SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA

Diplomski studij

**KORIŠTENJE PIX2PIX ZA RESTAURACIJU SLIKA, BOJENJE CRNO-BIJELIH SLIKA DUBOKIM UČENJEM S VEĆ ISTRENIRANIM ALGORITMOM**

Marko Jović

Osijek, 2023.

Sadržaj

[UVOD 3](#_Toc135151981)

[Biblioteke 4](#_Toc135151982)

[Neuronske mreže 4](#_Toc135151983)

[Pix2Pix 5](#_Toc135151984)

[PODATCI I NJIHOVA OBRADA 6](#_Toc135151985)

[U-NET 8](#_Toc135151986)

[Generator 11](#_Toc135151987)

[Diskriminator 12](#_Toc135151988)

[TRENIRANJE MODELA 15](#_Toc135151989)

[ZAKLJUČAK 16](#_Toc135151990)

# UVOD

U vremenu kada tehnologija postaje sve naprednija, algoritmi dubokog učenja kao što su neuronske mreže otvaraju nove mogućnosti u različitim područjima. Jedno takvo područje je digitalna obrada slike, u ovom konkretnom slučaju radi se o prelasku s crno-bijelih na slike u boji - proces poznat kao "kolorizacija". Ovaj rad bavi se istraživanjem korištenja neuronskih mreža, konkretno pix2pix modela, za “kolorizaciju“ slika u sivim tonovima. Cilj rada je formirati model koji se trenira na određenom skupu podataka te je nakon samog treniranja u mogućnosti “kolorizirat“ slike sivih tonova. Evo nekoliko problema koje ovaj rad nastoji riješiti. Tradicionalne metode “kolorizacije“ često zahtijevaju ručni rad, što može biti vremenski zahtjevno i sklono pogreškama. Korištenjem pix2pix modela ovaj se proces može automatizirati, čineći ga tako bržim i učinkovitijim. Jedan od glavnih problema u “kolorizaciji“ slika je postizanje visoke kvalitete i realističnih rezultata. Pix2pix model koristi uvjetovane generativne suparničke mreže (cGAN) koje omogućuju stvaranje detaljnih i preciznih slika u boji. Mnogi modeli dubokog učenja mogu imati problema s generalizacijom, odnosno s primjenom naučenog na nove, neviđene slike. Stoga ovaj rad istražuje kako pix2pix model može biti treniran da bolje generalizira i pruža konzistentne rezultate na različitim vrstama slika. Treniranje i testiranje modela vrši se nad skupom podataka koji se sastoji od parova slika različitih modela građevina odnosno točnije rečeno zgrada. Parovi se sastoje od jedne slike u sivim tonovima i njezine izvorne slike u pravim bojama.

## Biblioteke

Kako bi izvedba samog programa bila moguća potrebne su biblioteke koje omogućavaju izvedbe pojedinih operacija potrebnih za rad sa slikama i njihovom obradom. Neke od biblioteka korištenih u ovom radu su: TensorFlow (otvoreni softver za duboko učenje razvijen od strane Google Brain ekipe. Omogućava izgradnju i trening složenih modela neuronskih mreža. Koristi se za brojne primjene, uključujući prepoznavanje govora, prepoznavanje slika, preporuke i prevođenje jezika), Matplotlib (biblioteka za crtanje grafika u Pythonu, omogućava stvaranje različitih vrsta vizualizacija, uključujući linije, trake, točke, histograme, konture itd., također se često koristi za vizualizaciju slika prilikom rada s modelima dubokog učenja), cv2 (modul unutar OpenCV (Open Source Computer Vision Library), biblioteke otvorenog koda za računalni vid i strojno učenje, omogućava čitanje, pisanje, prikazivanje slika i provođenje različitih operacija na slikama), PIL ((Python Imaging Library) je biblioteka za obradu slika u Pythonu, omogućava otvaranje, manipuliranje i spremanje mnogih različitih formata slika).

## Neuronske mreže

Prije same razrade konkretnog zadatka potrebno je objasniti pojmove koji se često pojavljuju tokom prolaska kroz pojedine segmente koda odnosno rada.

