

# Analiza metoda za povećanje slike

---

Jovanović, Ivan

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:472416>

Rights / Prava: [Attribution-NonCommercial 4.0 International/Imenovanje-Nekomercijalno 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-07-29**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

# **Analiza metoda za povećanje slike**

Ivan Jovanović

Split, rujan 2023.

# Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

## Analiza metoda za povećanje slike

Ivan Jovanović

### SAŽETAK

U ovom radu predstavljene su metode za povećanje slike, krećući od tradicionalnih metoda i algoritama interpolacije, pa sve do najsuvremenijih metoda temeljenih na dubokome učenju, odnosno AI metoda za povećanje slike. Za referencu jedne od najsuvremenijih metoda odabran je i detaljnije prikazan Real-ESRGAN model, koji se kasnije u sklopu eksperimenta ovoga rada usporedio s nekoliko konkurenata različitih arhitektura. Eksperimentom se htjela provjeriti valjanost trenutnih metoda na tržištu, a istovremeno i ispitati dostupnost alata za primjenu tih metoda za proces povećanja slike lokalno, odnosno općenito pristupačnost krajnjem korisniku za osobnu upotrebu. Rezultati su zadovoljavajući i ukazuju na očigledan konstantni napredak, no kako je većina uobičajenih problema super-rezolucije još prisutna, izgleda da smo još daleko od prave super-rezolucije.

**Ključne riječi:** povećanje slike, super-rezolucija, interpolacija, AI metode, Real-ESRGAN

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

**Rad sadrži:** 91 stranica, 37 grafičkih prikaza, 2 tablica i 97 literaturnih navoda. Izvornik je na hrvatskom jeziku.

**Mentor:** **Dr. sc. Saša Mladenović**, *redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Ocjenjivači:** **Dr. sc. Saša Mladenović**, *redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Dr. sc. Divna Krpan**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Dino Nejašmić**, *predavač Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Rad prihvaćen: rujan, 2023. godine

# Basic documentation card

Thesis

University of Split

Faculty of Science

Department of computer science

Ruđera Boškovića 33, 21000 Split, Croatia

## Analysis of image upscaling methods

Ivan Jovanović

### ABSTRACT

In this thesis methods for image upscaling were introduced, going from traditional methods and interpolation algorithms, all the way to state-of-the-art methods based on deep learning, respectively known as AI image upscaling methods. For reference as one of the state-of-the-art methods Real-ESRGAN model was chosen, which was also presented in detail, and later on as a part of the experiment of this thesis this model was compared with few competitors with different architecture. Experiment served to check the validity of currently available methods on market, as well as to test the accessibility of tools for applying these methods for image upscaling locally, that is, in general the availability to end user for personal usage. Results are satisfactory and they indicate an obvious and constant progress, but due to common super-resolution issues still being present, it would seem we are still far away from real super-resolution.

**Key words:** image upscaling, super-resolution, interpolation, AI methods, Real-ESRGAN

Thesis deposited in library of Faculty of science, University of Split

**Thesis consists of:** 91 pages, 37 figures, 2 tables and 97 references

Original language: Croatian

**Mentor:** **Saša Mladenović, Ph.D.** *Full Professor of Faculty of Science, University of Split*

**Reviewers:** **Saša Mladenović, Ph.D.** *Full Professor of Faculty of Science, University of Split*

**Divna Krpan, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Dino Nejašmić, Lecturer of Faculty of Science, University of Split**

Thesis accepted: September 2023.

# IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom „*Analiza metoda za povećanje slike*“ izradio samostalno pod voditeljstvom redovitog profesora dr.sc. Saše Mladenovića. U radu sam primijenio metodologiju znanstvenoistraživačkog rada i koristio literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u diplomskom radu na uobičajen, standardan način citirao sam i povezao s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student:

Ivan Jovanović





## Sadržaj

Uvod .....	1
1. Pregled metoda za povećanje slike .....	3
1.1. Tradicionalne metode .....	4
1.2. Uvod u klasifikaciju AI metoda.....	10
2. Super-rezolucija.....	12
2.1. Metode metrike SR.....	14
2.2. Referentni skupovi podataka .....	21
3. Metode super-rezolucije .....	23
3.1. Metode povećanja uzorkovanja – US metode .....	23
3.2. Mreže dubokog učenja.....	25
3.3. Funkcije gubitka .....	32
3.4. Strategije učenja .....	35
3.5. Okviri SR.....	37
3.6. Primjeri najsuvremenijih metoda SR.....	40
3.7. Super-rezolucija bez nadgledanja.....	43
3.8. Primjene metoda SR u specifičnim područjima .....	45
4. Real-ESRGAN .....	48
4.1. Općenito .....	48
4.2. Metodologija.....	50
4.3. Eksperimenti .....	56
5. Eksperiment .....	61
5.1. Izbor alata .....	61
5.2. Dokumentacija.....	64
5.3. Implementacija .....	67
5.4. Evaluacija .....	69

5.5. Rezultati i rasprava .....	72
Zaključak .....	75
Literatura .....	76
Skraćenice.....	86
Privitak .....	87
Popis slika.....	90

# Uvod

Jedan od mnogih problema, odnosno bolje reći izazova, kojega imamo u digitalnome svijetu, a koji postaje i sve relevantniji sa samim razvojem tehnologije i njenim potrebama, no naravno, i potrebama samih ljudi, koji su ipak krajnji korisnici te tehnologije, je proces povećanja slike. Napomena odmah na početku za sami proces povećanja slike, pod ovim procesom indirektno se podrazumijeva i video(film), budući znamo da je to ništa drugo nego niz od više slika prikazanih u određenom vremenu, odnosno zapravo se pod ovim procesom indirektno misli na bilo koji vizualni sadržaj, koji se može prikazivati na nekakvom zaslonu, dobiven digitalnom tehnologijom. Da, to mogu na kraju biti i digitalne igrice ili bilo koji drugi oblik računalne grafike. No ipak, kako je slika ono najosnovnije i od toga sve kreće, fokus ovoga rada će biti na samoj slici, i pod pojmom povećanje slike misliti će se na sliku, osim ako drugačije nije navedeno. Potreba za povećanjem slika je postojala odavno, dapače sami koncept povećanja veličine slike je postojao i prije digitalnog doba, no uglavnom samo u kontekstu umjetnosti, fotografije i tiska. U sklopu ovoga rada, nas se ipak tiče samo onaj dio koji se odnosi na digitalne tehnologije.

