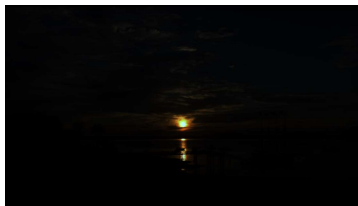
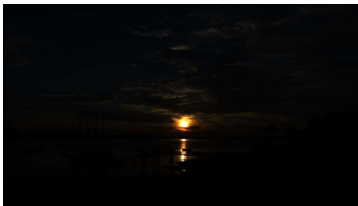


OE4DOS 2020/2021 – prvi domaći zadatak

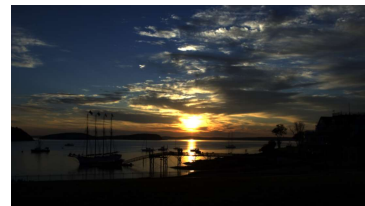
1. [20] U okviru ovog zadatka potrebno je odrediti što bolje mapiranje za prikaz slike *sea.hdr*. Slika predstavlja scenu širokog dinamičkog opsega direktno snimljenu sa senzora, bez gama korekcije. Ulaznu sliku pre transformacije skalirati u opseg $[0, 1]$. Na slici su prikazane tri varijante skaliranja ulaza pre transformacije. Vrednosti izlazne slike, nakon transformacije, koje izlaze van opsega $[0, 1]$ zasititi na 0 odnosno 1.



(a) Skaliranje bez gubitaka. Maksimalna vrednost ulaza je mapirana na vrednost piksela 1.

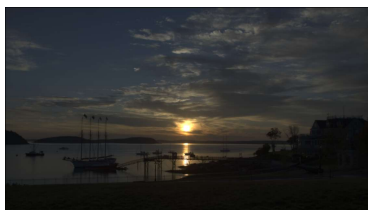


(b) 0.1% najsvetlijih ulaznih piksela je zasićen na vrednost 1. Ostali pikseli su linearno skalirani.

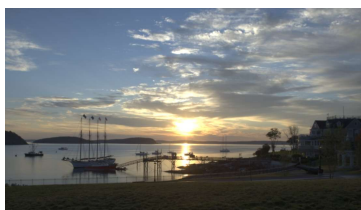


(c) 1% najsvetlijih ulaznih piksela je zasićen na vrednost 1. Ostali pikseli su linearno skalirani.

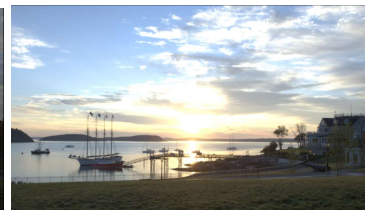
- a) Prikazati sliku korišćenjem linearnog mapiranja. Nakon linearne transformacije primeniti odgovarajuću gama korekciju (koristiti faktor $1/2.2$) pre čuvanja i prikaza izlazne slike. Na slici su prikazani rezultati linearnog skaliranja za 3 različita parametra skaliranja linearne funkcije. Odrediti parametre A, B i C.



(a) Linearno skalirana ulazna slika. Korišćen je parameter skaliranja A.



(b) Linearno skalirana ulazna slika. Korišćen je parameter skaliranja B.



(c) Linearno skalirana ulazna slika. Korišćen je parameter skaliranja C.

- b) Iskoristiti logaritamsku i stepenu funkciju za transformaciju slike. Testirati funkcije sa različitim parametrima i prokomentarisati rezultate. Na slici je prikazan rezultat logaritamske i stepene transformacije za jedan set parametara.



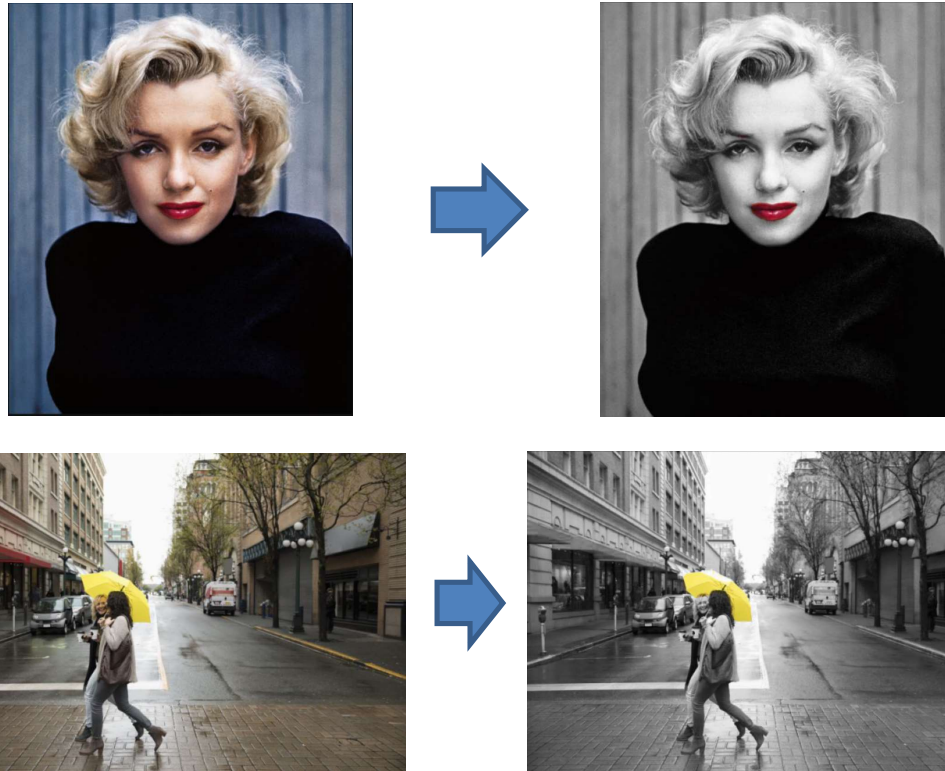
(a) Slika mapirana korišćenjem logaritamske funkcije.



