

Super rezolucija temeljena na generativnim suparničkim neuronskim mrežama

Momić, Ines

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:445057>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-28**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Sveučilišni diplomski studij Automobilsko računarstvo i komunikacije

**SUPER REZOLUCIJA TEMELJENA NA
GENERATIVNIM SUPARNIČKIM NEURONSKIM
MREŽAMA**

Diplomski rad

Ines Momić

Osijek, 2024.

Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju

Ocjena diplomskog rada na sveučilišnom diplomskom studiju

Ime i prezime pristupnika:	Ines Momić
Studij, smjer:	Sveučilišni diplomski studij Automobilsko računarstvo i
Mat. br. pristupnika, god.	D-71 ARK, 07.10.2022.
JMBAG:	0165083955
Mentor:	izv. prof. dr. sc. Ratko Grbić
Sumentor:	
Sumentor iz tvrtke:	Zvonimir Kaprocki
Predsjednik Povjerenstva:	prof. dr. sc. Mario Vranješ
Član Povjerenstva 1:	izv. prof. dr. sc. Ratko Grbić
Član Povjerenstva 2:	prof. dr. sc. Marijan Herceg
Naslov diplomskog rada:	Super rezolucija temeljena na generativnim suparničkim neuronskim mrežama
Znanstvena grana diplomskog rada:	Umjetna inteligencija (zn. polje računarstvo)
Zadatak diplomskog rada:	Tehnika super rezolucije koristi se u raznim područjima za poboljšanje kvalitete slika ili videa. Ova tehnika zasniva se na modernim algoritmima u području strojnog učenja (kao npr. generativnim suparničkim mrežama), pri čemu se iz niskokvalitetnih slika (niske razlučivosti) generiraju realistične slike visoke razlučivosti, čime se poboljšavaju oštrina i prikaz detalja slike. Ovo je čest zadatak u mnogim primjenama poput povećanja razlučivosti video signala koji se prenosi u nižoj rezoluciji do mjesta reprodukcije zbog manjih zahtjeva na propusnost mreže. U okviru
Datum ocjene pismenog dijela diplomskog rada od strane mentora:	17.09.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	30.09.2024.
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	30.09.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O IZVORNOSTI RADA**

Osijek, 30.09.2024.

Ime i prezime Pristupnika:

Ines Momić

Studij:

Sveučilišni diplomski studij Automobilsko računarstvo i komunikacije

Mat. br. Pristupnika, godina upisa:

D-71 ARK, 07.10.2022.

Turnitin podudaranje [%]:

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Super rezolucija temeljena na generativnim suparničkim neuronskim mrežama**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ratko Grbić

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 pristupnika:

SADRŽAJ

1. UVOD	4
2. PREGLED POSTOJEĆIH RJEŠENJA ZA SUPER REZOLUCIJU TEMELJENIH NA GENERATIVNIM SUPARNIČKIM MREŽAMA	1
2.1. Problem super rezolucije temeljene na generativnim suparničkim mrežama	1
2.2. Pregled postojećih metrika za procjenu kvalitete slika	4
2.3. Pregled postojećih rješenja za super rezoluciju	8
3. EVALUACIJA POSTOJEĆIH ALGORITAMA I PRIJEDLOG VLASTITOG ALGORITMA ZA POVEĆANJE RAZLUČIVOSTI LICA S DIGITALNIH SLIKA.....	15
3.1. Postojeći algoritmi za super rezoluciju i opis korištenog skupa podataka	15
3.2. Rezultati testiranja postojećih algoritama i evaluacija rješenja uz pomoć IQA metrika	17
3.2.1. Evaluacija i analiza rješenja postojećih super rezolucijskih algoritama	18
3.3. Razvoj vlastite neuronske mreže za super rezoluciju lica	27
3.3.1. Alati i programske biblioteke korištene za razvoj neuronske mreže	27
3.3.2. Osnovni model.....	27
3.3.3. Funkcije gubitaka	28
3.3.4. Skup podataka	29
3.3.5. Treniranje neuronske mreže	30
3.4. Izrada aplikacije za super rezoluciju i implementacija aplikacije na NVIDIA Jetson Nano.	34
3.4.1. Zahtjevi na aplikaciju	34
3.4.2. Alati korišteni za razvoj aplikacije	36
3.4.3. Predložena arhitektura aplikacije.....	37
3.4.4. Implementacija grafičkog korisničkog sučelja aplikacije	38
3.4.5. Implementacija modula za obradu videa	40
3.4.6. Implementacija modula za detekciju lica.....	42
3.4.7. Implementacija modula za super rezoluciju lica.....	44
4. EVALUACIJA I VERIFIKACIJA PREDLOŽENOG ALGORITMA I APLIKACIJE ZA POVEĆANJE RAZLUČIVOSTI LICA S DIGITALNIH SLIKA	46
4.1. Evaluacija istreniranog vlastitog modela za povećanje razlučivosti slike lica.....	46
4.2. Verifikacija aplikacije za povećanje razlučivosti lica	49
4.2.1. Testiranje aplikacije na videozapisima učitanim iz datotečnog sustava	49
4.2.2. Testiranje aplikacije na prijenosu s kamere u stvarnom vremenu	55

4.3. Evaluacija rješenja aplikacije za povećanje razlučivosti lica.....	57
5. ZAKLJUČAK.....	60
LITERATURA	62
SAŽETAK.....	65
ABSTRACT	66
ŽIVOTOPIS.....	67

1. UVOD

Strojno učenje je grana umjetne inteligencije i računarske znanosti koja je posljednjih godina postala temelj za algoritme super rezolucije (engl. *super-resolution*), čiji je cilj poboljšati kvalitetu slika povećanjem njihove rezolucije. Proces super rezolucije omogućio je unaprjeđenja u područjima primjene kao što su medicinske slike, satelitske slike i prepoznavanje lica, gdje su izrazito važni detalji i jasnoća slika. Super rezolucija također pronalazi široku primjenu u području televizije, gdje se sadržaj niže rezolucije (npr. 1080p) prikazuje u višoj rezoluciji (npr. 4K) kada on nije dostupan u 4K formatu. Primjerice, broj nadzornih kamera eksponencijalno raste svake godine zbog brojnih razloga kao što su nadzor prometa, unutarnjih prostora ili vanjskih događaja. Glavni izazov u podacima dobivenih sa nadzornih kamera je loša kvaliteta slika do koje dolazi zbog raznih šumova, degradacija i vremenskih nepogoda. Algoritmi super rezolucije pronalaze ključnu primjenu u području nadzornih kamera, gdje je poboljšanje kvalitete slika osobito važno u kontekstu prepoznavanja lica sa nadzornih snimaka, registarskih oznaka i sl.

Algoritmi super rezolucije mogu se podijeliti u dvije glavne kategorije: tradicionalne tehnike temeljene na interpolaciji i tehnike temeljene na dubokom učenju. Tehnika interpolacije odnosi se na povećanje rezolucije slike dodavanjem novih elemenata slike na nisko rezolucijsku sliku. Tri najčešće korištene tehnike interpolacije uključuju: interpolaciju najbližih susjeda, bilinearnu interpolaciju i bikubičnu interpolaciju. Tehnike interpolacije procjenjuju nove vrijednosti elemenata slike izračunavanjem prosjeka okolnih elemenata slike. Tehnike zahtijevaju minimalne proračune što proces super rezolucije temeljen na interpolaciji čini iznimno brzim. Unatoč brzom procesu, visoko rezolucijske slike generirane interpolacijskim tehnikama često izgledaju neprirodno i mutno. Razlog tome je što interpolacijskim metrikama nedostaje sposobnost prepoznavanja složenih uzoraka, tekstura i vizualnih informacija na slici, što je ključno za generiranje visokokvalitetnih slika.

S druge strane, algoritmi za povećanje rezolucije temeljeni na dubokom učenju, osobito oni temeljeni na neuronskim mrežama, mogu istaknuti detalje i dodati teksture koje nisu prisutne u originalnoj slici, tako da slika izgleda vrlo realistično. Navedeni algoritmi postižu super rezoluciju korištenjem tehnika strojnog učenja, koje uključuju treniranje modela na parovima ulaznih nisko rezolucijskih (engl. *Low Resolution* – LR) i visoko rezolucijskih (engl. *High Resolution* – HR) slika, gdje model uči kako predvidjeti visoko rezolucijsku sliku iz njenog nisko rezolucijskog para. Najčešći pristup razvoju ovakvih modela temelji se na generativnim suparničkim neuronskim mrežama (engl. *Generative Adversial Neural Networks* – GANs).

Generativne suparničke neuronske mreže sastoje se od dvije mreže: mreže generatora i mreže diskriminatora. Navedene mreže treniraju se istovremeno i zajedno stvaraju realistične i visokokvalitetne slike. Generator nastoji generirati visoko rezolucijski izlaz iz nisko rezolucijskog ulaza, dok diskriminator procjenjuje realističnost generiranih slika (procjenjuje jesu li generirane slike stvarne ili lažne). Mreže utječu jedna na drugu, uče jedna od druge te se neprestano poboljšavaju. S vremenom, generator postaje sposoban generirati uvjerljive visoko rezolucijske slike ili super rezolucijske slike (engl. *Super Resolution* – SR) te uspješno zavarava diskriminator koji te slike procjenjuje kao stvarne slike (engl. *Ground Truth* – GT) [1].

U ovom diplomskom radu proučeni su i uspoređeni postojeći algoritmi za super rezoluciju temeljeni na generativnim suparničkim neuronskim mrežama. Postojeća rješenja detaljno su opisana te testirana na odgovarajućim skupovima podataka. Prikupljeni su svi kvalitativni rezultati testiranih super rezolucijskih algoritama, nakon čega je provedena evaluacija rješenja pomoću metrika za procjenu kvalitete slika. Prikupljena su sva kvantitativna rješenja dobivena pomoću metrika te su rezultati uspoređeni i analizirani kako bi se pronašao algoritam s najboljim performansama za super rezoluciju lica.

Nadalje, u okviru diplomskog rada razvijen je algoritam super rezolucije pomoću generativnih suparničkih mreža usmjeren na povećanje rezolucije lica iz oštećenih nisko rezolucijskih slika ili izravno iz snimaka dobivenih pomoću kamere. Za razvoj algoritma korištena je tehnika generativnih suparničkih mreža. Razvijena neuronska mreža trenirana je na velikom skupu podataka koji sadrži raznolike slike lica. Nakon postupka treniranja mreže za super rezoluciju izvršena je evaluacija mreže na nisko rezolucijskom skupu podataka korištenjem metrika za procjenu kvalitete slike. Konačno, neuronska mreža je implementirana na ugradbeni računalni sustav NVIDIA Jetson Nano gdje je demonstrirana sposobnost generiranja visoko rezolucijskih slika lica u stvarnom vremenu. Razvijeni algoritam ima potencijalnu primjenu u sustavima nadzornih kamera, gdje bi olakšao identificiranje i analizu lica sa nadzornih snimaka.

Ovaj rad je strukturiran na sljedeći način. U drugom poglavlju predstavljena je metoda super rezolucije i opisana su postojeća rješenja za super rezoluciju korištenjem generativnih suparničkih mreža kao i postojeće metrike za procjenu kvalitete slika. U trećem poglavlju prikazano je detaljno testiranje i evaluacija postojećih algoritama te je opisan razvoj vlastitog algoritma za super rezoluciju slika lica. Opisan je i razvoj aplikacije za super rezoluciju lica u stvarnom vremenu koja je prilagođena za rad na ugradbenom računalnom sustavu NVIDIA Jetson

Nano. U četvrtom poglavlju dani su rezultati testiranja i evaluacije vlastitog modela te je provedena verifikacija rada aplikacije. Na kraju rada je izveden zaključak na temelju rezultata testiranja.

2. PREGLED POSTOJEĆIH RJEŠENJA ZA SUPER REZOLUCIJU TEMELJENIH NA GENERATIVNIM SUPARNIČKIM MREŽAMA

U ovom poglavlju predstavljena je metoda super rezolucije te su detaljno opisana postojeća rješenja za super rezoluciju temeljena na generativnim suparničkim mrežama.

2.1. Problem super rezolucije temeljene na generativnim suparničkim mrežama

Super rezolucija je algoritam koji se temelji na strojnom učenju, točnije dubokim neuronskim mrežama s ciljem povećanja razlučivosti digitalnih slika ili videozapisa. Algoritmi za super rezoluciju generiraju slike visoke razlučivosti iz seta slika niske razlučivosti. Procesom povećanja rezolucije povećava se oštrina i jasnoća slike te je moguće rekonstruirati detalje slike koji su izgubljeni u nisko razlučivim slikama zbog posljedica digitalnog šuma i kompresijskih artefakata, a da se pri tome zadržava originalan sadržaj slike. Na slici 2.1 prikazan je primjer primjene algoritma super rezolucije gdje je na slici a) LR slika rezolucije 62x90 elemenata slike, a na slici b) SR slika rezolucije 248x360 elemenata slike. .



a)

b)

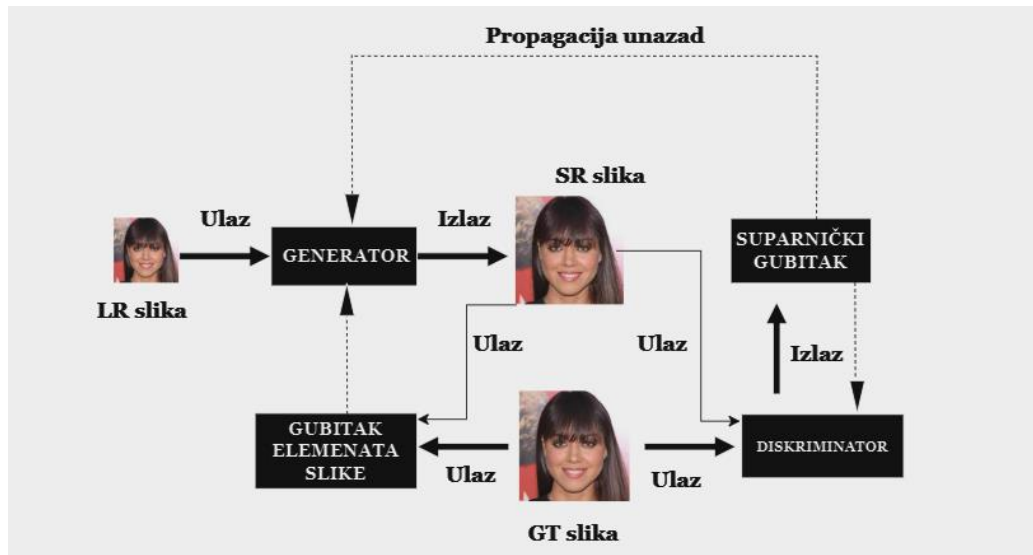
Slika 2.1. Primjer primjene algoritma super rezolucije za nisko rezolucijski ulaz: a) LR slika, b) SR slika

Moderni algoritmi iz područja dubokog učenja zajedno s generativnim suparničkim mrežama (eng. *Generative Adversarial Networks – GANs*) omogućili su značajna poboljšanja u

području super rezolucije slika i videozapisa. Rezultati korištenja navedenih algoritama iz područja strojnog učenja nisu samo doveli do povećanja prostorne rezolucije slika već i do razvoja novih područja istraživanja. Metoda super rezolucije danas se široko koristi za rekonstrukciju starih i oštećenih fotografija ili videozapisa kao i za poboljšanje performansi algoritama za detekciju objekata i prepoznavanje lica. Osim navedenih primjena bitno je naglasiti i primjenu super rezolucije u području medicine, gdje poboljšanje razlučivosti medicinskih slika može dovesti do sigurnije dijagnoze i liječenja. Algoritmi super rezolucije posebno su važni i korisni u području analize snimaka nadzornih kamera. Glavna ograničenja u slikama dobivenih iz nadzornih kamera su niskokvalitetne slike koje nastaju zbog brojnih razloga poput velikog šuma ili lošeg osvjetljenja kao i problema u okolišu poput magle, kiše ili snijega. Tehnika super rezolucije korištenjem generativnih suparničkih mreža omogućuje izgradnju modela koji uspješno poboljšava kvalitetu slike i poboljšanje lica sa slika dobivenih s nadzornih kamera [2]. U ovom radu, primaran fokus je na povećanju razlučivosti lica s digitalnih slika i videozapisa korištenjem generativnih suparničkih mreža.

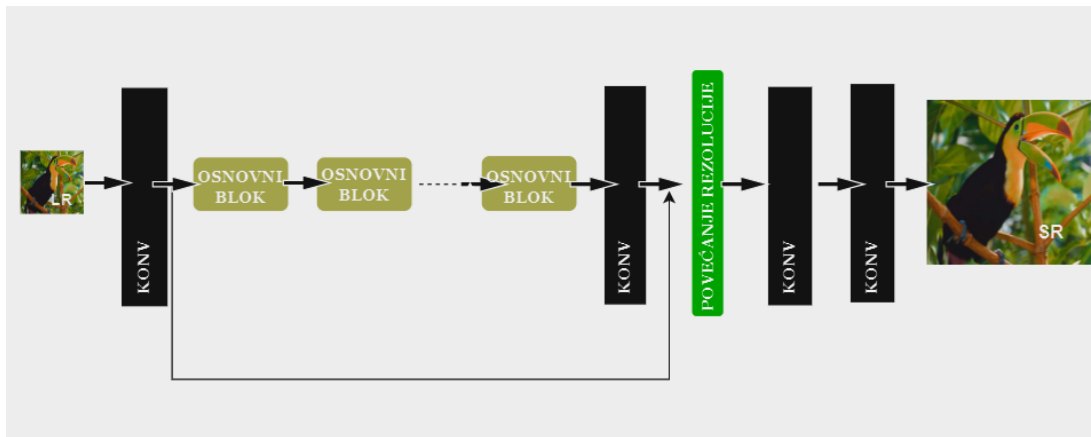
Generativni modeli su vrsta modela strojnog učenja koji za cilj imaju naučiti temeljne obrasce ili distribucije podataka te su usmjereni na razumijevanje načina na koji se podaci generiraju, kako bi generirali nove, slične podatke. Generativni modeli često se temelje na GAN-ovima. Generativne suparničke mreže čine dvije suparničke neuronske mreže: generator i diskriminator. Dvije suparničke neuronske mreže treniraju se istovremeno pri čemu generator generira nove podatke dok diskriminator nastoji razlikovati stvarne od generiranih podataka. U kontekstu super rezolucije generator radi na način da povećava rezoluciju nisko rezolucijske slike i stvara visoko rezolucijske slike koje bi trebale što je moguće više nalikovati stvarnim slikama, potom ih predaje diskriminatoru koji procjenjuje koliko dobro generirane slike odgovaraju stvarnim visoko rezolucijskim ili referentnim slikama. Suparničke mreže uče jedna od druge, generator uči kako stvoriti detalje i teksture, dok diskriminator pomaže u osiguravanju autentičnosti slike. Ovaj proces kao rezultat stvara generator koji proizvodi visoko rezolucijske slike ili super rezolucijske slike iz nisko rezolucijskih ulaznih slika. Na slici 2.2. prikazan je proces treniranja super rezolucijskog algoritma temeljenog na GAN mreži. Proces započinje ulaskom LR slike u mrežu generatora, nakon čega generator obrađuje LR sliku i generira SR sliku. Generirana SR slika uspoređuje se sa referentnom GT slikom kako bi se izračunao gubitak elemenata slike, koji mjeri razliku između GT i SR slike. Generirana SR slika zajedno sa GT slikom prosljeđuju se diskriminatoru, koji daje ocjenu vjerojatnosti koja pokazuje je li ulazna slika generirana (SR) ili referentna (GT). Diskriminator šalje povratnu informaciju generatoru u obliku suparničkog

gubitka, kojeg generator koristi za poboljšanje performansi generiranja visoko rezolucijskih izlaza. Cilj generatora je proizvesti SR slike koje diskriminator ne može razlikovati od GT slika. Proces uključuje propagaciju unazad (eng. *backpropagation*) izračunatih gubitaka kako bi se ažurirali parametri generatora i diskriminatora. Generator nastoji smanjiti gubitak elemenata slike i suparnički gubitak, dok diskriminator nastoji povećati točnost razlikovanja GT i SR slika.



Slika 2.2. Proces treniranja super rezolucijskog algoritma temeljenog na GAN mreži [3]

Na slici 2.3. prikazana je osnovna arhitektura mreže generatora za super rezoluciju. Proces započinje ulaskom nisko rezolucijske slike u mrežu. Konvolucijski slojevi izvlače značajke iz ulazne slike primjenom filtera, početni konvolucijski sloj izvlači značajke niske razine kao što su rubovi i teksture. Osnovni blokovi obično uključuju nekoliko konvolucijskih slojeva u kombinaciji s nelinearnim aktivacijskim funkcijama (npr. ReLu) i često uključuju konekcije preskakanja (eng. *skip connections*). Osnovni blokovi mogu biti rezidualni blokovi, gusti blokovi ili druge vrste blokova dizajniranih za duboko učenje. U procesu super rezolucije najčešće se koriste RRDB blokovi (eng. *Residual-in-Residual Dense Block*), zbog njihove učinkovitosti u izdvajanju i učenju značajki koje se koriste za rekonstrukciju SR slike iz LR ulaza. RRDB blokovi koriste se za progresivno poboljšanje značajki LR slike prije povećanja rezolucije. Povećanje rezolucije može uključivati brojne tehnike poput interpolacijskih tehnika, transponirane konvolucije, miješanje elemenata slike (eng. *Pixel Shuffle*) ili drugih tehnika dubokog učenja dizajniranih za generiranje slika visoke rezolucije. Nakon povećanja rezolucije, dodatni konvolucijski slojevi pročišćavaju značajke, poboljšavaju detalje i generiraju konačnu sliku visoke rezolucije.



Slika 2.3. Osnovna arhitektura mreže generatora za proces super rezolucije [4]

2.2. Pregled postojećih metrika za procjenu kvalitete slika

Kvaliteta slike može se smanjiti zbog izobličenja tijekom snimanja i obrade slike te tijekom transporta i pohrane slike. Izobličenja mogu uključivati šumove, zamućenja i artefakte kompresije. U brojnim područjima poboljšanja slike potrebne su metrike koje brojčano iskazuju kvalitetu slike. Procjena kvalitete slika (IQA) je područje obrade slika čiji je cilj iskazati subjektivnu i objektivnu procjenu kvalitete slika. Svrha procjene kvalitete slika je osigurati automatizirano i kvantitativno mjerenje kvalitete slika. IQA uključuje razvoj algoritama i modela koji mjere percipiranu kvalitetu slika koja je usklađena sa subjektivnom percepcijom kvalitete od strane ljudskog promatrača. U području super rezolucije krajnji cilj je generirati slike koje su vizualno što jasnije i realističnije ljudskom oku, stoga je ključno da su IQA metrike usklađene s ljudskom percepcijom kvalitete slike. Ako IQA metrika ne odražava ljudsku percepciju, može doći do netočnog ocjenjivanja kvalitete slika, što dovodi do slika koje imaju visoke ocjene, ali se ljudima čine vizualno neprivlačnima. Usklađivanje procjene kvalitete sa ljudskom percepcijom, odnosno razumijevanje onoga što ljudi percipiraju kao kvalitetu slike, osigurava da generirane slike koje proizvode super rezolucijski algoritmi uistinu poboljšavaju vizualni doživljaj.

IQA algoritmi analiziraju različite aspekte slika, uključujući vizualni sadržaj, strukturne informacije i artefakte te uzimaju u obzir čimbenike poput oštine, zamućenja, kontrasta, izobličenja i sl. IQA metrike uključuju dvije vrste metrika: metrike ocjene kvalitete bez potrebe za referentnim signalom (engl. *no reference* – NR) i metrike ocjene kvalitete s potpunim pristupom referentnom signalu (engl. *full reference* – FR). Navedene metrike mogu se temeljiti na matematičkim modelima, statistici, tehnikama strojnog učenja ili modelima ljudskog oka [5].

Ako je dostupna referentna slika bez izobličenja (GT) može se koristiti kao referenca za mjerenje kvalitete drugih slika. U tim slučajevima koriste se FR metrike za izravnu usporedbu generirane slike i referentne slike. Primjeri FR metrika uključuju: srednju kvadratnu pogrešku (eng. *Mean-squared error* – MSE), omjer vršnog signala i šuma (eng. *Peak signal-to-noise ratio* – PSNR), indeks strukturne sličnosti (eng. *Structural Similarity Index* – SSIM), indeks strukturne sličnosti s više skala (eng. *Multi-scale structural similarity index* – MS-SSIM), normalizirana udaljenost Laplaceove piramide (eng. *Normalized Laplacian Pyramid Distance* – NLPD), naučena perceptivna sličnost slike (eng. *Learned Perceptual Image Patch Similarity* – LPIPS) i duboka sličnost strukture i tekture slike (eng. *Deep Image Structure and Texture Similarity* – DISTTS).

MSE metrika na jednostavan način računa prosječnu kvadratnu razliku između stvarnih i idealnih vrijednosti elemenata slike, ali nije najbolje usklađena s ljudskom percepcijom kvalitete. U nastavku je prikazana jednadžba (2-1) za MSE izračun:

$$MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (2-1)$$

gdje I_1 i I_2 predstavljaju ulazne slike, dok M i N predstavljaju broj redaka i stupaca u ulaznim slikama. I_1 predstavlja generiranu SR sliku, a I_2 predstavlja GT sliku. Što je MSE vrijednost niža, manja je pogreška tj. odstupanje generirane slike od GT slike.

PSNR izveden je iz srednje kvadratne pogreške i predstavlja omjer maksimalnog intenziteta elemenata slike i snage izobličenja. Poput MSE metrike jednostavno se računa, ali također nije dovoljno usklađena s percipiranom kvalitetom. U nastavku je prikazana jednadžba (2-2) za PSNR izračun [5]:

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right) \quad (2-2)$$

gdje R predstavlja maksimalnu vrijednost intenziteta elemenata slike. Veća PSNR vrijednost ukazuje na bolju kvalitetu slike I_1 .