Primjera potrebe i primjene povećanja slika je mnogo. Štoviše, s razvojem tehnologije danas imamo toliko puno razloga i potreba za primjenom procesa povećanja slike na tolikoj razini da gotovo da postaje apsolutna nužnost i potreba na razini svakodnevnice, a u skoroj budućnosti će to vrlo vjerojatno biti i istina, ako već i nije. Jedan od osnovnih primjera danas bi bio razvoj zaslona s visokim rezolucijama kao što su to 4K, 8K itd., dokle sadržaj, tipa neka slika ili film, koji se prikazuje na tim zaslonima vrlo često je znatno manjih rezolucija. Konkretno, želite prikazati film rezolucije 720p ili 1080p na zaslonu rezolucije 4K, kako je to riječ o razlici u veličini rezolucije omjera 3:1 u odnosu filma na zaslon, logično je da će slika biti mutna, pikselizirana i ne baš ugodna ljudskome oku ako je razvučete preko cijelog zaslona, odnosno pokušate forsirano prilagoditi toj većoj rezoluciji. Nadalje, u slučajevima video igara, procesorska snaga i hardverski resursi koji bi bili potrebni za glatko igranje pri jako velikim rezolucijama često nisu lako ostvarivi većini ljudi, ili uopće, te bi bilo izvrsno kada bi mogli pokrenuti igre u manjim rezolucijama da troše manje resursa, a prikazivati ih u većim rezolucijama bez prevelike razlike u opterećenju. Primjer izvedbe ove ideje na malo specifičan način već imamo od strane NVIDIA-e s njihovim DLSS-om, koji je pokazao dosta

dobre rezultate, zapravo dajući na uvid da je ova ideja izvediva. [1] Jednako tako i za sve starije video igrice s lošijom grafikom i manjom rezolucijom da budu prikazane u boljoj grafici i većoj rezoluciji. Još bih spomenuo jedan primjer gdje bi povećanje slike moglo uštediti i promet internetom pored samih hardverskih resursa, vezan s *online streamingom*, gdje bi bilo poželjno kada bi mogli *streamati* sadržaj online, ili pak generalni *upload* na neki servis tipa YouTube, s manjom rezolucijom, odnosno manjim *bitrateom*, a da se ipak taj sadržaj onda na strani samog korisnika, ili možda bolje od strane servisa *online*, može prikazati u punoj rezoluciji ili povećati. Na ovaj način bi se znatno olakšalo opterećenje onoga tko želi *streamati*, i zapravo onoga tko želi gledati taj sadržaj na samome kraju, naravno ovisno o izvedbi. Donekle ovo možda zvuči kao fantazija, no već je dosta ljudi i pomislilo na ovu ideju, jer iako živimo u 2023. godini, nemaju svi ljudi jednake hardverske i internetske uvjete. Dapače, s ovime bi olakšali internet promet i skladištenje općenito, budući kako je već spomenuto prethodno, rezolucije sadržaja su sve veće, posljedično s time i njihove memorijske veličine, i one zapravo drastično rastu, npr. video rezolucije 1080p neka iznosi 7MB, u 2K je već 23MB, a u 4K je 40MB, što je zapravo povećanje od gotovo 6 puta. Naravno, ovisno o ostalim specifikacijama videa, razlike u memorijskim veličinama naspram rezolucije često su i veće. Ima još puno primjera, od tipičnog poboljšanja starijih slika koje ljudi čuvaju kao uspomene, do područja poslova u kojima je to poprilično bitno, kao medicina i promet, otprilike sve što vam može pasti na pamet a da će se prikazivati na nekom digitalnom zaslonu, zapravo se indirektno tiče procesa povećanja slike.

Uglavnom, osnovna ideja procesa povećanja slike je da imate sliku manje veličine, točnije rezolucije, budući je ipak riječ o digitalnoj slici rasterske grafike, koju biste htjeli povećati da bude jasnije vidljiva i preglednija. Naravno, trebalo bi pripaziti na to i da imate što manji gubitak kvalitete slike, idealno bez ikakvog gubitka kvalitete slike i zapravo s njenim poboljšanjem. Zvuči poprilično jednostavno, no sama činjenica kako se ljudi dan danas s ovime muče govori suprotno. Napredak kroz godine je očigledan, no ima još puno prostora za ići naprijed. Zadnjih nekoliko godina smo svjedočili znatnom razvoju umjetne inteligencije, koji općenito iz godine u godinu daje sve bolje rezultate, i koja je omogućila razvoj raznoraznih tehnologija, ona je jednako tako imala i veliki utjecaj na one tehnologije koje se vežu za proces povećanja slika. U ovome radu će tako biti predstavljene metode i tehnike kojima se danas koristimo za povećanje slika, kako one rade, te će se provjeriti koliko su one zapravo dobre, i na samome kraju koliko su pristupačne običnome korisniku.