Neuronske mreže su modeli dubokog učenja inspirirani biološkim neuronskim mrežama koje se nalaze u mozgu. Sastoje se od slojeva umjetnih neurona, ili "čvorova", koji su međusobno povezani. Svaki čvor obrađuje informacije koje primi i šalje rezultat na sljedeći sloj. Treniranjem neuronske mreže moguće je prilagoditi težine veza između čvorova kako bi se optimizirala točnost modela na nekom zadatku, kao što je prepoznavanje slika ili prijevod teksta.

Konvolucijske neuronske mreže (CNN) su specijalizirana vrsta neuronskih mreža dizajnirana za obradu podataka s višestrukim dimenzijama, kao što su slike. Ključni element CNN-a je operacija konvolucije, koja efektivno skenira ulazne podatke malim "prozorima", identificirajući lokalne značajke koje se zatim mogu kombinirati na višim razinama da bi se prepoznale kompleksnije značajke. Ovo ih čini posebno učinkovitima u obradi slika, gdje lokalne značajke mogu biti rubovi, teksture, boje i sl.

Uvjetovane generativne suparničke mreže (cGAN) su također vrsta neuronske mreže, ali s posebnom strukturom dizajniranom za generiranje novih podataka koji su slični nekom postojećem skupu podataka. cGAN mreže sastoje se od dvije komponente: generatora, koji stvara nove podatke, i diskriminatora, koji pokušava razlikovati stvarne podatke od onih koje je generirao generator. U cGAN mrežama, oba modela se dodatno "uvjetuju" s nekim dodatnim informacijama, što im omogućava da generiraju podatke koji ne samo da su slični originalnom skupu podataka, već su i specifični za neki uvjet.

## Pix2Pix

Pix2pix je model dubokog učenja koji koristi koncept uvjetovanih generativnih suparničkih mreža (cGAN) za transformaciju jednog seta slika u drugi. Ovaj model razvijen je od strane istraživača na Sveučilištu u Berkeleyu i koristi se u nizu različitih aplikacija, uključujući “kolorizaciju“ slika, prevođenje slike u slike, sintezu slika i slično.

Pix2pix model sastoji se od dvije glavne komponente: generatora i diskriminatora.

Generator je konvolucijska neuronska mreža koja ima zadatak transformirati ulaznu sliku u izlaznu sliku. U kontekstu colorizacije slika, ulazna slika je slika u sivim tonovima, a izlazna slika je slika u boji. Generator koristi arhitekturu koja je poznata kao "encoder-decoder". Encoder smanjuje dimenzionalnost ulazne slike, dok decoder povećava dimenzionalnost izlazne slike.

Diskriminator je također konvolucijska neuronska mreža, ali njegova uloga je da klasificira slike kao stvarne (izvorne slike u boji) ili lažne (slike u boji koje je generirao generator). Diskriminator je treniran da maksimizira točnost svoje klasifikacije, dok je generator treniran da maksimizira pogrešku diskriminatora.

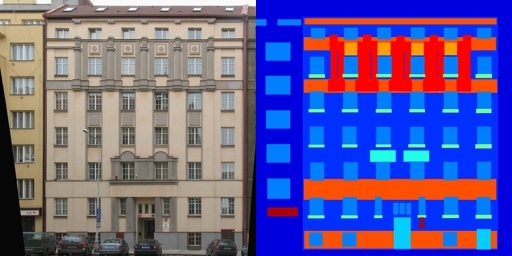
Pix2pix model trenira se kroz proces koji se sastoji od dvije faze. Prva faza je treniranje diskriminatora, gdje se on uči razlikovati između stvarnih i lažnih slika. Druga faza je treniranje generatora, gdje se on uči stvarati slike koje diskriminator ne može prepoznati kao lažne.

Jedan od ključnih aspekata pix2pix modela je uvjetovana GAN struktura, što znači da se proces generiranja slika ne događa nasumično, već je vođen ulaznom slikom. Ovo omogućava modelu da stvara precizne i detaljne izlazne slike koje odgovaraju ulaznim slikama.

Iako pix2pix model omogućuje impresivne rezultate, on također ima svojih izazova. Primjerice, model može imati poteškoća s generalizacijom i može stvarati neželjene artefakte na slikama. Unatoč ovim izazovima, pix2pix je jedan od najučinkovitijih alata za transformaciju slika dostupan danas.

