Uklanjanje objekata sa slika pomoću dubokog učenja

Varšava, Marko

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

UKLANJANJE OBJEKATA SA SLIKA POMOĆU DUBOKOG UČENJA

Diplomski rad

Marko Varšava

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 20.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Marko Varšava
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1101R, 06.10.2019.
OIB studenta:	89013915927
Mentor:	izv. prof. dr. sc. Časlav Livada
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	prof. dr. sc. Krešimir Nenadić
Član Povjerenstva 1:	izv. prof. dr. sc. Časlav Livada
Član Povjerenstva 2:	Robert Šojo, mag. ing. comp.
Naslov diplomskog rada:	Uklanjanje objekata sa slika pomoću dubokog učenja
Znanstvena grana diplomskog rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak diplomskog rada:	U radu je potrebno objasniti temeljne principe rada dubokog učenja te primjenu istog na uklanjanje neželjenih objekata u slici. Tema rezervirana za: Marko Varšava
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	20.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 14.10.2023.

Ime i prezime studenta:	Marko Varšava
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1101R, 06.10.2019.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Uklanjanje objekata sa slika pomoću dubokog učenja**

izrađen pod vodstvom mentora izv. prof. dr. sc. Časlav Livada

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Fotografija je postala neizostavan način za dijeljenje i dokumentiranje važnih trenutaka. Međutim, ponekad ti trenuci mogu biti narušeni prisustvom neželjenih objekata koji remete njihovu estetiku ili poruku. Tehnologije za obradu slike pomogle su nam transformirati vizualnu stvarnost na jednostavan i djelotvoran način. Uklanjanje objekata s fotografija postalo je moćan alat koji se koristi u raznim industrijama, uključujući dizajn, fotografiju te vizualne efekte u filmskoj industriji, kako bi se postigao željeni rezultat. Na primjer, u situacijama kada je potrebno ukloniti neželjeni objekt iz scene ili promijeniti izgled određenog dijela okoline. Uklanjanje objekata iz slikovnih podataka koristi se u geoinformacijskim sustavima i kartografiji kako bi se uklonile smetnje i neželjeni elementi ili prilagodile karte specifičnim potrebama. Na primjer, uklanjanje vozila, ljudi ili drugih objekata sa satelitskih snimki može pomoći u stvaranju preciznijih karti.

Inpainting je proces u kojem se oštećeni, narušeni ili dijelovi koji nedostaju na slici popunjavaju kako bi se prikazala potpuna slika. Rane metode inpaintinga fokusirale su se na jednostavne zamjene piksela temeljene na susjedstvu. Ove metode su se počele razvijati u 1970.-ima, a jedna od najranijih tehnika bila je takozvani kopiraj-zalijepi (engl. *copy-paste*) pristup, gdje su nedostajući dijelovi slike popunjavani kopiranjem dijelova iz drugih područja slike i prenosili na dijelove koji nedostaju.

Postoji nekoliko različitih pristupa ovom problemu, na primjer algoritmi temeljeni na teksturi. Ova kategorija algoritama koristi informacije o teksturi slike kako bi obnovila oštećena područja. Metode analiziraju teksture na slici kako bi identificirale slične teksture koje se mogu koristiti za popunjavanje praznina. Ove metode često daju bolje rezultate za popunjavanje većih praznina na slikama koje obuhvaćaju velike površine s mnogo manjih detalja.

Algoritmi temeljeni na prijenosu stila koriste tehnike za prijenos stila kako bi popunili praznine na slici. Prijenos stila uključuje preuzimanje stilskih karakteristika iz jedne slike i primjenu tih karakteristika na drugu sliku. U inpaintingu, stilski podaci se prenose iz okolnih dijelova slike na oštećena područja kako bi se obnovile. Ovi algoritmi mogu proizvesti vrlo realistične rezultate, ali su često računski zahtjevni.

U kasnim 1990.-ima i ranim 2000.-ima, počele su se razvijati metode koje su koristile primjere iz same slike ili baze podataka slika kako bi se obnovili dijelovi slike koji nedostaju. Ove metode temeljile su se na pretpostavci da se slični dijelovi slike mogu koristiti za popunjavanje praznina.

Primjeri ovakvih metoda inpaintinga uključuju sintezu tekstura pomoću zakrpa (engl. *patch-based texture synthesis*) [13].

S razvojem strojnog učenja, inpainting je dobio snažan poticaj. Ovi algoritmi uče statistike i obrasce iz velikog skupa podataka slika. U posljednjem desetljeću počele su se primjenjivati metode temeljene na generativnim modelima, posebno konvolucijskim neuronskim mrežama (CNN). Generativne protivničke mreže (egnl. *Generative Adversarial Networks* - GAN) [2] i varijacije autoenkodera (VAE) postale su popularne tehnike za inpainting slike. Ove metode omogućuju modelima naučiti kompleksne obrasce i obnove visokofrekventne detalje.

Prvi inpainting model čiji je algoritam baziran na generativnim protivničkim mrežama je Kontekstni koder (engl. *Context Encoder*) objavljen 2016. godine [1]. Context Encoder donosi korisne osnovne koncepte za zadatak popunjavanja slika. Pojam „konteksta“ odnosi se na razumijevanje same cjelokupne slike i osnovna ideja Context Encoder je sloj potpuno povezanih kanala. Glavna svrha je da svi prostorni položaji značajki u prethodnom sloju doprinose svakom prostornom položaju značajke u trenutnom sloju. Na taj način mreža može naučiti odnos između svih prostornih položaja značajki i dobiti dublje semantičko razumijevanje cijele slike.

U posljednjih nekoliko godina, razvijene su brojne napredne tehnike inpaintinga. Primjeri uključuju primjenu konvolucijskih neuronskih mreža s dodatnim modulima za vodstvo konteksta, primjenu generativnih modela koji uključuju prostornu pažnju (engl. *Spatial Attention*) i rekurzivne mehanizme, kao i kombinaciju inpaintinga s drugim zadacima poput segmentacije objekata.

Modeli se suočavaju s raznim izazovima. Duboko učenje može imati poteškoća u razumijevanju šireg konteksta slike. Ako dio slike koji nedostaje ima složenu strukturu koju je potrebno razumjeti kako bi se dobro popunio, modeli mogu imati poteškoće u generiranju prirodnih i uvjerljivih rezultata.

Važno je da popunjeni dio slike bude konzistentan s ostatkom slike, kako u smislu boje, teksture, oblika i drugih karakteristika. Ponekad modeli mogu stvoriti dijelove koji su vizualno primjetni i neusklađeni s okolnim dijelovima slike. Također jedan od problema kod generiranja dijelova slike događa se ako je dio slike koji se treba generirati velik i obuhvaća važne detalje, modeli mogu imati poteškoća u stvaranju uvjerljivog popunjavanja, često dolazi do zamućivanja tog dijela slike.

Za popunjavanje slika dubokim učenjem potrebni su veliki skupovi. Takvi skupovi podataka mogu biti ograničeni, teško dostupni ili vremenski zahtjevni za prikupiti, što može utjecati na

performanse modela. Vremenski zahtjevi za treniranje modela mogu varirati ovisno o složenosti modela i veličini slika, što može rezultirati dugotrajnim procesom treniranja.

U sklopu diplomskog rada, model je treniran za uklanjanje ljudi s fotografija. Kako bi se postigla precizna i jednostavna selekcija objekata, primijenjena je selekcija klikom na objekt. Nakon izrade modela, provedena je evaluacija kako bi se procijenila njegova učinkovitost, kroz testiranje u različitim scenarijima.

U drugom poglavlju objašnjeni su osnovni pojmovi vezani za strojno učenje koji se koriste u radu i napravljen je kratki pregled nekoliko postojećih rješenja za uklanjanje objekata sa slika. U trećem poglavlju detaljnije je opisan model korišten u ovom radu za generiranje maske pomoću semantičke segmentacije i detaljnije opisuje model odabran za popunjavanje slike. U četvrtom poglavlju opisana je izrada i pripremanje podatkovnog skupa za treniranje. U petom poglavlju opisana je evaluacija rješenja. Na kraju je dan zaključak.

2. POSTOJEĆA RJEŠENJA

U ovom poglavlju diplomskog rada predstavljeni su osnovni pojmovi iz područja strojnog učenja. Zatim su predstavljene različite metode koje se koriste za popunjavanje praznina na slikama.

2.1. Osnovni pojmovi

Strojno učenje predstavlja područje umjetne inteligencije koje se usredotočuje na razvoj algoritama koji svoju djelotvornost unapređuju na temelju empirijskih podataka [9]. Dijele se u tri glavne vrste: nadzirano, nenadzirano i podržano učenje (engl. *supervised*, *unsupervised* i *reinforcement learning*). U ovom istraživanju koristi se nadzirano učenje. U nadziranom učenju, kod treninga modela, ulazni podaci se kombiniraju s odgovarajućim izlaznim podacima. U ovom radu, ti ulazni podaci su slike s primijenjenom maskom, dok su izlazni podaci originalne slike.