# 1. Pregled metoda za povećanje slike

Kako je kratko spomenuto i u samome uvodu već, osnovna ideja ovog procesa je da želimo promijeniti rezoluciju neke slike na veću. Kada bi bili malo specifičniji, ono što želimo napraviti je zapravo povećati broj piksela na slici, a da što bolje zadržimo vjerodostojnost sadržaja koji slika predstavlja. Drugim riječima, cilj procesa povećanja slike je znači povećanje njene rezolucije (broja piksela), poželjno s poboljšanjem njene generalne kvalitete ili barem zadržavanjem trenutne kvalitete, tj. ono najbitnije bez gubitaka detalja i kvalitete. Kao i uvijek u digitalnome svijetu, trebamo pripaziti i da sačuvamo vrijeme i resurse potrebnih za taj proces, odnosno općenito na optimizaciju. Ovisno o tome kako radimo sve ovo, metode za povećanje slike se mogu podijeliti u dvije osnovne kategorije; tradicionalne (klasične) i metode koje koriste tehnologiju umjetne inteligencije (AI, eng. *Artificial Intelligence*). Zanimljivo je to da zapravo kod nekih metoda, pogotovo onih vezanih za umjetnu inteligenciju, prvo sliku umanjujemo da bi je mogli povećati, malo zvuči kontradiktorno ali ima smisla, što će se vidjeti detaljnije u samome poglavlju vezanom za te AI metode. Inače, najučestalija tehnika koja se koristi kod tradicionalnog povećanja slike je interpolacija, koja koristi matematičke algoritme da odredi vrijednosti piksela koji nedostaju na slici, odnosno koji se nadodaju slici. Interpolacija daje donekle prihvatljive rezultate, no oni često rezultiraju popraćeni s problemom pikselizacije i gubitkom detalja na slici, što baš i nije nešto što želimo postići, odnosno to želimo izbjeći. S druge strane imamo AI metode koje koriste strojno učenje, duboko učenje i duboke neuronske mreže pored ostalog, da bi postigli svoje rezultate, odnosno dodali te nove piksele na sliku i povećali sliku. AI metode se danas još često poistovjećuju s nazivom super-rezolucija (SR, eng. *Super-resolution*), koji označava cijelo jedno područje umjetne inteligencije koji se bavi procesom povećanja slike, restauracijom i slično. Slike dobivene AI metodama su često puno boljih kvaliteta nego one od tradicionalnih metoda, tako da ih većina ljudi, pogotovo u industriji grafičkih dizajnera i umjetnika, danas zapravo koristi. AI metode usput mogu i popraviti druge probleme i greške na slikama, tipa uklanjanje nepotrebnog šuma, izoštravanje, ispravljanje dijelova koji nedostaju na slici i slično. Naravno, ovo ovisi o tome kako su one izvedene i o kojem modelu je riječ, ali uglavnom imamo modele posebno posvećene za određene zadatke, rijetko kada one koji rade po više stvari odjednom, i još manje da ih rade jednako dobro, barem zasad. Važno je isto napomenuti da AI metode su dosta zahtjevnije u odnosu na tradicionalne, pogotovo što se tiče hardverskih resursa i vremena potrebnog, a detaljnije o ovome u

poglavlju posvećenom njima. Sve uvijek ima svoje prednosti i mane, jednako to vrijedi i za ove metode, tako da ćemo u nastavku vidjeti kada je možda bolje koristiti tradicionalne a kada one metode koje koriste AI tehnologije. Prvo malo o tradicionalnim metodama, budući je ipak to bila i startna točka za AI metode, zapravo dosta njih i koristi dijelom te tradicionalne metode poduprte s umjetnom inteligencijom. Naravno, kako i tradicionalnih metoda ima dosta, a AI metoda još i više, izdvojene su samo one najrelevantnije. [2, 3, 30]

## 1.1. Tradicionalne metode

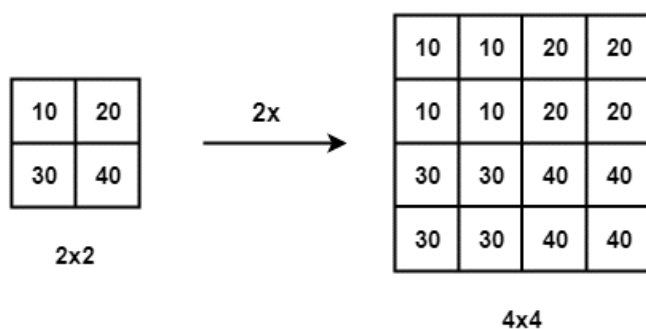
Tradicionalne metode, koje se još mogu pronaći i pod nazivom klasične ili matematičke metode, koriste matematičku osnovu, odnosno matematičke algoritme, da bi *rastegli* postojeće piksele na nove, ili popunili nove prazne piksele s novim vrijednostima prema nekim pravilima, pri povećanju slika. Donekle logično, budući već znamo kako su same slike zapravo ništa drugo nego matrice dimenzija oblika  $N \times M$ , gdje  $N$  predstavlja broj stupaca a  $M$  broj redaka, u kojima se nalaze vrijednosti pojedinih piksela za prikaz boje u tome pikselu na toj slici, a za ostalo se sjetimo samo kako rasterska grafika radi općenito. Najveći problem ovih metoda je definitivno to kako rade dosta kruto i bez puno prilagodbe, možda bolje reći bez konkretne adaptacije raznolikosti pojedinih slika, tako da često rezultiraju s tom zamućenosti i izobličenosti, pogotovo što je slika kompleksnija. Većina ovih metoda spada u kategoriju tehnika bez adaptacije. Ima i iznimki, odnosno adaptivnih tehnika u sklopu interpolacije, no one su kompleksnije i rjeđe. Znači, imamo tradicionalne metode koje se mogu podijeliti u tehnike bez adaptacije i s adaptacijom. Ipak, zbog generalno jednostavne implementacije one su dosta brze i ne traže puno resursa, barem u usporedbi s AI metodama. Gdje nam nije moguća velika procesorska moć i ostali resursi, te ako je riječ o jednostavnijim slikama, one i dan danas mogu poslužiti dobro. [2, 30]

### Interpolacija najbližeg susjeda

Pojam interpolacije je spomenut već par puta, a to je čisto naziv procesa kojim se neka manja slika želi uvećati na način da joj se poveća broj piksela, i ovaj pojam se direktno veže za tradicionalne metode. Algoritam koji se koristi u ovoj metodi je algoritam najbližeg susjeda, za kojeg ste vjerojatno čuli i u drugim područjima znanosti već, odnosno nekim njegovim izvedbama bilo u matematici, programiranju ili nekakvoj analitici. Uglavnom, u sklopu interpolacije i samog povećanja slike, možda je malo drukčiji ali generalni princip je isti, a

to je da se traži vrijednost najbližeg susjeda da bi se ta vrijednost mogla dalje pridonijeti točki za koju se tražio taj susjed, i slično.

