

## OE4DOS 2024/2025 – treći domaći zadatak

U arhivi uz domaći zadatak dat je skup slika sata. Vreme pokazuju kazaljke za sate, minute i sekunde. Potrebno je napisati funkciju *extract\_time* na osnovu ulazne slike analognog sata vraća informaciju o vremenu koje sat pokazuje.

Rešavanje zadatka je potrebno podeliti u više celina:

- 1) [20] Napisati funkciju *canny\_edge\_detection* koja za ulaznu sliku (**img\_in**), standardnu devijaciju Gausove funkcije (**sigma**) i apsolutne vrednosti nižeg (**thr\_low**) i višeg praga (**thr\_high**) detektuje ivice korišćenjem Kanijevog algoritma. Parametri funkcije su:

Ulazni parametri:

- img\_in** – Ulazna slika.
- sigma** – Standardna devijacija Gausove funkcije. Radijus prozora Gausovog filtra se određuje kao  $3 \cdot \sigma$ .
- thr\_low** – Apsolutna vrednost nižeg praga za histerezisno poređenje.
- thr\_high** – Apsolutna vrednost višeg praga za histerezisno poređenje.

Izlazni parametri:

- edges** – Mapa detektovanih ivica. Na pozicijama gde su detektovani ivični pikseli ima vrednost 1 dok na ostalim pozicijama ima vrednost 0.

Procesiranje Kanijevim algoritmom se sastoji iz sledećih koraka:

1. Filtriranje ulazne slike Gausovom funkcijom zadate standardne devijacije. Za radijus Gausove funkcije usvojiti vrednost od 3 sigma.
2. Određivanje horizontalnih i vertikalnih gradijenata tako filtrirane slike. Koristiti Sobelov operator za izdvajanje gradijenata.
3. Određivanje magnitude i ugla gradijenta.
4. Kvantizacija gradijenta na jedan od 4 pravaca  $-45^\circ$ ,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ .
5. Potiskivanje vrednosti gradijenata koje ne predstavljaju lokalne maksimume.
6. Određivanje mapa jakih i slabih ivica na osnovu vrednosti nižeg i višeg praga.
7. Uključivanje u izlaznu mapu ivica onih slabih ivica koje su povezane sa nekom jakom ivicom.

U izveštaju detaljno opisati svaki od ovih navedenih koraka i prikazati međurezultate za sliku **lena.tif**. Testirati funkciju nad slikama **lena.tif**, **shapes\_noisy.tif**, **camerman.tif**, **house.tif**, **van.tif**. Testirati funkciju sa različitim vrednostima ulaznih parametara i iskomentarisati rezultate.

- 2) [15] Napisati funkciju ***get\_line\_segments*** koja na osnovu matrice izdvojenih ivica i pravca linije zadate u normalnoj reprezentaciji izdvaja sve segmente zadate minimalne dužine koje se nalaze na datom pravcu.

Ulazni parametri:

- img\_edges*** – Slika sa detektovanim ivičnim pikselima sa koje se detektuju linije.
- line*** – Niz od dva elementa (*theta*, *rho*) kojim se zadaje pravac linije.
- min\_length*** – Minimalna dužina segmenta da bi duž bila detektovana.
- max\_gap*** – Maksimalna dužina prekida koji se može ignorisati.
- tolerancy*** – Radijus okoline u okviru koje se traže ivični pikseli

Izlazni parametri:

- segments\_num*** – Broj detektovanih segmenata.
- segments\_size*** – Niz koji sadrži veličine detektovanih segmenata.
- segments\_coord*** – Lista detektovanih koordinata detektovanih segmenata pri čemu je svaki segment predstavljen sa koordinatama početne i krajnje tačke.

Segment predstavlja skup ivičnih tačaka koje su susedne (8-susedstvo) i sve se nalaze na zadatom pravcu. Svaki segment je određena početnom i krajnjom tačkom. Da bi segment bio validan potrebno je da ispunjava sledeće kriterijume:

1. **Sve tačka segmenta koje se nalaze na pravcu između početne i krajnje tačke predstavljaju ivične piksele.** Ovaj kriterijum se uzima sa određenom tolerancijom, odnosno smatra se da element segmenta predstavlja ivični piksel ukoliko se u njegovoj okolini nalazi ivični piksel. Radijus ove okoline se zadaje kao ***tolerancy*** parametar funkcije.
2. Segment se smatra validnim ukoliko postoje neki elementi koji ne predstavljaju ivične piksele ukoliko je maksimalna (pojedinačna) dužina nedostajućih segmenta manja od zadatog parametra ***max\_gap***. **Segment počinje i završava se ivičnim pikselom (uzimajući u obzir toleranciju).**
3. Ukupna dužina segmenta koji ispunjava predhodna dva kriterijuma je veća od minimalne dužine linije zadate parametrom ***min\_length***.