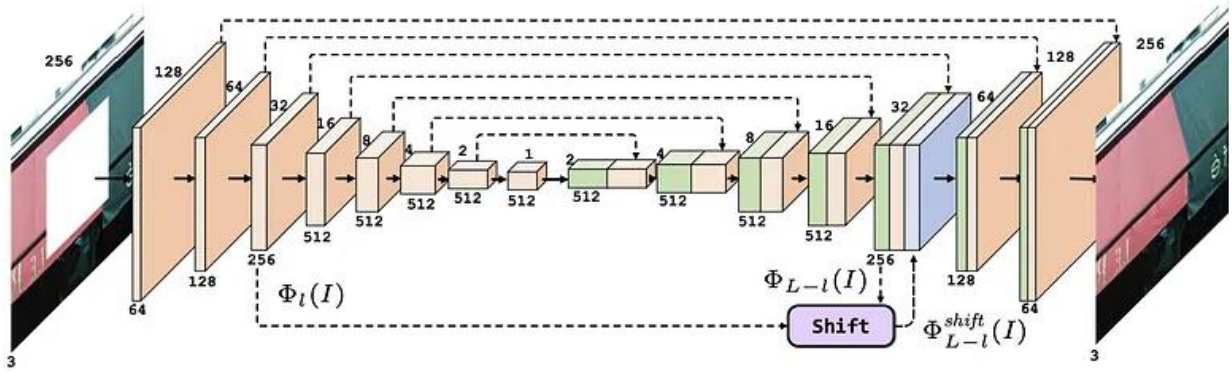
Neuronska mreža je temeljni algoritam koji se koriste u strojnom učenju. Neuronska mreža je struktura koja se sastoji od slojeva neurona, gdje svaki sloj obrađuje informacije od ulaza prema izlazu (engl. *feedforward neural network*). Potpuno povezana neuronska mreža (engl. *fully connected network*) je vrsta mreže u kojoj je svaki neuron u jednom sloju povezan sa svakim neuronu u prethodnom sloju. Duboko učenje se temelji na korištenju velikog broja slojeva kako bi postupno izvuklo značajke iz sirovih podataka. Konvolucijska neuronska mreža (engl. *Convolutional Neural Network* – CNN) je jedna od najčešće primjenjivanih arhitektura u dubokom učenju. Sastoji se od potpuno povezanih slojeva, konvolucijskih slojeva i slojeva sažimanja. Primjerice, konvolucijska neuronska mreža se često koristi u obradi slika za detekciju rubova. Kombiniranjem više CNN-ova mogu se postići složeniji koncepti, kao što je prepoznavanje lica.

2.2. Pregled postojećih metoda

2.2.1. Shift-Net

Shift-Net metoda objavljena 2018. godine, tehnika je koja kombinira prednosti korištenja modernih podatkovno vođenih CNN-ova i konvencionalne metode kopiranja i lijepljenja (engl. *copy-paste*) za popunjavanje slika [5]. Konvencionalni način popunjavanja dijelova slike koji nedostaju svodi se na traženje najsličnijih dijelova, a zatim izravno kopiranje i lijepljenje tih dijelova na dijelove koji nedostaju, metoda kopiranja i lijepljenja. Ova metoda stvara dobre detalje, međutim, ti dijelovi možda ne odgovaraju savršeno kontekstu cijele slike, a moguća su i ponavljanja uzorka, što može dovesti do lošeg rješenja u cijelini. Shift-Net koristi *shift-connection*

sloj kako bi se koncept kopiranja i lijepljenja ugradio u moderne konvolucijske neuronske mreže. Time model može pružiti rezultate popunjavanja praznina s ispravnom globalnom semantičkom strukturom i finim detaljnim teksturama. Na slici 2.1. prikazana je mrežna arhitektura Shift-Net mreže, *shift-connection* sloj dodan je na rezoluciji 32x32. Za treniranje njihove Shift-Net mreže korišten je gubitak navođenja (engl. *guidance loss*). Jednostavnije rečeno, ovaj gubitak izračunava razliku između dekodirane značajke ulazne slike na dijelu koji je nedostajao i kodirane značajke stvarne slike (engl. *ground truth*) unutar regije koja nedostaje. Model koristi značajke izvan regije koja nedostaje kako bi dodatno poboljšao mutnu procjenu unutar regije koja nedostaje. Za svaku dekodiranu značajku unutar područja koji nedostaje, nakon pronalaska najsličnije kodirane značajke izvan područja koji nedostaje, formira još jedan skup značajki temeljen na vektoru pomaka.



Sl. 2.1. Mrežna arhitektura Shift-Neta. *Shift-connection* sloj dodan je na rezoluciji 32×32 [5]

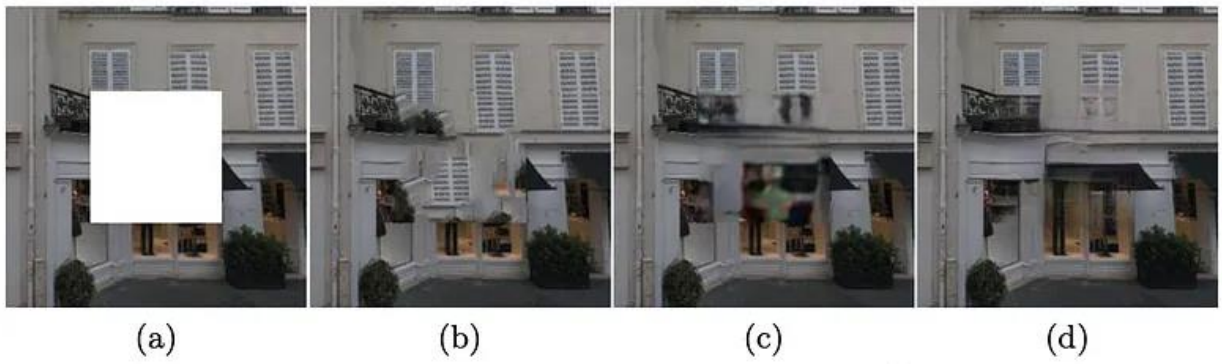
Funkcija gubitka (engl. *loss function*) koja se koristi osim *guidance loss-a* kojeg uvodi, također koristi L1 gubitak i standardni protivnički gubitak (engl. *adversarial loss*). Ukupna funkcija gubitka je sljedeća:

$$L = L_{l1} + \lambda_g L_g + \lambda_{adv} L_{adv} \quad (2\text{-}1)$$

Unutar Tablica 2.1. Usporedba PSNR (*Peak Signal-to-Noise Ratio*), SSIM (*Structural Similarity Index*) i srednje kvadratne pogreške na skupu podataka Paris StreetView. [5] nalaze se rezultati metrika PSNR, SSIM te srednje kvadratne pogreške za opisanu metodu Shift-Net, Content-Aware Fill metode i prve metode bazirane na GAN mreži opisane u uvodu, Context encoder. Shift-Net metoda dobila je najbolji rezultat za obje metrike. Na slici 2.2. prikazana je usporedba rezultata

Tablica 2.1. Usporedba PSNR (*Peak Signal-to-Noise Ratio*), SSIM (*Structural Similarity Index*) i srednje kvadratne pogreške na skupu podataka Paris StreetView. [5]

Metoda	*PSNR*	*SSIM*	*Mean l_2 loss*
Content-Aware Fill [4]	23.71	0.74	0.0617
Context encoder [1] (l_2 + adversarial loss)	24.16	0.87	0.0313
Shift-Net	26.51	0.90	0.0208

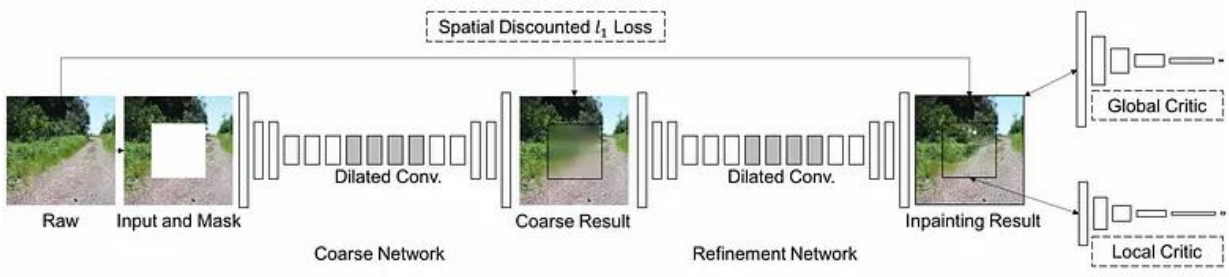


Sl. 2.2. Kvalitativna usporedba rezultata popunjavanja slika različitim metodama. (a) Ulazna slika (b) Konvencionalna metoda (temeljena na kopiranju i lijepljenju) (c) Prva metoda bazirana na GAN-u, Context Encoder (d) Shift-Net. [5]

dobivenih Shift-Net mrežom s rezultatima metode temeljene na kopiranju i lijepljenju i Context encoder mreže.