Ovo je općenito najjednostavnija tradicionalna metoda i ujedno najbrža, znači s najmanjom procesorskom zahtjevnošću i vremenom obrade. Načini na koji ona radi su se međusobno malo razlikovali u ovisnosti o kojoj je verziji riječ, no dovoljno je znati dvije osnovne ideje za razumjeti način rada. Prvo i ono najosnovnije, ideja je da imamo dvije slike, onu originalnu i tu novu sliku koja bi trebala predstavljati povećani original. Zatim da tu novu sliku popunjavamo vrijednostima piksela uspoređujući ih s originalnom, odnosno po principu algoritma najbližeg susjeda tražili bi vrijednost piksela na originalnoj slici koji je bio najbliži onome na uvećanoj pa mu pridonijeli tu vrijednost. Ta druga izvedba bi bila da gledamo na to kao da proširujemo prostor originalne slike, i te nove praznine popunjavamo vrijednostima dobivenih prateći algoritam najbližeg susjeda. Poanta je da, neovisno o izvedbi, novom nepoznatom pikselu želimo dodijeliti vrijednost onog poznatog piksela njemu najbližeg. Za ovo odrediti trebamo prvo odrediti središta poznatih piksela i znati njihove koordinate, zatim uspoređujemo središta novih nepoznatih piksela i njihove udaljenosti od središta poznatih piksela. Na kraju uzimamo udaljenosti tih dvaju središta s najmanjom vrijednosti, te tom novom pikselu dajemo vrijednost koja se nalazila unutar tog poznatog piksela. Na Slici 1. možemo vidjeti jedan primjer kakav rezultat nam daje ovaj način povećanja slike. Odmah možemo i uvidjeti potencijalne probleme ove metode, a to bi bili ti dosta nazubljeni rubovi objekata na slici i sama pikselizirana slika, budući nije bilo dodatne adaptacije za korigiranje dijelova i detalja na slici. [6]



Slika 1. Primjer povećanja slike i vrijednosti njenih piksela tehnikom najbližeg susjeda [6]

Najbolja primjena ove tehnike bi bila na slikama s objektima jednostavnih oblika, kao što su to npr. QR i bar-kodovi, gdje uglavnom imamo kvadratne oblike i slično, i još su u crno-bijeloj boji što nadalje pojednostavljuje stvari.

## Bi-linearna interpolacija

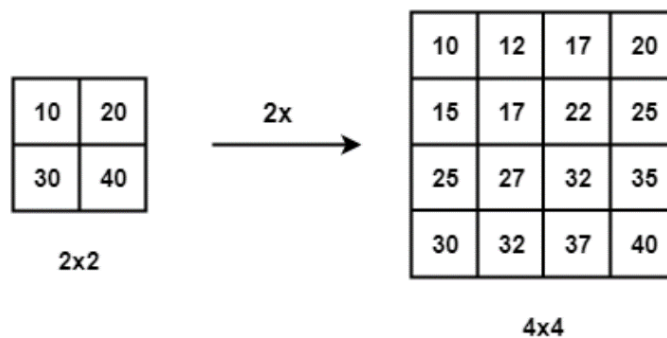
Direktno se nastavlja na ideju prethodne tehnike, samo malo kompleksnije. Za razliku od tehnike najbližeg susjeda, gdje smo gledali samo jednog susjeda da zadovoljimo uvjet algoritma, ovdje gledamo njih četiri. Zašto, jer je riječ o interpolaciji u 2D, odnosno linearna interpolacija napravljena u dvije dimenzija ili dva smjera. Sama linearna interpolacija je već bolja nego čisto gledanje najbližeg susjeda, ali kako je riječ o slikama koje su zapravo dvodimenzionalne, moramo napraviti tu linearnu interpolaciju u dva smjera, zato se ova tehnika i zove bi-linearna interpolacija. U nastavku će biti prikazan princip rada ove tehnike na primjeru računanja vrijednosti piksela uz pomoć same bi-linearna interpolacije, no kreće se s pretpostavkom da je rad linearne interpolacije već poznat prethodno.

Prvo za sami algoritam bi-linearne interpolacije i njegove korake:

1. Identificiraj koordinate – moramo prvo odrediti koordinate točke za koju želimo izvršiti bi-linearnu interpolaciju
2. Izračunaj težinske faktore – računamo težinske vrijednosti točke koju promatramo u odnosu na četiri rubne točke, konkretno misli se na vrhove pravokutnika(prostora) koji se promatra
3. Napravi linearnu interpolaciju u horizontalnom smjeru – računamo vrijednosti linearne interpolaciju u odnosu na gornji i donji rub tog prostora
4. Napravi linearnu interpolaciju u vertikalnom smjeru – računamo vrijednosti linearne interpolacije u odnosu na lijevi i desni rub, odnosno na točke dobivene prethodnom linearnom interpolacijom

Način na koji se dobivaju težinski faktori se zna razlikovati od alata do alata, ovisno o načinu distribucije težina, što npr. ovisi o tome koji vrhovi se gledaju prvo i slično, i eventualno o redosljedu linearnih interpolacija, ali princip je isti. Na Slici 2. možemo vidjeti kako dobijemo drukčiji rezultat ovom tehnikom na istoj slici u usporedbi s tehnikom najbližeg susjeda. [5]





Slika 2. Primjer povećanja slike i vrijednosti njenih piksela tehnikom bi-linearne interpolacije [5]

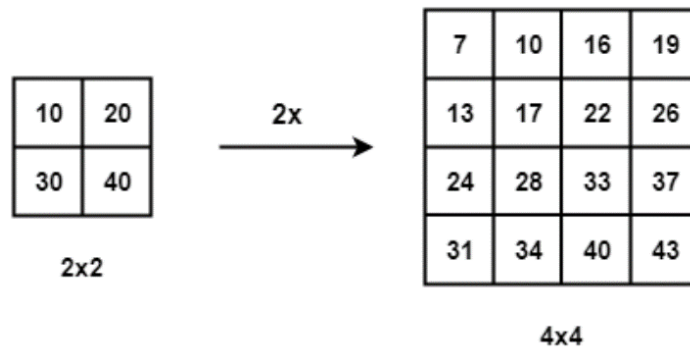
Odmah je isto vidljivo kako ova tehnika daje finije i manje upadljive rubove nego tehnika najbližeg susjeda, ali i dalje nije dovoljno dobra na kompleksnijim slikama, pogotovo onima s puno sitnih detalja i slično.

### Bi-kubična interpolacija

Koncept je jako sličan prethodno navedenim tehnikama, jasno vidljivo i iz samog naziva, s glavnom razlikom da interpolaciju činimo kompleksnijom. To radimo na način da više ne gledamo samo četiri najbliža susjeda kao što smo to radili u bi-linearnoj tehnici, već sada gledamo njih šesnaest, odnosno 4x4 susjedstva. Podjela težinskih faktora se isto radi na malo drukčiji način, no ovo opet ovisi od alata do alata koji se koristi za obradu, npr. Matlab ili OpenCV i slično. Kako vidimo da se u ovoj tehnici koristi matrica susjedstva od 4x4 veličine, možemo pretpostaviti odmah da za ovu tehniku postoji i minimalna preporučana veličina slike da bi nam ova tehnika dala konkretne rezultate, odnosno uopće se ne isplati koristiti ovu tehniku ako ne možemo zadovoljiti niti apsolutni minimum, što bi logično bilo veličina slike od barem 4x4 piksela. Preporuča se inače raditi na slikama s dimenzijama od 256x256, 512x512 i naviše za ovu tehniku.

