Duboko učenje 2

Siječanj, 2025.

Kolorizacija crno-bijelih slika pix2pix GAN-om

Tim:

Magda Radić

Antonia Meštrović

Bernard Ibrović

Juraj Skokandić

Sadržaj

Uvod 3
Pix2pix 3
Implementacija 4
Rezultati 6

Uvod

Kolorizacija crno-bijelih slika jedna je od zanimljivih zadataka dubokog učenja. Uz tradicionalne regresijske modele kolorizacije, u novije vrijeme sve se više koriste i generativni modeli, posebice GAN odnosno cGAN modeli. Želimo ispitati kako dodavanje adversarijalnog gubitka može poboljšati rezultate.

Pix2pix

Pix2pix arhitektura je uvjetnog GAN-a (en. cGAN) namijenjena za zadatke prevođenja slika-u-sliku (image-to-image) predstavljena u radu **Image-to-Image Translation with Conditional Adversarial Networks.** U radu se detaljno opisuje primjena arhitekture za razne zadatke uključujući i kolorizaciju.

U-Net Generator

Kao generator koristi se U-Net mreža sljedeće arhitekture

- Enkoder :
  - 8 konvolucijskih slojeva – 4x4, stride=2 , padding= 1
  - Svaki sloj popraćen BatchNormalizacijom i LeakyReLU(0.2) aktivacijom (osim ulaznog)
- Dekoder:
  - 8 dekonvolucijskih slojeva – 4x4, stride=2, padding=1
  - Svaki sloj popraćen BatchNormalizacijom i LeakyReLU aktivacijom (osim izlaznog koji koristi tangens hiperbolni)
  - Skip veze svakoj sloja s odgovarajućim poljem značaki enkodera

PatchGAN Diskriminator

Kao diskriminator koristi se PatchGAN mreža sljedeće arhitekture:

- 5 konvolucijskih slojeva – 4x4, stride=2, padding=1
- BatchNorm nakon svake konvolucije osim u prvom i zadnjem sloju
- LeakyReLU(0.2) nakon svih slojeva osim zadnjeg

Izlaz ovakvog PatchGAN-a je matrica 30x30 gdje svaki element odgovara „istinitosti“ jednog kvadrata u izvornoj slici. Receptivno polje tj. veličina kvadrata je 70x70 – preporučeno od strane autora u radu.

Gubitak

Gubitak GAN-a računa se u odnosu na oznake odnosno matrice jedinica ili nula. Računa se binarna unakrsna entropija (BCEWithLogitsLoss), odnosno srednja kvadratna greška (MSELoss).

Generator dodatno, uz gubitak GANa dobiva i regresijski L1 gubirak koji se množi s parametrom Lambda-L1.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

Implementacija

Skup podataka

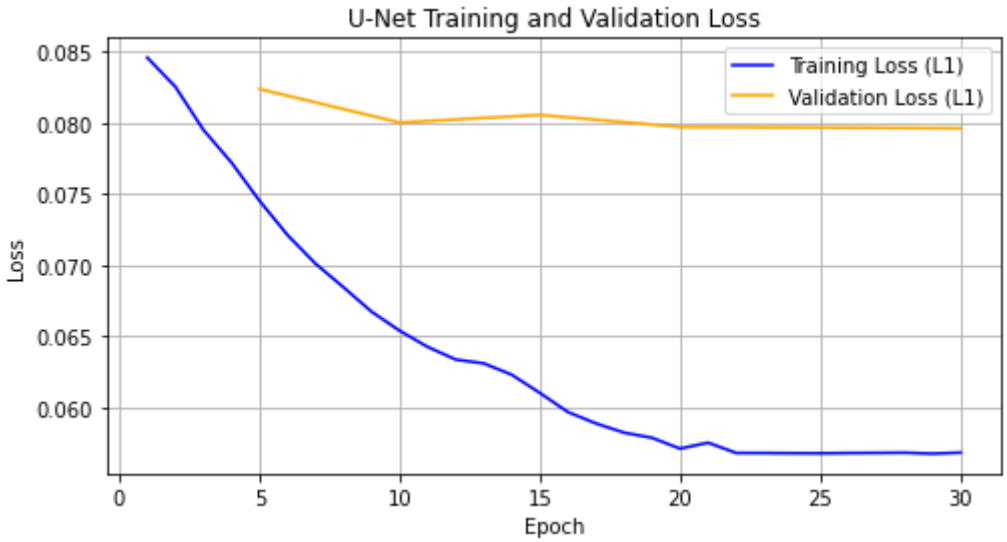
Za treniranje i validaciju koristili smo podskup COCO skupa od 10 000 uzoraka. Za treniranje je korišteno 8000, a validaciju 2000 slika.