2.2.2. DeepFill

DeepFill v1 model objavljen je 2018. godine [14]. Arhitektura modela sastoji se od dvije generatorske mreže i dvije diskriminatorske mreže. Dva generatora slijede potpune konvolucijske mreže (engl. *fully convolutional networks*) s proširenim konvolucijama (engl. *dilated convolutions*). Jedan generator generira grubu rekonstrukciju, a drugi služi za doradu iste. Na slici 2.3. prikazana je arhitektura DeepFill v1 mreže. Takva mreža poznata je kao standardna struktura mreže od grube do fine dorade (engl. *standard coarse-to-fine network*). Dva diskriminatora također gledaju dovršene slike globalno i lokalno; globalni diskriminator uzima cijelu sliku kao ulaz, dok lokalni diskriminator uzima ispunjeno područje kao ulaz.



Sl. 2.3. Arhitektira modela DeepFill v1 za popunjavanje slika. [14]

Za funkciju gubitka koristi se protivnički gubitak (engl. *adversarial loss*) odnosno GAN gubitak i L1 gubitak, za pikselski gledanu točnost rekonstrukcije. Za gubitak L1 koriste se prostorno smanjeni gubitak L1 (engl. *spatially discounted L1 loss*) u kojem se težina dodjeljuje svakoj razlici u pikselu, a težina se temelji na udaljenosti piksela od najbližeg poznatog piksela. Za GAN gubitak koristi se WGAN-GP gubitak. WGAN protivnički gubitak temelji se na mjerenju L1 udaljenosti, stoga je mrežu lakše trenirati i proces obuke je stabilniji.

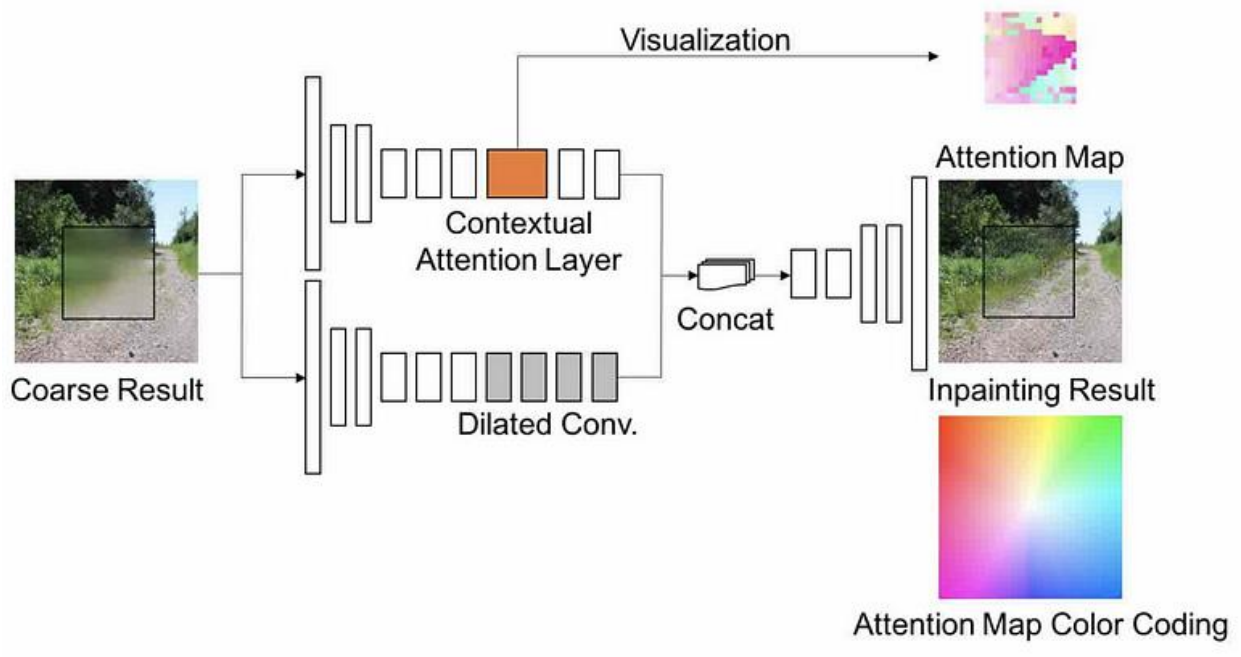
Mehanizam kontekstualne pažnje (engl. *Contextual Attention mechanism*) koristi se kako bi se učinkovito koristile kontekstualne informacije iz udaljenih prostornih mjesta za rekonstrukciju nedostajućih piksela. *Contextual Attention* primjenjuje se na drugu mrežu za doradu. Prva grublja mreža za rekonstrukciju odgovorna je za grubu procjenu nedostajućih područja. Koriste se globalni i lokalni diskriminatori kako bi se dobili bolji detalji lokalnih tekstura generiranih piksela.

Propagacija pažnje (engl. *attention propagation*) koristi se za fino podešavanje mapa značajki pažnje. Ključna ideja ovdje je da susjedni pikseli obično imaju sličniju vrijednost piksela. To znači da uzimaju u obzir vrijednosti piksela susjedstva kako bi mogli prilagoditi rezultat.

U usporedbi sa Shift-Net-om koji je opisan ranije, dodjeljuje se težina svakom poznatom dijelu značajke kako bismo označili njegovu važnost za rekonstrukciju svakog prostornog položaja značajke unutar područja koji ne dostaje (meko dodjeljivanje), umjesto da samo zadržimo najbliži poznati dio značajke za svaki prostorni položaj značajke unutar nedostajućeg područja (tvrdo dodjeljivanje).

Slika 2.4. prikazuje kako je integriran sloj kontekstualne pažnje u drugu mrežu za doradu. Uvedena još jedna grana kako bi se primijenila kontekstualna pažnja, a zatim se dvije grane spajaju kako bi se dobili konačni rezultati popunjavanja. Na slici se vidi prikaz boja za mapu pažnje koja se koristi za vizualizaciju. Na primjer, srednja bijela boja znači da piksel fokusira na sebe, roza na donje-lijevo područje, zelena na gornje-desno područje i tako dalje. Ovaj primjer ima mapu pažnje

ispunjenu rozom bojom. To znači da popunjeno područje uzima puno informacija iz donje-lijeve regije.



Sl. 2.4. Ilustracija ugradnje sloja kontekstualne pažnje u drugu mrežu za doradu. [14]

Tablica 2.2. Rezultati pogreške srednje vrijednosti l_1, pogreške srednje vrijednosti l_2, PSNR i navodi neke objektivne evaluacijske metrike za referencu na skupu podataka Places2. Ove metrike ne mogu u potpunosti prikazati kvalitetu rezultata popunjavanja, budući da postoji više mogućih rješenja za popunjavanje dijelova slike koji nedostaju. PatchMatch pruža niži TV gubitak (engl. *total variation loss*) jer izravno kopira originalne podatke na mjesta rupa.

Tablica 2.2. Rezultati pogreške srednje vrijednosti l_1, pogreške srednje vrijednosti l_2, PSNR i TV *loss* izmjeren na skupu podataka Places2. [14]

Metoda	*l_1 loss*	*l_2 loss*	*PSNR*	*TV loss*
PatchMatch [5]	16.1%	3.6%	16.62	25.0%
DeepFill v1	8.6%	2.1%	18.91	25.3%

2.2.3. PartialConv

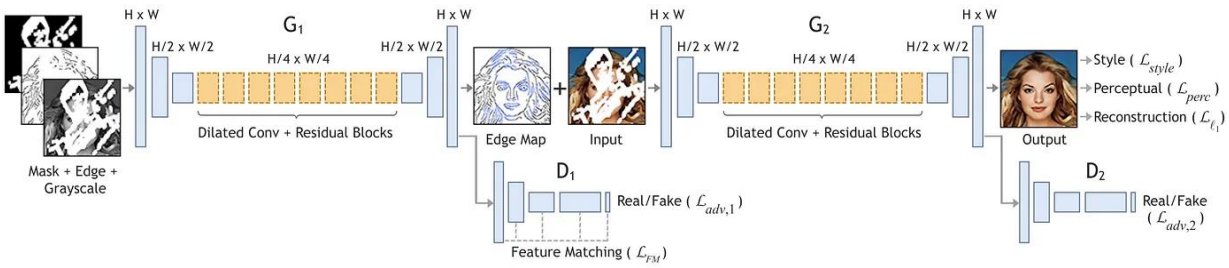
PartialConv model također objavljen 2018. godine, mreža slična U-Net mreži s preskočnim vezama (engl. *skip connections*) u kojoj su svi standardni konvolucijski slojevi zamijenjeni djelomičnim konvolucijskim slojevima (engl. *partial convolutional layers*) [15]. U ovom modelu

se ne koristi diskriminator. Ključna ideja je odvojiti piksele koji nedostaju od valjanih piksela tijekom konvolucija tako da rezultati konvolucija ovise samo o važećim pikselima. To je razlog zašto je predložena konvolucija nazvana parcijalna konvolucija. Konvolucija se djelomično izvodi na ulazu na temelju slike binarne maske koja se može automatski ažurirati.