SSIM metrika kombinira lokalnu strukturu slike, svjetlinu i kontrast u jednu lokalnu ocjenu kvalitete. Budući da je ljudski vizualni sustav dobar u opažanju strukture, SSIM se više slaže sa subjektivnom percepcijom kvalitete nego MSE i PSNR [5]. SSIM može bolje razlikovati slike koje

su percipirane kao vrlo različite, čak i ako imaju slične vrijednosti MSE-a i PSNR-a. SSIM temelji se na izračunu izraza osvjetljenja, izraza kontrasta i strukturalnog izraza. Ukupni SSIM indeks je multiplikativna kombinacija navedena tri izraza. U nastavku je prikazana jednačba (2-3) za SSIM izračun [5]:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2-3)$$

gdje su μ_x i μ_y srednje vrijednosti, σ_x i σ_y standardne devijacije i σ_{xy} unakrsna kovarijanca za dijelove slike x, y. C_1 i C_2 su regulacijske konstante koje se koriste kako bi se izbjegla nestabilnost pri računanju SSIM-a kad su srednje vrijednosti ili standardne devijacije vrlo blizu nuli. SSIM vrijednosti kreću se u rasponu od -1 do 1. U većini praktičnih primjena, SSIM vrijednosti obično su u rasponu između 0 i 1, gdje 1 označava savršeno podudaranje između dvije slike (slike su identične), a 0 ukazuje kako nema strukturne sličnosti između slika. Negativne vrijednosti su rijetke i pojavljuju se u slučajevima ekstremnog izobličenja ili inverzije slike.

MS-SSIM proširuje SSIM indeks kombiniranjem informacija o svjetlini na najvišoj razini rezolucije s informacijama o strukturi i kontrastu na nekoliko smanjenih rezolucija (skala). Višestruke skale uzimaju u obzir varijabilnost u percepciji detalja slike [5].

NLPD, LPIPS i DISTIS metrike u usporedbi s tradicionalnim metrikama pružaju bolju korelaciju s ljudskom percepcijom kvalitete slika, što ih čini preferiranim izborom za primjene gdje je važna percepcijska kvaliteta kao što su nadzorne kamere, dijagnostičke slike ili poboljšanje slike u digitalnim medijima. NLPD je metrika za procjenu kvalitete slike koja se koristi za određivanje percepcijskih razlika između dvije slike koristeći normaliziranu *Laplaceovu* piramidu. NLPD tehnika računa korijen srednje kvadratne pogreške između transformiranih reprezentacija referentne (GT) i generirane slike te koristi koncepte iz vizualne percepcije, uključujući oduzimanje lokalne srednje vrijednosti i normalizaciju kontrasta. Navedeni koncepti ključni su za povećanje osjetljivosti na percepcijski važne promjene u slici [6]. LPIPS metrika koristi duboke značajke iz konvolucijskih neuronskih mreža za procjenu percepcijske sličnosti između slika na način koji bolje odgovara ljudskoj percepciji. Osim prednosti koju donosi s percepcijskom mjerljivosti, također se može koristiti kao funkcija gubitka u treniranju. Implementacija LPIPS funkcije gubitka pomaže u razvoju modela koji generira vizualno bolje rezultate, što je posebice važno kod raznih primjena super rezolucije. LPIPS također radi s različitim arhitekturama mreža, što ga čini fleksibilnom metodom za različite primjene u računalnom vidu [7]. DISTIS je metrika

koja procjenjuje kvalitetu slike kombinirajući sličnost strukture i teksture. DISTS model treniran je kako bi se maksimalno približio ocjenama kvalitete koju daju ljudski promatrači, stoga omogućuje bolje razumijevanje onoga što ljudi percipiraju kao kvalitetu slike. Model se temelji na korištenju konvolucijskih neuronskih mreža za pretvorbu slika u višerazinske reprezentacije koje odražavaju i strukturu i teksturu slika. DISTS također pokazuje značajnu otpornost na blage lokalne i globalne geometrijske transformacije koje su često neprimjetne ljudskom vizualnom sustavu [8].

Ako referentna slika (GT) nije dostupna koriste se NR metrike. Ove metrike računaju ocjene kvalitete na temelju očekivane statistike slike [5]. Najčešće korištene metrike uključuju: procjenitelj prostorne kvalitete bez referentne slike (eng. *Blind/Referenceless Image Spatial Quality Evaluator* - BRISQUE), procjenitelj kvalitete prirodnih slika (eng. *Natural Image Quality Evaluator* – NIQE), procjenitelj kvalitete slike temeljen na percepciji (eng. *Perception based Image Quality Evaluator* - PIQE), MOS (eng. *Mean Opinion Score* – MOS), i NIMA (eng. *Neural Image Assessment* – NIMA). BRISQUE model treniran je na skupu podataka koji sadrži slike za koje su poznate subjektivne ocjene kvalitete dobivene od strane ljudskih promatrača. Samim time BRISQUE model ocjenjuje kvalitetu slike na način koji odgovara ljudskoj procjeni kvalitete. Nedostatak ove metode je ograničenje na evaluaciju kvalitete slika sa istom vrstom oštećenja koja se nalazi na slikama iz skupa podataka za treniranje. Suprotno BRISQUE-u, NIQE i PIQE metrike ne koriste poznate subjektivne ocjene kvalitete slika, ali procjenjuju kvalitetu slika u korelaciji s percepcijskom kvalitetom slike. NIQE model treniran je na prirodnim, neoštećenim slikama, ali može mjeriti kvalitetu slika s proizvoljnim oštećenjem. PIQE u većini slučajeva radi slično kao NIQE, ali ne zahtijeva istrenirani model. PIQE algoritam procjenjuje oštećenja po blokovima i mjeri lokalnu varijancu primjetno oštećenih blokova kako bi izračunao ocjenu kvalitete. PIQE prvo računa MSCN koeficijente za svaki element slike, a zatim dijeli sliku u 16x16 blokove i identificira blokove s visokom prostornom aktivnošću. Algoritam zatim procjenjuje i klasificira blokove na temelju prisutnosti neželjenih artefakata i šumova te u konačnici generira PIQE rezultat kao prosjek rezultata oštećenih blokova. MOS je metoda pri kojoj grupacija ljudi daje subjektivne ocjene kvalitete slika od 1 do 5, gdje je 5 izvrsna kvaliteta, a 1 iznimno loša kvaliteta. NIMA je model baziran na konvolucijskim neuronskim mrežama koji efikasno ocjenjuje slike uz visoku prilagođenost s ljudskim percepcijama kvalitete i estetike, umjesto da se fokusira na prosječnu ocjenu mišljenja. Osim za procjenu estetske kvalitete slika koristi se i za procjenu tehničke kvalitete slika (šum, zamućenost, kompresijski artefakti) [9]. Sve NR metrike obično nadmašuju FR metrike u smislu slaganja sa subjektivnom ljudskom ocjenom kvalitete [5].

2.3. Pregled postojećih rješenja za super rezoluciju

U radu [10] prvi puta je predložen okvir za super rezolucijsku generativnu suparničku mrežu (eng. *Super Resolution Generative Adversial Network – SRGAN*) sposoban za stvaranje realističnih prirodnih slika s faktorom povećanja 4×. Rad donosi ključnu inovaciju u perceptivnoj funkciji gubitka koja kombinira suparnički gubitak i gubitak sadržaja, ističući perceptivnu sličnost umjesto sličnosti elemenata slike. Koristi se duboka rezidualna mreža (*ResNet*) s povezanim prijelazima, izbjegavajući tradicionalnu optimizaciju srednjom kvadratnom pogreškom (eng. *Mean Square Error - MSE*). Umjesto gubitka sadržaja temeljenog na MSE-u, koristi se gubitak izračunat na značajkama VGG (eng. *Very Deep Convolutional Network - VGG*) mreže, postižući poboljšanja u foto-realističnim rezultatima. Rješenje je testirano na tri poznata skupa podataka za super rezoluciju slika, a to su Set5 [11], Set14 [12] i BSD100 [13]. Za usporedbu korišteni su standardne IQA (eng. *Image Quality Assesment – IQA*) metrike: PSNR i SSIM te subjektivna procjena kvalitete putem MOS (eng. *Mean Opinion Score*) testa, u kojem su 26 ocjenjivača dali ocjene od 1 do 5 različitim verzijama slika koje su dobivene primjenom predloženog algoritma super rezolucije. Kombinacija VGG gubitka s višim razinama značajki (SRGAN-VGG54) ističe se kao najuspješnija varijanta u usporedbi s drugim postojećim SRGAN i SRResNet modelima razvijenim prije opisanog SRGAN modela. Iako rezultati imaju visok PSNR i SSIM, pokazalo se kako ove standardne metrike ne uspijevaju točno procijeniti kvalitetu slike s obzirom na ljudsku percepciju kvalitete. Stoga autori uvode opsežna MOS testiranja, pomoću kojih potvrđuju visoku kvalitetnu rekonstrukciju SR slika za velike faktore povećanja rezolucije (4x). Autori naglašavaju kako je fokus SRGAN rada postići perceptivnu kvalitetu SR sliku, a ne računalnu učinkovitost. SRGAN rad [10] i danas se koristi kao polazna točka u istraživanju novih algoritama i nastavlja oblikovati smjer istraživanja u ovom području.

U radu [4] predstavljen je poboljšani SRGAN (eng. *Enhanced SRGAN – ESRGAN*) gdje su struktura mreže, funkcija gubitka i diskriminator temeljito poboljšani u odnosu na SRGAN iz [3]. Mrežna arhitektura poboljšana je uvođenjem RRDB-a te je uklonjena normalizacija serija (eng. *batch normalization - BN*). BN slojevi normaliziraju značajke koristeći srednju vrijednost i varijancu serije tijekom treniranja te procijenjenu srednju vrijednost i varijancu cijelog trening skupa podataka tijekom testiranja. Kada se statistike skupova podataka za treniranje i testiranje jako razlikuju, BN slojevi mogu proizvesti neželjene artefakte. Autori dokazuju kako će BN slojevi vjerojatnije proizvesti artefakte kada je mreža dublja i trenirana pod GAN okvirom. Dokazano je da uklanjanje BN slojeva povećava performanse modela i smanjuje složenost računanja i upotrebe memorije u zadacima orijentiranim na povećanje PSNR-a, uključujući zadatke super rezolucije i

uklanjanje zamućenja u slikama. Na temelju opažanja da više slojeva uvijek može poboljšati performanse modela, predloženi RRDB blok koristi dublju i složeniju strukturu od izvornog rezidualnog bloka u SRGAN-u. Predloženi RRDB blok ima „rezidualno u rezidualnom“ strukturu, gdje se rezidualno učenje koristi na različitim razinama. Uz poboljšanu arhitekturu uvode se i tehnike poput rezidualnog skaliranja i manje inicijalizacije parametara koje sprječavaju nestabilnost i omogućavaju lakše treniranje. Osim poboljšane arhitekture generatora, poboljšan je i diskriminator koji autori temelje na relativističkom GAN-u. Za razliku od standardnog diskriminatora u SRGAN-u, koji procjenjuje vjerojatnost da je ulazna slika x stvarna, relativistički diskriminator pokušava predvidjeti vjerojatnost da je stvarna slika x_r relativno realističnija od generirane (lažne) slike x_f . Na taj način diskriminator procjenjuje relativnu stvarnost umjesto apsolutne. Autori također poboljšavaju perceptivni gubitak \mathcal{L}_{percep} korištenjem značajki prije aktivacijskih slojeva. Za treniranje je korišten DIV2K [14] i Flickr2K [15] skup podataka, dok je za evaluaciju korišten Set5, Set14, BSD100, Urban100 [16] i PIRM skup podataka. Prema rezultatima testiranja ESRGAN nadmašuje SRGAN [10] u smislu vizualne kvalitete, oštine i detalja te predstavlja značajan korak naprijed u području super rezolucije. Zbog izvrsnih performansi i izuzetnih rezultata ESRGAN se i danas koristi kao polazna točka i osnovni model u brojnim novim super rezolucijskim algoritmima.

Tradicionalne metode super rezolucije, poput [10] i [4], koriste bikubičnu interpolaciju kako bi stvorili slike visoke rezolucije (HR) iz slika niske rezolucije (LR). Metode super rezolucije trenirane na takvim skupovima podataka često ne daju dobre rezultate kada se primjene na stvarne visoko oštećene nisko rezolucijske slike. Kako bi stvorili model super rezolucije koji je otporan na oštećenja prisutna u stvarnim slikama, u radu [17] predložen je novi okvir degradacije za generiranje realističnih LR/HR parova slika za treniranje. Navedeni okvir uključuje najčešće vrste degradacije slika iz stvarnih slika lica, te kao takav uključuje procjenu realističnih filtera zamućenja, procjenu distribucije šuma i artefakte JPEG kompresije. Predložena RWSR (eng. *Real World Super Resolution* – RWSR) metoda [17] primjenjuje se na slike lica proizvoljnih rezolucija, ali je također primjenjiva i u drugim domenama. Za procjenu realističnih filtera zamućenja autori koriste KernelGAN metodu. KernelGAN metoda predstavlja treniranje GAN mreže za smanjenje ulazne slike uz očuvanje distribucija segmenata (eng. *patch*) slike kroz različite skale. Nadalje, realistični filteri zamućenja procjenjuju se iz slika iz odabranog skupa podataka za trening (X domena) kako bi se formirao skup filtera koji se može koristiti za degradaciju HR slika u Y domeni. Za degradaciju s realističnim šumom koristi se metoda izdvajanja segmenata šuma iz slika u X domeni. Pri izdvajanju šuma pretpostavlja se da se približan segment šuma može dobiti iz

oštećene slike izdvajajući područje sa slabom pozadinom i zatim oduzimajući srednju vrijednost. Na taj način stvara se skup segmenata šuma. Proces degradacije uključuje nasumično biranje filtera zamućenja i šuma iz navedenih skupova te primjenu JPEG kompresije s jačinom kompresije 30. Kao osnovni model koriste ESRGAN [4], te ga treniraju na parovima LR/HR slika generiranih novim okvirom degradacije. Za razliku od ESRGAN modela koji se fokusira na smanjenje srednje kvadratne pogreške (MSE) između rekonstruirane slike visoke rezolucije (SR) i referentne slike (GT), rad [17] fokusira se na funkcije gubitka koje povećavaju perceptivnu kvalitetu. ESRGAN model koristi nekoliko različitih funkcija gubitaka tijekom treninga. Konkretno, generator koristi suparnički gubitak \mathcal{L}_{adv} u kombinaciji s VGG perceptivnim gubitkom \mathcal{L}_{vgg} i gubitkom piksela ili gubitkom elemenata slike \mathcal{L}_{pix} , dok diskriminator koristi VGG-128 gubitak \mathcal{L}_{vgg} . Kako bi poboljšali perceptivnu kvalitetu, autori uvode izmjene u generatoru i diskriminatoru. Umjesto VGG-128 gubitka korištenog u diskriminatoru, uvodi se PatchGAN gubitak \mathcal{L}_{patch} . PatchGAN pokazuje se kao detaljniji tip diskriminatora koji procjenjuje autentičnost manjih segmenata slike (npr. 16x16 elemenata slike) za razliku od VGG-128 diskriminatora koji procjenjuje autentičnost cijele slike. Računa se prosjek gubitaka svih segmenata slike i predaje generatoru. Nadalje, dolazi do izmjene u generatoru gdje se VGG gubitak mijenja za LPIPS perceptivni gubitak \mathcal{L}_{lrips} . Rezultati dovode do slika s manje šuma i bogatijim teksturama u usporedbi s korištenjem VGG gubitka. Glavni razlog je taj što je VGG mreža trenirana za klasifikaciju slika, dok je LPIPS metrika trenirana za ocjenjivanje i procjenu perceptivne sličnosti između slika na temelju ljudske percepcije sličnosti. Za treniranje i testiranje korišteno je više različitih skupova podataka. U domeni X korišten je Chokepoint [18] skup podataka koji sadrži slike 29 različitih osoba snimljenih s tri kamere u stvarnim uvjetima video nadzora. Chokepoint skup podataka korišten je isključivo za procjenu filtera zamućenja i procjenu distribucije šuma za okvir degradacije (72282 slika) te testiranje (3805 slika), a ne za treniranje SR modela. U Y domeni za treniranje slika korištena su tri skupa podataka: 571 slika iz SiblingsDB [19], 8040 slika lica iz Radboud Faces Database [20] i 5000 nasumično odabranih slika iz FFHQ [21] skupa podataka. Kako bi dobili što raznovrsniji skup podataka, primjenjuje se bikubična interpolacija za 25, 50 i 75% na sve slike u Y domeni, nakon čega se primjenjuje okvir degradacije kako bi dobili LR/HR parove slika za trening. Za evaluaciju modela na umjetno oštećenim slikama lica koristi se prvih 1000 slika iz FFHQ skupa podataka, parovi LR/GT slika dobivaju se pomoću okvira degradacije. Za procjenu kvalitete autori koriste tri tradicionalne IQA metrike: PSNR, SSIM i MS-SSIM. Budući da ove metrike nisu idealne za procjenu perceptivne kvalitete, evaluacija se provodi i na tri novije metrike usmjerene na ljudsku procjenu kvalitete slika: NLPD, LPIPS i DISTSS. Za evaluaciju rezultata

dobivenih testiranjem SR modela na slikama s nadzorne kamere, nije bilo moguće koristiti tradicionalne metrike zbog nedostataka GT slika. Kvaliteta dobivenih SR slika se stoga određivala pomoću sljedećih IQA metrika: NIQE, BRISQUE, PIQE, NQRM i PI. U slučaju super rezolucije navedene metode često slabo koreliraju s ljudskom procjenom pa autori protokol evaluacije nadopunjuju s MOR (eng. *Mean Opinion Rank*) i NIMA procjenom. NIMA je tehnika trenirana na ljudskim ocjenama kvalitete slika te se u radu koristi unaprijed istrenirani model za ocjenjivanje kvalitete slike. Tijekom MOR procjene, sudionici ocjenjuju ukupnu kvalitetu SR slike. Usporedbom rezultata dobivenih predloženom metodom u okviru ovog rada s rezultatima SoTA (eng. *State-of-the-Art*) modela dokazano je kako se konvencionalne IQA metrike više fokusiraju na vjernost signala nego na perceptivnu kvalitetu slike, stoga nisu najbolji pokazatelj kvalitete RWSR slika. Metrike poput NIME (za skup bez GT slika) i DISTSS metrike (za skup umjetno oštećenih slika), pokazuju kako je RWSR metoda daleko superiornija u odnosu na postojeće SoTA metode.

U radu [22] predstavljena je metoda za super rezoluciju slike lica koristeći jedinicu za pozornost lica (eng. *Facial Attention Unit* – FAU) s ciljem stvaranje rezidualne mreže s prostornom pozornošću (eng. *Spatial Attention Residual Network* – SPARNet). Predloženi SPARNet algoritam sastoji se od tri modula: modul za smanjenje rezolucije, modul za izvlačenje značajki i modul za povećanje rezolucije. Svaki od modula sastoji se od više FAU jedinica. Korištenje rezidualnih blokova pokazalo se uspješnim u tradicionalnim metodama super rezolucije i u metodama za super rezoluciju lica, stoga autori dodaju mehanizam prostorne pozornosti zajedno s rezidualnim blokovima. Mehanizam prostorne pozornosti omogućuje mreži da se usredotoči na ključne značajke lica, što pomaže mreži da bolje nauči lokalne detalje lica. Unutar jedinice za pozornost lica predložena je grana koja proširuje originalne rezidualne blokove – grana prostorne pozornosti lica. Sekvencijalnim spajanjem više jedinica (FAU) zajedno, neprestano se poboljšavaju važne značajke za super rezoluciju lica. Unutar grane prostorne pozornosti lica nalazi se blok pješčanog sata (eng. *hourglass block*) s dodatnim konvolucijskim slojem, a grana služi za generiranje mape pozornosti. *Hourglass* blok koristi se u grani pozornosti zbog sposobnosti hvatanja značajki različitih razmjera, dok se u grani za značajke koristi pre-aktivacijski rezidualni blok s PReLU aktivacijskom funkcijom.

SPARNet model treniran je na CelebA [23] skupu podataka. Za HR trening skup odabrane su slike u rezoluciji 128x128 elemenata slike, a LR trening skup dobiven je smanjenjem rezolucije HR trening skupa na rezoluciju 16x16 elemenata slike pomoću bikubične interpolacije. Korišteno je otprilike 179,000 parova LR/HR slika. Istrenirani model je testiran na Helen [24] skupu

podataka te su za evaluaciju korištene tradicionalne FR metrike PSNR i SSIM. Osim navedenih kvantitativnih metrika za evaluaciju rezultata korišten je i model za detekciju ključnih točaka lica FAN (eng. *Facial Alignment Network*) i metoda mjerenja sličnosti identiteta. Prema usporedbi PSNR i SSIM ocjena s drugim odabranim SoTA metodama, SPARNet pokazuje bolje performanse. Zbog korištenja mehanizma za prostornu pozornost, SPARNet generira slike na kojima su ključne značajke lica bolje očuvane nego u drugim metodama za super rezoluciju lica. SPARNet uspijeva generirati oštrije detalje sličnije onima na GT slikama. Prema rezultatima mjerenja sličnosti identiteta i detekciji ključnih točaka lica, SPARNet također nadmašuje druge metode odabrane za usporedbu u radu. Modeli poput SPARNet-a čiji je glavni cilj smanjenje gubitka elemenata slike L_{pix} , uvijek imaju visoke PSNR i SSIM ocjene, ali u isto vrijeme generiraju pretjerano zaglađene i zamagljene izlazne slike s niskom percepcijskom kvalitetom. Iz tog razloga autori proširuju SPARNet mrežu i predlažu poboljšani algoritam – SPARNetHD.

SPARNetHD proširuje SPARNet korištenjem višestruke diskriminatorske mreže, gdje diskriminatori D1, D2 i D3 ovisno rezoluciji ulaznih slika generiraju slike u super rezoluciji u rezolucijama: 128x128, 256x256 i 512x512 elemenata slike. Za razliku od SPARNet koji je treniran samo s jednom funkcijom gubitka – gubitak elemenata slike \mathcal{L}_{pix} , SPARNetHD treniran je s četiri funkcije gubitke: gubitak elemenata \mathcal{L}_{pix} , suparnički gubitak generatora \mathcal{L}_{GAN_G} i diskriminatora \mathcal{L}_{GAN_D} , gubitak podudaranja značajki \mathcal{L}_{fm} i percepcijski gubitak \mathcal{L}_{pcp} . Korištenjem navedene kombinacije gubitaka postiže se bolja percepcijska kvaliteta izlaznih slika. SPARNetHD model treniran je kako bi povećao rezoluciju stvarnih nisko rezolucijskih slika s ciljem postizanja što bolje percepcijske kvalitete. Slično kao u radu [17], autori stvaraju degradacijski model za dobivanje LR trening skupa iz HR trening skupa. Degradacijski model sastoji se od filtera zamućenja k_q , operacije smanjenja rezolucije s faktorom razmjera s koji je nasumično izabran iz $[\frac{16}{512}, \frac{128}{512}]$, aditivnog bijelog Gaussovog šuma n_δ i JPEG kompresije u rasponu [60, 85]. Za treniranje modela korišten je FFHQ skup podataka koji se sastoji od 70,000 slika lica u rezoluciji 1024x1024 elemenata slike. Sve slike smanjenje su na rezoluciju 512x512 elemenata slike bilinearnom interpolacijom te je tako stvoren HR trening skup, dok je LR trening skup dobiven pomoću degradacijskog modela iz HR trening skupa. Za testiranje je korišten CelebAHQ [25] skup podataka, koji sadrži 30,000 slika lica u rezoluciji 1024x1024 elemenata slike. Sve slike smanjene su na rezoluciju 128x128 elemenata slike pomoću degradacijskog modela. Osim na CelebAHQ skupu podataka, istrenirani model testiran je i na skupu prirodnih nisko rezolucijskih slika prikupljenih s interneta. Za evaluaciju rješenja na CelebAHQ skupu podataka koristi se LPIPS metrika, dok se za evaluaciju rješenja na prirodnom skupu podataka koristi NR metriku FID (eng.

Frechet Inception Distance) jer za prirodan skup ne postoje GT slike. LPIPS metrika, trenirana na ljudskoj procjeni, dokazala je da SPARNetHD generira percepcijski kvalitetnije slike lica od općenitog algoritma za super rezoluciju ESRGAN [4], koji za razliku od SPARNetHD modela ne koristi višestruke diskriminatorske mreže. Performanse SPARNetHD modela uspoređene su s još 3 metode koje su koristile višestruke diskriminatore te se pokazalo kako SPARNetHD daje daleko bolje rezultate od odabranih metoda, kako prema LPIPS, tako i prema FID metrici.

