

# Obrada slike u medicini

## Laboratorijska vežba 13: Segmentacija primenom praga

Potrebne biblioteke: cv2, numpy, matplotlib.pyplot, os, scipy.ndimage.convolve1d, scipy.signal.find\_peaks

### 1. Traženje praga za segmentaciju analizom histograma

- 1.1 U skripti v13\_1.py u promenljivu im učitati jednu od slika *mr\_0X.png*. Prikazati sliku. Cilj je segmentacija bele moždane mase iz MRI slika glave. Bela masa je prilično homogeno tkivo koje se nalazi u središnjem delu mozga i karakteriše je visok intenzitet u MRI T1 slikama. Zato se da lako segmentirati kroz jednostavnu klasifikaciju intenziteta.
- 1.2 Napraviti i prikazati histogram slike im. Koliko modova ima histogram?
- 1.3 Na osnovu izgleda histograma izaberite donji prag t1 i gornji prag t2 za koje vam se čini da će najbolje odvojiti belu masu od ostalih tkiva. Podeliti sliku na dva segmenta pomoću pragova (binarna slika koja je 1 tamo gde je bela masa, a 0 drugde). Da li vam se čini da je segmentacija dobra?
- 1.4 Sada ćemo pokušati da automatski pronađemo prag iz histograma. Prvi korak je ublažavanje, tj. NF filtriranje (Gausovim filtrom) histograma kako bi se suzbile sitne lokalne varijacije.
- 1.5 Pronaći pikove u isfiltriranom histogramu upotrebom funkcije find\_peaks. Prilagoditi parametre (height, distance, prominence...) tako da funkcija pronađe lokalne maksimume (pikove) modova sive mase i bele mase. Prikazati pikove na histogramu.
- 1.6 Prag za segmentaciju izračunati kao srednju vrednost između intenziteta na kojem se nalazi lokalni maksimum moda sive mase i lokalni maksimum moda bele mase. Segmentirati sliku upotrebom ovog praga. Da li je pronađeni prag blizu ručno definisanog praga t1?

### 2. Procena kvaliteta segmentacije i optimalni prag

- 2.1 Da bismo proverili kvalitet naše segmentacije potrebno je da segmentiranu sliku uporedimo sa referentnom slikom. Svaka *mr\_0X.png* slika ima odgovarajuću *lab\_0X.png* sliku na kojoj su ručno označeni različiti segmenti. Prikazati odgovarajuću *lab\_0X.png* sliku. Koliko segmenata je označeno na slici?
- 2.2 Izdvojiti belu masu desne i leve hemisfere u referentnu sliku IS.
- 2.3 U skripti osum.py napraviti funkciju tanimoto koja kao ulaz uzima dve binarne slike, a vraća njihov Tanimoto presek kao skalarnu vrednost. Tanimoto presek je količnik preseka dva skupa i njihove unije. Veća vrednost ukazuje na veću sličnost između slika. U slučaju binarnih slika, viđenih kao skupova, presek je suma minimalnih vrednosti na svakom pikselu (ili logičko  $\wedge$ ), dok je unija maksimalna vrednost (ili logičko  $\vee$ ).
- 2.4 Izračunati tanimoto presek između referentne slike IS i slika dobijenih u 1.3 i 1.6. Da li slike liče?

- 2.5 Prikazati sliku apsolutnih razlika između referentne slike i slika iz 1.3 i 1.6. Na kojim regionima je došlo do najvećih grešaka u segmentaciji?
- 2.6 Sada ćemo pokušati da pronađemo optimalni prag za segmentaciju. Raditi segmentaciju u petlji čija je promenljiva vrednost. Segmentiranu sliku onda uporediti sa referentnom pomoću tanimoto indeksa. Optimalni prag je onaj za koji je dobiven najveći tanimoto indeks. Koliko se taj prag razlikuje od onog iz 1.6? Prikazati sliku nakon segmentacije.

### 3. Traženje praga za segmentaciju analizom ivičnih piksela

- 3.1 U skripti v13\_2.py u promenljivu im učitati sliku *hand.tif*. Smanjiti rezoluciju slike, npr. na 2 nivo Gausove piramide i tu sliku nazvati im\_g. Prikazati im i im\_g.
- 3.2 Izračunati gradijente po x i y, magnitudu gradijenta i orijentaciju koristeći funkciju `osum.sobel_grad`. Prikazati sve dobijene slike i izanalizirati šta koja predstavlja.
- 3.3 Normalizovati magnitudu gradijenta na opseg [0, 1] i napraviti binarnu masku im\_gs gde su prikazane lokacije na kojima se nalazi jak gradijent, odnosno gde je magnitude gradijenta veća od 0.1 (vrednost može da se menja). Prikazati tu masku.
- 3.4 Primeriti morfološku operaciju dilatacije na im\_gs koristeći kvadrat 3x3 kao strukturni element. Tako dobijena slika d predstavlja ivice i susedne piksele oko ivica.
- 3.5 Napraviti binarnu sliku d2 na kojoj su predstavljeni samo susedni pikseli oko ivica. To se dobija operacijom logičko i između d i invertovane im\_gs.
- 3.6 Prag t0 se procenjuje kao srednja vrednost susednih piksela, tj. srednja vrednost na lokacijama definisanim u d2.
- 3.7 Napraviti binarne mape g\_svetlo i g\_tamno koje predstavljaju lokacije piksela koji imaju veći (anatomija) odnosno manji intenzitet (pozadina) od t0.
- 3.8 Prag t\_svetlo se računa kao prosečna vrednost na lokacijama definisanim u g\_svetlo, a t\_tamno kao prosečna vrednost na lokacijama definisanim u g\_tamno. Konačni prag za segmentaciju tc dobija se kao prosek t\_svetlo i t\_tamno.
- 3.9 Primeriti prag na sliku im i prikazati rezultat segmentacije. Da li je rezultat zadovoljavajuć?
- 3.10 Ponoviti algoritam za sliku *shoulder.png*. Da li je rezultat segmentacije zadovoljavajuć?