(b) Slika mapirana korišćenjem stepene funkcije.

- c) Predložiti još neka mapiranja, komentarisati rezultate i odabrati najbolji prikaz. Naći što bolji kompromis između kompleksnosti procesiranja i kvaliteta izlazne slike.

2. [20] Potrebno je realizovati selektivnu dekolorizaciju slika *marlyn.jpg* i *street.jpg*. Selektivna dekolorizacija podrazumeva da se slika u boji konvertuje u sivu sliku osim određenih delova koji treba da ostanu u boji. Nije dozvoljeno fizički zadavati koordinate maske, već je selekciju potrebno obaviti isključivo na osnovu boje oblasti.



Detaljno opisati ideju. Postupno predstaviti rešenje generisanja odgovarajućih maski.

3. [20] Realizovati funkciju *bilateral_filter* koja filtrira ulaznu sliku izbegavajući zamućivanje značajnih ivica. Težine bilateralnog filtra prilikom filtriranja piksela na poziciji $[m,n]$ su date sledećim izrazima:

$$w[k,l] = e^{-\frac{k^2+l^2}{2\sigma_s^2}} e^{-\frac{(x[m+k,n+l]-x[m,n])^2}{2\sigma_r^2}}$$

$$k,l \in [-r,r]$$

Radijus filtra je r odnosno prostorna maska je dimenzija $(2r+1) \times (2r+1)$. Dimenzije slike su $M \times N$. Vodite računa da je u pitanju prostorno varijabilna maska, odnosno da će vrednost koeficijenata biti različita za različite položaje u slici (zavisi od koordinata m i n). Ne zaboravite da normalizujete koeficijente maske!!! Realizovana funkcija treba da ima sledeći interfejs:

bilateral_filter(x, radius, sigma_s, sigma_r)

Funkcija može da primi sliku x koja može biti tipa uint8 ili double i tada se podrazumeva da je opseg vrednosti $[0,255]$ odnosno $[0,1]$ respektivno. Unutar funkcije proveriti validnost unetih parametara i u slučaju nevalidnih parametara vratiti poruku o grešci. Sva interna izračunavanja se obavljaju nad brojevima u pokretnom zarezu. Izlazna slika treba da ima isti format kao ulazna slika. Standardna devijacija za intenzitete *sigma_r* podrazumeva opseg intenziteta $[0,1]$. Izlazna slika je istog tipa kao i ulazna.

Proširivanje slike obaviti unutar same funkcije ponavljanjem odgovarajućeg broja redova i kolona. Izlazna slika je istih dimenzija kao ulazna.

Testirati napisanu funkciju za različite vrednosti ulaznih parametara i prikazati ove rezultate u izveštaju. Odabrati pogodnu sliku za testiranje i nju poslati uz rešenje.

Izmeriti brzinu rada ove funkcije za različite vrednosti radijusa. Obavezne vrednosti radijusa za testiranje brzine su: 2, 4, 20, 40. Brzina se može izmeriti direktno iz Jupyter-a. Potrebno je uključiti biblioteku **time** (import time). Dužina trajanja određenog dela koda se meri na sledeći način:

```
start = time.time()

## deo koda za koji se meri vreme ##

end = time.time()
execution_time = (end - start)

print("Vreme izvršavanja: " + str(round(execution_time,3)) + "s \n")

execution_time_norm = execution_time/np.size(x)
print("Vreme izvršavanja: " + str(round(execution_time*1e6,3)) + " us/pix \n")
```

Vreme izvršavanja normalizovati sa brojem piksela u slici koja je procesirana.

Broj poena između ostalog zavisi od robusnosti funkcije, kvaliteta komentara i brzine rada.

Uporediti rezultate kao i vreme izvršavanja sa ugrađenom funkcijom **skimage.restoration.denoise_bilateral**.

Rešenje za sve tačke ovog domaćeg napisati u okviru jedne sveske **domaci1_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, kao i svi fajlovi koji vam nisu bili zadati treba da budu u okviru direktorijuma **domaci1_gg_bbb** pri čemu se podrazumeva da se ulazne sekvence nalaze na relativnoj putanji **../sekvence**.

Na primer: `I = imread('../sekvence/marlyn.jpg');`

Fajlove **domaci1_gg_bbb.ipynb**, korišćene ulazne slike koje nisu bile u zadatku (ako ih ima), **izveštaj (domaci1_gg_bbb.pdf)**, kao i sve dodatne fajlove potrebne za pokretanje glavnog programa zapakovati u **domaci1_gg_bbb.zip** i okačiti na OneDrive. Link ka rešenju poslati na adresu elmezeni@etf.rs i meja@etf.rs sa subjectom **OE4DOS prvi domaci**.

Rok za predaju rešenja domaćeg zadatka je **nedelja 06.12.2020.**

Svaki dan kašnjenja povlači -10% osvojenih poena!