U radu [26] predstavljena je duboka metoda za super rezoluciju lica koja se temelji na suradnji procjene ključnih točaka lica (eng. *landmark estimation*) i super rezoluciji lica. Autori predlažu mrežu koja na ulaznim slikama iterativno provodi metodu generiranja visoko rezolucijskih slika i mapiranje ključnih točaka. Kako bi se poboljšala suradnja između metode super rezolucije i procesa poravnanja lica (eng. *face alignment*), dizajnira se fuzijski modul koji učinkovito integrira navedene procese. Glavna ideja je omogućiti istovremeno i rekurzivno mapiranje ključnih točaka i rekonstrukciju lica sa slike. Mape ključnih točaka točnije se procjenjuju ako postoje visoko kvalitetne ulazne slike, a s pouzdanim mapama proces super rezolucije može generirati bolje visoko rezolucijske slike. Oba procesa međusobno se poboljšavaju i progresivno ostvaruju bolje rezultate. Kako bi se ostvarilo generiranje što realističnijih slika lica, uvodi se suparnički gubitak \mathcal{L}_{adv} te percepcijski gubitak \mathcal{L}_{perc} kako bi se povećala percepcijska kvaliteta generiranih super rezolucijskih slika. Treniranja i evaluacija modela provodi se na CelebA i Helen skupu podataka. Za stvaranje HR trening skupa prvo se koristi OpenFace podatkovni skup [27] za detekciju 68 ključnih točaka te se na temelju detektiranih točaka izrezuju dijelovi lica bez pozadine. Izrežanim područjima se zatim smanjuje rezolucija na 128x128 elemenata slike. LR trening skup dobiva se smanjenjem rezolucije HR skupa na rezoluciju 16x16 elemenata slike pomoću bikubične interpolacije. CelebA skup podataka podijeljen je na 168854 slika za treniranje i 1000 slika za testiranje, dok je Helen skup podataka podijeljen na 2005 slika za treniranje i 50 slika za testiranje. Kvaliteta dobivenih visoko rezolucijskih slika predloženom metodom procjenjuje se pomoću PSNR i SSIM metrika, a NRMSE metoda je korištena za ocjenu točnosti poravnanja lica. U usporedbi s drugim odabranim metodama u radu, kvantitativni i kvalitativni rezultati ukazuju na visoku učinkovitost predložene metode.

U radu [28] predložena je metoda za super rezoluciju lica, koristeći tri glavna pristupa: progresivno treniranje, gubitak pozornosti lica i destilacija metode za poravnanje lica (eng. *Facial Alignment Network – FAN*). Uvodi se gubitak pozornosti lica $\mathcal{L}_{attention}$ kako bi se mreža fokusirala na rekonstrukciju detalja lica oko označenih ključnih točaka lica. Destilacija FAN metode radi se s ciljem generiranja što točnijih mapa koje su prilagođene za super rezoluciju lica

te kako bi se smanjilo vrijeme treniranja. Osim gubitka pozornosti lica, koristi se i MSE gubitak \mathcal{L}_{pixel} , percepcijski gubitak $\mathcal{L}_{feat/i}$, suparnički gubitak \mathcal{L}_{WGAN} i gubitak mape $\mathcal{L}_{heatmap}$. U radu su korišteni CelebA i AFLW skup podataka za treniranje i testiranje. Evaluacija rješenja provedena je korištenjem IQA metrika PSNR, SSIM i MS-SSIM. Eksperimenti na navedenim skupovima podataka su pokazali da predložena metoda, u usporedbi s drugim odabranim metodama u radu, stvara visokokvalitetne slike te precizno rekonstruira detalje lica. U tablici 2.1. prikazana je usporedba opisanih postojećih rješenja na temelju rezolucije LR i HR slika korištenih za treniranje, korištenom trening i test podatkovnom skupu te evaluacijskim metrikama korištenim u radu.

Tablica 2.1. *Usporedba opisanih postojećih rješenja za super rezoluciju*

Algoritam	LR	HR	Trening skup	Test skup	Evaluacijske metrike
SRGAN[10]	24x24 (proizvoljno)	96x96 (proizvoljno)	ImageNet	Set5, Set14, BSD100	PSNR, SSIM, MOS
ESRGAN[4]	npr. 510x350	npr. 2040x1400	Div2k, Flickr2k	Set5, Set14, BSD100, Urban100	PSNR, SSIM, MOS
RWFSR[17]	128x128	512x512	SiblingsDB, Radboud Faces Database, FFHQ	Chokepoint FFHQ	PSNR, SSIM, MS-SSIM, DISTS,LPIPS, NLPD,NIMA, MOR, NIQE, BRISQUE, PIQE, NQRM PI
SPARNet[22]	16x16 128x128	128x128 512x512	CelebA, FFHQ	CelebA, CelebAHQ	PSNR, SSIM, FID, LPIPS
DICFFSR[26]	16x16	128x128	CelebA, Helen	CelebA, Helen	PSNR, SSIM, NRMSE
Progressive FSR[28]	16x16	128x128	CelebA, AFLW	CelebA, AFLW	PSNR, SSIM, MS-SSIM

3. EVALUACIJA POSTOJEĆIH ALGORITAMA I PRIJEDLOG VLASTITOG ALGORITMA ZA POVEĆANJE RAZLUČIVOSTI LICA S DIGITALNIH SLIKA

3.1. Postojeći algoritmi za super rezoluciju i opis korištenog skupa podataka

Kako bi se dobio bolji uvid u rad i performanse algoritama opisanih u poglavlju 2.1., postojeća rješenja testirana su na CelebAHQ [25] skupu podataka. CelebAHQ je visokokvalitetna verzija CelebA [23] skupa podataka, koja se sastoji od 30,000 slika lica poznatih osoba u rezoluciji 1024x1024 elemenata slike. CelebAHQ skup podataka je dostupan na *TensorFlow* službenoj stranici i u drugim rezolucijama (1024x1024, 512x512, 256x256, 128x128, ..., 2x2, 1x1 elemenata slike). Za potrebe ovog rada korišten je skup podataka u rezoluciji 512x512 elemenata slike. Eksperimenti testiranja odvijaju se na 6 *SoTA* super rezolucijskih algoritama s faktorom povećanja 4x. Prvi algoritam predstavlja trenutno najutjecajniji algoritam u području super rezolucije – Real-ESRGAN [4]. ESRGAN model treniran je na DIV2k i Flickr2k skupu podataka, koji sadrže slike različitih scena, okruženja i sadržaja. Iako ESRGAN model nije prilagođen za super rezoluciju slika lica, koristi se kao polazna točka u brojnim algoritmima za super rezoluciju lica te je iz tog razloga esencijalan za usporedbu performansi s drugim algoritmima. Osim ESRGAN algoritma, testiranje je odvijeno na još jednom općenitom algoritmu za super rezoluciju – SRGAN-tensorflow [29], koji se temelji na SRGAN [10] modelu. Ostali modeli uključuju algoritme prilagođene za super rezoluciju na slikama lica: Face-SPARNet [22], FSR (engl. *Face Super-Resolution* – FSR) [30], Progressive FSR (engl. *Progressive Face Super-Resolution* – Progressive FSR) [28] i DICFFSR (engl. *Deep Iterative Collaboration for Face Super-Resolution* – DICFFSR) [26].

Istrenirani modeli testirani su na 1000 slika iz skupa CelebAHQ. Algoritmi ESRGAN, SRGAN-Tensorflow, Face-SPARNet i FSR trenirani su na parovima HR i LR slika u rezolucijama 512x512 i 128x128 elemenata slike, stoga je stvoren testni skup u rezoluciji 128x128 elemenata slike. Progressive FSR i DICFFSR algoritmi trenirani su na parovima LR i HR slika u rezolucijama 32x32 elemenata slike i 128x128 elemenata slike te je za potrebe testiranja navedena dva algoritma stvoren dodatni testni skup u rezoluciji 32x32.

Preuzet je skup u rezoluciji 512x512 elemenata slike, nakon čega je testni skup slika niske rezolucije dobiven je koristeći određeni okvir degradacije, preuzet iz rada [30]. Okvir degradacije simulira scenarije iz stvarnog svijeta u kojem su fotografije različitih razina kvalitete, što je posebice često u području video nadzora i u aplikacijama za prepoznavanje lica. Kako bi se stvorio

testni skup s raznovrsnim slikama niske rezolucije, okvir degradacije koristi nasumično odabrane vrijednosti za smanjenje kvalitete slike.

Primarni programski jezik korišten pri testiranju postojećih rješenja i pri razvoju vlastitog rješenja je *Python*. Python programski jezik široko se koristi u području strojnog i dubokog učenja zbog svoje jednostavnosti, opsežnih biblioteka i aktivne zajednice. Za manipulaciju i obradu slika korištena je *OpenCV* [31] biblioteka. *OpenCV* (eng. *Open Source Computer Vision Library – OpenCV*) je softverska biblioteka otvorenog koda koja sadrži više od 2500 optimiziranih algoritama u području računalnog vida i strojnog učenja. Algoritmi koje pruža *OpenCV* mogu se koristiti za detekciju i prepoznavanje lica, identifikaciju objekata, praćenje pokretnih objekata, izdvajanje 3D modela objekta, povezivanja slika kako bi se proizvela visoka rezolucijska slika cijelog prizora, itd. U okviru super rezolucije, *OpenCV* ima brojne primjene. *OpenCV* može se koristiti za učitavanje i spremanje slika u različitim formatima s različitim kvalitetama te promjenu veličine slika s funkcijama poput `cv2.resize`. Navedene funkcionalnosti korisne su za stvaranje skupova slika za trening koji sadrži slike niske rezolucije i visoke rezolucije. *OpenCV* biblioteka sadrži brojne funkcije za detekciju rubova i uglova koje se mogu koristiti za obradu slika prije primjene algoritma za super rezoluciju, a također nudi i razne filtere poput Gaussovog filtera za smanjenje šuma prije procesa super rezolucije. *OpenCV* također nudi mogućnost korištenja jednostavnih metoda za povećanje rezolucije poput bikubične interpolacije te pruža funkcije za izračun metrika kao što su MSE i PSNR. *OpenCV* podržava integraciju s drugim bibliotekama poput *TensorFlow* ili *PyTorch* biblioteka, što može biti korisno za prethodno ili naknadno procesiranje podataka kada se koriste duboke neuronske mreže za super rezoluciju. *NumPy* se koristi za razne numeričke operacije kao što su rukovanje nizovima, izvođenje množenja matrica i druga matematička izračunavanja potrebna za obradu podataka [32].

Okvir degradacije s kojim se od skupa podataka u rezoluciji 512x512 elemenata slike stvara testni skup podataka implementiran je na sljedeći način. Slike iz CelebAHQ skupa u rezoluciji 512x512 elemenata slike, učitavaju se koristeći *OpenCV* funkciju `cv2.imread`, nakon čega se pomoću `cv2.resize` funkcije linearnom interpolacijom njihova rezolucija smanjuje na 128x128 elemenata slike te se za drugi testni skup rezolucija smanjuje na 32x32 elemenata slike. Postavlja se raspon JPEG kompresije od 15 do 75 te se za svaku sliku kvaliteta smanjuje na temelju nasumičnog izbora jačine kompresije unutar intervala [15, 75]. Degradirane slike spremaju se koristeći `cv2.imwrite` funkciju, pri čemu se slika dodatno degradira koristeći opciju `cv2.IMWRITE_JPEQ_QUALITY` (ili `cv2.IMWRITE_PNG_COMPRESSION` za PNG). Navedena opcija definira finalnu razinu kompresije i kvalitetu slike. Na slici 3.1. prikazan je

programski kod okvira degradacije, dok su na slici 3.2. prikazane nisko rezolucijske slike dobivene okvirom degradacije za dva različita raspona kvalitete (eng. *quality range*). Okvir degradacije primijenjen je na GT slici iz CelebAHQ skupa u rezoluciji 512x512 elemenata slike.

```
all_list = glob('ground_truth_512')
target_size = 128
quality_ranges = [(15, 75)]
output_path = 'data/lr-128'

def saving(path):
    assert '.jpg' in path
    img = cv2.imread(path)
    img = cv2.resize(img, (target_size, target_size))
    for qr in quality_ranges:
        quality = int(random.random() * (qr[1] - qr[0]) + qr[0])
        cv2.imwrite(output_path + '/' + path.split('/')[-1], img,
                    [int(cv2.IMWRITE_JPEG_QUALITY), quality]) #.replace('.jpg', '_%d.jpg' % quality)
```

Slika 3.1. Programski kod okvira degradacije iz [30]



a)

b)

c)

Slika 3.2. Prikaz degradiranih slika pomoću okvira degradacije a) GT slika, b) zadana kvaliteta slike = 65, c) zadana kvaliteta slike = 20

3.2. Rezultati testiranja postojećih algoritama i evaluacija rješenja uz pomoć IQA metrika

U ovom poglavlju napravljena je detaljna kvalitativna i kvantitativna usporedba izlaznih rezultata postojećih super rezolucijskih algoritama. Procjena kvalitete generiranih slika pomoću odabranih algoritama super rezolucije izvršena je uz pomoć IQA metrika.

3.2.1. Evaluacija i analiza rješenja postojećih super rezolucijskih algoritama

U ovom radu za evaluaciju postojećih SR algoritama korištene su tri tradicionalne FR IQA metrike: PSNR, SSIM i MS-SSIM, te DISTS metrika koja se u radovima za super rezoluciju stvarnih slika, poput RWFSR [17], dokazala kao najbolji pokazatelj ljudske procjene perceptivne kvalitete slika.

Iz priloženih rezultata u tablici 3.1. vidljivo je kako su bikubična interpolacija, ESRGAN [4] i Tensorflow-SRGAN [29] metoda vrednovane s najvišim ocjenama prema klasičnim metrikama koje nisu temeljene na ljudskoj procjeni. U usporebi sa svim testiranim algoritmima bikubična interpolacija postiže najbolje PSNR, SSIM i MS-SSIM ocjene, što ukazuje na očuvanje i obnovu detalja te strukturnu sličnost s GT slikama. Bikubična interpolacija također postiže jako dobre ocjene prema DISTS metrici. Međutim, DISTS metrika pokazuje prosječne rezultate za rezultate ESRGAN-a, SRGAN-a što je učestalo kod super rezolucijskih metoda koje su najviše fokusirane na smanjenje gubitka elemenata slike te se pri treniranju neuronske mreže ne koncentriraju na poboljšanje percepcijskog gubitka. Visoki DISTS rezultati ukazuju na manje realistične teksture u usporedbi s GT slikom. Iako je ESRGAN algoritam ocijenjen sa nižim ocjenama u usporedbi sa bikubičnom interpolacijom, kvalitativni rezultati na slici 3.6. pokazuju bolju perceptivnu kvalitetu.

Prema svim metrikama, Face-SPARNet [22] pokazuje slabije performanse u usporedbi s ostalim algoritmima za super rezoluciju te je na slici 3.7. vidljivo kako su neke izlazne SR slike zrcaljenje, što može biti primarni razlog niskih ocjena.

Progressive FSR [28] algoritam pokazuje srednje dobre rezultate prema PSNR i SSIM MS-SSIM metrici. Prema DISTS metrici algoritam je dobio prosječne ocjene, iako je na slici **Pogreška! Izvor reference nije pronađen.**vidljivo kako su lica izobličena i neprepoznatljiva. Na slici **Pogreška! Izvor reference nije pronađen.**prikazane su SR slike algoritma Progressive FSR koje su dobile max i min vrijednosti prema DISTS metrici.

DICFFS [26] metoda dobila je podjednake rezultate kao i Progressive-FSR algoritam, ali na slici 3.7. je vizualno uočljiva najlošija kvaliteta rekonstruirane slike. Uspoređujući ova dva algoritma koji povećavaju rezoluciju slike s 32x32 na 128x128 elemenata slike, Progressive FSR se pokazao kao pouzdaniji algoritam.



LR

SR

a)



LR

SR

b)

Slika 3.3. Prikaz SR izlaza algoritma *Progressive FSR* za LR ulaz a) Prikaz SR slike sa max vrijednosti *DISTS* = 0.12, b) Prikaz SR slike sa min vrijednosti *DISTS* = 0.35

Iako FSR [30] metoda ima prosječne PSNR, SSIM i MS-SSIM ocjene, u usporedbi sa ostalim modelima dobio je najbolje *DISTS* ocjene, što ukazuje na to da je ova metoda najbolja u stvaranju realističnih tekstura i detalja u usporedbi s drugim metodama. Na slici 3.8 vidljivo je kako FSR pruža vizualno najrealističnije i najprihvatljivije rezultate bez neželjenih artefakata koji se nalazi na izlaznim slikama drugih algoritama. Na slici 3.4. nalaze se SR slike koje su dobile pod a) max i pod b) min *DISTS* vrijednosti.



LR

SR

a)



LR

SR

b)

Slika 3.4. Prikaz SR izlaza algoritma FSR za LR ulaz a) Prikaz SR slike sa max vrijednosti $DISTS = 0.13$, b) Prikaz SR slike sa min vrijednosti $DISTS = 0.29$

Na slici 3.5. kvalitativno su uspoređeni SR izlazi bikubične interpolacije i FSR algoritma. Usporedba se provodi kako bi se detaljnije prikazala sposobnost generiranja vizualno privlačnih SR slika lica. Prema rezultatima FR metrika bikubična interpolacija dobila je najbolje PSNR ocjene u usporedbi sa ostalim algoritmima te je dobila i jako visoke DISTS ocjene. Na slici 3.5. predstavljena je pod a) SR slika sa najvišom PSNR ocjenom ($PSNR = 34.29$) i pod b) sa najvišom DISTS ocjenom ($DISTS = 0.12$) generirana uz pomoć bikubične interpolacije. Na istoj slici nalaze se i SR slike dobivene uz pomoć FSR algoritma, koje su dobile niže ocjene prema FR metrikama ($PSNR = 20.07$, $DISTS = 0.15$). Kvalitativna usporedba se provodi kako bi se detaljnije prikazala perceptualna kvaliteta FSR super rezolucijskih izlaza. Priloženi kvalitativni rezultati u generiranju SR slika lica čine ga superiornijim u pogledu perceptivne kvalitete, unatoč nižim rezultatima u drugim metrikama. Zbog toga je u ovom radu kao osnovni model i polazna točka odabran Face Super Resolution (FSR) [30] algoritam.

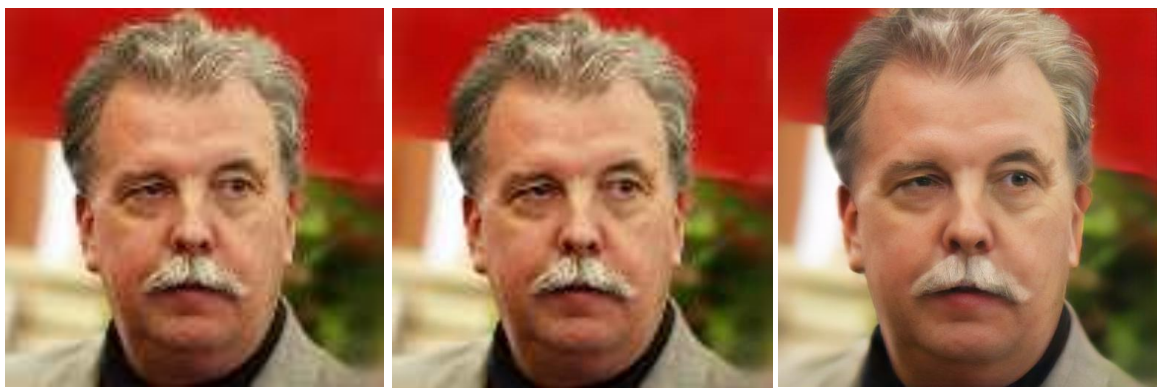


LR

Bikubična interpolacija

FSR

a)



LR

Bikubična interpolacija

FSR

b)

Slika 3.5. Usporedba generiranih SR izlaza algoritma bikubične interpolacije i FSR algoritma

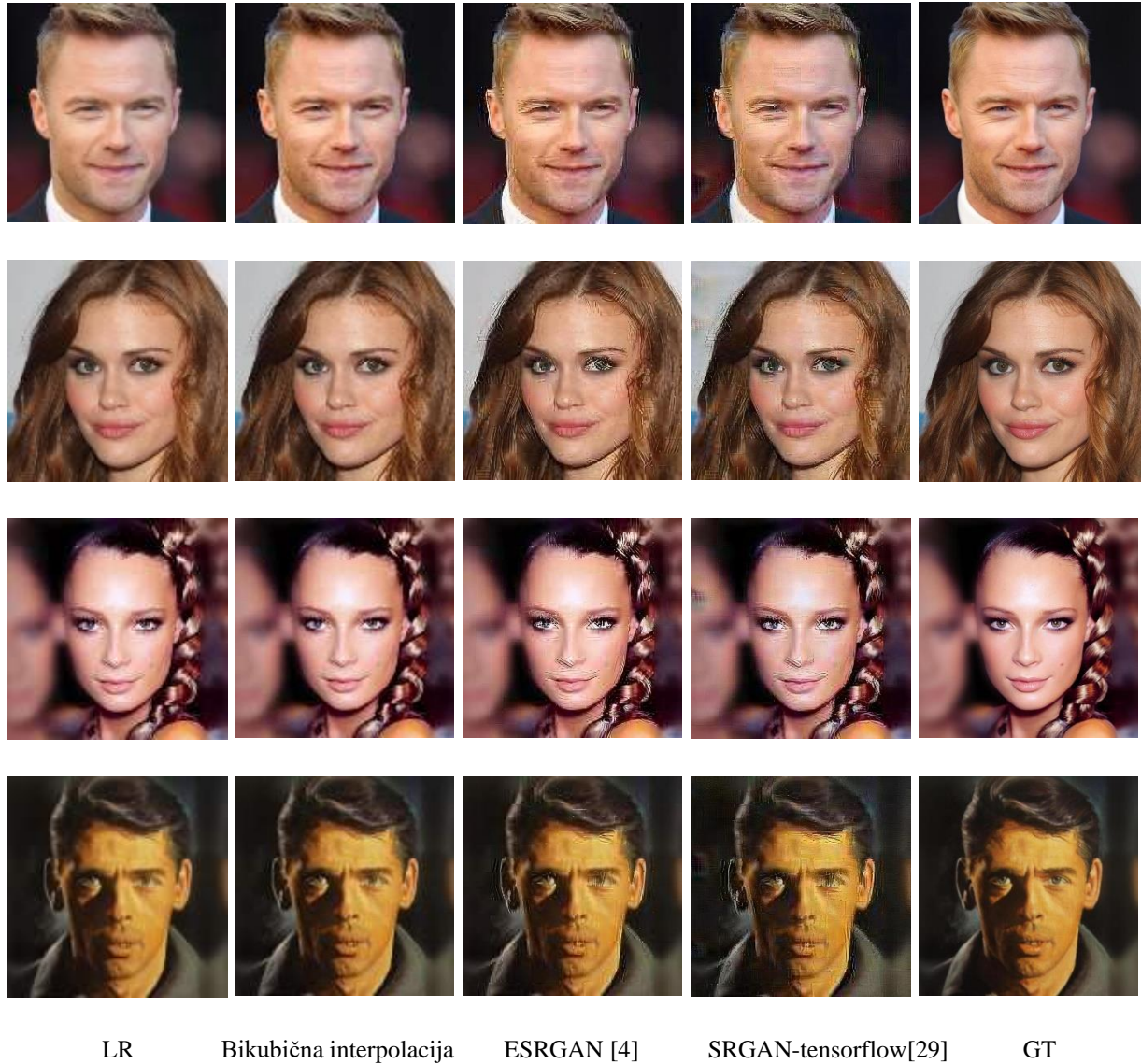
U tablici 3.1. prikazani su kvantitativni rezultati testiranja postojećih super rezolucijskih algoritama na 1000 slika iz CelebAHQ skupa podataka. Rezultati su dobiveni pomoću FR metrika (PSNR, SSIM, MS-SSIM i DISTS). U tablici su dane srednje vrijednosti, standardna devijacija te minimalne i maksimalne vrijednosti SR izlaza svakog algoritma. Strelice prikazane uz imena metrika pokazuju jesu li poželjnije više ili niže vrijednosti ocjena.