Koraci algoritma bi-kubične interpolacije su praktički identični bi-linearnoj interpolaciji, samo što sada promatramo veći broj točaka i ne koristimo linearnu interpolaciju nego kubičnu interpolaciju. Znači, redoslijed je isti, samo što promatramo veći broj točaka te je zato riječ o kubičnim funkcijama i kubičnoj interpolaciji. Jedini dodatni korak koji bi trebali uzeti u obzir bi bio slučaj kada vrijednosti prelete raspon vrijednosti koji pikseli mogu poprimiti, a ovo ovisi o tipu slike, tako da npr. gdje imamo 8bitnu sliku moramo pripaziti da vrijednosti budu od 0-255, pa u slučajevima kada budu van tog raspona da ih onda ograničimo na jedan od ta dva ekstremna slučaja, bilo 0 ili 255. Donekle razumljiva posljedica budući je riječ o kubičnim funkcijama. Inače, neovisno i o ovom dodatnom koraku

koji je često potreban za provjeru, jasno je da kako gledamo veći broj točaka i računamo kompliciranije funkcije, ova tehnika zahtjeva veću procesnu moć nego li prethodne, ali zato i daje bolje rezultate. Na Slici 3. sada možemo vidjeti kakvi rezultat daje ova metoda povećanjem iste slike kao i prethodne tehnike za usporedbu. [4]



Slika 3. Primjer povećanja slike i vrijednosti njenih piksela tehnikom bi-kubične interpolacije [4]

Ova tehnika unatoč većoj kompleksnosti nego prethodne je svejedno dosta brza, i ono važnije je da daje zapravo poprilično dobre rezultate. Naravno, i dalje ima svojih problema, kao oni tipični artefakti na slikama i halo efekt. No kako ipak balansira vrijeme potrebno za obradu i kvalitetu samog rezultata, ona ima široku primjenu danas, pa čak i u alatima kao Adobe Photoshop. [3]

### Lanczos interpolacija

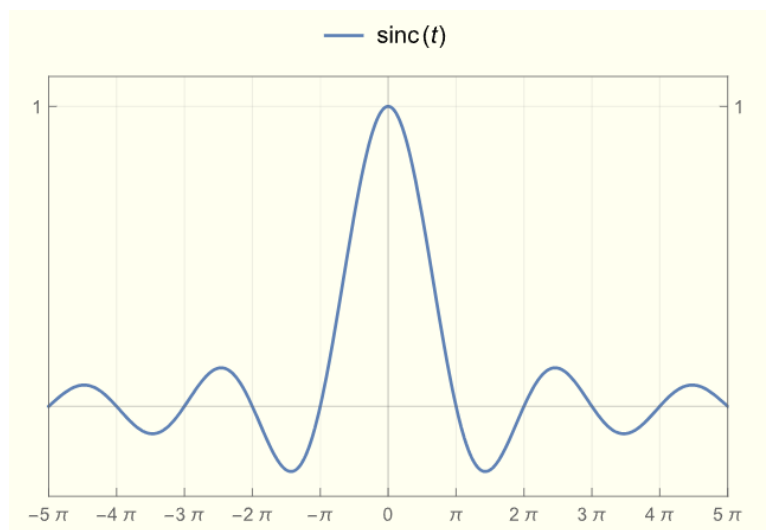
Isto jedna jako popularna tehnika, no nažalost kako je dosta kompleksnija nego prethodno spomenute tehnike, ne samo u smislu razumijevanja kako radi, već potrebne procesorske moći i resursa za izvršavanje, barem u usporedbi s dosad navedenim, ne primjenjuje se toliko u normalnoj upotrebi. Pod normalnom upotrebom, misli se na krajnjeg korisnika u smislu osobne i svakodnevne upotrebe. Ona se uglavnom koristi u specijaliziranim područjima, kao što je to pregled medicinskih snimaka organa i slično.

Za osnovu ove tehnike koristi se jedna funkcija koja se naziva sinc funkcija, skraćeno za latinski *sine cardinal*. Sinc se definira kao

$$\text{sinc}(t) = \frac{\sin t}{t} \quad (1)$$

gdje je  $t$  neki realan broj.

Ova sinc funkcija postiže maksimum za  $t=0$ , znači vrijedi  $\text{sinc}(0) = 1$ , ili bolje vidjeti direktno na Slici 4. ispod.



Slika 4. Primjerak grafa *sinc* funkcije [7]

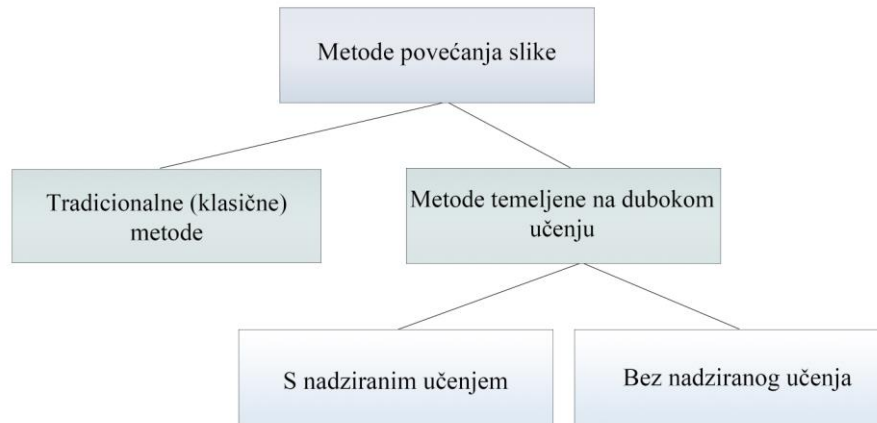
Poprilično pojednostavljeno, svrha ove funkcije je da bi se kreirali određeni okviri, bolje reći filteri, da bi se mogle bolje raspodijeliti težinske vrijednosti potrebne za samu interpolaciju (sjetimo se prethodnih metoda i koraka u algoritmima ondje). Ovo je zahvaljujući tome kako ona ima dobra svojstva interpolacijske funkcije, te se nadalje može primijeniti Fourierova transformacija u obradi skupa s njome, budući je zapravo sve riječ o valovima, odnosno signalima koji predstavljaju vrijednosti piksela na slici. Naravno, s ovime se nadalje komplicira i zahtjeva veća procesorska moć i vrijeme potrebno za obradu, no to se isplati pogotovo za područja kao medicina, gdje im kvaliteta slike veći prioritet. [7]