2.2.4. EdgeConnect

EdgeConnect model je objavljen 2019. godine, a glavna ideja ovog modela je podijeliti zadatak u dva jednostavnija koraka [6]. Prvi korak je predviđanje rubova, to jest linija na dijelovima koji nedostaju, zatim drugi dio mreže za dovršavanje slike na temelju predviđenih linija. Koristi strategiju prvo linije, zatim boja. Postojeći pristupi inpaintinga pomoću dubokog učenja obično proizvode mutna područja. EdgeConnect model koristi mrežu sličnu DeepFill-u, dvostupanjsku mrežu, to jest dva generatora i dva diskriminatora. Generator linija za funkciju gubitka koristi dva termina, protivnički gubitak (engl. *adversarial loss*) i gubitak usklađivanja značajki (engl. *feature matching loss*). Drugi dio mreže za generiranje slike koristi četiri pojma funkcije gubitka, stilski gubitak (engl. *style loss*), perceptivni gubitak (engl. *perceptual loss*), L1 rekonstrukcijski gubitak (engl. *reconstruction loss*) i protivnički *gubitak*.

Na slici 2.5. može se vidjeti da prvi generator G1 uzima sliku maske, maskiranu sliku rubova i maskiranu sliku u sivim tonovima kao ulaz, a za izlaz daje sliku s predviđenim rubovima. Drugi generator G2 uzima sliku s predviđenim rubovima i maskiranu RGB sliku kao ulaz i pomoću njih



Sl. 2.5. Mrežna arhitektura EdgeConnect-a. Ima dva generatora i dva diskriminatora. [6]

generira potpunu RGB sliku.

2.2.5. DeepFill v2

Još jedan model koji je objavljen 2019. godine Deepfill v2 usvaja više ideja za neuronsku mrežu u jedan model, djelomične konvolucijske slojeve predstavljenje u PartialConv modelu, način razdvajanja valjanih i nevaljanih piksela kako bi rezultati konvolucije ovisili samo o valjanim pikselima koje prati generator rubova predstavljen u EdgeConnect modelu. Ta dva pristupa žele spojiti s *Contextual Attention* slojem iz prve verzije DeepFill v1 modela. DeepFill v2 je poboljšana verzija njihovog prethodnog modela DeepFill v1. DeepFill v2 slijedi dvostupanjsku strukturu mreže od grubog do fino modeliranja. Prva mreža generatora odgovorna je za grubu rekonstrukciju, dok je druga mreža generatora odgovorna za doradu grubo popunjene slike. Arhitektura mreže je vrlo slična DeepFill v1, jedina razlika je zamjena standardnih konvolucijskih slojeva sa zaključanim konvolucijama (engl. *gated convolutions*). Jednostavno rečeno, umjesto pravila temeljenog na ažuriranju maske kao u PatiralConv-u, koristi se standardni sloj konvolucije sa sigmoidnom aktivacijskom funkcijom za ažuriranje maske.

Grubi generator uzima maskiranu sliku, masku slike i opcionalnu korisničku skicu kao ulaz za grubu rekonstrukciju nedostajućih područja. Zatim, grubo popunjena slika prosljeđuje se drugoj mreži generatora za doradu. C*ontextual attention* sloj koji je predložen u DeepFill v1 koristi se u mreži za doradu. Kao diskriminator koristi se PatchGAN.

Kao u EdgeConnect modelu primjenjuje se spektralna normalizacija (engl. *Spectral Normalization* - SN) [6] na svaki standardni konvolucijski sloj diskriminatora radi postizanja stabilnosti tijekom treninga. Funkcija gubitka za treniranje modela sastoji se od dva gubitka: gubitka rekonstrukcije piksela (L1 *loss*) i SN-PatchGAN gubitka (formula (2.2.)). Hiperparametri za balansiranje ova dva gubitka su postavljeni u omjeru 1:1. SN-PatchGAN gubitak je negativna srednja vrijednost izlaza SN-PatchGAN diskriminatora. Zapravo, ovo je gubitak na margini (engl. *hinge loss*) koji se često koristi u mnogim GAN okruženjima.

$$L_G = -\mathbb{E}_{z\sim\mathbb{P}_z(z)}\left[D^{sn}\left(G(z)\right)\right] \quad (2\text{-}2)$$

3. ODABIR MODELA

U ovom poglavlju opisana je model za segmentaciju, detaljnije je pojašnjen model korišten za inpainting i podatkovni skup korišten za treniranje.

3.1. Semantička segmentacija

Objekt koji se želi maknuti potrebno je selektirati, to jest generirati masku koja prekriva taj objekt te se ta maska koristi za inpainting. U ovom radu koristi se semantička segmentacija pomoću modela Segment Anything Model od tvrtke Meta AI [8]. Semantička segmentacija je postupak računalnog vida koji ima za cilj pridruživanje semantičke kategorije, kao što su osobe, vozila, drveće i slično svakom pikselu ili regiji na digitalnoj slici. To je oblik računalnog vizualnog prepoznavanja koji pomaže računalima razumjeti sadržaj slike na razini pojedinačnih objekata. Semantička segmentacija koristi algoritme strojnog učenja kako bi naučila razlikovati različite objekte ili regije na slici (slika 3.1.). Razlog odabranog modela je taj što je za treniranje ovog modela korišteno 11 milijuna slika visoke rezolucije s preko 1.1 milijarde segmentacijskih maski,

Sl. 3.1. Primjer semantičke segmentacije pomoću Segment Anything modela. [8]

te je rezultat selektiranja izuzetno precizan. Selektiranje se odvija pomoću sučelja u kojem se otvori slika i klikne na objekt koji se želi maknuti.

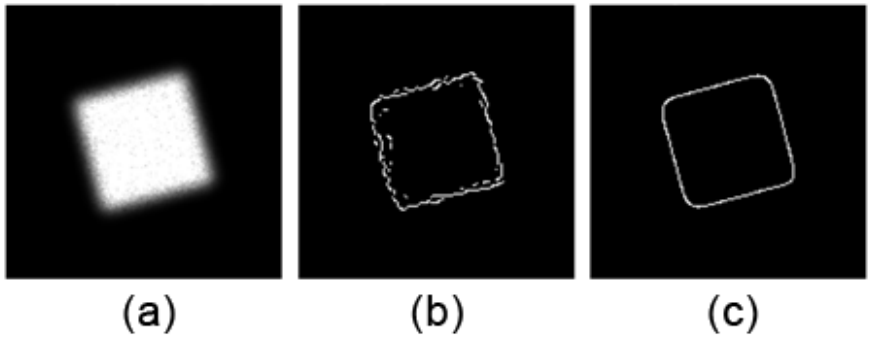
3.2. Odabir modela za inpainting

U ovom radu odabran je dvostupanjski protivnički model EdgeConnect koji se sastoji od generatora rubova, a potom mreže za dovršavanje slike. Generator rubova halucinira rubove područja slike koji nedostaje, dok mreža za dovršavanje slike popunjava područja koja nedostaju koristeći halucinirane rubove kao apriori, to jest uzima rubove predviđene od strane prvog dijela

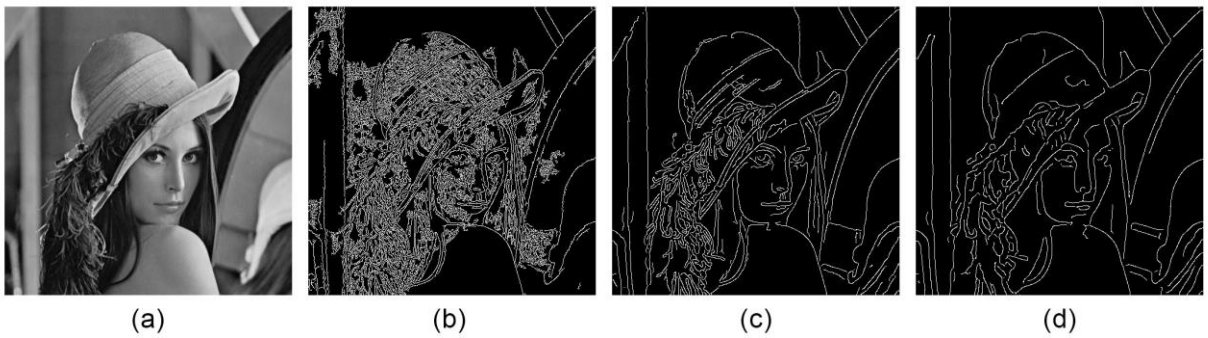
mreže kao osnovu prije nego što popuni područje koje nedostaje. Generator se sastoji od enkodera koji dva puta smanjuje veličinu slike, nakon čega slijedi osam rezidualnih blokova, te dekodera koji povećavaju veličinu slike natrag na originalnu veličinu. Umjesto regularnih konvolucija, u rezidualnim slojevima koriste se proširene konvolucije (engl. *dilated convolutions*). Za diskriminator se koristi arhitektura PatchGAN veličine 70×70 piksela, koja određuje jesu li preklapajući dijelovi slike veličine 70×70 piksela stvarni ili ne. Ova metoda je korištena zato što je dovoljno napredna a s druge strane postoji dovoljno resursa otvorenog koda za razliku od dvije spomenute novije metode.