Tablica 3.1. Evaluacija super rezolucijskih algoritama na CelebAHQ podatkovnom skupu uz pomoć FR metrika

ALGORITAM	Bikubična interpolacija				ESRGAN [4]			
IQA	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓
\bar{x}	27.48	0.75	0.89	0.19	24.11	0.67	0.86	0.28
σ	1.96	0.07	0.03	0.03	2.26	0.07	0.03	0.03
Min	19.83	0.50	0.79	0.12	13.63	0.44	0.76	0.18
Max	34.29	0.94	0.96	0.28	32.12	0.89	0.95	0.36
ALGORITAM	SRGAN-Tensorflow [29]				FaceSPARNet [22]			
IQA	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓
\bar{x}	22.50	0.57	0.81	0.27	13.76	0.42	0.43	0.37
σ	2.54	0.12	0.07	0.03	3.66	0.11	0.15	0.03
Min	11.58	0.20	0.55	0.21	5.46	0.16	0.0	0.28
Max	28.94	0.84	0.92	0.39	27.06	0.83	0.91	0.47
ALGORITAM	Progressive FSR [28]				DICFFSR [26]			
IQA	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓
\bar{x}	21.17	0.55	0.79	0.21	20.15	0.55	0.79	0.29
σ	1.45	0.07	0.04	0.04	1.84	0.06	0.04	0.03
Min	15.26	0.31	0.64	0.12	15.24	0.36	0.67	0.20
Max	25.85	0.77	0.90	0.35	26.32	0.76	0.89	0.38
ALGORITAM	Face Super-Resolution [30]							
IQA	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓				
\bar{x}	16.00	0.48	0.50	0.18				
σ	2.2	0.09	0.1	0.02				
Min	9.04	0.25	0.26	0.13				
Max	24.17	0.76	0.76	0.29				

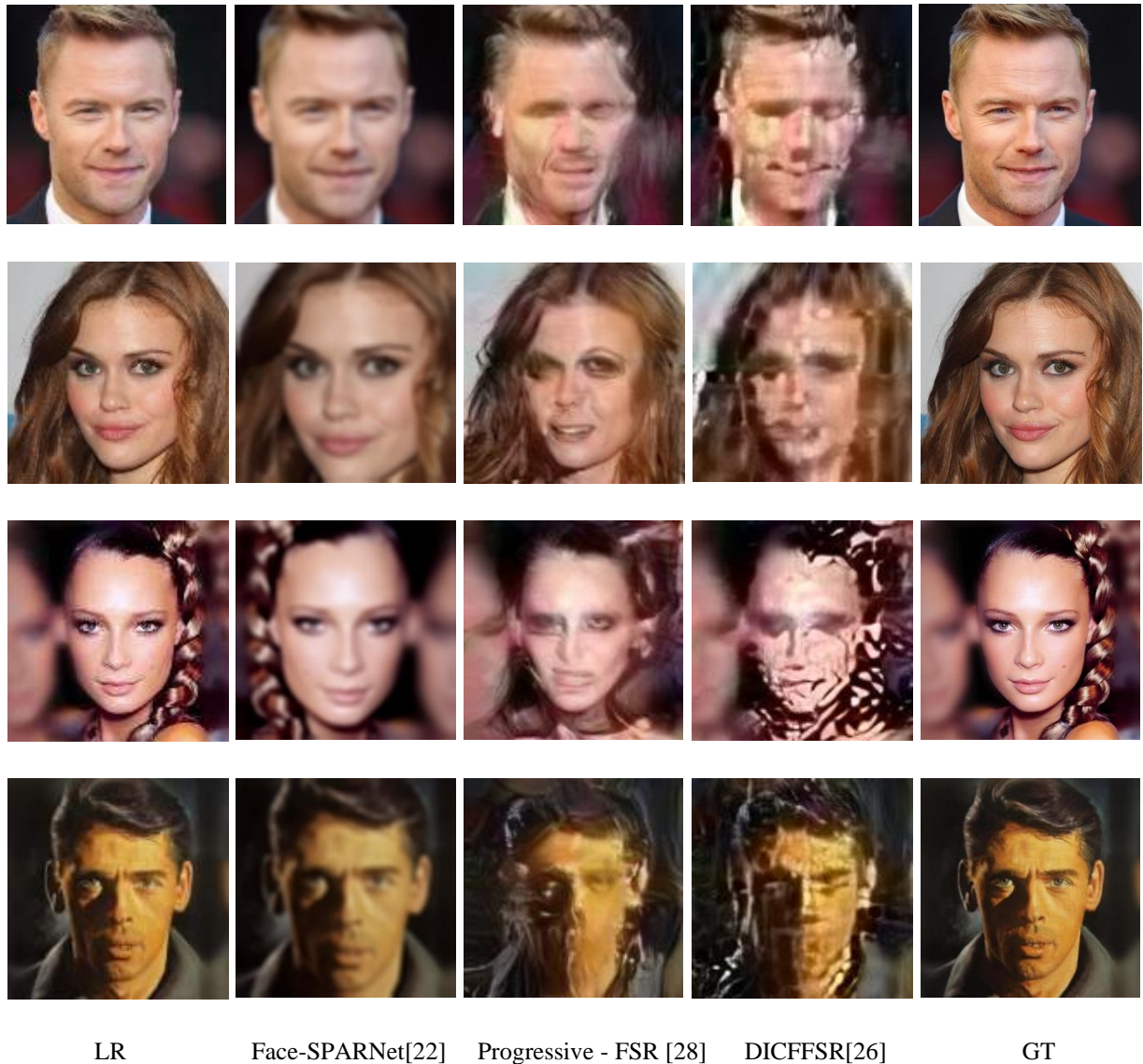
Na slikama 3.6. ,3.7. , i 3.8 prikazane je nekoliko primjera SR slika za ulazne LR slike te odgovarajuća GT slika iz CelebAHQ skupa podataka, s ciljem detaljnije analize super rezolucijskih izlaza testiranih algoritama.

Na slici 3.6. prikazane su SR slike dobivene pomoću algoritma bikubične interpolacije, ESRGAN i SRGAN-Tensorflow algoritma.



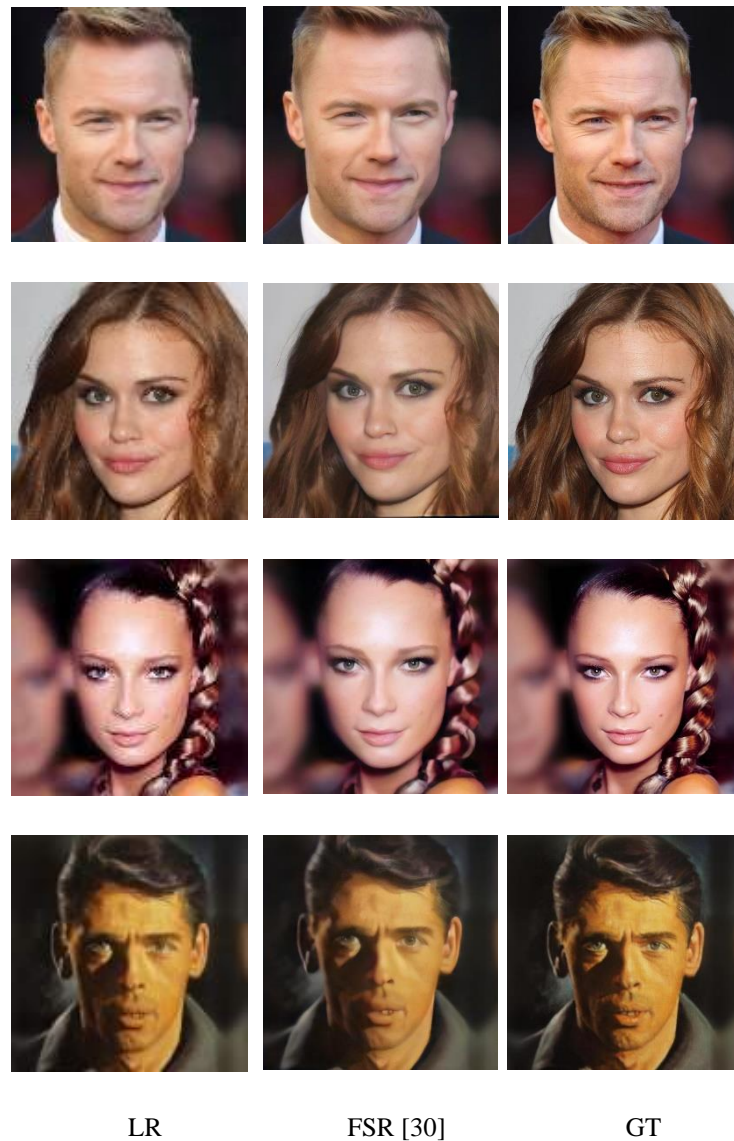
Slika 3.6. Kvalitativni rezultati izlaza super rezolucijskih algoritama (bikubična interpolacija, ESRGAN i SRGAN-tensorflow) za nisko rezolucijski ulaz LR

Na slici 3.7. prikazane su SR slike dobivene pomoću Face-SPARNet, Progressive-FSR i DICFFSR algoritma.



Slika 3.7. Kvalitativni rezultati izlaza super rezolucijskih algoritama (Face-SPARNet, Progressive-FSR i DICFFSR) za nisko rezolucijski ulaz LR

Na slici 3.8. prikazane su SR slike dobivene pomoću Face Super-Resolution algoritma.



Slika 3.8. Kvalitativni rezultati izlaza super rezolucijskog algoritama (*Face Super Resolution*) za nisko rezolucijski ulaz *LR*

Tablica 3.2. Evaluacija super rezolucijskih algoritama na četiri proizvoljno odabrane slike iz CelebAHQ podatkovnog skupa pomoću IQA metrika

ALGORITAM	Bikubična interpolacija				ESRGAN [4]			
SLIKA	Slika 1	Slika 2	Slika 3	Slika 4	Slika 1	Slika 2	Slika 3	Slika 4
PSNR ↑	29.74	27.71	28.47	31.69	25.28	23.13	22.81	28.18
SSIM ↑	0.84	0.66	0.87	0.87	0.75	0.56	0.77	0.83
MS – SSIM ↑	0.93	0.88	0.95	0.93	0.89	0.83	0.84	0.91
DISTS ↓	0.20	0.23	0.21	0.20	0.28	0.29	0.26	0.23
ALGORITAM	SRGAN-Tensorflow [29]				FaceSPARNet [22]			
SLIKA	Slika 1	Slika 2	Slika 3	Slika 4	Slika 1	Slika 2	Slika 3	Slika 4
PSNR ↑	23.96	23.13	23.25	25.29	18.42	12.88	10.57	17.91
SSIM ↑	0.64	0.52	0.75	0.64	0.61	0.31	0.35	0.49
MS – SSIM ↑	0.86	0.81	0.90	0.85	0.68	0.33	0.33	0.48
DISTS ↓	0.28	0.28	0.24	0.28	0.31	0.40	0.32	0.32
ALGORITAM	Progressive FSR [28]				DICFFSR [26]			
SLIKA	Slika 1	Slika 2	Slika 3	Slika 4	Slika 1	Slika 2	Slika 3	Slika 4
PSNR ↑	22.28	21.69	19.29	21.70	21.55	21.67	17.68	21.11
SSIM ↑	0.49	0.51	0.63	0.50	0.64	0.55	0.62	0.57
MS – SSIM ↑	0.84	0.78	0.82	0.75	0.83	0.80	0.83	0.78
DISTS ↓	0.20	0.12	0.21	0.31	0.29	0.31	0.28	0.34
ALGORITAM	Face Super-Resolution [30]							
SLIKA	Slika 1	Slika 2	Slika 3	Slika 4				
PSNR ↑	17.99	18.36	13.09	20.18				
SSIM ↑	0.65	0.43	0.54	0.61				
MS – SSIM ↑	0.65	0.56	0.53	0.60				
DISTS ↓	0.13	0.17	0.20	0.15				

U tablici 3.2. prikazane su vrijednosti dobivene pomoću odabranih FR metrika. Iako je algoritam bikubične interpolacije dobio najbolje rezultate u usporedbi sa ostala dva tradicionalna algoritma za super rezoluciju na priloženim slikama se ipak može uočiti kako ESRGAN i SRGAN Tensorflow algoritam generiraju izoštrnije detalje. Iako su Progressive-FSR i DICFFSR algoritmi dobili bolje ocjene u usporedbi sa FaceSPARNet algoritmom na priloženim slikama su uočljiva visoka oštećenja i izobličenja te su lica na slikama neprepoznatljiva. Iz navedenog se može zaključiti kako je FaceSPARNet algoritam pouzdaniji od Progressive-FSR i DICFFSR algoritma unatoč nižim ocjenama. FSR algoritam generira vizualno najkvalitetnije slike te je moguće uočiti kako se na slikama generiranim uz pomoć FSR algoritma ne uočavaju artefakti koji se nalaze na

izlaznim SR slikama drugih algoritama. Zbog najboljih DISTSS ocjena i izvrsnih kvalitativnih rezultata, FSR algoritam je istaknut kao najbolji algoritam u usporedbi sa ostalim testiranim algoritmima.

3.3. Razvoj vlastite neuronske mreže za super rezoluciju lica

U ovom potpoglavlju detaljno su opisani koraci razvoja vlastite neuronske mreže za super rezoluciju, pri čemu se opisuje arhitektura neuronske mreže, funkcije gubitaka i postupak treniranja mreže.

3.3.1. Alati i programske biblioteke korištene za razvoj neuronske mreže

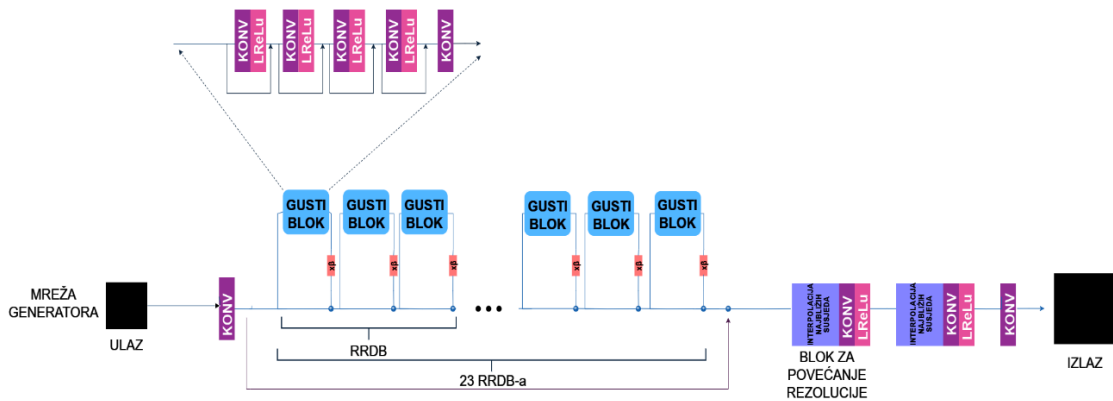
Primarni programski jezik korišten pri razvoju vlastite neuronske mreže je *Python*. Za izgradnju modela neuronske mreže, obradu podataka i treniranje modela korišten je *Python* programski jezik te optimizirana tenzor biblioteka *PyTorch* [33]. *PyTorch* biblioteka pruža potrebne alate za stvaranje slojeva neuronske mreže, definiranje funkcije gubitaka i optimizaciju modela. Konkretno u ovom radu, *PyTorch* biblioteka se koristi za definiranje SRGAN modela, uključujući mrežu generatora i mrežu diskriminatora. Mreže su izrađene korištenjem *PyTorch* modula *nn.Module* i drugih komponenata neuronske mreže koje pruža *PyTorch*. *PyTorch* biblioteka upravlja cijelim procesom treniranja. Korišten je i NVIDIA CUDA alat za ubrzano treniranje mreže na GPU-u [34]. Za praćenje i vizualizaciju procesa treniranja u stvarnom vremenu, korišten je *TensorBoard* alat. *PyTorch*-ov program *SummaryWriter* korišten je za bilježenje metrika treniranja (kao što su gubici i stope učenja) na *TensorBoard* [35].

3.3.2. Osnovni model

Za razvoj vlastitog super rezolucijskog modela u ovom radu kao temelj se koristi Face Super Resolution (FSR) algoritam [30], koji je jedan od SoTA algoritama za percepcijsku super rezoluciju lica s faktorom povećanja 4x. Model se trenira na uparenim LR i HR slikama generiranim s okvirom degradacije opisanim u potpoglavlju 3.1.

Face Super Resolution algoritam kao i brojne druge metode za super rezoluciju lica koristi mrežnu arhitekturu temeljenu na ESRGAN metodi, stoga koristi rezidualne blokove za generiranje visoko rezolucijskih izlaza iz nisko rezolucijskih ulaza. Arhitektura generatora temelji se na RRDBNet mreži (eng. *Residual-in-Residual Dense Block Network* – RRDBNet). RRDBNet arhitektura sastoji se od 23 RRDB bloka. Svaki RRDB blok sadrži tri rezidualna gusta bloka (eng. *Residual Dense Block* – RDB), dok svaki RDB sadrži 5 konvolucijskih slojeva s konekcijama preskakanja i LeakyReLU aktivacijskom funkcijom. Za povećanje rezolucije koristi se metoda

interpolacije najbližih susjeda umjesto PixelShuffle tehnike korištene u ESRGAN arhitekturi mreže. Na slici 3.9. nalazi se grafički prikaz arhitekture mreže generatora.



Slika 3.9. Prikaz arhitekture mreže generatora [30]

Diskriminator koristi modificiranu VGG arhitekturu, konkretno VGG-128 arhitekturu mreže. Mreža diskriminatora se sastoji od 14 konvolucijskih slojeva s *batch* normalizacijom i LeakyReLU aktivacijskom funkcijom. Osim konvolucijskih slojeva koristi i potpuno povezane slojeve. Na slici 3.10. nalazi se grafički prikaz arhitekture mreže diskriminatora.



Slika 3.10. Prikaz arhitekture mreže diskriminatora [30]

3.3.3. Funkcije gubitaka

Generator koristi kombinaciju gubitka elemenata slike (eng. *pixel loss* – \mathcal{L}_{pix} , percepcijskog gubitka (eng. *perceptual loss* – \mathcal{L}_{vgg}) i suparničkog gubitka (eng. *adversial loss* – \mathcal{L}), dok diskriminator koristi suparnički gubitak za treniranje. Ukupni gubitak generatora prikazan je u jednadžbi (3-1):

$$\mathcal{L}_{generator} = \lambda_{pix} \cdot \mathcal{L}_{pix} + \lambda_{vgg} \cdot \mathcal{L}_{vgg} + \lambda_{adv} \cdot \mathcal{L}_{adv} \quad (3-1)$$

gdje je \mathcal{L}_{pix} gubitak elemenata slike, \mathcal{L}_{vgg} percepcijski gubitak i \mathcal{L}_{adv} suparnički gubitak. λ_{pix} , λ_{vgg} i λ_{adv} su parametri skaliranja koji balansiraju doprinos svakog gubitka.

Funkcija gubitka elemenata slike je klasična funkcija koja mjeri razliku između generirane slike i stvarne slike na razini elemenata slike te na taj način osigurava da generirana slika bude što bliža stvarnoj slici na razini pojedinih elemenata slike. Najčešće se koristi MSE ili MAE za izračun ove razlike. Funkcija gubitka elemenata slike također doprinosi ošttrini i preciznosti generirane slike.

Funkcija percepcijskog gubitka pomaže u održavanju perceptivne kvalitete slike te osigurava da generirana slika bude vizualno slična stvarnoj slici u smislu strukture i tekstura, a ne samo na razini elemenata slike. Funkcija percepcijskog gubitka koristi značajke iz unaprijed istrenirane VGG mreže (najčešće VGG19) za mjerenje sličnosti između generirane i stvarne slike.

Funkcija suparničkog gubitka potječe iz suparničkog treniranja generativnih suparničkih mreža (GAN) gdje generator nastoji zavarati diskriminator stvaranjem realističnih slika. Suparnički gubitak poboljšava realističnost generiranih slika stvaranjem uvjerljivih tekstura i detalja.

Cilj diskriminatora je mogućnost razlikovanja stvarnih od generiranih slika. Diskriminator koristi suparnički gubitak za treniranje, koristi se isti suparnički gubitak \mathcal{L}_{adv} kao i za generator, ali se ulazi i izlazi razlikuju. Mreža diskriminatora se trenira tako da predviđa 1 za stvarne slike i 0 za generirane slike. Na ovaj način se osigurava da generator kontinuirano poboljšava kvalitetu generiranih slika.

Kombinacijom ovih gubitaka nastoji se generirati slike koje su točne na razini elemenata slike (\mathcal{L}_{pix}), vizualno slične stvarnim slikama (\mathcal{L}_{vgg}) i realistične i uvjerljive ljudskom oku (\mathcal{L}_{adv}).

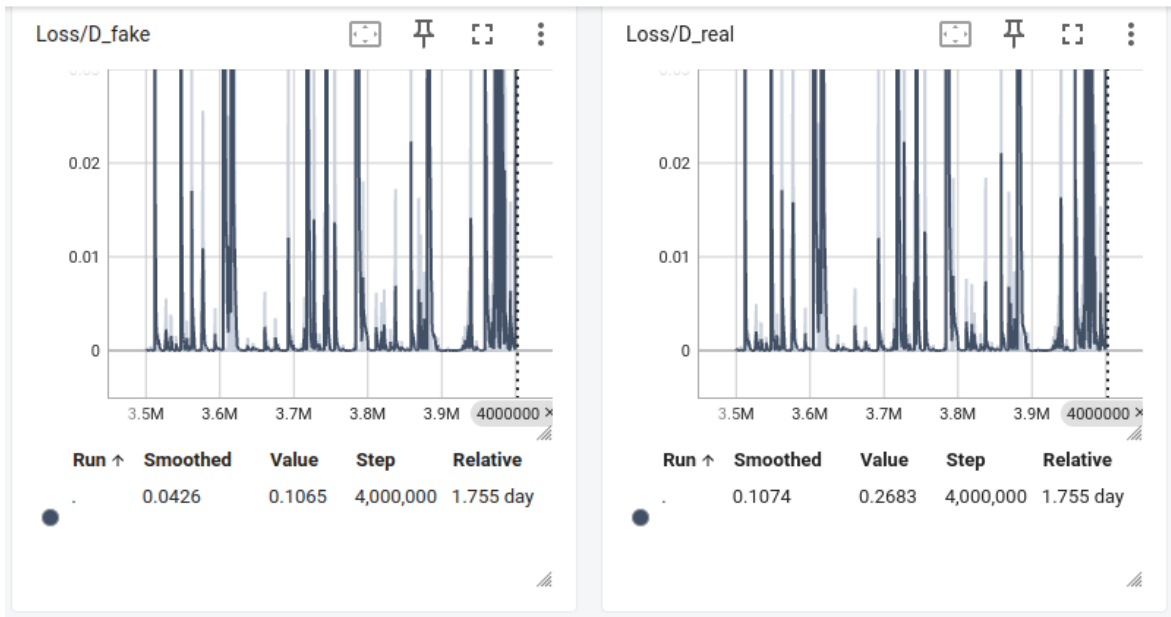
3.3.4. Skup podataka

Skup podataka korišten za treniranje neuronske mreže za super rezoluciju je FFHQ [21] skup podataka. FFHQ skup podataka sastoji se od 70,000 visoko rezolucijskih slika u rezoluciji 1024x1024 elemenata slike. Na slikama se nalaze lica koja su već izrezana i poravnata, odnosno usklađena. Usklađivanje slika osigurava da su ključne značajke lica dosljedno postavljene na svim slikama. Na primjer, u usklađenom skupu podataka oči su u istoj visini na svim slikama, nos je centriran i usta se nalaze u sličnom položaju. Usklađivanje je važno u skupovima podataka koji se koriste za treniranje modela super rezolucije jer osigurava da se model fokusira na učenje relevantnih značajki lica, a ne na način kako su lica smještena unutar slike. FFHQ skup podataka

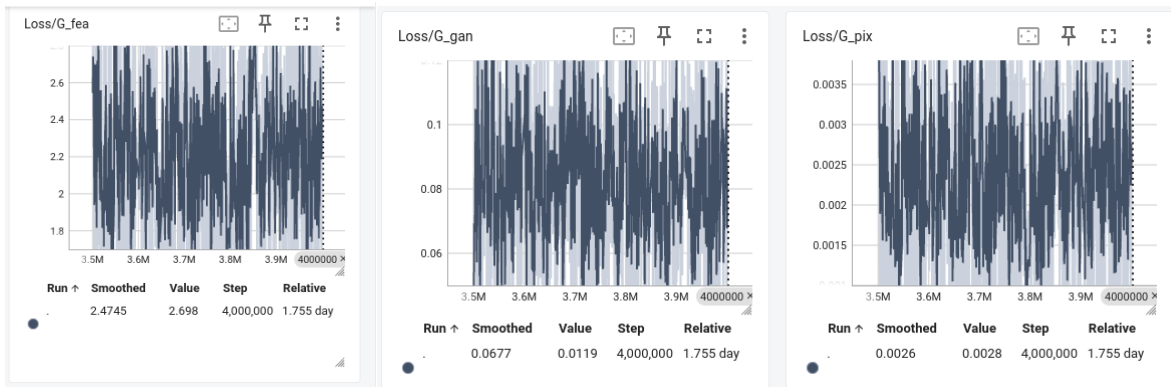
je vrlo prikladan skup podataka za treniranje mreža namijenjenih super rezoluciji lica jer sadrži značajne varijacije dobi, etničke pripadnosti i pozadine slike. Također ima dobru pokrivenost dodacima kao što su naočale, šeširi, modni dodaci, itd. Za potrebe razvoja vlastitog algoritma najprije su iz FFHQ kreirani parovi HR i LR slika. HR slike u rezoluciji 512x512 elemenata slike, dobivene su bilinearnom interpolacijom originalnih slika. LR slike u rezoluciji 128x128 elemenata slike, dobivene su koristeći okvir degradacije predstavljen u poglavlju 3.1. Ukupno je kreirano 70,000 parova slika.