Slike su transformirane u veličinu 256x256 i pretvorene iz RGB formata u Lab format. Ovo je učinjeno s namjerom olakšavanja modelu rekonstrukcije slika. Generator na ulaz dobiva 1 kanal (L) te na izlazu daje rekonstrukciju a i b kanala. Ovo smanjuje prostor mogućih „rješenja“ (model rekonstruira samo 2 a ne 3 kanala) i stabilizira treniranje.

Veličina minigrupa u dataloaderima bila je 16.

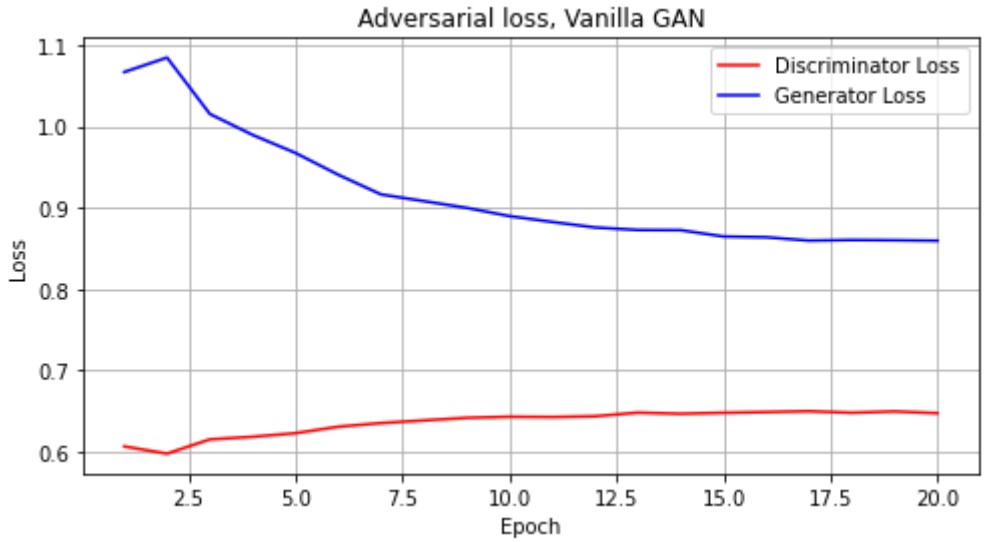
U-Net predtreniranje

Kako bismo ubrzali treniranje odlučili smo najprije regresijski predtrenirati U-Net Generator običnim L1 gubitkom. Osim toga, inicijlaizirali smo težine enkoderskog dijela U-Neta težinama preuzetim od ResNet18 mreže. Ovakav U-Net trenirali smo 30 epoha.