# PODATCI I NJIHOVA OBRADA

Treniranje i testiranje programa vrši se nad skupom podataka koji se u ovom slučaju sastoji od slika zgrada odnosno njihovih prednjih strana. Postoje dvije skupine podataka u datom skupu, jedna skupina predstavlja podatke namijenjene za treniranje dok druga skupina sadrži podatke nad kojima se vrši testiranje istreniranog modela. Prva skupina sastoji se od 400 parova slika od kojih je jedna u svojim stvarnim bojama dok je druga u paleti boja kojima su naznačene značajke na slici(slika 1.). Druga skupina sadrži 106 parova slika istih obilježja kao slike prve skupine.



Slika 1.

Prvi korak u obradi podataka je učitavanje i nasumično odsijecanje svake od slika koje se koriste u treniranju. Primjena nasumičnih odsijecanja za svrhu augmentacije podataka korisna je iz dva razloga:

Povećanje raznolikosti podataka: Nasumična odsijecanja omogućuju generiranje novih primjera podataka iz postojećih slika. Time se povećava raznolikost skupa podataka, što može pomoći u poboljšanju performansi modela strojnog učenja.

Smanjenje prenaučenosti: Primjenom nasumičnih odsijecanja, model će vidjeti različite dijelove slika tijekom treninga, što mu pomaže da nauči općenitije značajke umjesto specifičnih detalja.

Uz to potrebno je sve slike postaviti na iste vrijednosti visine i širine kako bi imali skup jednakih slika, u ovom slučaju te vrijednosti iznose 256px. Nakon toga se provodi normalizacija slika, tj. postavljanje vrijednosti svakog piksela u raspon od -1 do 1 radi olakšanog treniranja neuronskih mreža (Slika 2).

Slika na kojoj se prikazuje tekst, Font, broj, crta

Opis je automatski generiran

Slika 2.

Nakon toga slike je potrebno prebaciti u paletu sivih tonova kako bi dobili željene ulazne slike. Za to se koristi funkcija tf.image.rgb\_to\_grayscale(re). Ova transformacija pretvara trokanalnu RGB sliku u jednokanalnu sivu sliku.

Kada su podatci obrađeni slijedi raspored i priprema istih za treniranje modela. Prvo se stvara putanja do datoteke s podatcima za treniranje. Zatim se podatci miješaju što utječe na raznolikost primjera pri svakoj epohi treninga. Uz to važno je i postaviti putanju do slika koje se koriste za testiranje modela.

# U-NET

U-Net je arhitektura neuronske mreže koja se često koristi u zadacima segmentacije slika, posebno u biomedicinskom slikanju. Ime "U-Net" dolazi od oblika arhitekture koji ima oblik slova "U".

U-Net ima karakteristične značajke koje ga čine popularnim u segmentaciji slika:

**Encoder** (smanjenje rezolucije): U početnom dijelu U-Neta, ulazna slika prolazi kroz konvolucijske slojeve kako bi se smanjila rezolucija i naučile niske i apstraktne značajke. Ovi slojevi se često nazivaju "downsampling" ili "encoder" slojevi. Svaki sloj smanjuje rezoluciju slike i povećava broj značajki kako bi model mogao naučiti reprezentaciju slike na različitim apstrakcijskim razinama. U ovom slučaju encoder se sastoji od sljedećih operacija: Conv(konvolucijski sloj za izdvajanje različitih karakteristika), BatchNorm (stabilizacija i ubrzani proces učenja), LeakyRelu(zadržava sve pozitivne vrednosti nepromenjene).

**Decoder** (povećanje rezolucije): U drugom dijelu U-Neta, izlaz iz "encoder" dijela se koristi za rekonstrukciju originalne rezolucije slike. Ovi slojevi se često nazivaju "upsampling" ili "decoder" slojevi. Svaki sloj povećava rezoluciju slike i smanjuje broj značajki kako bi se rekonstruirala originalna slika. U ovom slučaju decoder se sastoji od sljedećih operacija: TConv(koristi se za povećanje dimenzija izlaza), BatchNorm, Dropout (isključuje nasumične jedinice kako bi se smanjio “overfitting“) i ReLU (koristi za stvaranje neprekidne i nelinearne reprezentacije slike).