Sve dosad navedene tradicionalne metode su one koje spadaju u kategorije bez adaptacije, jer su koristile fiksne parametre i ostalo. Kako je bilo i spomenuto, ima i adaptivnih, neke od primjera bi bili usmjerena interpolacija (DI, eng. *Directional Interpolation*) i njene izvedenice, adaptivna bi-kubična interpolacija, interpolacija bazirana na uzorku itd. [31] One su jako specifične, često nepraktične zbog kompleksnosti i dublje analize potrebne da bi se uklonili problemi koji se uobičajeno javljaju s tradicionalnim metodama, npr. zamućenost, artefakti, oštri rubovi i ostalo. Isto je bitno napomenuti kako su radile samo za specifične tipove slika, npr. slike morskog dna, snimke u medicini, slike svemira itd. Ukratko, kada se već dođe do ove razine, bolje je odmah prijeći na metode za povećanje slike koje koriste AI, jer su relevantnije i zapravo će se govoriti o jako sličnim konceptima, a ovdje su i ti koncepti bolje riješeni s ne tako velikom razlikom u vremenu i resursima potrebnim. Napomena, ovo se misli već nakon uvježbavanja potrebnih modela i slično, budući je to ipak najzahtjevniji dio za odraditi u metodama AI, a detaljnije o svemu u nastavku.

## 1.2. Uvod u klasifikaciju AI metoda

Kako smo u tradicionalnim metodama imali naglasak na interpolaciji i kako su se sve one fokusirale na taj pojam, za AI metode tako imamo super-rezoluciju, pored mnogih drugih pojmova koji su bitni za AI metode. Već smo spomenuli super-rezoluciju kao područje umjetne inteligencije koje se bavi povećanjem slika niskih rezolucija (LR, eng. *Low-resolution*) na slike visokih rezolucija (HR, eng. *High-resolution*), no sami koncept super-rezolucije je već postojao od 1974. godine, gdje je R.W. Gerchberg prvi put predstavio super-rezoluciju u časopisu *Optica Acta*, časopis za područje optike, kao način za poboljšanjem rezolucije optičkog sustava izvan granice difrakcije, što je više zapravo u kontekstu same optike, a time i fizike. [8] Kroz povijest koncept super-rezolucije je dobivao drugačije značenje ovisno o primjenama i razvoju ostalih tehnologija i područja znanosti. Tako da nakon uključivanja u računalnu grafiku, medicinsko snimanje, sigurnost i nadzor općenito, a pogotovo nakon razvoja umjetne inteligencije za koju se danas najviše i veže, dobila je značenje koje ima danas. [2] Čemu sve ova kratka priča o razvoju super-rezolucije? Čisto da se da na uvid koliko daleko u prošlost su ljudi bili već mislili o ovome, konkretno povećanju slike, no s nedostatkom tehnologije u to vrijeme nisu je još mogli ostvariti kako su je zamislili. Istina, i dan danas se još ljudi muče s istim problemom, ali samo gledajući na napredak posljednjih dvadeset godina, a tek pogotovo zadnjih pet godina, napredak je poprilično zastrašujući. Trenutno smo zapravo na razini gdje samo unutar jedne godine vidimo velike napretke u umjetnoj inteligenciji, za koje je prije recimo trebalo po barem par godina.

Što se tiče podjele AI metoda, najjednostavnije ih je podijeliti ovisno o dubokom učenju, da li su s nadziranom učenjem ili bez nadziranog učenja. Dalje ih onda možemo dijeliti ovisno o arhitekturi i vrsti neuronskih mreža, metodama povećanja uzorkovanja (US, eng. *Upsampling*), strategijama učenja, SR okvirima i drugim poboljšanjima na koje se ona odnose itd. Ukratko, podjela se može poprilično zakomplicirati, jer ima više karakteristika prema kojima se one razlikuju. [2, 30]



Slika 5. Klasifikacija metoda za povećanje slike - SR

Na Slici 5. možemo vidjeti sažetak podjela metoda za povećanje slike (super-rezolucije), odnosno tradicionalnih (klasičnih) metoda i AI metoda (metoda baziranih na dubokom učenju) za povećanje slike.

U nastavku ovog rada će na naglasak biti stavljene metode s dubokim učenjem, a razlog zašto su prethodno bile obrađene najosnovnije tradicionalne metode (tehnika) je upravo zato što su AI metode imale njih za osnovu, i/ili barem koriste te tehnike u nekom pogledu.

No prije svega toga, prvo bi trebali spomenuti detaljnije pojmove po kojima se one razlikuju i koji su općenito jako bitni za AI metode, a to su:

- Metrike kvalitete slike – načini na koji se ocjenjuju, mjeri i uspoređuju rezultati kvalitete povećanja slika,
- SR metode – npr. metoda koje uključuje uklanjanje šuma,
- Skup podataka (eng. *dataset*) – na kojem se skupu podataka radilo, da li su to ljudi, auta, životinje, crno-bijele slike i slično,
- Primjene - s kojom svrhom je napravljena koja metoda, npr. povećanje slika registracijskih tablica automobila u prometu i slično,
- Izazovi – ovdje je više riječ o natjecajima i na zadacima gdje se traži poboljšanje, npr. fokus na napretku restauracije.