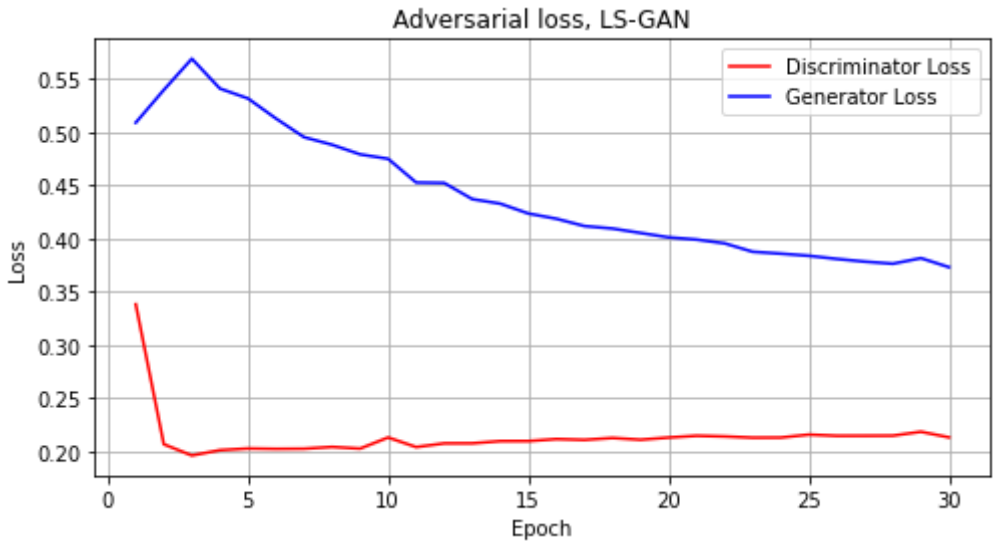
Vanilla-GAN

Inicijalizirali smo pix2pix GAN s U-Net generatorom treniranim 30 epoha, i za gubitak odabrali „vanilla“ opciju. U ovoj varijanti gubitak diskriminatora u odnosnu na ispravne oznake (matrica jedinica ili nula) jednak je binarnoj unakrsnoj entropiji (**nn.BCEWithLogitsLoss**()). Ova GAN trenirali smo 20 epoha



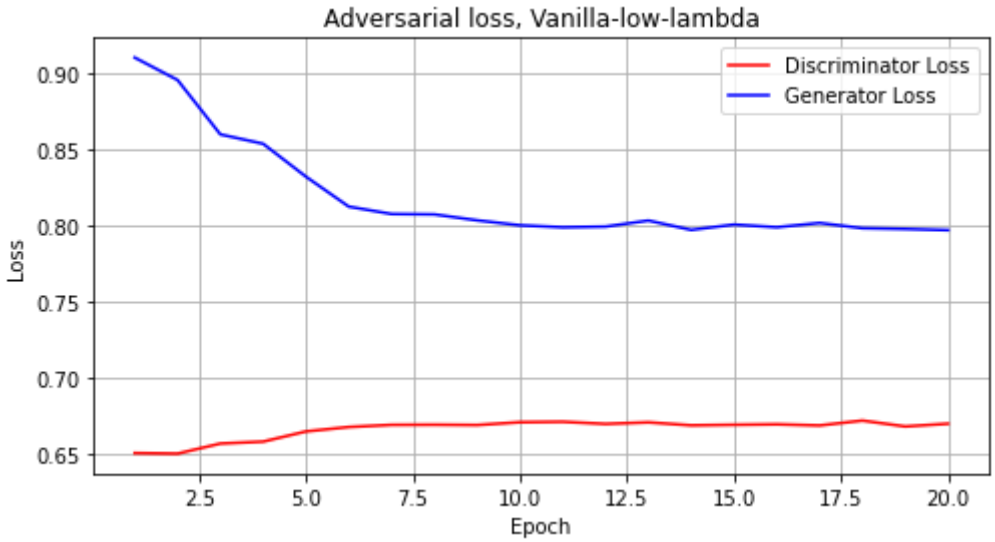
LS-GAN

Inicijalizirali smo pix2pix GAN s U-Net generatorom treniranim 30 epoha, i za gubitak odabrali „ls_gan“ opciju. U ovoj varijanti gubitak diskriminatora u odnosnu na ispravne oznake (matrica jedinica ili nula) jednak je srednjoj kvadratnoj pogrešci (**nn.MSELoss**()). Ovaj GAN trenirali smo izvorno 20, a zatim još 10 epoha, jer je nakon 20 epoha pokazivao bolje rezultate od ostalih.