**Skip konekcije:** U-Net koristi “skip“ konekcije koje omogućuju prijenos informacija između slojeva smanjenja i povećanja rezolucije. Ove skip konekcije služe za prenošenje detaljnih i lokalnih informacija iz nižih slojeva u više slojeve, što pomaže u preciznijoj segmentaciji slika.

**Finalni slojevi:** Na kraju arhitekture U-Neta nalaze se finalni slojevi koji generiraju izlaznu sliku ili mapu segmentacije. Obično se koriste konvolucijski slojevi s odgovarajućim aktivacijskim funkcijama za generiranje željenog izlaza.

U-Net ima sposobnost naučiti detaljne značajke slike na različitim razinama rezolucije.

Prikaz arhitekture U-Neta(slika 3.).

Prikaz funkcija “downsample“ i “upsample“(slika 4.).

Slika na kojoj se prikazuje tekst, dijagram, račun, dokument

Opis je automatski generiran

Slika 3.

Slika na kojoj se prikazuje tekst, Font, softver, web-stranica

Opis je automatski generiran

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, softver

Opis je automatski generiran

Slika 4.

## Generator

Generator je jedna od komponenti generativnih modela, poput generativnih suparničkih mreža (GAN). Odgovoran je za generiranje novih uzoraka koji sliče na uzorke iz originalnog skupa podataka.

Trenira se zajedno s diskriminatorom kako bi se postigla željena generacija slika.

Cilj generatora je stvoriti uvjerljive i visokokvalitetne slike koje se približavaju stvarnim primjerima iz skupa podataka. Trenira se iterativno, poboljšavajući svoju sposobnost generiranja realističnih slika kako treniranje napreduje.

Arhitektura generatora korištenog u ovom radu bazirana je na U-Net(*U-Mreža*) strukturi. On prima sliku niske rezolucije kao ulaz i generira sliku visoke rezolucije. Generator koristi kombinaciju slojeva za smanjenje i povećanje rezolucije, kao i skip konekcije za prijenos informacija između slojeva. Prikaz smanjivanja i povećavanja rezolucije slika unutar generatora (Slika 5.).

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Font

Opis je automatski generiran

Slika 5.

## Diskriminator

Diskriminator je model u strojnom učenju koji se koristi za klasifikaciju između stvarnih i lažnih primjera. U kontekstu generativnih neuronskih mreža (GAN), diskriminator je jedan od dva ključna dijela.

U arhitekturi GAN-a, diskriminator ima zadatak razlikovati stvarne primjere od generiranih primjera koje proizvodi generator. Cilj diskriminatora je naučiti razlikovati između tih dviju klasa. Uobičajeno je da diskriminator bude binarni klasifikator koji izlaznim vrijednostima pokušava predvidjeti je li primjer stvaran ili lažan.

Diskriminator se trenira zajedno s generatorom u GAN-u. Generator generira primjere, a diskriminator ocjenjuje koliko su ti primjeri stvarni. Na temelju povratnih informacija od diskriminatora, generator se prilagođava kako bi poboljšao svoje generiranje i pokušao prevariti diskriminatora da prihvati generirane primjere kao stvarne.

Cilj GAN-a je postići ravnotežu između generatora i diskriminatora, gdje generator generira primjere koji su dovoljno uvjerljivi da diskriminator ne može pouzdano razlikovati stvarne od lažnih primjera. Na taj način, generator stvara realistične primjere koji odgovaraju distribuciji stvarnih primjera.

Arhitektura diskriminatora korištenog u ovom radu temelji se na “PatchGAN-u“. Pomoću njega diskriminator klasificira manje dijelove slike umjesto cijele slike. Time se omogućava bolje hvatanje lokalnih informacija i detalja slike. Osnovna jedinica samog diskriminatora sastoji se od konvolucijskog sloja, sloja za normalizaciju te aktivacijske funkcije ReLu.

Zadnji sloj Diskriminatora ima izlazni oblik bs, 30, 30, 1, gdje bs predstavlja veličinu serije (broj primjera u jednom prolazu kroz Diskriminator). Ovaj sloj se koristi za klasifikaciju kvadratnih područja dimenzija 70x70 piksela ulazne slike.

Diskriminator prima dva ulaza. Prvi ulaz je stvarna slika koja se koristi kao referentna vrijednost prilikom klasifikacije. Drugi ulaz je generirana slika koju je proizveo Generator i trebala bi biti klasificirana kao lažna.