3.3.5. Treniranje neuronske mreže

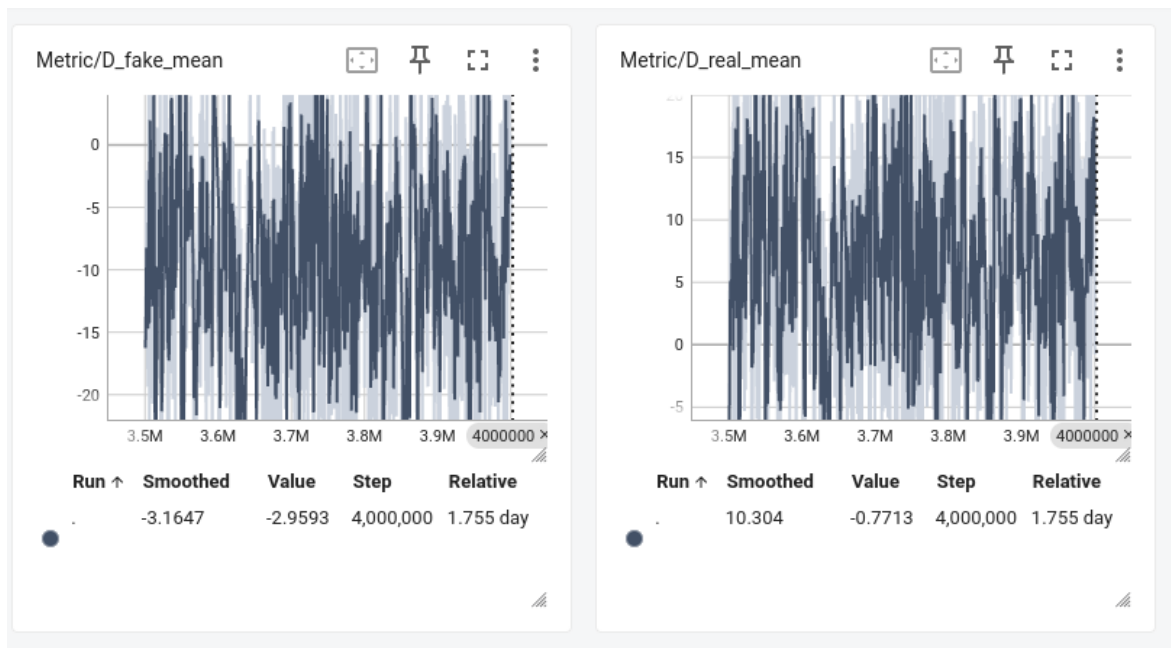
Tok treniranja započinje s inicijalizacijom modela i parametara. Inicijalizira se generator i diskriminator nakon čega se postavlja Adam optimizator s početnim stopama učenja za generator i diskriminator te se dodaje planer stope učenja (eng. *learning rate scheduler*) za prilagodbu stope učenja tijekom treniranja. Trening modela odvija se kroz unaprijed definirani broj epoha. Tijekom svake epohe, podaci se predaju modelu u serijama, rade se predikcije te se postepeno podešavaju parametri mreže. Korišteni parametri treniranja uključuju broj iteracija $niter = 4,000,000$, veličinu serije $batch_size = 2$ te stope učenja za generator i diskriminator $lr_G, lr_D = 1e-5$. Za Adam optimizator postavlja se $beta1_G$ i $beta1_D = 0.9$, te $beta2_G$ i $beta2_D = 0.99$. Korištene su već navedene funkcije gubitaka iz djela 3.3.2., a težine gubitaka postavljene su na sljedeći način: za gubitak piksela $pixel_weight = 1e-2$, za percepcijski gubitak $feature_weight = 1$ te ta suparnički gubitak $gan_weight = 5e-3$. Ukupan broj epoha izračunat prema iteracijama i veličini serije je 115 epoha. Tijekom treniranja provodi se i validacija modela. *TensorBoard* alat se koristi za vizualizaciju metrika treniranja. *TensorBoard* [35] omogućuje praćenje gubitaka, stope učenja, te generiranje i vizualizaciju rezultata treniranja i validacije. Svaki log zapis se pohranjuje u direktorij definiran u argumentima. Na slikama 3.11., 3.12. i 3.13. se nalazi *TensorBoard* vizualizacija procesa treniranja neuronske mreže za super rezoluciju.



Slika 3.11. TensorBoard vizualizacija gubitaka diskriminatora



Slika 3.12. TensorBoard vizualizacija gubitaka generatora



Slika 3.13. TensorBoard vizualizacija metrika treniranja

Na slici 3.11. grafički su prikazane vrijednosti gubitaka za diskriminator prilikom predviđanja je li slika lažna (D_{fake}) ili stvarna (D_{real}). $Loss/D_{fake}$ gubitak trebao bi se smanjivati tijekom vremena, što ukazuje na to da diskriminator postaje sve bolji u prepoznavanju lažnih (generiranih) slika. $Loss/D_{real}$ gubitak također bi se trebao s vremenom smanjivati, kako bi se pokazalo da diskriminator poboljšava svoju sposobnost prepoznavanja stvarnih slika. Na priloženim grafovima je uočljivo kako vrijednosti gubitaka osciliraju, što je tipično za treniranje GAN mreže. Vidljivi oštri skokovi također mogu ukazivati na nestabilnost u treningu. Međutim, ova nestabilnost uobičajena je u GAN mrežama, gdje su generator i diskriminator u neprekidnoj borbi kako bi nadmudrili jedan drugoga. Ako diskriminator postane prejak, lako će razlikovati prave od lažnih slika, što može zaustaviti učenje generatora. Iz grafa je vidljivo kako je D_{fake} gubitak nešto manji u usporedbi s D_{real} gubitkom, što sugerira da je diskriminatoru lakše identificirati generirane slike kao lažne. Na oba grafa vidljive su niske vrijednosti gubitaka i relativna stabilnost pri kraju, što ukazuje na to da diskriminator dolazi do stanja gdje je sposoban s konzistentnošću razlikovati lažne slike od stvarnih.

Na slici 3.12. grafički su prikazane vrijednosti gubitaka generatora za tri različita gubitka: percepcijski gubitak G_{fea} , suparnički gubitak G_{gan} i gubitak elemenata slike G_{pix} . G_{fea} percepcijski gubitak predstavlja koliko dobro generator čuva značajke visoke razine prilikom generiranja slika. Ovaj gubitak pomaže osigurati da generirane slike ne izgledaju samo vizualno

slično ciljanim slikama, već održavaju semantički i percepcijski integritet. Graf pokazuje velike oscilacije, što ukazuje na to da se generator bori s održavanjem dosljedne kvalitete značajki.

G_{gan} predstavlja suparnički gubitak generatora. Manji suparnički gubitak sugerira da je generator sve bolji u zavaravanju diskriminatora da misli da su lažne slike stvarne. Oscilacije na grafu ukazuju na kontinuirano učenje, ali također naglašavaju izazove s kojima se generator suočava u stalnom zavaravanju diskriminatora. Oscilacije također ukazuju na to da se generator neprestano prilagođava kako bi prevario diskriminator.

G_{pix} predstavlja gubitak elemenata slike generatora, obično L1 ili L2 gubitak, koji mjeri koliko su generirane slike blizu stvarnih slika na razini piksela. Graf gubitka elemenata slike pokazuje najmanje vrijednosti, ali i dalje varira. Stalna velika varijabilnost na grafu sugerira da iako se generator poboljšava u smislu kontradiktornih gubitaka, bori se s proizvodnjom dosljednih slika visoke vjernosti na razini elemenata slike.

Na slici 3.13. grafički su prikazane srednje vrijednosti metrika koje predstavljaju prosječnu izlaznu ocjenu koju diskriminator dodjeljuje slikama koje su lažne ili stvarne. D_{fake_mean} idealno bi trebao biti nizak, što ukazuje na to da diskriminator prepoznaje generirane slike kao lažne. Vrijednosti na grafovima pokazuju značajne promjene oko negativnih vrijednosti što sugerira da je diskriminator siguran pri identificiranju lažnih slika. D_{real_mean} vrijednosti trebale bi biti visoke, pokazujući da diskriminator pouzdano identificira stvarne slike kao stvarne. Promjene na grafu su prilično značajne te ukazuju da izvedba diskriminatora na stvarnim slikama znatno varira.

Izražene promjene u vrijednostima na svim grafovima sugeriraju da je treniranje GAN mreže prilično nestabilno, što je uobičajeno u GAN mrežama zbog njihove suparničke prirode, ali može biti štetno za konačnu kvalitetu slike. Iz svega navedenog može se zaključiti kako se performanse generatora poboljšavaju, ali se generator i dalje bori održati dosljednu kvalitetu u različitim mjerama. Diskriminator pokazuje različitu razinu pouzdanosti u svojim predviđanjima, što znači da diskriminator još uvijek uči, ali još nije dosegao stabilno stanje performansi. Ovi grafovi ističu izazove u treniranju GAN mreža, posebice za zadatke kao što je super rezolucija, gdje su važni i točnost elemenata slike i percepcijska kvaliteta na visokoj razini. Super rezolucija predstavlja računalno zahtjevan proces treniranja neuronskih mreža, što može uzrokovati nestabilnost u procesu treniranja. Iako je na grafovima vidljivo da proces treniranja napreduje, vidljive oscilacije pokazuju da bi trening mogao imati korist od dodatnih tehnika stabilizacije. Zahtjevan proces treniranja primarno je posljedica velike količine memorije potrebne za obradu

visoko rezolucijskih slika iz HR skupa podataka. Osnovni FSR model na kojem se temelji ova mreža bio je optimiziran za treniranje na čak 8 GPU-ova. Korištenje većeg broja GPU-ova omogućava korištenje većeg broja slika u svakoj seriji (*batch size* = 32), što pridonosi stabilnijem i efikasnijem treniranju. Međutim, za razvoj vlastitog algoritma za super rezoluciju lica nije bilo pristupa tako velikom broju GPU resursa. Na raspolaganju je bio GPU s 12GB slobodne memorije, što je dozvoljavalo *batch size* = 2. Ovakav *batch size* ne može učinkovito generalizirati značajke iz velikog skupa podataka, što dovodi do varijabilnosti u učenju i manje kvalitetnih rezultata. Kako bi se kompenzirao *batch size*, povećan je broj iteracija na 4,000,000, čime je postignuto 115 epoha treniranja. Osim toga, smanjena je stopa učenja i implementirana je prilagodba stope učenja i *mixed precision* metoda. Nedostatak memorije onemogućio je povećanje *batch size*-a, što je rezultiralo većim oscilacijama u gubitcima prikazanim na grafovima iako su implementirane dodatne tehnike kako bi se pospješio proces učenja.

3.4. Izrada aplikacije za super rezoluciju i implementacija aplikacije na NVIDIA Jetson Nano

U ovom potpoglavlju detaljno je opisan postupak izrade aplikacije za super rezoluciju s ciljem poboljšanja prikaza lica osoba prisutnih u video signalu krajnjem korisniku. Aplikacija je razvijena s ciljem obrade videozapisa u stvarnom vremenu, koristeći tehnike detekcije lica i super rezolucije pomoću neuronskih mreža. Aplikacija je implementirana i optimizirana za rad na NVIDIA Jetson Nano ugradbenom računalnom sustavu.

3.4.1. Zahtjevi na aplikaciju

U ovom djelu detaljno se obrađuju zahtjevi na aplikaciju dizajniranu za detekciju lica u stvarnom vremenu i naknadnu primjenu super rezolucije kako bi se poboljšala kvaliteta slika lica. Ključne funkcije i značajke koje aplikacija mora sadržavati navedene su u nastavku.

- Učitavanje videozapisa

Primarni zahtjev na aplikaciju je omogućiti korisniku učitavanje video datoteka iz direktorija. Nakon što korisnik odabere opciju "*Load Video*" otvara se dijaloški okvir koji korisniku omogućuje pretraživanje direktorija i odabir željenog videozapisa. Nakon pronađenog videozapisa od interesa, korisnik ga učitava klikom na gumb "*Open*".

- Učitavanje *live* kamere

Aplikacija bi također trebala podržavati učitavanje video prijenosa uživo. Kada korisnik odabere opciju "*Start Video*", kamera na sustavu NVIDIA Jetson Nano počinje snimati prijenos uživo.

- Video reprodukcija i kontrola video reprodukcije

Korisnik bi unutar aplikacije trebao imati mogućnost upravljanja učitanim videozapisima ili prijenosom s kamere. Kada korisnik učitava željeni videozapis iz direktorija, videozapis se prikazuje u grafičkom korisničkom sučelju. Korisnik ima mogućnost zaustavljanja videozapisa i ponovnog pokretanja videozapisa pomoću opcije "*Pause/Resume Video*". Osim navedene opcije, korisnik može koristiti klizač za lakšu orijentaciju u videozapisu. Također, korisnik u bilo kojem trenutku *live* prijenosa kamere može odabrati opciju "*Pause/Resume Video*".

- Detekcija lica

Primarni zahtjev na aplikaciju je omogućiti detekciju lica u stvarnom vremenu iz video prijenosa s kamere ili iz unaprijed snimljenih videozapisa. Detekcija lica izvodi se samo kada je videozapis zaustavljen kako bi se smanjila latencija. Kada je videozapis zaustavljen pomoću opcije "*Pause/Resume Video*" i ako je lice detektirano unutar trenutnog okvira (eng. *frame*), oko detektiranih lica u trenutnom okviru će se prikazati granični okviri (eng. *bounding box*) te postotak pouzdanosti detekcije.

- Super rezolucija

Kada se videozapis pauzira i lica se detektiraju u određenom okviru, korisnik dobiva obavijest da je videozapis zaustavljen i da može kliknuti na bilo koje detektirano lice kako bi primijenio super rezoluciju. Ako korisnik pokuša primijeniti super rezoluciju prije pauziranja videozapisa, prikazat će se poruka koja ga upućuje da najprije pauzira videozapis.

Korisnik može primijeniti super rezoluciju na dva načina:

- 1) Klikom na opciju "*Process Frame*": algoritam super rezolucije automatski će se primijeniti na sva detektirana lica u trenutnom *frame*-u
- 2) Klikom na određeno lice : ako je u *frame*-u detektirano više lica, korisnik može odabrati određeno lice izravnim klikom na njega kako bi povećao njegovu rezoluciju

Ako je lice odabrano klikom, prikazat će se poruka s koordinatama odabranog lica i započet će se obrada slike. Ako u trenutnom okviru nisu detektirana lica, informativna poruka će obavijestiti korisnika da unutar okvira nema detektiranih lica. Također, ako su lica detektirana

unutar okvira, ali korisnik klikne izvan graničnog okvira bilo kojeg detektiranog lica, poruka će obavijestiti korisnika da nijedno lice nije odabrano i uputiti ga da klikne unutar granica okvira lica.

- Pohrana SR slika

Generirane visoko rezolucijske slike (512x512 elemenata slike), zajedno s odgovarajućim nisko rezolucijskim slikama (128x128 elemenata slike), trebaju se pohraniti na unaprijed definiranu lokaciju.

Ispunjavanjem ovih zahtjeva, aplikacija će biti prikladna za različite slučajeve upotrebe, uključujući poboljšanje snimaka nadzornih kamera usmjerenih na lica.

3.4.2. Alati korišteni za razvoj aplikacije

U ovom djelu opisani su programski jezik, alati i biblioteke korištene za izradu aplikacije. Kako bi se ispunili svi zahtjevi navedeni u dijelu 3.4.1., aplikacija je izrađena korištenjem *Python* programskog jezika i koristi nekoliko biblioteka, uključujući *PyQt* za izradu grafičkog korisničkog sučelja, *OpenCV* za obradu videa i detekciju lica te *PyTorch* za integraciju modela super rezolucije. Aplikacija je razvijena unutar *Visual Studio Code*-a.

Aplikacija je u potpunosti prilagođena za rad na NVIDIA Jetson Nano ugradbenoj platformi [36]. NVIDIA Jetson Nano je jednokartično računalo razvijeno od strane NVIDIA-e koje pruža snagu moderne umjetne inteligencije. Omogućava paralelno pokretanje višestrukih neuronskih mreža i obradu podataka s nekoliko senzora visoke rezolucije istovremeno. Široko je korišten u projektima koji zahtijevaju obradu videozapisa u stvarnom vremenu, računalni vid i strojno učenje zbog GPU-a i četverojezgrenog ARM Cortex-A57 procesora. Pri razvoju aplikacije korišten je GStreamer kanal (eng. *pipeline*) za snimanje videa s modula kamere spojenog preko MIPI-CSI sučelja. *GStreamer* [37] je multimedijски okvir koji se koristi za upravljanje audio i video tokovima podataka (eng. *stream*). Prikladan je za ugradbene računalne sustave poput NVIDIA Jetson Nano jer omogućuje učinkovitu obradu videozapisa visoke razlučivosti i neometan tok podataka. Za kameru je korišten modul Raspberry Pi Camera Module V2, popularan zbog kompatibilnosti s NVIDIA Jetson Nano platformom i jednostavnoj integraciji s GStreamer *pipeline*-om, što omogućuje snimanje i obradu videa u stvarnom vremenu.

PyQt5 [38] predstavlja Python sučelje za *Qt*, jednu od najpopularnijih GUI (engl. *Graphical User Interface* – GUI) biblioteka. *Qt* je skup višeplatformskih C++ biblioteka koje implementiraju API-je visoke razine za pristup mnogim aspektima modernih desktop i mobilnih sustava. *PyQt5* je mješavina programskog jezika *Python* i biblioteke *Qt*. Implementiran je kao više

od 35 proširenih modula i omogućuje korištenje *Pythona* kao alternativnog jezika za razvoj aplikacija na svim podržanim platformama uključujući *iOS* i *Android*. *PyQt5* također može biti ugrađen u aplikacije temeljene na C++ jeziku kako bi korisnicima tih aplikacija omogućio poboljšanje ili konfiguraciju funkcionalnosti tih aplikacija.

OpenCV [31] biblioteka koristi se za upravljanje videozapisima iz datotečnog sustava i prijenosom kamere uživo. U aplikaciji je također korišten *cv2.dnn* modul duboke neuronske mreže za učitavanje i pokretanje prethodno istreniranog modela detekcije lica. Pri obradi slike, koriste se *OpenCV* funkcije za promjenu rezolucije, pretvorbu formata slike i crtanje graničnih okvira oko detektiranih lica. Obradeni okviri pretvaraju se u format prikladan za prikaz unutar *PyQt* korisničkog sučelja pomoću *OpenCV* biblioteke.

Uz *PyQt* i *OpenCV*, korištena je *PyTorch* biblioteka [33]. Pomoću *PyTorch* biblioteke učitava se, prilagođava i pokreće model super rezolucije. *PyTorch* upravlja tenzorskim proračunima i transformacijama potrebnim za unos slika u neuronsku mrežu.

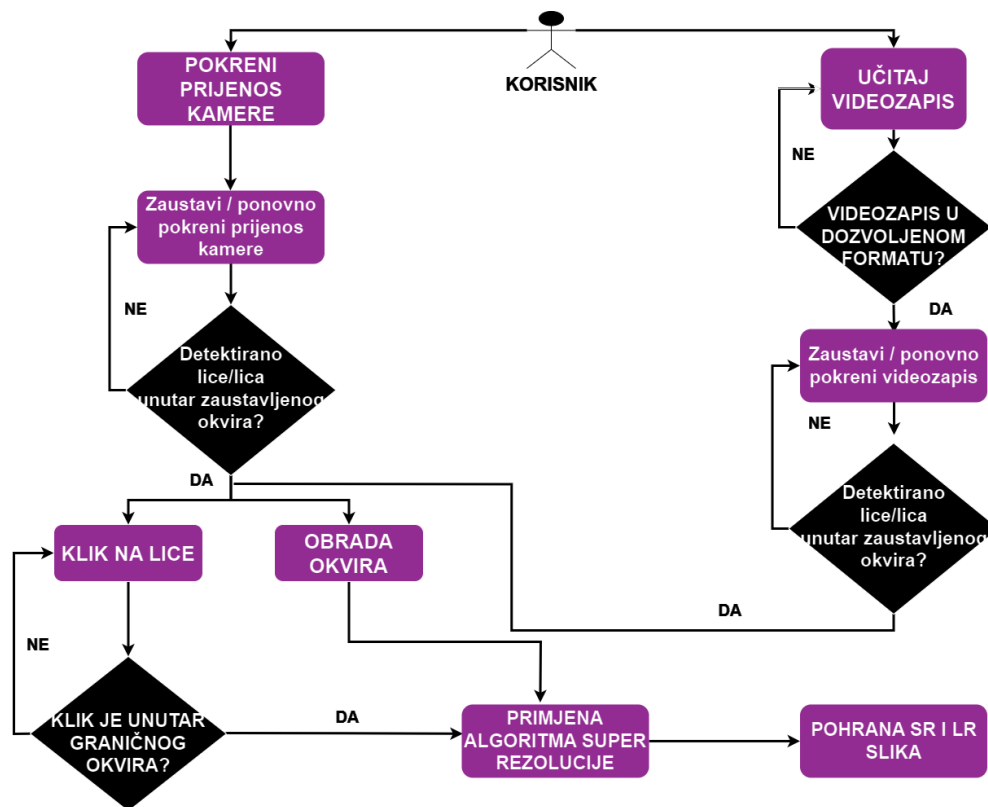
Uz biblioteku *PyTorch*, korištena je biblioteka *torchvision* koja je dio *PyTorch* projekta [30]. *Torchvision* biblioteka pruža razne skupove podataka, transformacije i modele za zadatke računalnog vida izgrađene na *PyTorch*-u. Pri razvoju aplikacije korišten je *torchvision* modul *transforms* za prethodnu obradu slika prije nego se unesu u SRGAN model. Prethodna obrada slika uključuje pretvaranje slika u tenzore i normalizaciju slika.

PIL (Pillow) biblioteka je *Python* slikovna biblioteka za obradu slika [39]. Omogućuje otvaranje, rukovanje i spremanje raznih formata slikovnih datoteka. *PIL* biblioteka korištena je za učitavanje i manipuliranje slikama prije nego što prođu kroz neuronsku mrežu, također se koristi za pohranu izlaznih slika koje generira SRGAN model. Omogućava pretvaranje *OpenCV* slika (koje su u BGR formatu) u RGB format prije nego što ih obradi SRGAN model. *PIL* upravlja pretvorbom slika u format kompatibilan s *PyTorch* tenzorima, olakšavajući proces super rezolucije.

3.4.3. Predložena arhitektura aplikacije

U ovom dijelu opisana je arhitektura aplikacije. Aplikacija je strukturirana u nekoliko ključnih komponenti: grafičko korisničko sučelje (eng. *Graphical User Interface* – GUI), modul za obradu videa, modul za detekciju lica, modul za super rezoluciju i modul za prikaz rezultata. Arhitektura aplikacije je modularna, što omogućava jednostavno održavanje i nadogradnju funkcionalnosti. Kako bi se ispunili svi zahtjevi navedeni u dijelu 3.4.1., aplikacija je izrađena

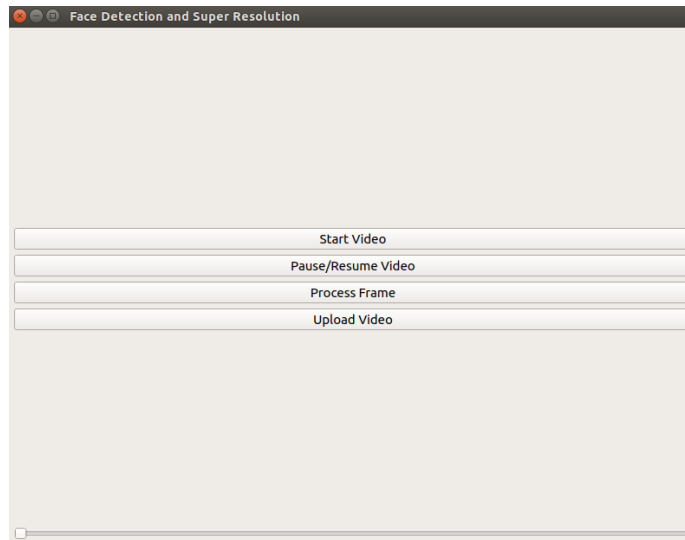
korištenjem *Python* programskog jezika i koristi nekoliko biblioteka, uključujući *PyQt* za izradu grafičkog korisničkog sučelja, *OpenCV* za obradu videa i detekciju lica te *PyTorch* za integraciju modela super rezolucije. Na slici **Slika 3.14.** nalazi se funkcionalni blok dijagram aplikacije za super rezoluciju lica.



Slika 3.14. Funkcionalni blok dijagram aplikacije za super rezoluciju lica

3.4.4. Implementacija grafičkog korisničkog sučelja aplikacije

GUI se sastoji od područja za prikaz videa, kontrolnih gumbi (*Start Video*, *Pause/Resume Video*, *Process Frame*, *Load Video*) i klizača za navigaciju kroz video. Područje prikaza slike omogućuje korisnicima da kliknu na detektirana lica kako bi pokrenuli proces super rezolucije na odabrano lice. Neposredno ispod područja prikaza slike prikazuju se informativne poruke namijenjene za korisnika, kao što je upućivanje korisnika da pauzira videozapis prije odabira lica za obradu. Na slici 3.15. prikazan je dizajn GUI-a aplikacije.



Slika 3.15. Dizajn grafičkog korisničkog sučelja aplikacije za super rezoluciju lica

Grafičko korisničko sučelje izrađeno je koristeći *PyQt5* alat te je implementirano u skladu sa zahtjevima na aplikaciju. *PyQt5* biblioteka ključna je za stvaranje i upravljanje korisničkim sučeljem u ovoj aplikaciji. Upravlja svim zahtjevima za korisničko sučelje, od prikaza glavnog prozora i upravljanja gumbima do interakcije s korisnikom. U nastavku su opisani svi korišteni *PyQt5* moduli za razvoj grafičkog korisničkog sučelja aplikacije.

- *QApplication* – osnovni modul za bilo koju *PyQt* aplikaciju. Upravlja postavkama cijele aplikacije te se u programskom kodu ove aplikacije koristi za inicijalizaciju aplikacije i pokretanje petlje događaja koja čeka interakcije korisnika.
- *QMainWindow* – pruža glavni prozor aplikacije s trakom izbornika, statusnom trakom i *dock widgetima*. U ovom programskom kodu, *QMainWindow* služi kao glavni prozor aplikacije na kojem su smješteni svi ostali *widgeti* (poput gumba i oznaka).
- *QPushButton* – *widget* koji predstavlja gumb u korisničkom sučelju. Koristi se za izradu gumba na koje se može kliknuti kao što su na slici 3.7. "Upload Video", "Start Video", "Pause/Resume Video" i "Process Frame" gumb. Svaki gumb povezan je sa specifičnom funkcijom koja se izvršava kada se gumb pritisne.
- *QLabel* – *widget* koji prikazuje tekst ili slike u korisničkom sučelju. Unutar programskog koda ove aplikacije koristi se za prikaz videozapisa ('image_label') i prikaz informativnih poruka ('result_label').