Low-Lambda Vanilla-GAN

S obzirom da smo primijetili da po dosta pokazatelja Vanilla-GAN i LS-GAN ne daju rezultate značajno drugačije od običnog U-Neta odlučili smo smanjiti hiperparametar LAMBDA_L1 s kojim se množi L1 gubitak. Ovime smo htjeli smanjiti utjecaj L1 gubitka u ukupnom gubitku generatora, odnosno povećati utjecaj gubitka od samog diskriminatora. LAMBDA_L1 u prethodna dva GAN-a iznosila je 100, a u ovom modelu 50. Ostalo je isto kao i kod Vanilla-GANa. Ovaj model trenirali smo također 20 epoha.



Svi su modeli koristili Adam optimizator s parametrima (beta_1, beta_2)= (0.5, 0.99) i stopom učenja i za generator i diskriminator 2x10^-4.

S time smo završili treniranje modela i tako dobili 5 verzija generatora koje smo nazvali:

U-Net_30, Vanilla_GAN_20, LS_GAN_20, LS_GAN_30, VANILLA_LOW_LAMBDA_20,

sukladno s opisanim postupcima treniranja.

Rezultati

Prethodno opisanih 5 modela evaluirali smo koristeći 6 različitih metrika te rezultate usporedili međusobno i sa „savršenom rekonstrukcijom“ (en. ground truth). Sve metrike za sve modele evaluirane su na skupu za validaciju od 2000 256x256 slika, uzetog iz COCO skupa (https://cocodataset.org/), uspoređujući izvorne i rekonstruirane slike u boji (ili samo rekonstruirane, ovisno o metrici)

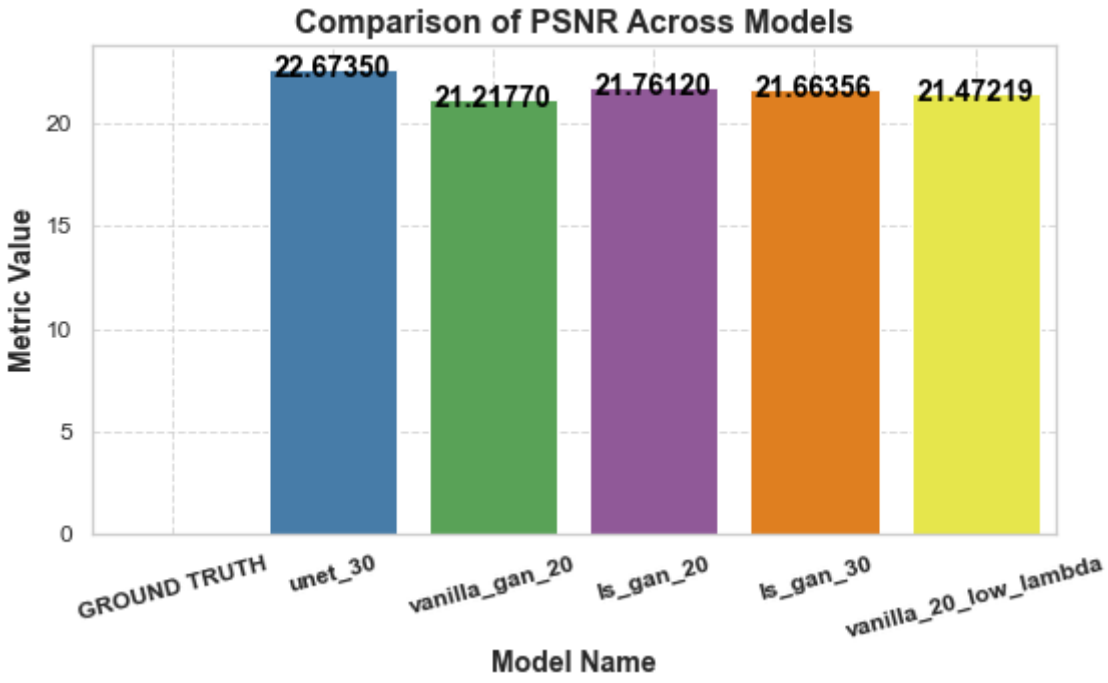
Metrike

Sve navedene metrike računali smo u RGB prostoru, s tim da je za izračun prve 4 potrebna referentna istinita RGB slika u boji, dok za zadnje dvije nije, te metrike jednostavno dodjeljuju ocjenu RGB slikama, bez usporedbe sa „stvarnom“. Sve ih navodimo i ukratko objašnjavamo kako funkcioniraju:

- PSNR (en. Peak Signal-to-Noise Ratio) mjeri sličnost rekonstruirane i izvorne slike

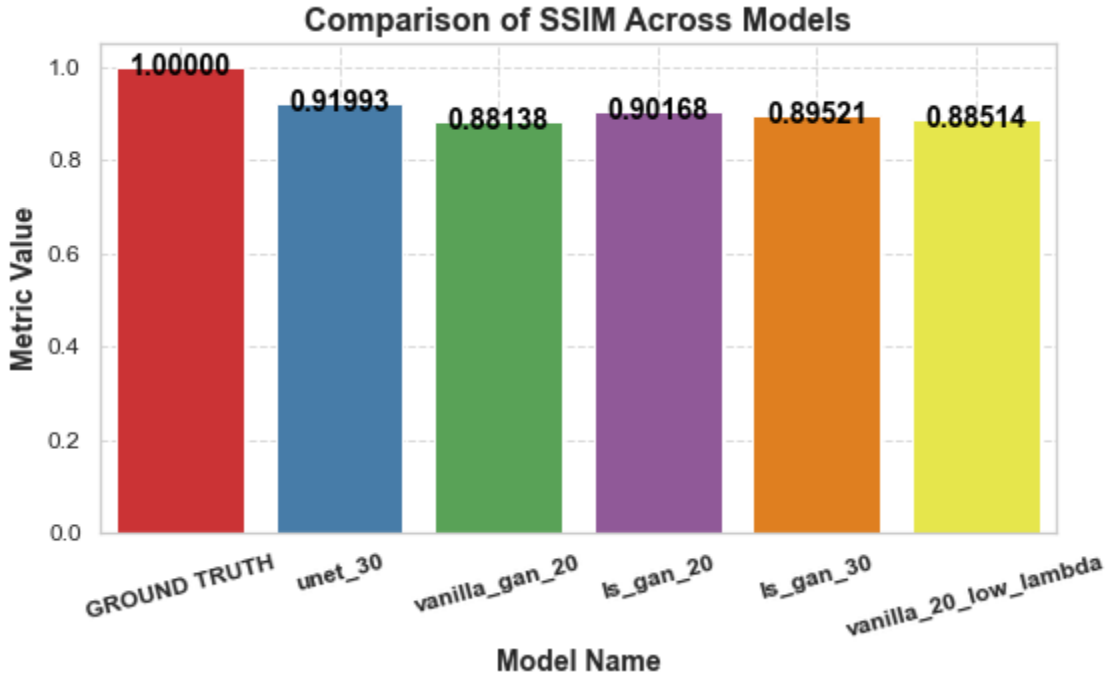
$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - K(i,j))^2 \qquad PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

  Veće vrijednosti indiciraju bolju rekonstrukciju. PSNR vrijednost slike sa samom sobom je ∞.