Slika 6 prikazuje strukturu diskriminatora dok slika 7 prikazuje njegov način rada.

Slika na kojoj se prikazuje tekst, Font, snimka zaslona

Opis je automatski generiran

Slika 6.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, paralelno

Opis je automatski generiran

Slika 7.

Optimizacija

Za potrebe optimizacije rada modela korišten je optimizator po imenu Adam. Adam (Adaptive Moment Estimation) je popularan optimizacijski algoritam. On adaptivno prilagođava stopu učenja za svaki parametar na temelju trenutnog gradijenta i prethodnih gradijenata, što omogućuje efikasniju i stabilniju optimizaciju.

Prilikom treniranja GAN modela, generator i diskriminator se izmjenično treniraju. Generator generira primjere, a diskriminator ocjenjuje njihovu autentičnost. Na temelju povratnih informacija od diskriminatora, generator se ažurira kako bi poboljšao svoje generiranje i pokušao prevariti diskriminatora.

Optimizatori igraju ključnu ulogu u postizanju konvergencije GAN-a i postizanju ravnoteže između generatora i diskriminatora. Pravilno postavljanje stopa učenja i ostalih parametara može biti ključno za postizanje stabilnog i uspješnog treniranja GAN modela.

# TRENIRANJE MODELA

Model se trenira kroz određen broj epoha koji je postavljen na 150. U svakoj epohi, prolazi se kroz skup podataka za treniranje i izvršava se korak treninga za svaki primjer u skupu podataka.

Koristi se Adam optimizator za optimizaciju generatora i diskriminatora. Gradijenti se izračunavaju pomoću gradijentnih traka, a primjenjuju se na odgovarajuće trenirajuće varijable. Gubitci generatora i diskriminatora se zapisuju u zapisnike pomoću funkcije tf.summary.scalar.

Tijekom treninga, generirane slike se prikazuju pomoću funkcije generate\_images. Također se koriste kontrolne točke (checkpoints) za spremanje stanja modela svakih 20 epoha i na kraju treninga.

Cilj ovog treninga je naučiti generativnu neuronsku mrežu da generira realistične slike na temelju ulaznih slika. Kombinacijom generatora i diskriminatora u GAN arhitekturi, model se uči generirati slike koje su dovoljno uvjerljive da diskriminator ne može pouzdano razlikovati stvarne slike od generiranih slika.

Jedan od rezultata nakon treniranja (Slika 8).

Slika na kojoj se prikazuje tekst, snimka zaslona, aparat za slatkiše, gazirano piće

Opis je automatski generiran

Slika 8.

# ZAKLJUČAK

U ovom radu prikazana je implementacija generativne neuronske mreže (GAN) za generiranje slika. GAN arhitektura sastoji se od generatora i diskriminatora koji se međusobno natječu tijekom treninga. Cilj je naučiti generator da generira realistične slike koje diskriminator ne može pouzdano razlikovati od stvarnih slika.

Model je treniran kroz više epoha, a tijekom svake epohe izvršavaju se koraci treninga. Optimizatori se koriste za prilagođavanje težina generatora i diskriminatora, s ciljem minimiziranja gubitka i poboljšanja kvalitete generiranih slika.

Implementirani su i pomoćni dijelovi koda koji omogućuju praćenje napretka treninga, generiranje slika za evaluaciju i spremanje kontrolnih točaka modela. To pruža dodatne mogućnosti za analizu i ponovno pokretanje treninga u budućnosti.

Prema dobivenom rezultatu jasno je vidljivo da je model u mogućnosti “kolorizirati“ slike s vrlo velikom točnošću. Kada se usporede stvarna i predviđena slika moguće je primijetiti da se boje fasada vrlo malo razlikuju a to se smatra zadovoljavajućim rezultatom. Uz to moguće je vidjeti da se na predviđenoj slici vidi područje koju model nije uspio dobro prikazati ali je ono zanemarive veličine.

Kroz eksperimente s ovim GAN modelom, moguće je generirati slike koje su vizualno slične stvarnim primjerima iz skupa podataka za trening. Kvaliteta generiranih slika može se dalje poboljšati povećanjem broja epoha.