## 2. Super-rezolucija

Problem kojim se super-rezolucija bavi, i koji joj je glavni fokus, je dobivanje HR slike iz ulazne LR slike, a ovo je zapravo ništa drugo nego i osnova cijelog procesa povećanja slike. Kako zapravo znače gotovo jedno te isto, često ljudi znaju poistovjetiti proces povećanja slike s pojmom super-rezolucije, što se zapravo može smatrati i jednim oblikom miskoncepcije. Razlog zašto to nije potpuno ispravno je jer zapravo ima bitne razlike između tih dvaju pojmova. Super-rezolucija nije samo povećanje slike kao što su to tradicionalne metode bile s interpolacijom, već ona ima sve one dodatne stvari kao čišćenje šuma, uklanjanje artefakata, popunjavanje rupa na slici pretpostavljajući koji dijelovi nedostaju na slici ovisno o onome što je prikazano, što ima veze i s detekcijom objekata, i još mnogo toga. Puno je ispravnije poistovjetiti pojam super-rezolucije s AI metodama za povećanje slike, jer u suštini one i jesu to. U svrhu toga će se dalje i u nastavku rada obrađati AI metodama s nazivom super-rezolucija. Tako da, super-rezolucija je novija i puno naprednija metoda od onih tradicionalnih s interpolacijom, koja koristi raznorazne koncepte iz umjetne inteligencije (kao što su to strojno i duboko učenje, neuronske mreže i ostalo) da bi povećala sliku i istovremeno poboljšala kvalitetu slike.

U osnovi, načelni princip s kojim se super-rezolucija bavi je dobivanje HR slike iz ulazne LR slike, a ta LR slika se može predstaviti kao izlaz degradacijske funkcije prikazane izrazom ispod

$$I_{xLR} = d(I_{yHR}, \partial) \quad (2)$$

gdje je  $I_{xLR}$  LR slika,  $I_{yHR}$  HR slika,  $d$  degradacijska funkcija SR koja je odgovorna za konverziju HR slike (referentna slika) u LR sliku, a  $\partial$  predstavlja parametre unosa za tu degradacijsku funkciju. Ti parametri su uglavnom faktor povećanja, tip zamućenosti i šum. U samoj praksi, proces degradacije i odgovarajući parametri su nepoznati, već su poznate samo LR slike koje se koriste da bi se dobile HR slike metodom SR. Tako da ono za što je proces SR rezolucije odgovoran je zapravo predviđanje inverzne funkcije od degradacijske funkcije  $d$ , odnosno da je  $g = d^{-1}$ , vidljivo u izrazu

$$g(I_{xLR}, \delta) = d^{-1}(I_{xLR}) = I_{yE} \approx I_{yHR} \quad (3)$$

gdje je  $g$  funkcija SR, a  $\delta$  predstavlja unosne parametre za nju, a  $I_{yE}$  je procijenjena HR slika koja odgovara ulaznoj  $I_{xLR}$  slici. Ovdje je bitno napomenuti kako SR funkcija

predstavljena u izrazu (3) nije dobro postavljena, odnosno kako  $g$  funkcija nije injektivna imamo beskonačno mnogo mogućnosti  $I_{yE}$  za koju bi vrijedilo  $d(I_{yE}, \partial) = I_{xLR}$ .

Proces degradacije za unosne LR slike je nepoznat, i on je pod direktnim utjecajem brojnih faktora kao što su to sensorima izazvani šum, artefakti koji nastaju zbog kompresije s gubitkom, pjegavi šum, zamućenje pokreta, slike van fokusa i ostalo. Zbog svega ovoga, u većini metoda i studija često se koristila funkcija smanjivanja uzorkovanja (DS, eng. *Downsampling function*) kao degradacijska funkcija, prikazana izrazom

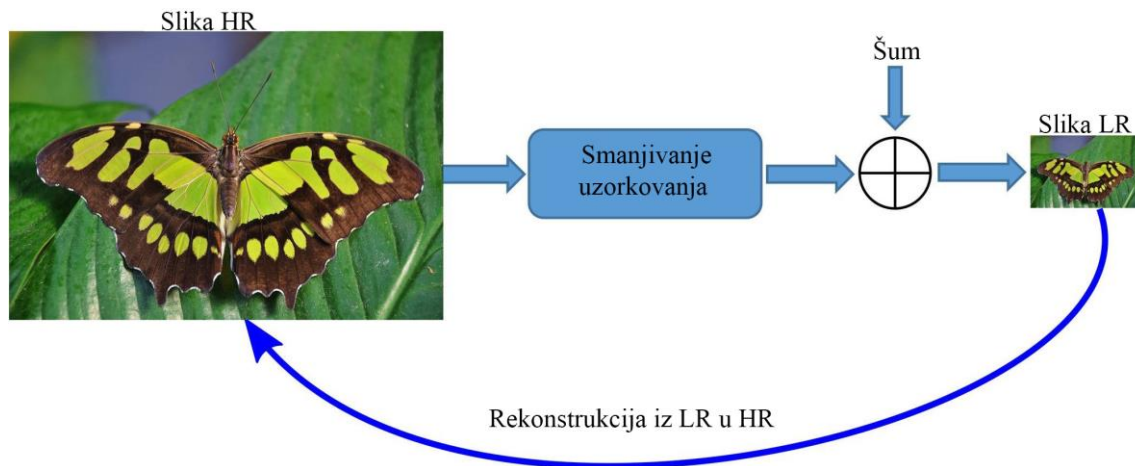
$$d(I_{yHR}, \partial) = (I_{yHR}) \downarrow_{s_f}, \{s\} \subseteq \partial \quad (4)$$

Gdje je  $\downarrow_{s_f}$  operator smanjivanja uzorkovanja, a  $s_f$  faktor skaliranja. Funkcija koja je predstavljala funkciju smanjivanja uzorkovanja često se je uzimala da je bila bi-kubična interpolacija s *antialiasing*-om. Inače, sada se da vidjeti zašto su bile spomenute osnovne tradicionalne metode, jer su bile osnova za funkcije koje su se koristile u SR, odnosno samim AI metodama za povećanje slike. Dalje su sve samo bile varijacije više manje iste ideje, no evo prikazanog još jednog načina gdje bi se dodale još neke operacije u izraz funkcije za smanjivanje uzorkovanja, npr.

$$d(I_{yHR}, \partial) = (I_{yHR} \otimes \kappa) \downarrow_{s_f} + n_\sigma, \{\kappa, s, \sigma\} \subseteq \partial \quad (5)$$

gdje  $I_{yHR} \otimes \kappa$  predstavlja konvoluciju HR slike  $I_{yHR}$  s kernelima za zamućivanje  $\kappa$ , a  $n_\sigma$  predstavlja dodatni bijeli Gaussian šum s standardnom devijacijom od  $\sigma$ . Ovaj izraz (5) i primjer smanjivanja uzorkovanja i povećanje uzorkovanja koje se koristi u SR, prikazan na Slici 6. je već puno bliže samoj praksi i pravilnije, barem u pogledu osnovne ideje. [2, 9]

Jer u pravilu SR metode sve imaju po puno dodataka i moraju uzeti u obzir greške da bi slike bile što realnije. Tako da možemo vidjeti i taj šum koji se dodaje u tom primjeru Slike 6. koji služi da bi simulirao realističnu degradaciju same slike.