U izveštaju detaljno opisati način realizacije funkcije i prikazati međurezultate. Testirati funkciju sa različitim vrednostima ulaznih parametara na slici ***shapes\_noisy.tif*** i iskomentarisati rezultate.

- 3) [25] Napisati funkciju *extract\_time* koja za ulaznu sliku (*img\_in*) vraća informaciju o vremenu.

Ulazni parametri:

*img\_in* – Ulazna slika sata.

Izlazni parametri:

*time\_out* – Niz od 2 elementa koji predstavlja detektovano vreme u formatu  $[h, m]$ .

Iskoristiti prethodno realizovane funkcije za detekciju ivica i duži kako bi se odredila pozicija i orijentacija kazaljki sata. Voditi računa o tome da *ScikitImage* funkcija *hough\_line* određuje ugao u odnosu na horizontalnu osu (pogledajte primer sa vežbi).

U izveštaju detaljno opisati način realizacije funkcije i prikazati međurezultate za karakteristične primere. Testirati realizovanu funkciju nad svim elementima zadatog skupa i prikazati detektovana vremena za sve zadate sekvence u izveštaju.

**Funkcija prihvata samo matricu ulazne slike, svi ostali parametri se automatski podešavaju u funkciji. Za pun broj poena nije dozvoljena ručna promena parametara od slike do slike.** Ukoliko generisano rešenje ne rešava sve test slučajeve analizirati posebno u čemu je problem i eventualno podesiti posebne parametre za podskup slika tako da se problem reši. Na taj način se može smanjiti broj izglubljenih poena za nerešene test slučajeve.

**[Bonus 10]** Proširiti funkciju *extract\_time* tako da vraća niz u kom su pored sata i minuta detektovane i sekunde  $[h, m, s]$ . Ukoliko na slici nije detektovana kazaljka za sekunde, za vrednost sekundi vraća vrednost -1 na poziciji *s*. Kreirati sopstveni test skup realnih slika analognih satova. Omogućiti da funkcija radi i u slučajevima kada se sat ne nalazi u centru slike, za različito osvetljene scene, veličine slika i modele satova. Smatrati da je sat kružnog oblika, da zauzima veći deo scene i da je orijentisan tako da je linija koja spaja brojeve 12 i 6 vertikalna. Prilikom kreiranja test skupa voditi računa da budu ispunjeni navedeni uslovi. Postoji skriveni test skup koji zadovoljava gore navedene kriterijume. Pun broj bonus poena će biti dodeljen ukoliko realizovana funkcija detektuje ispravna vremena i nad elementima skrivenog test skupa. Deo skrivenog skupa je dat u direktorijumu *clocksBonus*. Opisati način kreiranja sopstvenog test skupa i koji karakteristični slučajevi su pokriveni. Detaljno opisati rešenje. Potrebno je da funkcija za detekciju prihvata samo slike satova a da se svi parametri automatski podešavaju unutar funkcije.

Hint: Kazaljka za sekunde je znatno tanja od ostalih kazaljki, ili je obojena. Mogu se prvo detektovati sati i minuti a onda kad je poznata pozicija centra sata se može detektovati kazaljka za sekunde. Skalirati slike na ulazu na predefinisano veličinu po jednoj (obično manjoj) dimenziji. Na taj način se omogućava uniformnije podešavanje parametara unutar funkcije. Prave koje sadrže kazaljke se seku u centru sata, a centar sata je bar po jednoj koordinati relativno blizu polovine slike (sat zauzima veći deo scene).

Rešenje domaćeg zadatka napisati u okviru jedne sveske *domaci3\_gg\_bbb.ipynb* pri čemu je rešenje posebnih tačaka potrebno podeliti u posebne ćelije (ili više ćelija za jednu tačku ako ima više smislenih celina).

**Napomena:** Nemojte slati slike koje su date uz zadatak. Skripta za testiranje treba da bude u okviru direktorijuma **domaci3\_gg\_bbb** pri čemu se podrazumeva da se ulazne sekvence nalaze na relativnoj putanji **../sekvence**.

Fajlove **domaci3\_gg\_bbb.ipynb**, **izveštaj (domaci3\_gg\_bbb.pdf)**, kao i sve dodatne fajlove potrebne za testiranje (dodatni test skup satova ukoliko postoji) zapakovati u **domaci3\_gg\_bbb.zip** i okačiti na OneDrive. Link ka rešenju poslati na adresu [elmezeni@etf.rs](mailto:elmezeni@etf.rs) sa subjectom **OE4DOS treci domaci**.

Rok za predaju rešenja domaćeg zadatka je **nedelja 22.12.2024.**  
**Svaki dan kašnjenja povlači -10% osvojenih poena!**