3.2.1. Detekcija linija

Za treniranje generatora rubova G1, u ovom modelu korišten je poznati konvencionalni algoritam za otkrivanje rubova nazvan Canny detektor rubova [10]. Canny detektor rubova je algoritam za otkrivanje rubova koji koristi više faza algoritma za detekciju različitih vrsta rubova na slikama. Razvio ga je John F. Canny 1986. godine. Canny detekcija rubova koristi linearno filtriranje s Gaussovim kernelom za uklanjanje šuma, a zatim računa jačinu i smjer ruba za svaki piksel u zaglađenoj slici. Rubni pikseli kandidati identificiraju se kao pikseli koji prežive proces stanjivanja. U tom procesu, jačina ruba svakog dobivenog rubnog piksela postavljena je na nulu ako njegova jačina ruba nije veća od jačine ruba dvaju susjednih piksela u smjeru gradijenta. Zatim se određuje prag na slici stanjenog ruba pomoću histereze. U histerezi se koriste dva praga za jačinu ruba, donji i gornji prag. Svi rubni pikseli kandidati ispod donjeg praga ne predstavljaju rub, svi pikseli koji su iznad gornjeg praga su označeni kao rub, te svi pikseli iznad donjeg praga koji mogu biti povezani s bilo kojim pikselom iznad gornjeg praga putem lanca piksela rubova predstavljeni su kao rubni pikseli. Canny detektor rubova zahtijeva od korisnika unos tri parametra. Donji i gornji prag, te sigmu (σ), standardnu devijaciju Gaussovog filtra određenog u pikselima koji utječe na kvalitetu detektirane slike rubova. Autori EdgeConnect modela otkrili su da je σ=2 najbolja vrijednost za generiranje slike rubova u rješavanju problema popunjavanja slika.



Sl. 3.2. (a) ulazna slika koja ima šum, (b) Canny detektor rubova s σ=1, (c) Canny detektor rubova s σ=3. [20]



Sl. 3.3. (a) ulazna slika, (b) Canny detektor rubova s σ=0, (c) σ=2 i (d) σ=4 .

3.2.2. Kreiranje nedostajućih linija

Neka je I_{gt} oznaka za originalnu sliku. Mapa rubova i crnobijela verzija označene su s C_{gt} i I_{gray}. U generatoru rubova G1 (slika 2.5.) kao ulaz koristi se zamaskirana crnobijela slika $\tilde{I}_{gray} = I_{gray} \odot (1 - M)$, njezina mapa rubova $\tilde{C}_{gt} = C_{gt} \odot (1-M)$, te maska M kao preduvjet, u kojoj vrijednost 1 predstavlja područje koje fali, a 0 pozadinu. Simbol $\odot$ označava Hadamardov produkt. Generator predviđa mapu rubova za zamaskirano područje.

$$C_{pred} = G_1(\tilde{I}_{gray}, \tilde{C}_{gt}, M) \quad (3\text{-}1)$$

C_{gt} i C_{pred} (formula 3-1) se koriste za uvjetovanje na I_{gray} kao ulaz u diskriminator koji predviđa je li rubna mapa stvarna. Mreža je trenirana pomoću dvije funkcije gubitka, *adversarial loss* i *feature matching loss*. *Feature matching loss* uspoređuje aktivacijske mape u međuslojevima diskriminatora. Ovo stabilizira proces treniranja prisiljavanjem generatora da proizvodi rezultate sa sličnim značajkama kao na izvornim slikama izračunate od strane diskriminatora rubova i definiran je kao:

$$L_{FM} = \mathbb{E}\left[\sum_{i=1}^{L} \frac{1}{N_i} \left\| D_1^{(i)}(C_{gt}) - D_1^{(i)}(C_{pred}) \right\|_1\right] \quad (3\text{-}2)$$

gdje je L broj konvolucijskih slojeva diskriminatora, N_i je broj elemenata u i-tom aktivacijskom sloju, i $D_1^{(i)}$ je aktivacija u i-tom sloju diskriminatora.

3.2.3. Stvaranje nedostajućeg dijela slike

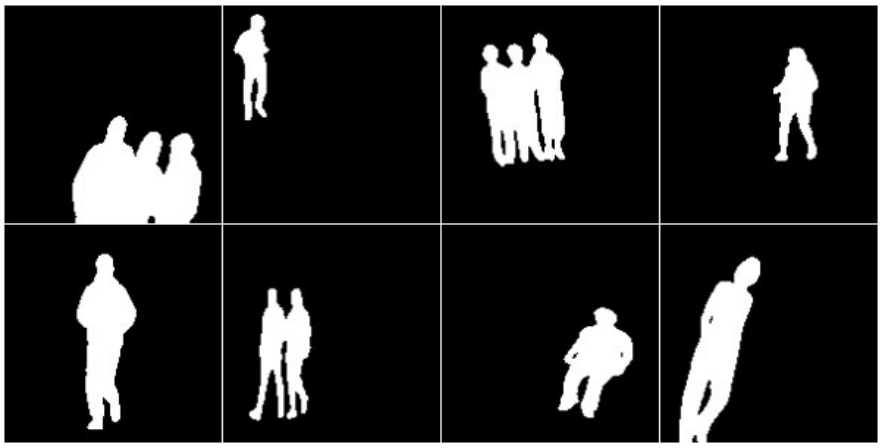
Mreža za dovršetak slike koristi nepotpunu obojenu sliku $\tilde{I}_{gt} = I_{gt} \odot (1 - M)$ kao ulaz, uvjetovanu s mapom rubova C_{comp}. Mapa rubova konstruira je kombiniranjem područja mape rubova originalne slike s mapom rubova generiranim u maskiranom području iz prethodne faze, tj. $C_{comp} = C_{gt} \odot (1 - M) + C_{pred} \odot M$. Mreža vraća obojenu sliku I_{pred}, na kojoj su popunjena područja koja su nedostajala, slika je iste rezolucije kao ulazna slika.

$$I_{pred} = G_2(\tilde{I}_{gt}, C_{comp}) \quad (3\text{-}3)$$

3.3. Podatkovni skup

Slike za treniranje modela preuzete su iz skupa podataka s Kaggle web stranice [12], slike su JPG formata. Izdvojene su slike grada i ulica bez ljudi (slika 3.4.). Razlog tome je što slike s puno detalja bi mogle loše utjecati na trening modela. Slike su 150x150 piksela. Za treniranje modela bilo je potrebno smanjiti ih na 148x148 piksela zbog toga što je modelu u drugom sloju generatora potrebna veličina slike djeljiva s 2. Ulazna maska koja se koristi za treniranje, testiranje i validaciju modela je PNG (engl. *Portable Network Graphics*) binarna matrica, sadrži isključivo crne i bijele pixele (engl. *grayscale*). Maske koju su korištene u treningu generirane su pomoću besplatnog Inkspace alata za vektorsko crtanje. Maske predstavljaju siluete ljudi. Nakon toga su te maske transformirane na nasumične pozicije, veličine i rotacije s dodatnom distorzijom po slici, u setu maski za treniranje nalazi se preko 2000 slika. Na slici 3.5. nalazi se primjer nekoliko takvih maski.

Sl. 3.4. Primjer slika za treniranje.



Sl. 3.5. Primjer generiranih maski.

4. TRENIRANJE MODELA

Treniranje ovog modela odvija se u dvije faze. U prvoj fazi trenira se model za generiranje rubova nedostajuće slike koji koristi Canny detektor rubova. U drugoj fazi trenira se model za inpainting koji koristi generirane rubove iz prve faze treninga. Odabrani model implementiran je u PyTorchu.

4.1. Priprema podatka

Skup preuzetih slika podijeljen je u tri dijela, skup za trening, validacijski skup te skup za testiranje. U skupu za trening nalazi se 2320 slika, dok se unutar validacijskog i testnog skupa nalaze po 50 slika. Validacijski skup koristi se za validaciju i odabir modela. Nekoliko testnih slika koristi se na kraju za evaluaciju modela.

4.2. Hiperparametri

Hiperparametri korišteni za treniranje modela grupirani su u nekoliko skupina. U nastavku slijedi detaljnije objašnjenje za svaku od tih skupina. Unutar Tablica 4.1 nalaze se hiperparametri koji se odnose na generalne postavke za model zajedno s njihovim objašnjenjima.