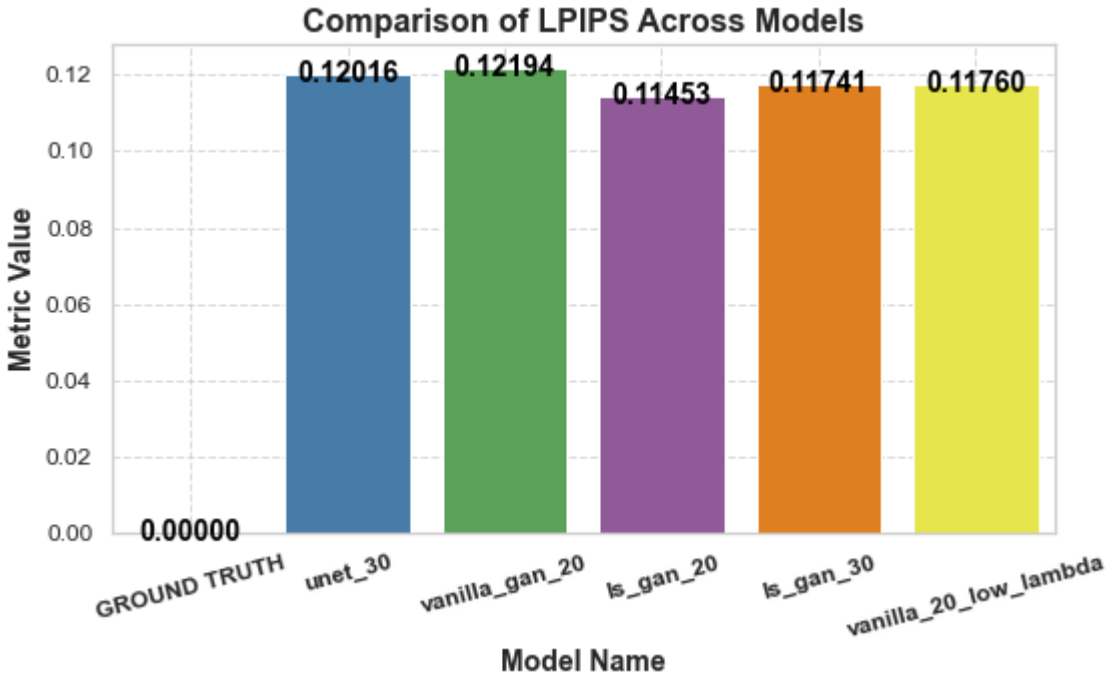
- SSIM (en. Structural Similarity Index) mjeri perceptualnu sličnost dviju slika na temelju svjetline, kontrasta i strukture. Vrijednosti su u rangu [0,1] ; 1 indicira savršenu sličnost (slike su identične), a 0 nepostojanje sličnosti.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

