zmieniony na przeciwny z prawd. 0.025, natomiast na obrazkach z nazwą “15” prawd. wynosiło 0.15

Spróbuj “odzyskać” obrazek używając używając metod MCMC. Dokładniej: skonstruuj łańcuch Markowa o rozkładzie związanym z modelem Isinga (szczegóły były na wykładzie). Uruchom go zapisując wyniki dla różnej liczby kroków (tak by mieć kilka obrazków).