Tablica 4.1. Generalni hiperparametri

Hiperparametar	Vrijednost	Pojašnjenje
GPU	0	Lista id-ova korištenih grafičkih kartica
EDGE	1	Odabir detektora rubova, 1 označava korištenje Canny detektora rubova
MASK	3	Vrsta maski koje su korištene pri treningu
NMS	True	Suzbijanje ne-maksimalnih vrijednosti na vanjskim rubovima
SEED	-1	Nasumični broj za sjeme generatora

Druga skupina hiperparametara odnosi se na hiperparametre za treniranje (Tablica 4.2), veličina ulazne slike, broj iteracija potrebnih prije spremanja modela, broj iteracija nakon koje će se spremiti vrijednost loss funkcije, broj iteracija nakon koje će se pohraniti uzorak slike, broj slika koji će se koritsti u uzorku. Hiperparametri za pripremu slike prije ulaska u model za generiranje rubova uključuju Standardnu devijaciju Gaussovog filtra i prag detekcije rubova u Canny detektoru rubova. Ovaj prag se koristi za razdvajanje izraženih rubova od slabijih rubova i šuma u slici. Što je veća težina *perceptual loss*-a, model će više paziti na percepcijske aspekte slike nego na strogu pikselsku točnost. NSGAN koristi nezasićujuću (engl. *non-saturating*) funkciju gubitka kako bi se riješio problem zasićenja. Umjesto da pokušava minimizirati vjerojatnost da diskriminator prepozna generirane slike kao lažne, generator s NSGAN gubitkom pokušava maksimizirati vjerojatnost da diskriminator pogrešno prepozna generirane slike kao prave. Ovo može potaknuti jače gradijente za generator čak i kada diskriminator dobro radi, čime se poboljšava stabilnost i učinkovitost treniranja.

Tablica 4.2. Hiperparametri za teniranje modela

Hiperparametar	Vrijednost	Pojašnjenje
INPUT_SIZE	148	Veličina ulaznih slika za treniranje
D2G_LR	0.1	Omjer stope učenja diskriminatora i generatora
SAVE_INTERVAL	1000	Broj iteracija prije spremanja modela.
LOG_INTERVAL	500	Broj iteracija prije spremanja trenutne loss funkcije
SAMPLE_INTERVAL	1000	Broj iteracija prije pohranjivanja uzorka slike.
SAMPLE_SIZE	8	Broj slika koji će se koristiti u uzorku.
SIGMA	2	Standardna devijacija Gaussovog filtra koji se koristi u Canny detektoru rubova
EDGE_TRESHOLD	0.5	Prag detekcije rubova u Canny detektoru rubova
L1_LOSS_WEIGHT	1	Hiperparametar koji određuje doprinos L1 regularizacije u ukupnoj funkciji gubitka
FM_LOSS_WEIGHT	10	Težina gubitka usklađivanja značajki (engl. *feature-matching loss weight*)
STYLE_LOSS_WEIGHT	250	Težina gubitka stila
CONTENT_LOSS_WEIGHT	0.1	Težina perceptivnog gubitaka
INPAINT_ADV_LOSS_WEIGHT	0.1	Težina protivničkog kontradiktornog gubitaka
GAN_LOSS	nsgan	NSGAN funkcija gubitka
GAN_POOL_SIZE	0	Slike se ne pohranjuju u bazen lažnih slika
MAX_ITERS	3e6	Broj iteracija nakon kojeg će se zaustaviti treniranje modela

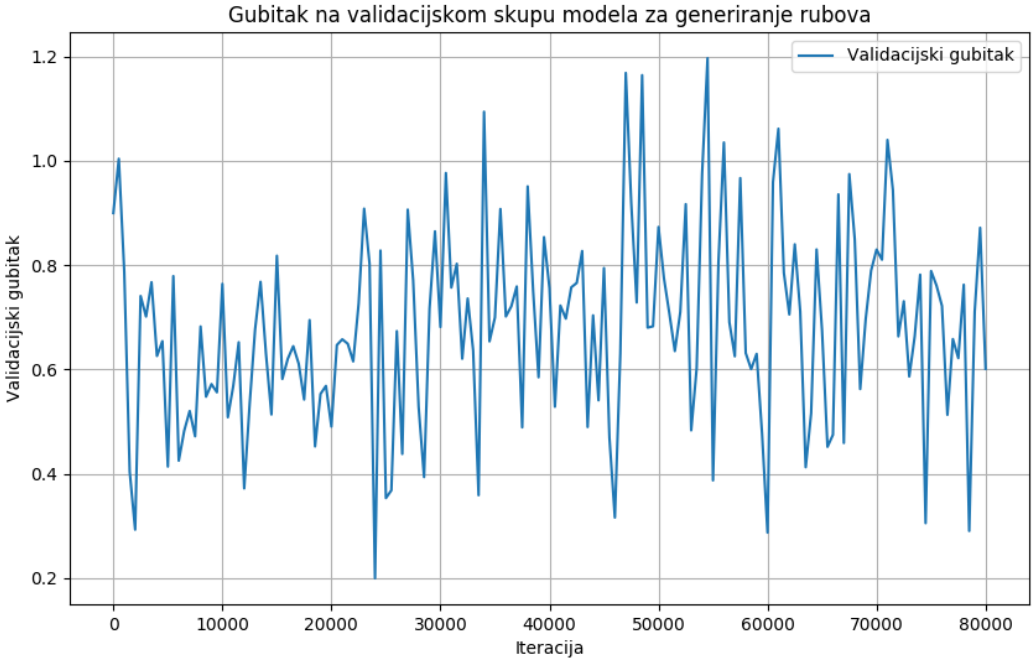
U Tablica 4.3 nalaze se optimizacijski hiperparametri, oni utječu na način na koji model prilagođava svoje težine tijekom treniranja kako bi minimizirao pogrešku. Stopa učenja je hiperparametar koji određuje veličinu koraka na svakoj iteraciji prilikom kretanja prema minimumu funkcije gubitka. Diktira koliko se model prilagođava u odgovoru na procijenjenu pogrešku svaki puta kada se težine modela ažuriraju. Kada je stopa premala može dovesti do sporog konvergiranja, model pravi vrlo male korake prema optimalnom rješenju, potrebno je mnogo iteracija da bi se do njega došlo. Također postoji rizik od zaglavljivanja u lokalnim minimumima. Dok prevelika stopa može uzrokovati da model preskoči optimalno rješenje ili divergira. Veličina serije odnosi se na broj primjera za treniranje koji se koriste u jednoj iteraciji. Optimizator Adam je algoritam optimizacije koji se koristi prilikom treniranja modela i ima dva hiperparamtra, β1 i β2.

Tablica 4.3. Hiperparametri optimizacije

Hiperparametar	Vrijednost	Pojašnjenje
LR	0.0001	Stopa učenja (engl. *learning rate*)
BATCH_SIZE	8	Veličina ulazne serije
BETA1	0.0	Adam optimizator beta1
BETA2	0.9	Adam optimizator beta2

4.3. Treniranje modela

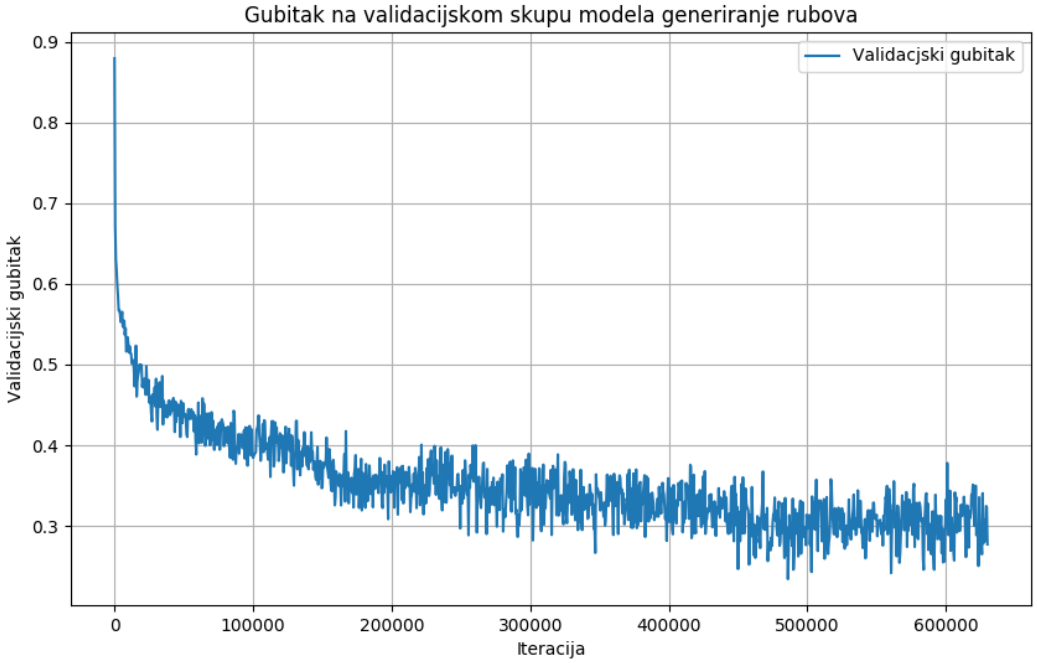
Model je dotreniran koristeći njihov predtrenirani model nad skupom podataka ParisStreetView kako bi se postigla brža konvergencija. Pokušano je treniranje bez njega, ali zbog male veličine podatkovnog skupa rezultati nisu bili zadovoljavajući. Treniranje prvog modela za generiranje rubova prekinuto je nakon 80200 iteracija jer je protivnički gubitak na validacijskom skupu podataka (engl. *validation loss*) rastao i model nije konvergirao. Na slici 4.1. prikazan je gubitak na validacijskom skupu podataka tijekom treniranja bez korištenja predtreniranog modela za *edge* model. *Edge* i *inpaint* model ovise jedan o drugome pa se drugi model nije ni trenirao u ovom slučaju.