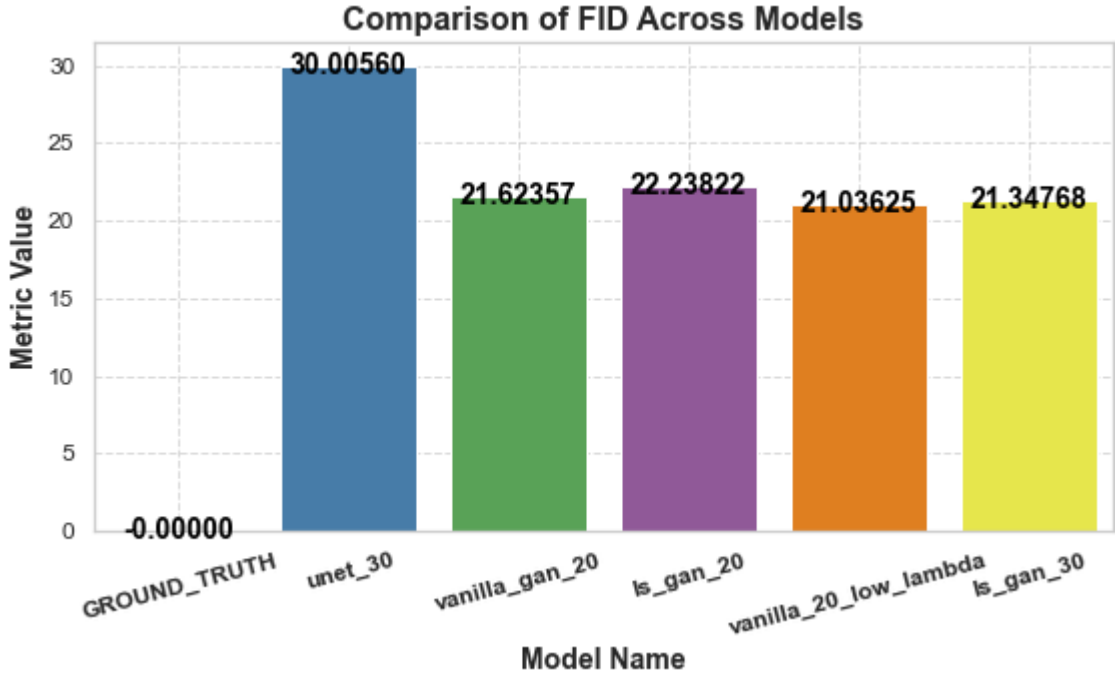


- LPIPS (en. Learned Perceptual Image Patch Similarity) mjeri sličnost tj. udaljenost dviju slika temeljem
značajki koje se ekstrahiraju korištenjem neuronske mreže poput AlexNet-a ili VGG-a.

Mi smo koristili AlexNet i računali udaljenost od istinitih slika; niže vrijednosti indiciraju bolje rezultate (manja udaljenost u prostoru značajki).



- FID (en.Frechet Inception Distance) je metrika koja također mjeri udaljenost korištenjem značajki neuronske mreže, ali se razlikuje po tome što ne uspoređuje značajke za parove slika već računa statistike o značajkama za stvarni i lažni skup slika, te na temelju očekivanja i varijance značajki tih dvaju skupova računa statističku udaljenost između stvarnog i lažnog skupa podataka. Mi smo koristili InceptionV3 neuronsku mrežu, te ekstrahirali 2048 značajki po slici. Niže vrijednosti indiciraju veću sličnost (FID skupa sa samim sobom je 0.)



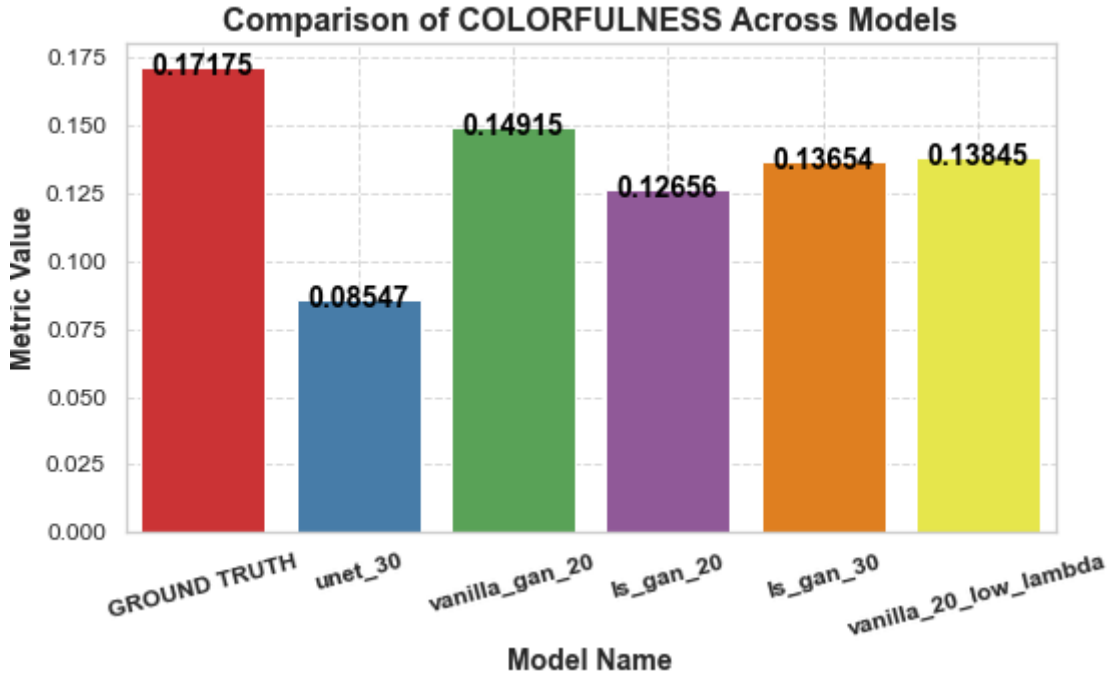
Sljedeće dvije metrike ne rade usporedbu s istinitom slikom u boji već daju ocjenu samoj slici.