- *QVBoxLayout* – koristi se za formaciju podređenih *widget*-a prema vertikalnom rasporedu (jedan ispod drugoga). Unutar ove aplikacije, koristi se za vertikalni poredak video zaslona, gumba i klizne trake.
- *QWidget* – osnovna klasa za sve GUI objekte u biblioteci *PyQt* te može služiti kao spremnik za druge *widgete*. U ovom programskom kodu *QWidget* se koristi kao središnji widget koji obuhvaća sve podređene *widgete* i njihov vertikalni raspored (*QVBoxLayout*).
- *QSlider* – *widget* koji nudi kliznu traku, omogućavajući korisniku podešavanje vrijednosti pomicanjem klizača. U ovom programskom kodu se koristi kao horizontalni klizač za kontrolu video reprodukcije, dopuštajući korisniku da traži različite okvire u videu.
- *QFileDialog* – dijalog koji korisniku omogućuje odabir datoteka ili direktorija iz datotečnog sustava. U ovoj aplikaciji se koristi za otvaranje dijaloškog okvira za odabir i učitavanje video datoteka.
- *QTimer* – pruža mogućnost izvršavanja funkcije nakon određenog vremenskog intervala ili više puta u pravilnim vremenskim intervalima. U ovom programskom kodu, *QTimer* se koristi za povremeno ažuriranje okvira i za upravljanje vremenom prikazivanja poruka u korisničkom sučelju.
- *Qt* – polje imena koje sadrži razne oznake koje se koriste u *PyQt*-u. U ovom programskom kodu koristi se za postavljanje orijentacije klizača (*'Qt.Horizontal'*) i za postavljanje svojstava poravnanja i prikaza.
- *pyqtSignal* – koristi se za stvaranje prilagođenih signala u *PyQt*-u. U ovoj se aplikaciji *pyqtSignal* koristi u klasi *ClickableLabel* za emitiranje signala kada korisnik klikne na sliku, omogućujući interakciju s detektiranim licima u okviru.
- *QImage* – klasa za rukovanje i manipuliranje slikama u *PyQt*-u. U programskom kodu se koristi za pretvaranje *OpenCV* slikovnih podataka (cv2 format) u format koji se može prikazati u *PyQt* GUI-u (*QImage* format)
- *QPixmap* – prikaz slike izvan zaslona koji se može koristiti za prikaz slika u *PyQt* widgetima. Koristi se u kombinaciji s *QImage* klasom za prikaz okvira na *QLabel widgetu* (*image_label*).

3.4.5. Implementacija modula za obradu videa

Za učitavanje, pokretanje i obradu videozapisa korištene su biblioteke *OpenCV*, *PyQt* i *GStreamer*. Kao što je već navedeno, modul *QFileDialog* iz *PyQt*-a koristi se kako bi korisnicima

omogućio odabir video datoteke iz datotečnog sustava. Korisnik može kliknuti na gumb "Upload Video" koji pokreće funkcija `upload_video`. Ova funkcija koristi `QFileDialog` za otvaranje dijaloškog okvira u kojem korisnik može odabrati video datoteku s ekstenzijama `.mp4`, `.avi`, `.mov` ili `.mkv`. Nakon što je datoteka odabrana, njezina se putanja u obliku niza predaje funkciji `load_video`, koja koristi `OpenCV` klasu `cv2.VideoCapture` za učitavanje videozapisa. Video okviri se zatim obrađuju u petlji, dopuštajući interakciju u stvarnom vremenu kao što je pauziranje i primjena super rezolucije. Dijaloški okvir se pokreće nakon što korisnik pritisne gumb "Upload Video". Na slici 3.16. prikazan je programski kod koji obuhvaća funkcije `upload_video` i `load_video`.

```
def upload_video(self):
    options = QFileDialog.Options()
    options |= QFileDialog.DontUseNativeDialog
    videoFileName, _ = QFileDialog.getOpenFileName(self, "QFileDialog.getOpenFileName()", "",
                                                "Video Files (*.mp4 *.avi *.mov *.mkv)", options=options)
    if videoFileName:
        self.load_video(videoFileName)

def load_video(self, video_path):
    if self.capture is not None:
        self.capture.release()
        self.timer.stop()

    self.capture = cv2.VideoCapture(video_path)
    total_frames = int(self.capture.get(cv2.CAP_PROP_FRAME_COUNT))
    self.video_slider.setMaximum(total_frames - 1)
    self.timer.start(20)
```

Slika 3.16. Prikaz funkcija `upload_video` i `load_video`

Ako korisnik pritisne gumb "Start Video", pokreće se funkcija `start_video`. Ova funkcija postavlja `GStreamer` cjevovod (eng. *pipeline*) kojeg definira `gst_str` niz i koristi `cv2.VideoCapture` klasu za početak video prijenosa s kamere. Niz `gst_str` definira `GStreamer pipeline` određujući izvor kamere, rezoluciju, format i broj okvira u sekundi. Ako se kamera uspješno otvori, video okviri obrađuju se u petlji, dopuštajući pauziranje video prijenosa i primjenu detekcije lica i super rezolucije. `OpenCV` u kombinaciji s `GStreamer` okvirom prikazuje video *stream*-ove izravno s kamere. Uz `OpenCV` i `GStreamer`, koristi se i `PyQt`-ov `QTimer` za povremeno osvježavanje i prikaz snimljenih video okvira s kamere u korisničkom sučelju. Prijenos video *stream*-a s kamere započinje nakon što korisnik pritisne gumb "Start Video". Na slici 3.17. nalazi se programski kod koji obuhvaća `start_video` funkciju.

```

def start_video(self):
    # GStreamer pipeline niz za prijenos stream-a kamere
    gst_str = (
        'nvarguscamerasrc ! '
        'video/x-raw(memory:NVMM), width=(int)1920, height=(int)1080, format=(string)NV12, framerate=(fraction)30/1 ! '
        'nvvidconv ! video/x-raw, format=(string)BGRx ! '
        'videoconvert ! video/x-raw, format=(string)BGR ! appsink'
    )

    # Otpuštanje prethodnog prijenosa ako postoji
    if self.capture is not None:
        self.capture.release()
        self.timer.stop()

    # Stvaranje video capture objekta sa GStreamer pipeline-om
    self.capture = cv2.VideoCapture(gst_str, cv2.CAP_GSTREAMER)

    if self.capture.isOpened():
        self.capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        self.capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
        self.timer.start(20) # Start the timer to periodically update frames
        print("Camera opened successfully")
    else:
        print("Failed to open the camera")
        self.result_label.setText("Failed to open the camera")

```

Slika 3.17. Prikaz *start_video* funkcije

3.4.6. Implementacija modula za detekciju lica

Modul za detekciju lica korišten pri razvoju aplikacije obuhvaća *cv2.dnn*, *OpenCV* modul za duboko učenje, u kombinaciji s unaprijed istreniranim *Caffe* modelom. Pomoću funkcije *cv2.dnn.readNetFromCaffe* učitava se model za detekciju lica iz dvije ključne datoteke: *prototxt_file* koja sadrži arhitekturu neuronske mreže i *caffemodel_file* koja sadrži unaprijed istrenirane težine. Detekcija lica pokreće se kada korisnik pauzira videozapis. Ova metoda osigurava da se detekcija vrši na okviru na kojem se video zaustavio, čime se smanjuje latencija videozapisa. Na slici 3.18. nalazi se funkcija *pause_video* koja se pokreće nakon što je korisnik pritisnuo gumb "Pause".

```

def pause_video(self):
    if self.is_paused:
        self.timer.start(20)
        self.is_paused = False
        self.result_label.setText("")
    else:
        self.timer.stop()
        self.is_paused = True
        self.result_label.setText("Video paused. Click on a face to apply super resolution.")
        self.detect_and_display_faces(self.current_frame) # Detekcija lica na pauziranom frame-u

```

Slika 3.18. Prikaz funkcije *pause_video*

Proces detekcije lica započinje sa predobradom okvira. Prije nego što model može detektirati lice, video okvir se mora obraditi u format prikladan za ulaz u neuronsku mrežu.

Predobrada uključuje promjenu rezolucije okvira na 128x128 elemenata slike kako bi odgovarala veličini ulaza koju očekuje model. Nakon promjene veličine vrši se oduzimanje srednjih vrijednosti kako bi se ulazna slika normalizirala te se na kraju pomoću funkcije `'cv2.dnn.blobFromImage'` smanjena i normalizirana slika pretvara u *blob* format. *Blob* format je ulazni format kojeg zahtijeva model dubokog učenja. Nakon što se okvir prethodno obradi, predaje se mrežu za detekciju lica. Funkcija `net.setInput(blob)` postavlja prethodno procesirani okvir kao ulaz u model, dok funkcija `net.forward()` proizvodi detekcije koje uključuju koordinate graničnih okvira i rezultate pouzdanosti za sva lica detektirana u okviru. Nakon prikupljanja svih detekcija, aplikacija obrađuje svaku detekciju kako bi izdvojila granične okvire i njihove rezultate pouzdanosti. Model proizvodi brojne detekcije, ali samo detekcije s ocjenom pouzdanosti višom od 0,5 smatraju se ispravnim. Koordinate graničnih okvira definiraju lokaciju detektiranog lica unutar okvira. Važeće detekcije (granični okviri) pohranjuju se na popis `self.current_detections` za daljnju obradu, kao što je crtanje okvira ili primjena super rezolucije. Granični okvir prikazan je kao crveni pravokutnik iscrtan oko svakog detektiranog lica. Svaki granični okvir iscrtava se koristeći prethodno izračunate koordinate. Rezultat pouzdanosti pretvara se u postotak i prikazuje iznad graničnog okvira. Konačno, okvir s detektiranim licima i graničnim okvirima prikazuje se u grafičkom korisničkom sučelju aplikacije. Na slici 3.19. prikazan je programski kod koji obuhvaća funkciju `detect_and_display_faces`.

```
def detect_and_display_faces(self, frame):
    #detekcija lica i crtanje bounding box-ova
    h, w = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (128, 128)), 1.0, (128, 128), (104.0, 177.0, 123.0))
    net.setInput(blob)
    detections = net.forward()
    self.current_detections = []

    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.5:
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (x_start, y_start, x_end, y_end) = box.astype("int")
            self.current_detections.append((x_start, y_start, x_end - x_start, y_end - y_start))

            # crtanje graničnih okvira i postotka
            cv.rectangle(frame, (x_start, y_start), (x_end, y_end), (0, 0, 255), 2)
            label = '{0:.2f}%'.format(confidence * 100)
            cv.rectangle(frame, (x_start, y_start - 18), (x_end, y_start), (0, 0, 255), -1)
            cv.putText(frame, label, (x_start + 2, y_start - 5), cv.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

    self.display_image(frame) # prikaz frame-a na koji je primijenjena detekcija
```

Slika 3.19. Prikaz funkcije `detect_and_display_faces`

3.4.7. Implementacija modula za super rezoluciju lica

Metoda super rezolucije u ovoj aplikaciji namijenjena je za poboljšanje rezolucije detektiranih lica u određenom okviru videozapisa. Proces super rezolucije pokreće prethodno istrenirani SRGAN model, koji je unutar aplikacije konfiguriran i učitani pomoću prilagođenih Python klasa. Proces započinje detekcijom lica, izdvajanjem detektiranih lica, a zatim korištenjem istreniranog modela za povećanje rezolucije lica. Aplikacije nudi automatsku ili ručnu metodu za primjenu super rezolucije. Automatska metoda super rezolucije primjenjuje se kada korisnik pritisne gumb "Process Frame". Metoda `process_frame()` prolazi kroz sva detektirana lica u trenutnom okviru, izdvaja svako detektirano lice te primjenjuje super rezoluciju na svaku izdvojenu nisko rezolucijsku sliku i pohranjuje odgovarajuće SR slike. Ručna metoda odnosi se na mogućnost odabiranja određenog lica u okviru, klikom na lice na koje korisnik želi primijeniti super rezoluciju. Metoda `on_image_click()` provjerava nalazi li se klik unutar graničnog okvira detektiranog lica. Ako je klik unutar graničnog okvira, primjenjuje se super rezolucija samo na odabrano lice. Na slici 3.20. prikazana je funkcija `on_image_click`.

```
def on_image_click(self, x, y):
    if not self.is_paused:
        self.result_label.setText("Please pause the video to select faces.")
        return

    display_scale_x = self.image_label.pixmap().width() / self.current_frame.shape[1]
    display_scale_y = self.image_label.pixmap().height() / self.current_frame.shape[0]
    original_x = int(x / display_scale_x)
    original_y = int(y / display_scale_y)

    face_found = False
    for (x1, y1, w, h) in self.current_detections:
        x2, y2 = x1 + w, y1 + h
        if x1 <= original_x <= x2 and y1 <= original_y <= y2:
            self.result_label.setText("Face selected at ({} , {}). Processing...".format(original_x, original_y))

            self.process_face_sr(self.current_frame, (x1, y1, w, h))
            face_found = True
            break

    if not face_found:
        self.display_message("No face selected. Click within a face boundary.", 1000)
```

Slika 3.20. Prikaz funkcije `on_image_click`

Proces super rezolucije se provodi korištenjem prethodno istreniranog SRGAN modela čija je mreža i treniranje opisano u poglavlju 3.3. Korišteni SRGAN model definiran je i učitani kroz klasu `SRGANModel`, koja upravlja konfiguracijom, učitavanjem i izvođenjem procesa super rezolucije. Konfiguracija za SRGAN model uključuje učitavanje različitih parametara kao što su mreža generatora i diskriminatora, funkcije gubitaka, stope učenja itd. Nakon konfiguracije modela, model se instancira i učitava s unaprijed istreniranim težinama. Klasa `SRGANModel`

upravlja inicijalizacijom SRGAN arhitekture dok *load()* metoda u klasi *SRGANModel* učitava unaprijed istrenirane težine modela (*pretrain_model_G*) u mrežu, pripremajući je za zadatak super rezolucije. Prije ulaska slike u mrežu SRGAN modela, ulazna slika prolazi kroz niz transformacija kako bi se prilagodila formatu prikladnom prema ulaznim zahtjevima mreže. Koristi se klasa *ToTensor* koja ulaznu sliku iz *PIL* slike ili *NumPy* polja pretvara u *PyTorch* tenzor. Vrijednosti elemenata slike se normaliziraju tako da imaju srednju vrijednost 0,5 i standardnu devijaciju 0,5 na svim kanalima (RGB). Detektiranom licu (*face_img*), koje je izdvojeno iz okvira, mijenja se veličina na rezoluciju 128x128 elemenata slike (*face_img_lr*) bilinearnom interpolacijom. Nisko rezolucijska slika lica (*face_img_lr*) predstavlja ulaz za SRGAN model. Slike niske rezolucije spremaju se u svrhu reference ili procjene. Prije testiranja SRGAN modela na nisko rezolucijskim slikama, slike niske rezolucije prvo se pretvaraju iz *OpenCV* BGR formata u RGB format, a zatim u *PIL* sliku (*pil_img_lr*). *PIL* slika pretvara u 4D ulazni tenzor (*input_img*). Naposljetku, pokreće se SRGAN model na ulaznom tenzoru, proizvodeći izlaznu sliku visoke rezolucije. Nakon primjene super rezolucije, izlazni tenzor se pretvara natrag u *NumPy* polje (*output_img*) i vrijednosti piksela mijenjaju se iz normaliziranog raspona [0,1] na [0,255]. Slika u rezoluciji 512x512 elemenata slike, pretvara se u *PIL* format i sprema se na predviđeno mjesto. Na slici 3.21. prikazana je funkcija *process_face_sr*, koja primjenjuje super rezoluciju na lice na koje je korisnik izravno kliknuo.

```
def process_face_sr(self, frame, face_rect):
    x, y, w, h = face_rect
    face_img = frame[y:y+h, x:x+w]

    face_img_lr = cv2.resize(face_img, (128, 128))
    cv2.imwrite('lr_face_{}_{}.png'.format(x, y), face_img_lr) #spremanje LR slike

    #pred obrada slike
    pil_img_lr = Image.fromarray(cv2.cvtColor(face_img_lr, cv2.COLOR_BGR2RGB))
    input_img = _transform(pil_img_lr).unsqueeze(0).to(sr_model.device)
    sr_model.var_L = input_img
    sr_model.test()
    output_img = sr_model.fake_H.squeeze(0).cpu().numpy()
    output_img = np.clip((np.transpose(output_img, (1, 2, 0)) + 1) / 2 * 255, 0, 255).astype(np.uint8)

    #spremanje SR slike
    output_hr_img = cv2.resize(output_img, (512, 512))
    sr_img = Image.fromarray(output_hr_img)
    sr_img.save('super_resolved_{}_{}.png'.format(x, y))
```

Slika 3.21. Prikaz funkcije *process_face_sr*

4. EVALUACIJA I VERIFIKACIJA PREDLOŽENOG ALGORITMA I APLIKACIJE ZA POVEĆANJE RAZLUČIVOSTI LICA S DIGITALNIH SLIKA

4.1. Evaluacija istreniranog vlastitog modela za povećanje razlučivosti slike lica

U ovom poglavlju provodi se detaljna evaluacija razvijenog modela za super rezoluciju lica na CelebAHQ testnom skupu podataka. Izvedena je kvalitativna i kvantitativna usporedba rezultata sa osnovnim modelom - Face Super-Resolution [30]. U tablici 4.1. nalaze se kvantitativni rezultati razvijenog algoritma i FSR algoritma dobiveni uz pomoć FR metrika.

Tablica 4.1. Evaluacija razvijenog super rezolucijskog algoritma na CelebAHQ podatkovnom skupu uz pomoć FR metrika i usporedba s FSR algoritmom

ALGORITAM	Face Super-Resolution [30]				Razvijeni algoritam			
IQA	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓	PSNR ↑	SSIM ↑	MS-SSIM ↑	DISTS ↓
\bar{x}	16.00	0.48	0.50	0.18	15.91	0.48	0.49	0.25
σ	2.2	0.09	0.1	0.02	2.24	0.09	0.1	0.03
Min	9.04	0.25	0.26	0.13	8.89	0.25	0.24	0.16
Max	24.17	0.76	0.76	0.29	25.01	0.75	0.75	0.36

Iz priloženih rezultata uočljivo je kako su razvijeni algoritam i Face Super Resolution algoritam podjednako ocjenjeni prema klasičnim metrikama za procjenu kvalitete slika. Algoritam Face Super-Resolution neznatno nadmašuje razvijeni algoritam u pogledu rezultata IQA metrika, što ukazuje na bolju kvalitetu slike i strukturnu sličnost. Međutim, razvijeni algoritam pokazuje sličnu pouzdanost u povećanju rezolucije lica. Oba algoritma detektiraju i izdvajaju lica prije povećanja rezolucije. Pri procesu izdvajanja lica koristi se proces zumiranja lica koji dovodi do neprimjetnog gubitka elemenata slike koji utječe na rezultate. Ovaj gubitak elemenata slike doprinosi prosječnim vrijednostima PSNR, SSIM i MS-SSIM metrike. U usporedbi sa prosječnim PSNR, SSIM i MS-SSIM rezultatima, oba algoritma postižu bolje DISTS rezultate. Unatoč podjednakim srednjim vrijednostima prema standardnim FR metrikama, može se uočiti kako razvijeni algoritam zaostaje za FSR algoritmom prema DISTS rezultatima. Prosječna DISTS ocjena na temelju svih prikupljenih rezultata testiranja razvijenog algoritma na CelebAHQ skupu podataka iznosi 0.25, dok prosječna ocjena za FSR algoritam iznosi 0.18. Niže DISTS ocjene u usporedbi s FSR algoritmom rezultat su nedovoljno raspoloživih GPU resursa za treniranje

modela, što je uzrokovalo određeni kompromis prilikom treniranja. Zbog nedostatka memorije za razvoj vlastitog algoritma korišten je manji *batch size*, što je glavni uzrok lošijih rezultata. Na slici 4.1. prikazane su SR slike za koje je razvijeni algoritam dobio maksimalnu PSNR i DISTSS ocjenu te minimalnu PSNR i DISTSS ocjenu.



a)



b)



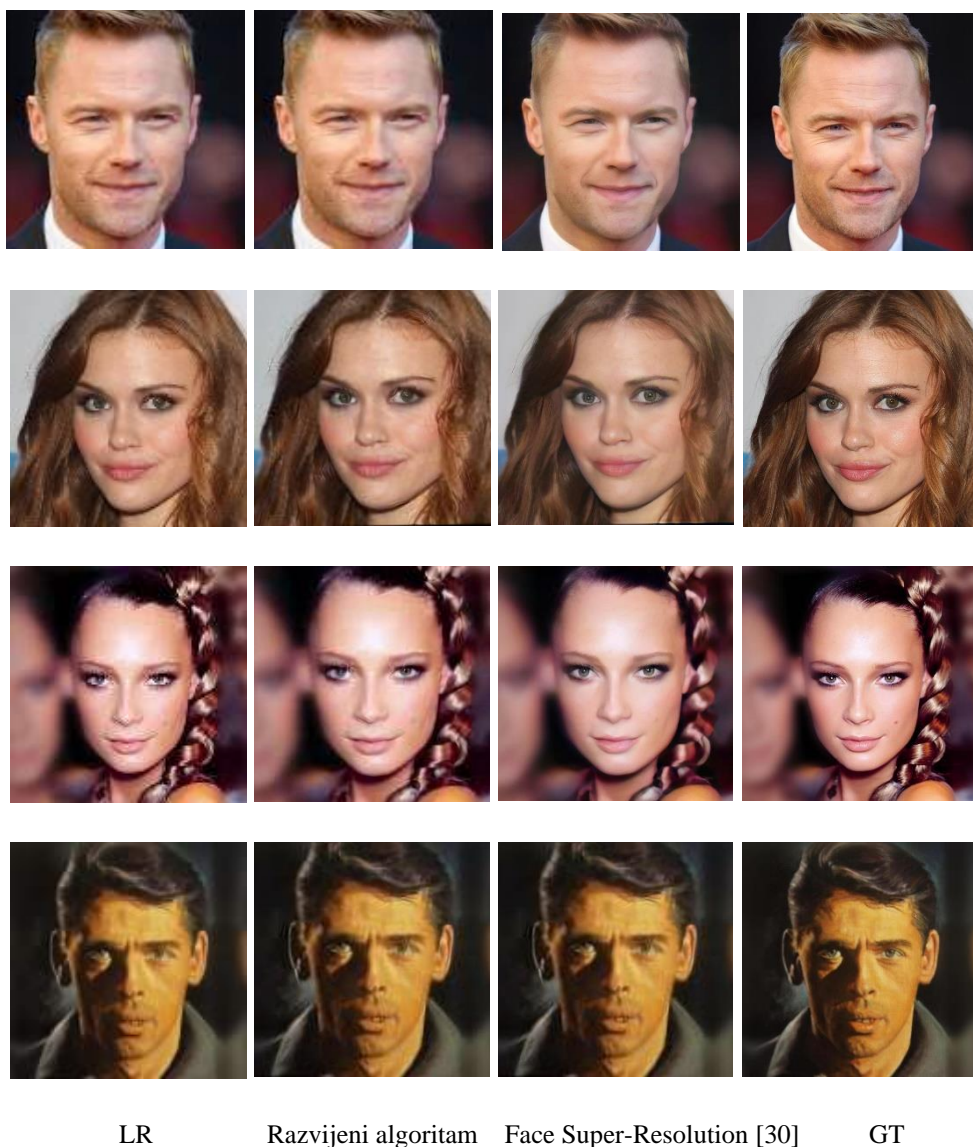
c)



d)

Slika 4.1. Prikaz SR izlaza razvijenog algoritma za LR ulaze a) Prikaz SR slike sa max vrijednosti PSNR= 25.01, b) Prikaz SR slike sa min vrijednosti PSNR = 8.89, c) Prikaz SR slike sa min vrijednosti DISTSS = 0.16, d) Prikaz SR slike sa max vrijednosti DISTSS = 0.36

Na slici 4.2. nalaze se kvalitativni rezultati povećanja rezolucije ulaznih LR slika s razvijenim modelom, te rezultati Face Super-Resolution [30] algoritma i odgovarajuća GT slika (na četiri odabrane slike). U priloženim rezultatima na slici 4.2. uočava se lošija perceptivna kvaliteta SR slika generiranih uz pomoć razvijenog algoritma u usporedbi sa FSR algoritmom. Na SR slikama generiranim uz pomoć razvijenog algoritma vidljivi su artefakti koji nedostaju na SR slikama FSR algoritma te je potrebno dodatno treniranje razvijenog modela kako bi generirao jednako dobre ili bolje SR izlaze od FSR modela. Prema rezultatima u tablici 4.2. vidljivo je kako su algoritmi dobili podjednake ocjene prema standardnim FR metrikama, ali razvijeni algoritam zaostaje za FSR algoritmom u pogledu DISTS rezultata.



Slika 4.2. Kvalitativni rezultati izlaza super rezolucijskog razvijenog algoritma i FSR algoritma za niskorezolucijski ulaz

Tablica 4.2. prikazuje kvantitativne rezultate četiri super rezolucijske slike generirane pomoću razvijenog algoritma i Face Super-Resolution algoritma koje se nalaze na slici 4.2.

Tablica 4.2. Kvantitativni rezultati super rezolucijskih izlaza dobiveni evaluacijom pomoću IQA metrika

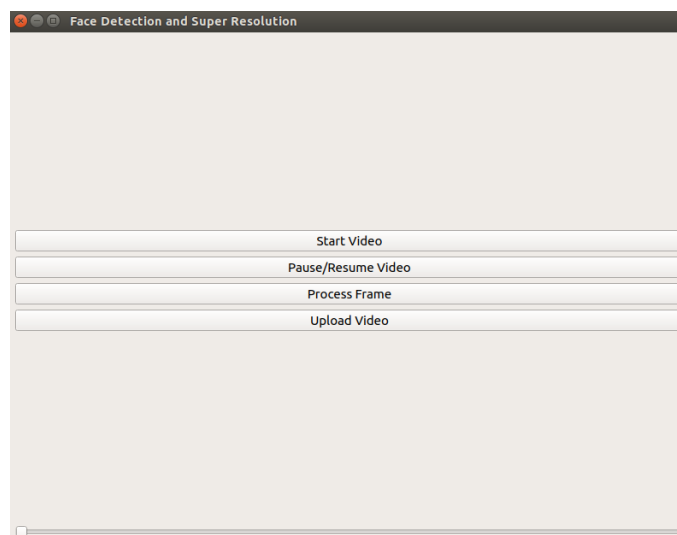
ALGORITAM	Face Super – Resolution [30]				Razvijeni algoritam			
SLIKA	Slika 1	Slika 2	Slika 3	Slika 4	Slika 1	Slika 2	Slika 3	Slika 4
PSNR ↑	17.99	18.36	13.09	20.18	18.08	18.18	13.85	20.06
SSIM ↑	0.65	0.43	0.54	0.61	0.65	0.43	0.53	0.61
MS – SSIM ↑	0.65	0.56	0.53	0.60	0.64	0.55	0.52	0.59
DISTS ↓	0.13	0.17	0.20	0.15	0.19	0.23	0.19	0.16

4.2. Verifikacija aplikacije za povećanje razlučivosti lica

U ovom poglavlju predstavljena je verifikacija rada aplikacije prema zahtjevima opisanim u trećem poglavlju. Verifikacija aplikacije se odvija testiranjem aplikacije na postojećim videozapisima i na video snimkama dobivenih s modula kamere spojenog na NVIDIA Jetson Nano platformu.