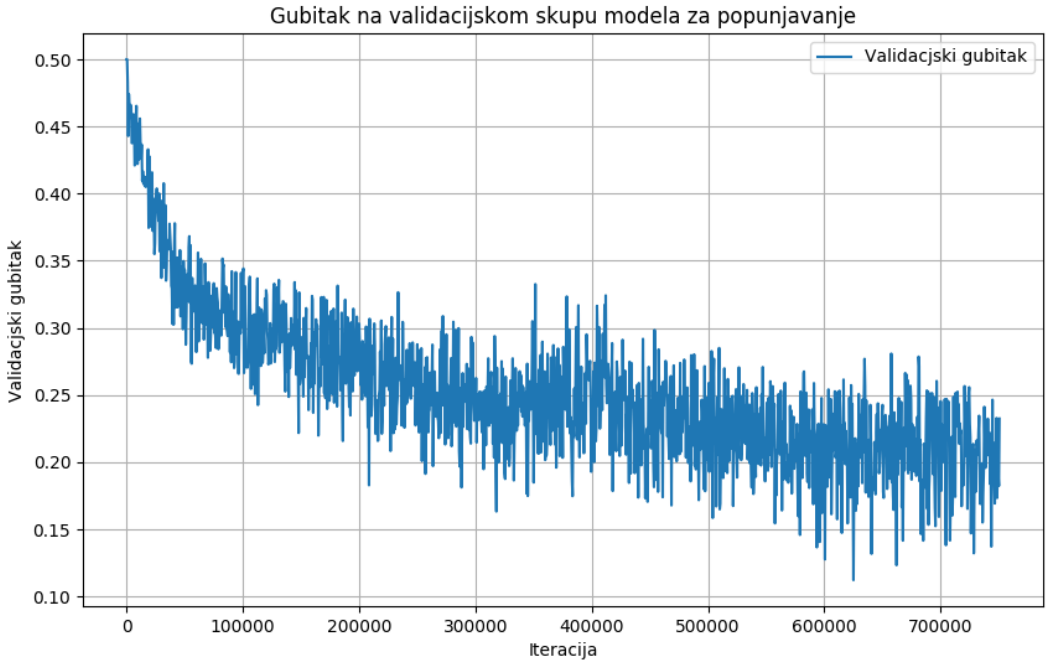
Sl. 4.1. Gubitak na validacijskom skupu podataka modela za generiranje rubova bez korištenja predtreniranog modela.

Na slici 4.2. se vidi protivnički gubitak na validacijskom skupu podataka tijekom treniranja modela s korištenjem predtreniranog modela treniranom na ParisStreetView podatkovnom skupu za *edge* model. Model je treniran 625100 iteracija jer je funkcija gubitka prestala opadati pa je treniranje zaustavljeno, vidljivo je da je funkcija gubitka uspješno konvergirala, a model nije počeo previše usklađivati na podatke za treniranje (engl. *overfiting*).

Treniranje duge faze modela zastavljeno je nakon 748400 iteracija, treniranje je zaustavljeno jer je funkcija gubitka počela sporo opadati, ukazujući na uspješnu konvergenciju bez vidljivog prevelikog usklađivati na podatke za treniranje. Ovo ukazuje da bi dodatnim treniranjem mogli dobiti još bolje rezultate.



Sl. 4.3. Gubitak na validacijskom skupu podataka modela za generiranje rubova uz korištenja predtreniranog modela.



Sl. 4.2. Gubitak na validacijskom skupu podataka modela za popunjavanje slike uz korištenja predtreniranog modela.

5. EVAULACIJA RJEŠENJA

5.1. Usporedba originalne slike s generiranom

Na slikama 5.1. i 5.2. prikazano je 20 originalnih slika, zatim originalnih slika s maskom, slika s generiranim rubovima s prvim modelom, te konačna slika popunjena s drugim modelom. Za dobivene slike izračunate su dvije metrike validacije sličnosti slika [17] PSNR (engl. *Peak Signal-to-Noise Ratio*) [18] i SSIM (engl. *Structural Similarity Index Measure)* [19]. PSNR je mjera koja uspoređuje originalnu sliku s rekonstruiranom slikom i ocjenjuje razinu šuma u odnosu na korisne informacije. Viša vrijednost PSNR-a obično ukazuje na veću sličnost između originalne i rekonstruirane slike. Međutim, PSNR nije uvijek dobar pokazatelj percepcijske kvalitete slike jer ne uzima u obzir ljudsku percepciju. Slika s visokim PSNR-om može izgledati loše ljudskom oku. SSIM je metrika koja uzima u obzir ljudsku percepciju slike. Ona uspoređuje strukturu, svjetlinu i kontrast između originalne i rekonstruirane slike. SSIM daje vrijednosti između -1 i 1, gdje je 1 savršena podudarnost između slika. Veće vrijednosti SSIM-a obično ukazuju na bolju percepcijsku sličnost između slika. U kontekstu inpaintinga, PSNR može ocijeniti kvalitetu rekonstrukcije na temelju fizičkih sličnosti, dok SSIM uzima u obzir i ljudsku percepciju, što ga čini boljom metrikom za ocjenu kvalitete inpaintinga, osobito kad je važno očuvati strukturu i detalje u slici. U Tablica 5.1 prikazane su vrijednosti PSNR i SSIM izračunate za testni skup slika.

Tablica 5.1. Metrike evaluacije SSIM i PSNR

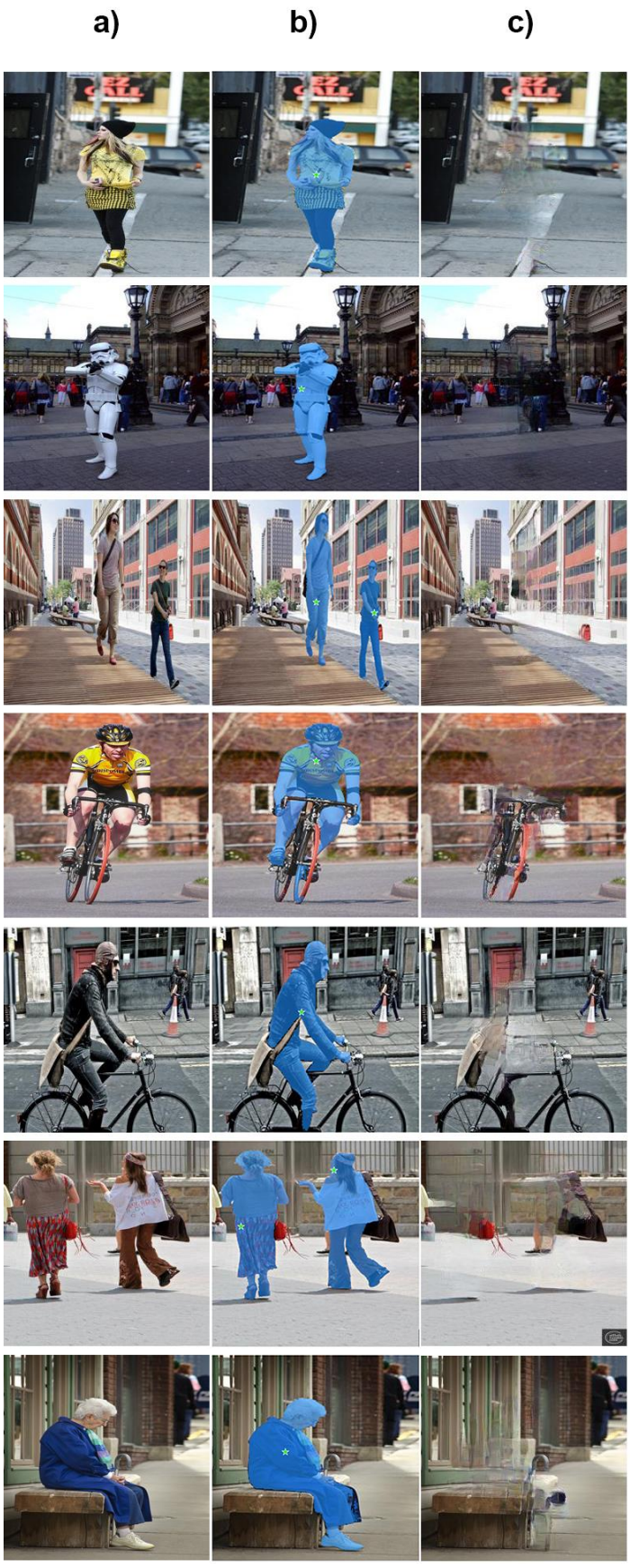
Metrika	Vrijednost
SSIM	0.937
PSNR	29.437

Sl. 5.1. Primjer popunjavanja slika bez SAM modela. Stupac a) original slika, b) slika s maskom, c) izlaz iz generatora rubova, d) konačna slika.