- Šarenilo (en. Colorfulness) metrika koja mjeri intenzitet boja u slici temeljem RG (Red-Green) i YB (Luminance-Blue) kontrasta:

$$\mathrm{RG} = R - G$$

$$\mathrm{YB} = 0.5 \times (R + G) - B$$

$$\text{Colorfulness} = \sigma_{\mathrm{RG}} + 0.3 \times \sqrt{\mu_{\mathrm{RG}}^2 + \mu_{\mathrm{YB}}^2}$$



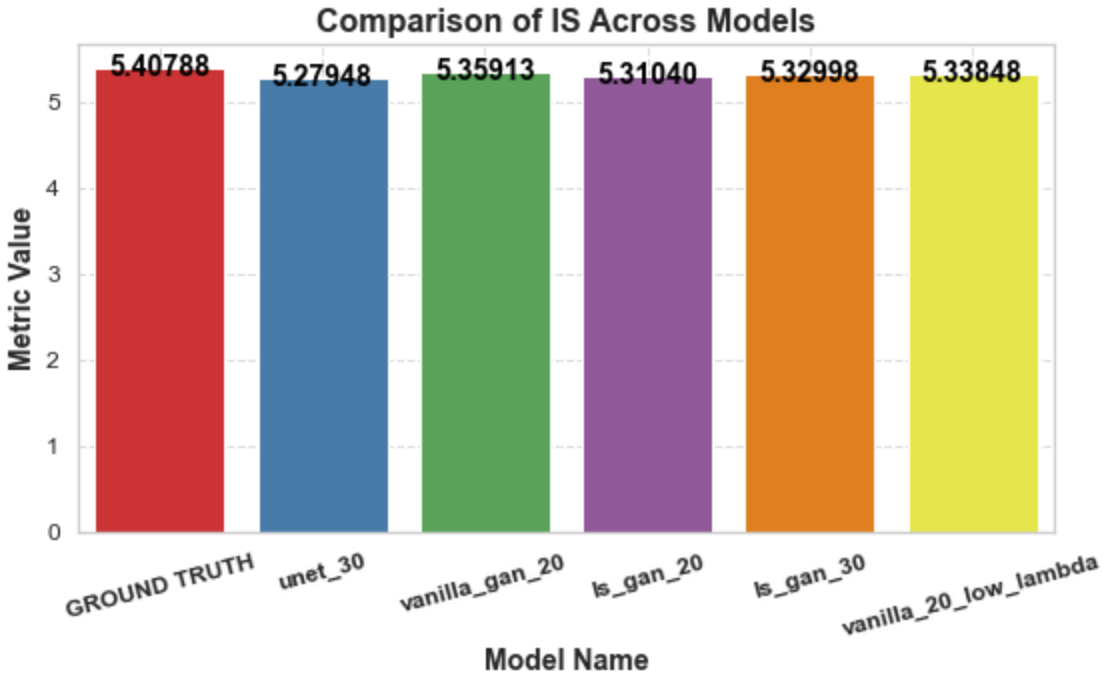
- IS (en. Inception Score) : ova metrika koristi InceptionV3 model koji je treniran na velikom skupu (ImageNet) za zadatke klasifikacije (detekcije vrsta objekata na slici). Visoka IS vrijednost indicira visoku kvalitetu skupa (prepoznatljivi i pouzdano klasificirani uzorci po modelu InceptionV3) i raznolikost (uzorci pokrivaju mnoge kategorije).

$$\mathrm{IS}(x) = \exp\left(\mathbb{E}_x\left[D_{\mathrm{KL}}\left(p(y|x) \| p(y)\right)\right]\right)$$

x is a generated image.

$p(y|x)$ is the conditional probability distribution of the class labels for image x according to the Inception model.

$p(y)$ is the marginal distribution over all generated images.



Vizualna inspekcija

Primjer- lijevo: kolorizacija ls_gan_20 modelom, desno: stvarne slike

Primjer: lijevo: crno-bijela slika, desno: kolorizacija ls_gan_20 modelom