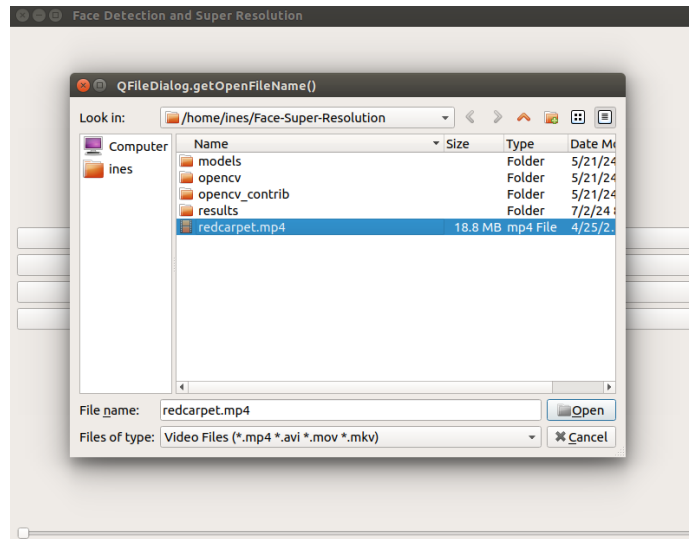
4.2.1. Testiranje aplikacije na videozapisima učitanim iz datotečnog sustava

Aplikacija se pokreće izravno iz *Visual Studio Code*-a ili terminala. Nakon pokretanja aplikacije otvara se grafičko korisničko sučelje koje nudi opcije učitavanja videozapisa ili pokretanja kamere. Grafičko korisničko sučelje prikazano je na slici 4.3.



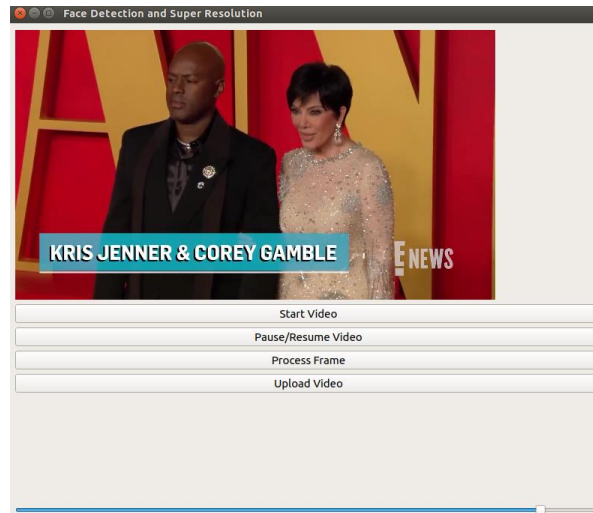
Slika 4.3. Izgled grafičkog korisničkog sučelja

Odabiranjem opcije "Upload Video", pokreće se dijaloški okvir koji korisniku pruža pretraživanje i učitavanje videozapisa iz datotečnog sustava. Korisniku je dozvoljeno učitavanje videozapisa sa ekstenzijama .mp4, .avi, .mov i .mkv. Nakon što korisnik odabere željeni videozapis, učitava ga klikom na gumb "Open". Na slici 4.4. prikazan je dijaloški okvir za učitavanje videozapisa.

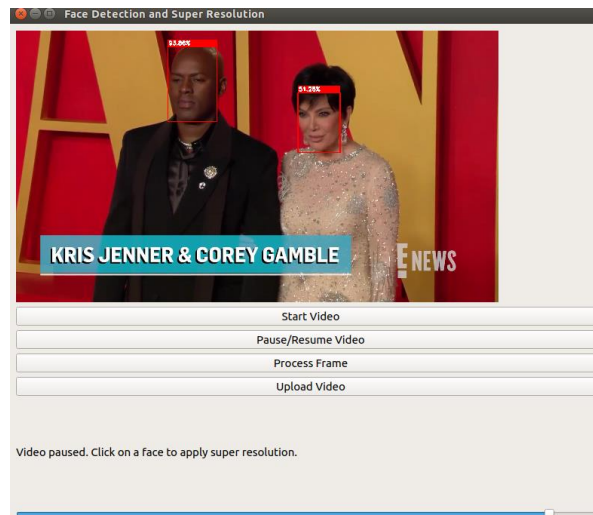


Slika 4.4. Verifikacija prikaza dijaloškog okvira za učitavanje videozapisa

Kada je videozapis sa odgovarajućom ekstenzijom ispravno učitana iz datotečnog sustava, prikazuje se u grafičkom korisničkom sučelju kao što je prikazano na slici 4.5. Pomoću klizača korisnik može prolaziti kroz video i jednostavnije pronaći okvir od interesa. U bilo kojem trenutku korisnik može pritisnuti gumb "Pause/Resume Video". Opcija *Pause/Resume Video* zaustavlja videozapis, a ponovnim odabirom iste opcije videozapis se nastavlja od okvira na kojem je zaustavljen. Kada je odabrana opcija *Pause/Resume Video*, videozapis se zaustavlja i izvršava se automatska detekcija lica unutar zaustavljenog okvira prikazana na slici 4.6. Aplikacija obavještava korisnika da je videozapis zaustavljen i da može kliknuti na lice unutar okvira kako bi primijenio implementirani algoritam super rezolucije.

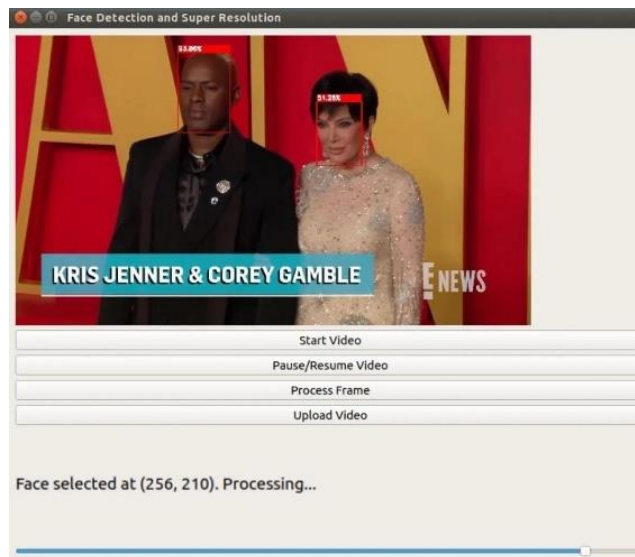


Slika 4.5. Verifikacija modula za učitavanje videozapisa i prikaz okvira videozapisa unutar GUI-a

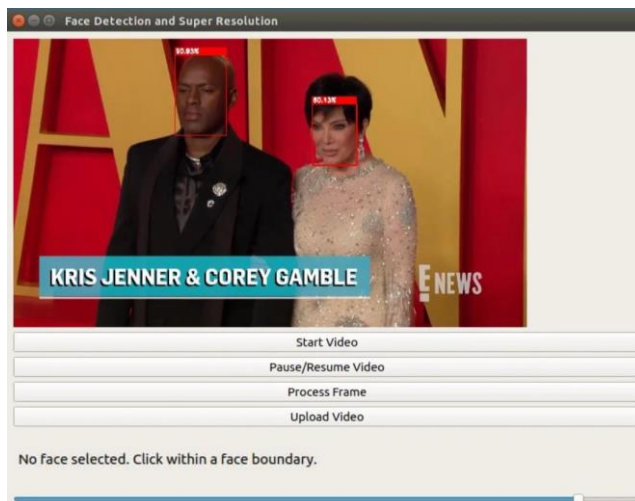


Slika 4.6. Verifikacija modula za detekciju lica i prikaz okvira sa detektiranim licima i graničnim okvirima

Sada korisnik može primijeniti super rezoluciju na detektirana lica klikom na gumb "Process Frame" ili izravnim klikom na lice unutar graničnog okvira. Odabirom opcije *Process Frame*, metoda super rezolucije primjenjuje se na sva detektirana lica u okviru. Klikom na određeno lice unutar okvira, metoda super rezolucije primjenjuje se isključivo na odabrano lice. Unutar korisničkog sučelja ispisuju se koordinate odabranog lica kao što je prikazano pod a) na slici 4.7. Ako korisnik klikne na lice izvan graničnog okvira, pojavljuje se informativna poruka koja obavještava korisnika da lice nije odabrano i da klikne na lice unutar okvira što je prikazano pod b) na slici 4.7.



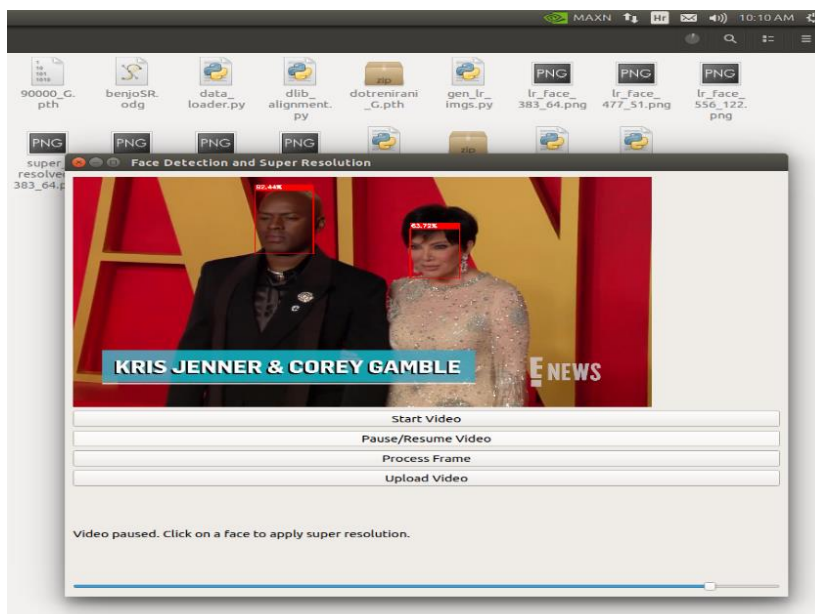
a)



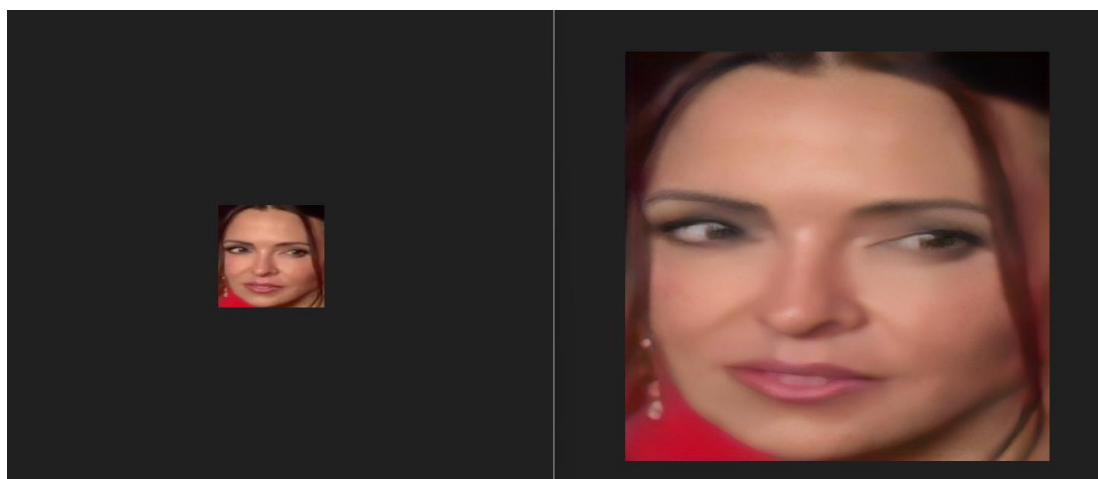
b)

Slika 4.7. Verifikacija modula za super rezoluciju: a) prikaz ispisa koordinata lica nakon što je korisnik odabrao lice b) prikaz informativne poruke koja obavještava korisnika da nije odabrao lice unutar graničnih okvira

Ako su slike lica iz okvira ispravno procesirane, super rezolucijske slike se zajedno sa nisko rezolucijskom slikom pohranjuju na predviđeno mjesto. Na slici 4.8. prikazan je okvir u kojem su detektirana dva lica na koje je primijenjena super rezolucija koristeći opciju *Process Frame*, te su LR i SR slike automatski pohranjene u direktorij u .png formatu. Na slici 4.9. prikazane su pohranjene LR i SR slike za jedno od lica za slike.



Slika 4.8. Prikaz direktorija za pohranu generiranih SR i nisko rezolucijskih LR slika

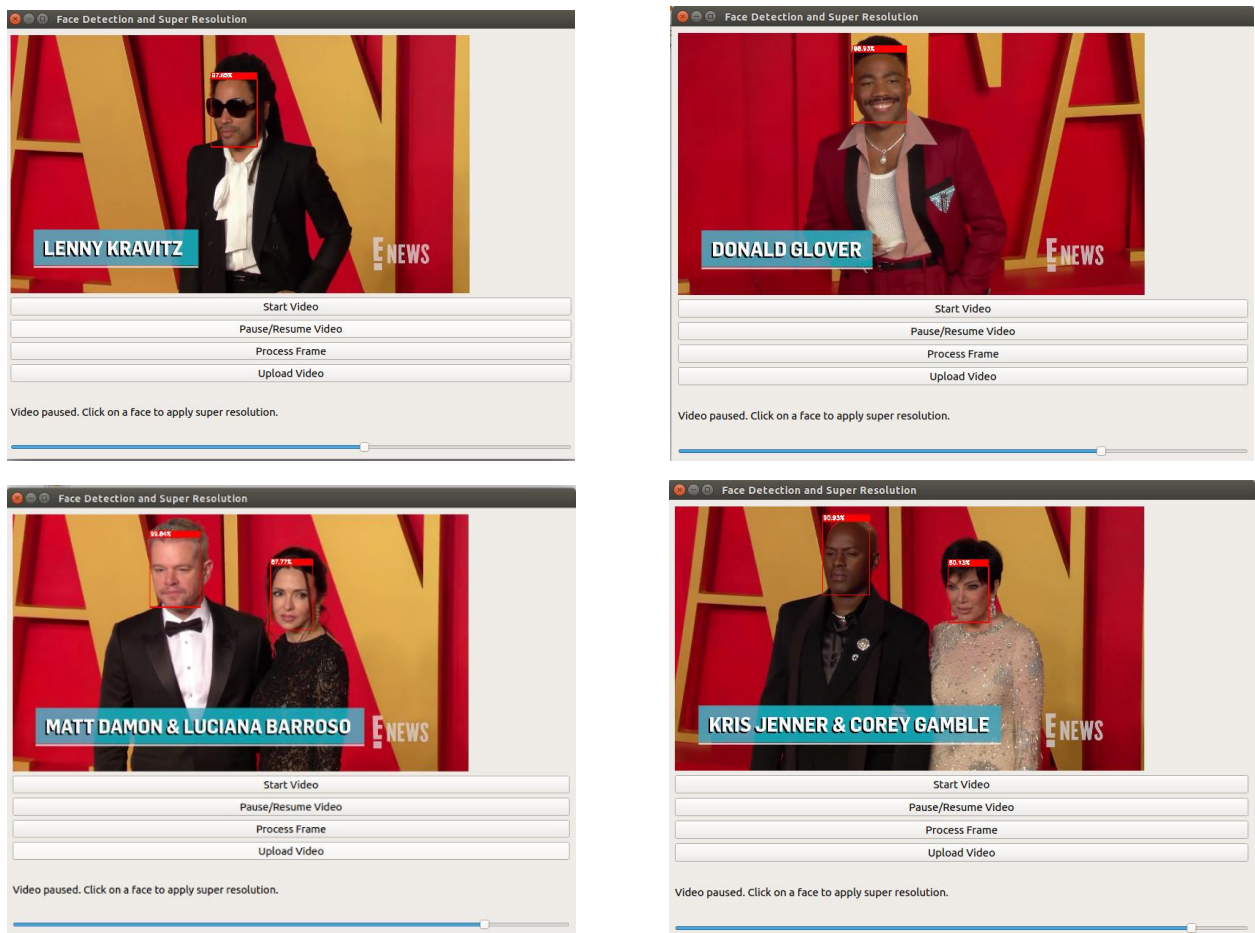


a)

b)

Slika 4.9. Prikaz a) pohranjene LR i b) SR slike

U nastavku je provedeno testiranje aplikacije na četiri nasumično odabrana okvira iz učitanoog videozapisa. Na slici 4.10. prikazani su odabrani okviri. Na slici 4.11. prikazani su kvalitativni rezultati dobiveni procesom super rezolucije, koristeći opciju Process Frame i izravan klik na lice. Lice sa crvenim obrubom dobiveno je izravnim klikom na detektirano lice unutar okvira.



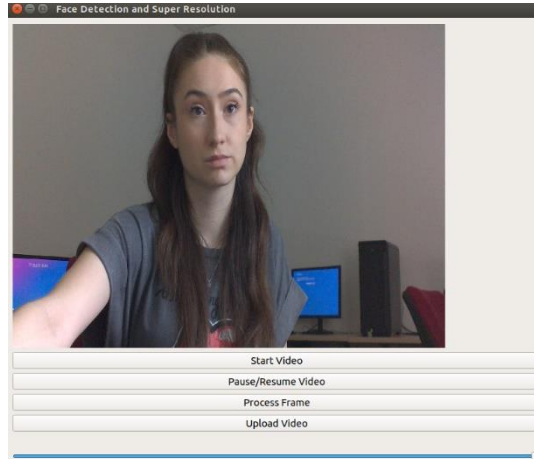
Slika 4.10. Prikaz zaustavljenih odabranih okvira



Slika 4.11. Kvalitativni rezultati izlaza super rezolucijskih algoritama za niskorezolucijski ulaz LR

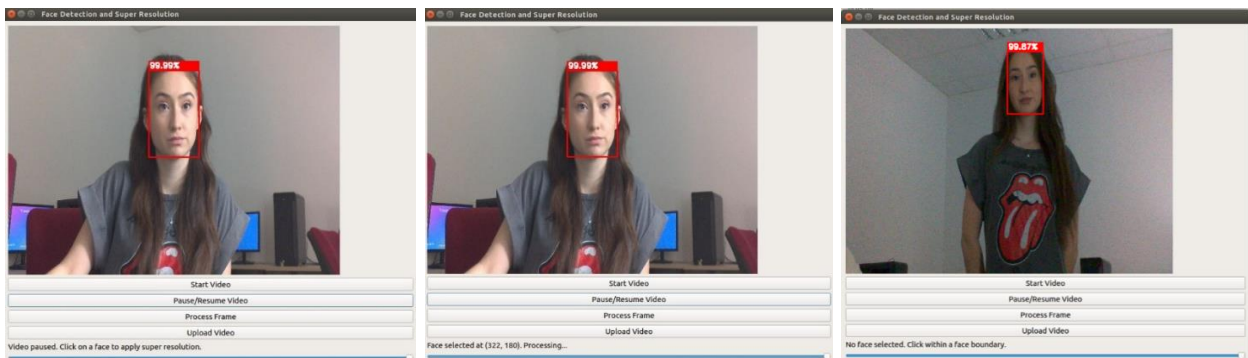
4.2.2. Testiranje aplikacije na prijenosu s kamere u stvarnom vremenu

Kada korisnik pritisne gumb "Start Video" pokreće se kamera. Ako je kamera ispravno implementirana, u grafičkom korisničkom sučelju započinje prijenos videa s kamere. Na slici 4.12. prikazan je prijenos uživo s kamere nakon pokretanja opcije Start Video.



Slika 4.12. Verifikacija prijenosa feed-a s kamere spojene na NVIDIA Jetson Nano

Pritiskom na gumb "Pause/Resume Video" video prijenos se zaustavlja, ponovnim pritiskom na "Pause/Resume Video" prijenos ponovno započinje. Kada je video zaustavljen, vrši se automatska detekcija lica te korisnik može primijeniti metodu super rezolucije pritiskom na gumb "Process Frame" ili izravnim klikom na lice. Na slici 4.13. prikazan je izgled zaustavljenog okvira prije i poslije izravnog klika na lice. Ako je korisnik kliknuo izvan okvira graničnog okvira, informativna poruka unutar sučelja ga obavještava da odabere lice unutar graničnog okvira.



a)

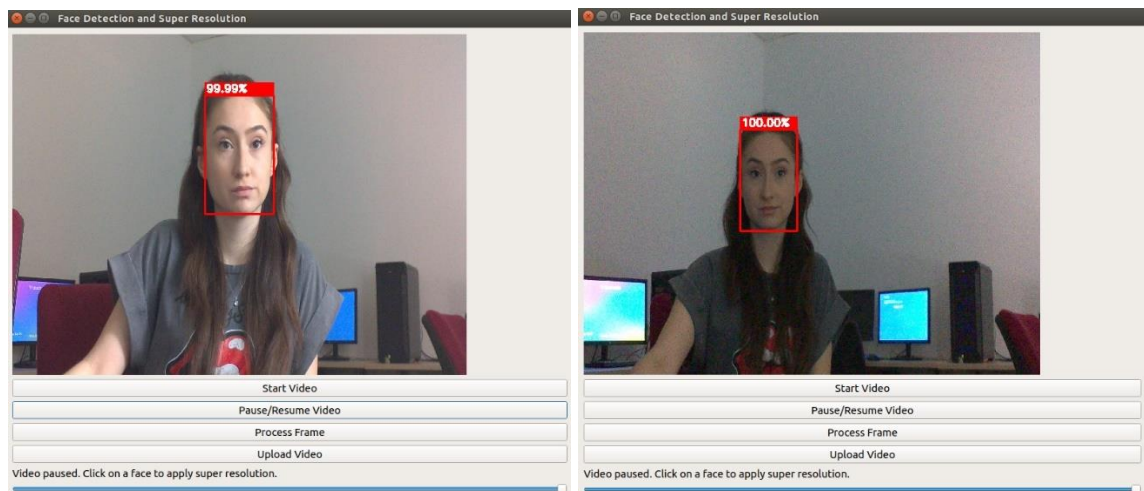
b)

c)

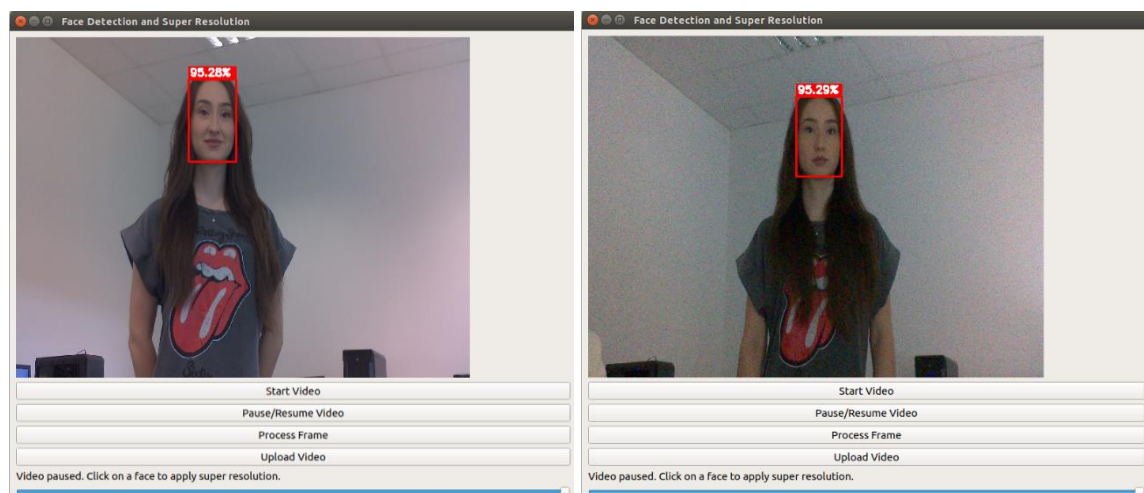
Slika 4.13. Verifikacija modula za detekciju lica i modula za super rezoluciju a) prikaz okvira nakon što je okvir zaustavljen b) prikaz ispisa koordinata odabranog lica c) prikaz informativne poruke koja obavještava korisnika da nije odabrao lice unutar okvira

U nastavku je opisano testiranje aplikacije na odabranim okvirima tokom *live stream*-a kamere. Testiranje je provedeno na tri različite udaljenosti od kamere u dva različita osvjetljenja. Prvi test slučaj predstavlja kratku udaljenost od kamere, gdje je lice udaljeno 1m od kamere, dok je u drugom test slučaju lice udaljeno 2,5m od kamere, a pri trećem test slučaju lice je udaljeno 4,5m od kamere. Super rezolucija je primjenjena pri dva različita osvjetljenja, a izgled pauziranih okvira i detekcija lica je prikazana na slici 4.12. Zbog detaljnije analize performansi osim razvijenog modela, postojeći Face Super-Resolution model je također testiran na ovim test slučajevima. Na slikama 4.14. , 4.15. i Slika 4.16. prikazani su test slučajevi za jedan od modela.