Sl. 5.2. Primjer popunjavanja slika bez SAM modela. Stupac a) original slika, b) slika s maskom, c) izlaz iz generatora rubova, d) konačna slika.

5.2. Uklanjanje objekata pomoću SAM modela

Na slici 5.3. prikazano je 7 slika koje su maskirane pomoću SAM segmentacijskog modela, gdje je maska generirana pomoću jedne točke, zatim je maska proširena za 5 piksela te popunjena korištenjem ranije objašnjenog modela. U primjeru je prikazano nekoliko slika na kojima SAM model selektira osobe, ali izostavi objekte koje bi trebalo ukloniti jer zbune model za popunjavanje slike.



Sl. 5.3. Primjer popunjavanja slika uz korištenje SAM modela. Stupac a) original slika, b) SAM segmentacija, c) konačna slika.

6. ZAKLJUČAK

U ovom radu opisana su postojeća rješenja za uklanjanje objekata sa slika. Odabran je model za segmentaciju i inpainting. Model za segmentaciju (SAM) generira dovoljno precizne maske za rješavanje ovog problema, iako model selektira objekt sa zadovoljavajućom pogreškom, mjesta za poboljšanje ima. Često sjena objekta ne ulazi u selekciju, isto tako ni razni objekti koje nose osobe poput torbi, štapa, bicikala, te nakon uklanjanja objekta to ostavlja trag postojanja objekta na slici. Odabrani model EdgeConnect pokazao se dobar za uža maskirana područja, dok se kod velikih područja u popunjavanju pojavljuju zamućeni dijelovi. Iako pomoću PSNR metrike nije dobivena visoka ocjena, pomoću SSIM metrike dobivena ocjena pri samom vrhu ljestvice. Rezultat bi se mogao poboljšati treniranjem modela na većem skupu podataka. To bi naravno produžilo vrijeme treniranja. Jedan od problema također je i veličina slika na kojem je treniran model, model je treniran na slikama niske rezolucije i zato uklanjanje objekata na slikama visoke rezolucije ne daje kvalitetne rezultate. To bi se moglo ispraviti tako da se modelu za treniranje predaju slike visoke rezolucije, što bi dodatno produžilo vrijeme treniranja. Iako ovakvi pristupi mogu poboljšati rezultat i dalje popunjavanje ovisi o samoj slici koju treba popraviti. Bolji rezultat bi dobili korištenjem novijih generativnih modela dubokog učenja. Poput modela za generiranje slike iz teksta (engl. *text-to-image*). Ovakvo popunjavanje praznih dijelova slike bi manje ovisilo o samoj slici već bi model mogao popuniti prazninu s više smislenog konteksta.

LITERATURA

[1] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei A. Efros, „Context Encoders: Feature Learning by Inpainting“ 2016.,
dostupno na: https://arxiv.org/pdf/1604.07379.pdf [30.6.2023.]

[2] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, „Generative Image Inpainting with Contextual Attention“, 2018.
dostupno na: https://arxiv.org/abs/1801.07892 [30.6.2023.]

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, „Generative Adversarial Nets“, 2014.
dostupno na: https://arxiv.org/pdf/1406.2661.pdf [30.6.2023.]

[4] Goldman D., Shechtman E., Barnes C., Belaunde I., Chien J, „Content-aware fill“.

[5] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, Shiguang Shan, Shift-Net: Image Inpainting via Deep Feature Rearrangement, dostupno na: https://arxiv.org/pdf/1801.09392.pdf [30.6.2023.]

[6] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, Mehran Ebrahimi, „EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning“, 2019. dostupno na: https://arxiv.org/abs/1901.00212 [30.6.2023.]

[7] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, „Generative Image Inpainting with Contextual Attention“, dostupno na: https://arxiv.org/pdf/1801.07892.pdf [30.6.2023.]

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick, „Segment Anything“, 2023., dostupno na: https://arxiv.org/abs/2304.02643 [30.6.2023.]

[9] E. Alpaydın, „Introduction to Machine Learning“, The MIT Press, Cambridge, Massachusetts, 2010., dostupno na: https://dl.matlabyar.com/siavash/ML/Book/Ethem%20Alpaydin-Introduction%20to%20Machine%20Learning-The%20MIT%20Press%20(2014).pdf [30.6.2023]

[10] John Canny, „A Computational Approach to Edge Detection“, 1986. dostupno na: https://www.researchgate.net/publication/224377985_A_Computational_Approach_To_Edge_Detection [30.6.2023.]

[11] Canny edge detector, dostupno na: https://en.wikipedia.org/wiki/Canny_edge_detector [30.6.2023.]

[12] Landscape color and grayscale images, dostupno na: https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization [30.6.2023.]

[13] Samuel Black, Somayeh Keshavarz, Richard Souvenir, „Evaluation of Image Inpainting for Classification and Retrieval“, Department of Computer and Information Sciences, Temple University, 2020.,

dostupno na: https://openaccess.thecvf.com/content_WACV_2020/papers/Black_Evaluation_of_Image_Inpainting_for_Classification_and_Retrieval_WACV_2020_paper.pdf [30.6.2023.]

[14] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, Jiaya Jia, „DeepFill v1, Image Inpainting via Generative Multi-column Convolutional Neural Networks“, 2018., dostupno na: https://arxiv.org/pdf/1810.08771.pdf [30.6.2023.]

[15] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, Bryan Catanzaro , „Image Inpainting for Irregular Holes Using Partial Convolutions“, 2018., dostupno na: https://arxiv.org/pdf/1804.07723.pdf [30.6.2023.]

[16] JND, Just-noticeable differencedostupno na: https://en.wikipedia.org/wiki/Just-noticeable_difference [12.8.2023.]

[17] Sayed Nadim, Image Quality Evaluation Metrics, dostupno na: https://github.com/SayedNadim/Image-Quality-Evaluation-Metrics [14.8.2023.]

[18] Peak signal-to-noise ratio, dostupno na: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio [15.8.2023.]

[19] Structural similarity index, dostupno na: https://en.wikipedia.org/wiki/Structural_similarity [15.8.2023.]

[20] Canny Edge Detector sigma, dostupno na: https://scikit-image.org/docs/stable/auto_examples/edges/plot_canny.html [18.9.2023.]

SAŽETAK

Ovaj diplomski rad istražuje kako kombinacija tehnika dubokog učenja segmentacija objekata sa slike i inpainting može ubrzati uklanjanje objekata sa slika. Na početku su objašnjenja postojeća rješenja za uklanjanje objekata sa slika. Za rješenje ovog problema koristi se model za semantičku segmentaciju SAM za generiranje maske, a za popunjavanje maskiranog dijela slike koristi se model EdgeConnect. Modeli su detaljnije objašnjeni. Rad je fokusiran na uklanjanju osoba s fotografija ulica, za trening su generirane maske silueta ljudi i preuzet skup podataka slika iz kojeg su izdvojene slike ulica i gradova. Model je dotreniran na predtreniranom modelu treniranom nad skupom podataka ParisStreetView.

Ključne riječi: EdgeConnect, inpainting, duboko učenje

ABSTRACT

REMOVING OBJECTS FROM IMAGES USING DEEP LEARNING

This thesis explores how the combination of deep learning techniques for object segmentation from images and inpainting can expedite the removal of objects from pictures. At the outset, existing solutions for object removal from images are explained. To address this issue, the SAM model for semantic segmentation is used to generate a mask, and the EdgeConnect model is utilized for filling in the masked portion of the image. The models are explained in detail. The research is focused on the removal of people from street photographs. For training, silhouette masks of people were generated, and a dataset containing images was acquired, from which street and city images were extracted. The model was fine-tuned on a pre-trained model trained on the ParisStreetView dataset.

Keywords: EdgeConnect, inpainting, deep learning.

ŽIVOTOPIS

Marko Varšava rođen je 18. srpnja 1996. godine u Đakovu. Nakon završene Osnovne škole Ivana Gorana Kovačića u Đakovu, 2011. godine upisuje Gimnaziju Antuna Gustava Matoša u Đakovu, smjer Prirodoslovno-matematički, te ju završava 2015. godine. Obrazovanje nastavlja iste godine na Elektrotehničkom fakultetu u Osijeku (današnji Fakultet elektrotehnike, računarstva i informacijskih tehnologija) na kojem upisuje preddiplomski studij računarstva. U 2019. godini stječe naziv univ.bacc.ing. te upisuje diplomski studij računarstva, izborni blok Informacijske i podatkovne znanosti, na Fakultetu elekrotehnike, računarstva i informacijkih tehnologija.

_____

Potpis autora