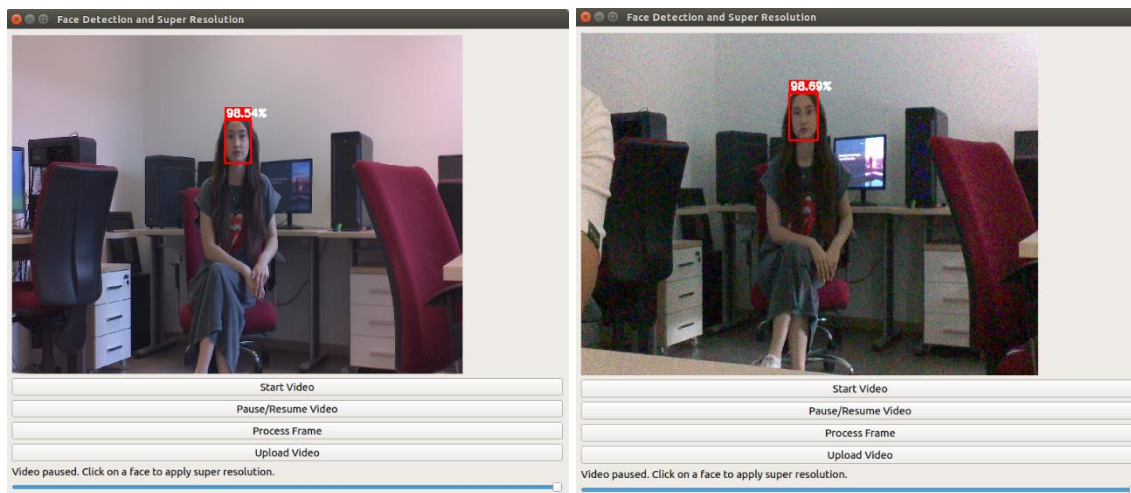
Na lica iz okvira primijenjena je metoda super rezolucije pomoću opcije Process Frame i izravnim klikom na lice. Na slici 4.17. prikazani su kvalitativni rezultati super rezolucijskih izlaza FSR modela za nisko rezolucijski ulaz LR. Na slici 4.18. prikazani su kvalitativni rezultati razvijenog modela.



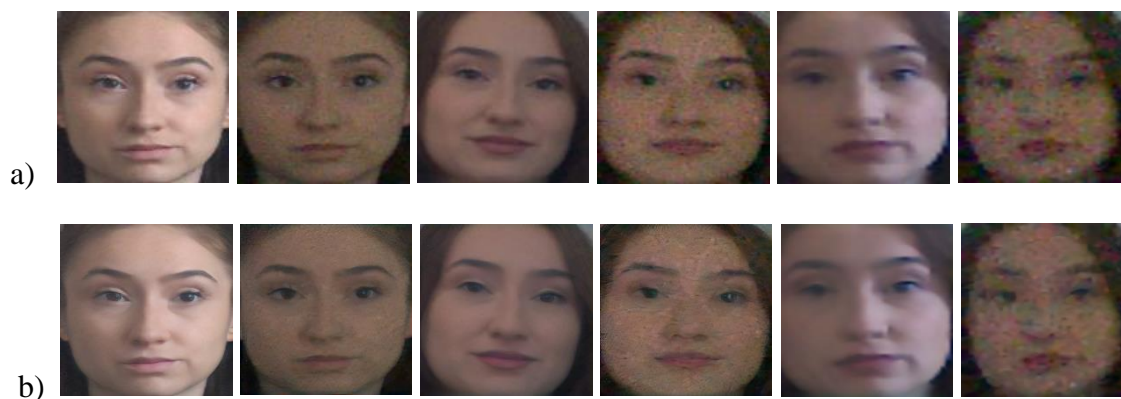
Slika 4.14. Prikaz zaustavljenih okvira na kojem je lice udaljeno 1m od kamere



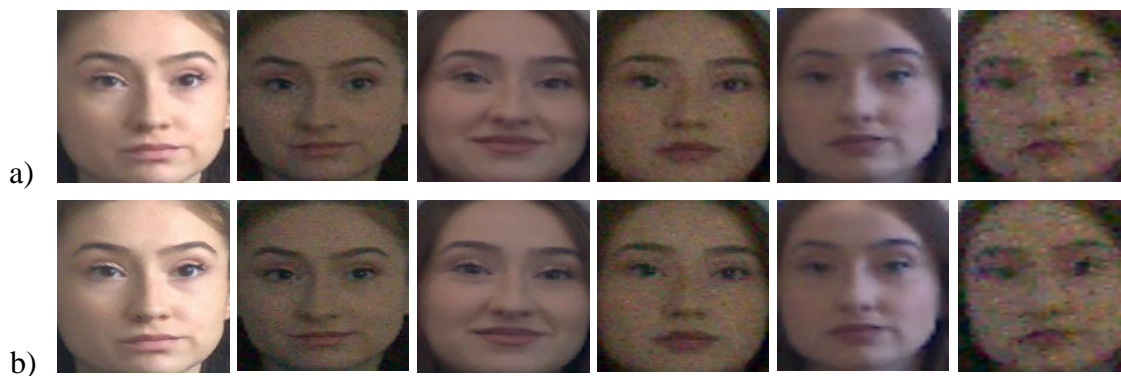
Slika 4.15. Prikaz zaustavljenih okvira na kojem je lice udaljeno 2,5m od kamere



Slika 4.16. Prikaz zaustavljenih okvira na kojem je lice udaljeno 4,5m od kamere



Slika 4.17. Kvalitativni rezultati izlaza FSR [30] modela za niskorezolucijski ulaz LR: a)LR slike, b)SR slike



Slika 4.18. Kvalitativni rezultati izlaza razvijenog modela za niskorezolucijski ulaz LR: a)LR slike, b)SR slike

4.3. Evaluacija rješenja aplikacije za povećanje razlučivosti lica

Dobivena kvalitativna rješenja iz poglavlja 4.2. evaluirana su pomoću *NR* IQA metrika BRISQUE, NIQE i PIQE opisanih u drugom poglavlju na pet različitih primjera. Dobivene ocjene

IQA metrika prikazane su u tablicama 4.3. i **Pogreška! Izvor reference nije pronađen.** Strelica pored polja "IQA" ukazuje na to da je poželjna što niža ocjena. U tablici 4.3. prikazani su kvantitativni rezultati za SR slike prikazane na slici 4.11. dok su u tablici 4.4. prikazani rezultati za SR slike prikazane na slikama 4.17. i 4.18.

Tablica 4.3. Kvantitativni rezultati evaluacije SR slika iz učitanoog videa uz pomoć NR metrika

SLIKA	Slika 1			Slika 2			Slika 3		
IQA ↓	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE
RAZVIJENI ALGORITAM	55.93	11.61	32.69	58.08	12.73	33.48	53.06	12.56	33.38
FSR [30]	40.16	10.27	17.36	41.16	12.10	15.66	34.14	15.78	15.05
SLIKA	Slika 4			Slika 5					
IQA ↓	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE			
RAZVIJENI ALGORITAM	60.05	13.02	23.56	58.53	14.57	36.91			
FSR	41.67	12.33	19.12	39.53	12.74	15.42			

Iz priloženih rezultata vidljivo je kako je razvijeni algoritam dobio sve BRISQUE ocjene u rasponu od 53 – 60.05. Najbolje je ocjenjena slika 3 s ocjenom 53.06. U usporedbi sa razvijenim modelom, FSR model postiže bolje BRISQUE ocjene u rasponu od 34 – 41.67. Kao i razvijeni model, FSR model najbolju ocjenu je dobio za sliku 3 s rezultatom 34.14. Prema NIQE metrici oba modela postižu podjednake rezultate, bez isticanja jednog od navedenih. Prema PIQE metrici, FSR model postiže primjetno bolje ocjene od razvijenog modela, s najboljom ocjenom 15.05. Iako je razvijeni model ocjenjen sa prosječno dobrim BRISQUE i PIQE ocjenama, pokazuje pouzdanost za primjenu metode super rezolucije na lica iz postojećih, učitanih videozapisa.

Tablica 4.4. Kvantitativni rezultati evaluacije SR slika dobivenih iz prijenosa s kamere

SLIKA	Slika 1			Slika 2			Slika 3		
IQA ↓	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE
RAZVIJENI ALGORITAM	40.55	15.08	25.37	51.22	24.45	35.57	57.56	14.15	27.78
FSR [30]	21.42	11.91	6.78	8.74	18.24	2.95	29.40	14.19	8.64
SLIKA	Slika 4			Slika 5			Slika 6		
IQA ↓	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE	BRISQUE	NIQE	PIQE
RAZVIJENI ALGORITAM	54.66	19.49	47.26	71.56	16.19	8.33	65.07	17.94	50.0
FSR [30]	7.65	22.14	2.33	55.24	12.72	25.93	45.10	13.58	17.78

Iz priloženih rezultata vidljivo je kako je kako razvijeni model najbolje BRISQUE ocjene dobiva za slike s jačim intenzitetom svjetline te je najbolju ocjenu dobio za sliku 1. Vidljivo je kako su NIQE ocjene podjednake za sve slike te je razvijeni model dobio ocjene u rasponu od 14.15 do 24.45, što predstavlja nizak NIQE rezultat. Razvijeni algoritam dobio je prosječno dobre PIQE ocjene te je dobio najbolju PIQE ocjenu za sliku 5. FSR model dobio je bolje BRISQUE ocjene te je najbolju ocjenu dobio za sliku 5. Dobivene NIQE ocjene slične su onima koje je dobio razvijeni model, sve ocjene su u rasponu od 11.91 do 22.14. Za razliku od razvijenog modela, FSR algoritam dobio je bolje PIQE ocjene, a najbolju PIQE ocjenu FSR model je dobio za sliku 4. Iako je razvijeni model dobio prosječno dobre ocjene prema NR metrikama, pokazao je visoku pouzdanost za rad u stvarnom vremenu.

Sudeći po dobivenim rezultatima, može se zaključiti kako je razvijeni model sposoban za visoko kvalitetnu primjenu super rezolucije za lica. Razvijeni model pokazao se sposobnim za rad u stvarnom vremenu na stvarnim slikama lica kao i na postojećim izrazito oštećenim slikama lica.

5. ZAKLJUČAK

U ovom diplomskom radu detaljno su opisani algoritmi za super rezoluciju te je razvijen vlastiti algoritam za super rezoluciju lica, koji je uspješno implementiran na NVIDIA Jetson Nano ugradbeni računalni sustav. Razvijeno rješenje pronalazi primjenu u poboljšanju kvalitete slika lica sa snimaka nadzornih kamera kod kojih najveći izazov predstavljaju nisko rezolucijske slike sa izraženim oštećenjima.

Postojeći modeli za super rezoluciju testirani su na CelebAHQ skupu podataka, nakon čega je provedena detaljna evaluacija modela pomoću metrika za procjenu kvalitete slika. Korištene su metrike : PSNR, SSIM, MS-SSIM i DISTS, koje obuhvaćaju procjenu sličnosti strukture, teksture i percipirane kvalitete slike. Nakon detaljne usporedbe i analize modela, model koji se pokazao kao najoptimalniji, korišten je kao osnovni model za razvoj vlastitog modela super rezolucije.

Za razvoj vlastite neuronske mreže, korištena je arhitektura mreže osnovnog modela zasnovana na tehnici generativnih suparničkih mreža. Odabran je skup podataka za treniranje i testiranje modela, te su parametri treniranja neuronske mreže prilagođeni dostupnim računalnim resursima. Nakon treniranja modela uslijedilo je testiranje modela na CelebAHQ skupu podataka te potom evaluacija modela PSNR, SSIM, MS-SSIM i DISTS metrikama kako bi se usporedile performanse razvijenog modela sa postojećim modelima za super rezoluciju. Razvijeni model procijenjen je kao efikasan i pouzdan model, čiji rezultati nadmašuju rezultate nekih od postojećih algoritama.

Sljedeći korak bio je implementacija modela na NVIDIA Jetson Nano platformu. Razvijena je aplikacija unutar koje je implementiran razvijeni model za super rezolucija lica i koja je potpuno prilagođena za rad na ugradbenom računalnom sustavu NVIDIA Jetson Nano. Uslijedilo je testiranje modela na postojećim snimkama te na prijenosu uživo sa kamere implementirane na NVIDIA Jetson Nano platformu. Model je pokazao kako može uspješno proizvoditi visoko rezolucijske slike u stvarnom vremenu, što dokazuje njegovu primjenjivost u stvarnim snimkama sa nadzornih kamera. Prikupljeni su svi izlazni rezultati te je provedena evaluacija modela na metrikama BRISQUE, NIQE i PIQE. Izvršena je usporedba razvijenog modela sa postojećim osnovnim modelom te je razvijeni model procijenjen kao stabilan i učinkovit model za rad u stvarnom vremenu.

Najveće izazove pri razvoju modela za super rezoluciju predstavljaju visoki računalni zahtjevi. Generativne suparničke mreže korištene za razvoj modela zahtijevaju značajne računalne

resurse, uključujući snažne grafičke procesore sa velikom količinom memorije. Idealno treniranje složenih mreža poput GAN-ova odvija se na više GPU-ova, gdje svaki GPU ima memorijski prostor od 16-32GB kako bi mogao stvoriti velike modele i obraditi visoko rezolucijske podatke. Veliki *batch size* i visoko rezolucijske slike zahtijevaju znatnu količinu memorije, što može stvoriti poteškoće kada se mreža trenira na slikama visoke rezolucije poput 512x512 elemenata slike ili više. Osim toga, skupovi podataka za treniranje GAN-ova su često jako veliki, pa tako i zahtijevaju veliki prostor za pohranu. GAN mreže treniraju se na iznimno velikom broju iteraciju (često milijuni) dok ne postignu stabilnost treninga, što dovodi do dugog vremena treniranja koje može uključivati i tjedne treniranja čak i na snažnom hardveru. Kako bi se optimizirala kompleksnost treninga uvode se tehnike poput akumulacije gradijenta i *mixed precision* treninga koje mogu upravljati računalnim opterećenjima, ali povećavaju složenost procesa treniranja.

Pri razvoju vlastitog modela za super rezoluciju lica najveći izazov bio je upravo nedostatak dovoljno snažnog grafičkog procesora, koji bi omogućio stabilniji i uspješniji proces treniranja. No usprkos nedostacima uzrokovanim ograničenim računalnim resursima, razvijeni model pokazao je mogućnost stvaranja visoko rezolucijskih slika iz oštećenih nisko rezolucijskih slika u stvarnom vremenu. Razvijeni algoritam pokazao je potencijal za daljnje unapređenje i poboljšanje performansi, koje se može postići korištenjem višestrukih grafičkih procesora kako bi se osigurala veća dostupna memorija te samim time omogućilo kvalitetnije treniranje neuronske mreže.

LITERATURA

- [1] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” Jun. 10, 2014, *arXiv*: arXiv:1406.2661. Accessed: Aug. 26, 2024. [Online]. Available: <http://arxiv.org/abs/1406.2661>
- [2] M. Seshaih, A. R. Nair, E. Mallya, and S. Dev, “CCTV Surveillance Camera’s Image Resolution Enhancement using SRGAN,” vol. 07, no. 06, 2020.
- [3] B. Z. Demiray, M. Sit, and I. Demir, “D-SRGAN: DEM Super-Resolution with Generative Adversarial Networks,” Apr. 16, 2020, *arXiv*: arXiv:2004.04788. Accessed: Sep. 01, 2024. [Online]. Available: <http://arxiv.org/abs/2004.04788>
- [4] X. Wang *et al.*, “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks,” Sep. 17, 2018, *arXiv*: arXiv:1809.00219. Accessed: Jul. 19, 2024. [Online]. Available: <http://arxiv.org/abs/1809.00219>
- [5] Mathworks, “Image Quality Metrics.” [Online]. Available: <https://www.mathworks.com/help/images/image-quality-metrics.html> (Accessed: Jul. 19, 2024.)
- [6] V. Laparra, J. Ballé, A. Berardino, and E. P. Simoncelli, “Perceptual image quality assessment using a normalized Laplacian pyramid,” *Electron. Imaging*, vol. 28, no. 16, pp. 1–6, Feb. 2016, doi: 10.2352/ISSN.2470-1173.2016.16.HVEI-103.
- [7] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 586–595. doi: 10.1109/CVPR.2018.00068.
- [8] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, “Image Quality Assessment: Unifying Structure and Texture Similarity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020, doi: 10.1109/TPAMI.2020.3045810.
- [9] H. Talebi and P. Milanfar, “NIMA: Neural Image Assessment,” *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3998–4011, Aug. 2018, doi: 10.1109/TIP.2018.2831899.
- [10] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” May 25, 2017, *arXiv*: arXiv:1609.04802. Accessed: Jul. 19, 2024. [Online]. Available: <http://arxiv.org/abs/1609.04802>
- [11] M. Bevilacqua, A. Roumy, C. Guillemot, and M. A. Morel, “Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding,” in *Proceedings of the British Machine Vision Conference 2012*, Surrey: British Machine Vision Association, 2012, p. 135.1-135.10. doi: 10.5244/C.26.135.
- [12] R. Zeyde, M. Protter, and M. Elad, “On Single Image Scale-Up using Sparse-Representation,” 2010.
- [13] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vancouver, BC, Canada: IEEE Comput. Soc, 2001, pp. 416–423. doi: 10.1109/ICCV.2001.937655.
- [14] E. Agustsson and R. Timofte, “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1122–1131. doi: 10.1109/CVPRW.2017.150.
- [15] R. Timofte *et al.*, “NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1110–1121. doi: 10.1109/CVPRW.2017.149.

- [16] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 5197–5206. doi: 10.1109/CVPR.2015.7299156.
- [17] A. Aakerberg, K. Nasrollahi, and T. B. Moeslund, “Real-World Super-Resolution of Face-Images from Surveillance Cameras,” Feb. 05, 2021, *arXiv*: arXiv:2102.03113. Accessed: Jul. 21, 2024. [Online]. Available: <http://arxiv.org/abs/2102.03113>
- [18] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell, “Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition,” in *CVPR 2011 WORKSHOPS*, Jun. 2011, pp. 74–81. doi: 10.1109/CVPRW.2011.5981881.
- [19] T. F. Vieira, A. Bottino, A. Laurentini, and M. De Simone, “Detecting siblings in image pairs,” *Vis. Comput.*, vol. 30, no. 12, pp. 1333–1345, Dec. 2014, doi: 10.1007/s00371-013-0884-3.
- [20] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. Van Knippenberg, “Presentation and validation of the Radboud Faces Database,” *Cogn. Emot.*, vol. 24, no. 8, pp. 1377–1388, Dec. 2010, doi: 10.1080/02699930903485076.
- [21] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 4396–4405. doi: 10.1109/CVPR.2019.00453.
- [22] C. Chen, D. Gong, H. Wang, Z. Li, and K.-Y. K. Wong, “Learning Spatial Attention for Face Super-Resolution,” *IEEE Trans. Image Process.*, vol. 30, pp. 1219–1231, 2021, doi: 10.1109/TIP.2020.3043093.
- [23] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *International Conference on Computer Vision (ICCV)*. 2015.
- [24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “A Semi-automatic Methodology for Facial Landmark Annotation,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, OR, USA: IEEE, Jun. 2013, pp. 896–903. doi: 10.1109/CVPRW.2013.132.
- [25] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” Feb. 26, 2018, *arXiv*: arXiv:1710.10196. Accessed: Jul. 23, 2024. [Online]. Available: <http://arxiv.org/abs/1710.10196>
- [26] C. Ma, Z. Jiang, Y. Rao, J. Lu, and J. Zhou, “Deep Face Super-Resolution with Iterative Collaboration between Attentive Recovery and Landmark Estimation,” Mar. 29, 2020, *arXiv*: arXiv:2003.13063. Accessed: Jul. 23, 2024. [Online]. Available: <http://arxiv.org/abs/2003.13063>
- [27] T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild,” in *2013 IEEE International Conference on Computer Vision Workshops*, Sydney, Australia: IEEE, Dec. 2013, pp. 354–361. doi: 10.1109/ICCVW.2013.54.
- [28] D. Kim, M. Kim, G. Kwon, and D.-S. Kim, “Progressive Face Super-Resolution via Attention to Facial Landmark,” Aug. 22, 2019, *arXiv*: arXiv:1908.08239. Accessed: Jul. 23, 2024. [Online]. Available: <http://arxiv.org/abs/1908.08239>
- [29] Juan-Ting Lin, Alexander Wu, Maciej Szpakowski, “SRGAN-tensorflow,” GitHub. [Online]. Available: <https://github.com/brade31919/SRGAN-tensorflow?tab=readme-ov-file> (Accessed: . 23, 2024.)
- [30] “Face Super Resolution,” GitHub. Accessed: Jul. 23, 2024. [Online]. Available: <https://github.com/ewrfcas/Face-Super-Resolution>
- [31] “OpenCV: Open Source Computer Vision Library.” [Online]. Available: <https://opencv.org> (Accessed: Jul. 26, 2024.)

- [32] “NumPy.” [Online]. Available: <https://numpy.org> (Accessed: Aug. 7, 2024.)
- [33] “PyTorch.” [Online]. Available: <https://pytorch.org> (Accessed: Aug. 7, 2024.)
- [34] “CUDA Toolkit.” [Online]. Available: <https://developer.nvidia.com/cuda-toolkit> (Accessed: Aug. 7, 2024.)
- [35] “TensorBoard: TensorFlow’s visualization toolkit.” [Online]. Available: <https://www.tensorflow.org/tensorboard> (Accessed: Aug. 4, 2024.)
- [36] “NVIDIA Jetson Nano.” [Online]. Available: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/> (Accessed: Aug. 7, 2024.)
- [37] “GStreamer.” [Online]. Available: <https://gstreamer.freedesktop.org> (Accessed: Aug. 7, 2024.)
- [38] “PyQt5.” [Online]. Available: <https://pypi.org/project/PyQt5/> (Accessed: Aug. 4, 2024.)
- [39] “Pillow.” [Online]. Available: <https://pypi.org/project/pillow/> (Accessed: Aug. 7, 2024.)

SAŽETAK

Diplomski rad opisuje algoritme za super rezoluciju slika korištenjem generativnih suparničkih mreža. U radu su detaljno testirani i analizirani postojeći algoritmi za super rezoluciju te je razvijeno vlastito rješenje za super rezoluciju lica. Za izradu rješenja korištene su tehnike strojnog učenja, poput generativnih suparničkih mreža koje se sastoje od mreže generatora i mreže diskriminatora. Mreže se treniraju istovremeno te neprestano uče jedna od druge. Mreža generatora nastoji generirati realistične visoko rezolucijske izlaze iz nisko rezolucijskih ulaza, dok mreža diskriminatora procjenjuje jesu li generirane slike dovoljno realistične. Razvijena mreža trenirana je na FFHQ skupu podataka, a razvijeni model testiran je na CelebAHQ skupu podataka. Nakon evaluacije rezultata na metrikama za procjenu kvalitete slike, model je implementiran na ugradbeni računalni sustav NVIDIA Jetson Nano gdje je testiran za rad u stvarnom vremenu. Pomoću NVIDIA Jetson Nano platforme omogućen je prijenos uživo sa kamere te je razvijeni model uspješno pokazao mogućnost stvaranja visoko rezolucijskih slika lica u stvarnom vremenu. Daljnji napredak ovog algoritma može se postići treniranjem neuronske mreže na višestrukom broju grafičkih procesora, što bi osiguralo veću dostupnu memoriju i poboljšalo ukupnu izvedbu modela.

Ključne riječi: strojno učenje, super rezolucija, generativne suparničke mreže, procjena kvalitete slika, NVIDIA Jetson Nano

SUPER RESOLUTION BASED ON GENERATIVE ADVERSIAL NETWORKS

ABSTRACT

The thesis describes algorithms for super resolution using generative adversarial networks (SRGAN). In this paper, the existing algorithms for super resolution were tested and analyzed in detail and a new solution for face super resolution was developed. Machine learning techniques, such as GANs were used to create the solution. GANs consist of two neural networks : generator network and discriminator network. Both networks are trained simultaneously and constantly learn from each other. The generator aims to generate realistic high-resolution outputs from low-resolution inputs, while the discriminator evaluates whether the generated images are realistic enough. The developed network was trained on the FFHQ dataset, while the developed model was tested on the CelebAHQ dataset. After evaluating the results using image quality assessment metrics (IQA), the model was implemented on a NVIDIA Jetson Nano embedded system where it was tested in real-time. Using the NVIDIA Jetson Nano platform, live stream from the camera was enabled, and the developed model successfully demonstrated the possibility of creating high-resolution facial images in real time. Further progress of this algorithm can be achieved by training the neural network on a multiple number of GPUs, which would ensure more available memory and improve the overall performance of the model.

Keywords: machine learning, super resolution, generative adversarial networks (GANs), image quality assesment metrics (IQA), NVIDIA Jetson Nano

ŽIVOTOPIS

Ines Momić rođena je 13. siječnja 2001. godine u Vinkovcima. Pohađala je Osnovnu školu „Mate Lovraka“ u Županji. Nakon završene osnovne škole, upisala je Prirodoslovnu – matematičku gimnaziju u Županji. Nakon završetka srednje škole, 2019. godine upisuje preddiplomski sveučilišni studij Elektrotehnike na FERIT-u, gdje se opredjeljuje za izborni blok „Elektroenergetika“. U rujnu 2022. godine završava preddiplomski studij i upisuje diplomski studij „Automobilsko računarstvo i komunikacije“ na istom fakultetu.

Potpis autora