Slika 6. Smanjivanje i povećanje uzorkovanja u SR prema [2]

Na samome kraju, kada uzmemo sve dodatke i probleme koje prate povećanje slika općenito, možemo reći da je cilj SR da minimizira funkciju gubitka prikazanu izrazom

$$\phi = [\min_{\phi} \mathcal{L}(I_{yE}, I_{yHR})]_{\phi} + h\psi(\phi) \quad (6)$$

gdje je  $\mathcal{L}(I_{yE}, I_{yHR})$  funkcija gubitka između izlazne HR slike od SR i referentne HR slike,  $h$  je *tradeoff* parameta, a  $\psi(\phi)$  izraz regulacije.

Najčešće korištena funkcija gubitka u SR je srednja kvadratna pogreška (MSE, eng. *Mean Square Error*) temeljena na pikselima, koja se kraće može zvati i samo gubitak piksela. Detaljnije o ovome u nastavku sa samom metrikom bez koje ne bi mogli uopće procijeniti koliko je koja metoda dobra ili bolja od neke druge metode i slično. [2, 30]

## 2.1. Metode metrike SR

Bez instrumenta za mjerenje kvalitete ne možemo kvantificirati napredak niti imati nešto po čemu bi smo mogli uspoređivati koliko je nešto bolje ili gore od nečega. Tako da u ovome dijelu će biti spomenute osnovne metrike i pojmovi koji se najčešće koriste za mjerenje kvalitete slike, a one se uglavnom odnose na vizualne attribute i percepciju od strane gledatelja, odnosno samih ljudi jer ipak su oni ti koji će gledati u te slike na samome kraju.

Metode metrike za procjenu kvalitete slike (IQA, eng. *Image Quality Assessment*) se mogu podijeliti u dvije kategorije: subjektivne metode i objektivne metode. Subjektivne su one koje se tiču percepcije ljudskog oka, znači procjene od strane samih ljudi, koliko je slika prirodna i kakva je kvaliteta slike, odnosno koliko je dobra kvaliteta. Objektivne metode su one za koje se kvaliteta slike može brojčano izračunati i izraziti, odnosno možda bolje reći i



da su to kvantitativne metode. Ovaj dio uglavnom je prepušten samome računalu i matematičkim algoritmima. Vizualni aspekti slike koji se odnose na kvalitetu slike su dosta dobra mjera kvalitete, ali ova metoda zahtjeva više resursa, a pogotovo ako nam je skup podataka na kojima želimo izvršiti mjeru kvalitete jako velik. Zbog ovoga metode koje se češće koriste i bolje odgovaraju uvjetima za mjerenje kvalitete su one objektivne metode. Inače, jedan od primjera subjektivne metode je kada imamo referentnu sliku nastalu povećanjem slike s SR metodom, i onu originalnu, pa tražimo od ljudi da procijene koja je bolja. Nadalje će ipak biti više istražene i prikazane one objektivne, budući kako je već i rečeno, one se više i koriste u praksi.

### PSNR metrika

Tako imamo omjer vršnog signala i šuma (PSNR, eng. *Peak signal-to-noise ratio*), koji općenito u informatičkim sustavima označava tehniku mjerenja za analizu snage signala u usporedbi s snagom šuma. U sklopu slika i povećanja slika, PSNR služi za kvantitativno mjerenje kompresije kvalitete slike, jer znamo kako se rasterska grafika ponaša i koje su nuspojave mijenjanja rezolucije slike. U samoj SR, PSNR za neku sliku je određen sa maksimalnom vrijednosti piksela i sa MSE između referentne slike i slike SR, a još se poznaje i pod nazivom snaga distorzije šuma slike. Za neki dani maksimum vrijednosti piksela  $M$  i referentnu sliku  $I_r$ , koja može imati  $t$  broj piksela, i sliku SR  $I_y$ , način na koji se PSNR definira dan je sljedećim izrazom

$$PSNR = 10 \log_{10} \left( \frac{M^2}{MSE} \right) \quad (7)$$

Gdje je  $M$  uglavnom 8-bitni dubinski prostor boja, tako da je npr. maksimalna vrijednost 255. Za sami MSE imamo sljedeći izraz

$$MSE = \frac{1}{t} \sum_{i=1}^t (I_r(i) - I_y(i))^2 \quad (8)$$

Iz ovih dvaju izraza (7) i (8) da se vidjeti zašto je ova metrika u kategoriji onih baziranih na pikselima, jer direktno ovisi o vrijednosti koju pikseli postižu na slici. Napomena za ovu metriku, u nekim slučajevima može zavarati, gdje recimo ova sama metrika može biti dobra, odnosno da daje dobre brojeve i rezultate, ali percipirana kvalitete same slike i ne bude tako dobra. Dapače, narušila se je vjerodostojnost slike i onda je teško reći da li je tehnika koja se koristila za povećanje slike uistinu bila dobra ili bolja, ili je zapravo gora. Unatoč ovome

ipak se jako često koristi za usporedbu algoritama i metoda SR, no eto na kraju je ipak sami čovjek potreban da potvrdi valjanost.

Dalje imamo varijaciju MSE metrike, samo ne više prema vrijednostima piksela, već prema prosječnim vrijednostima boja u kanalima boja, korišten zbog osvijetljenosti i/ili sivih kanala odvojeno, budući je ljudsko oko više osjetljivije na promjenu osvjetljenja u odnosu na promjenama u obojenosti. Ukratko, ljudsko oko je osjetljivije na samo svjetlo nego na konkretnu vrijednost boje, što nam je jasno iz same biologije ljudskog oka.

### SSIM metrika

Još jedna od bitnih metrika za SR bi bila metrika za indeks strukturalne sličnosti (SSIM, eng. *Structural Similarity Index Metrics*). Ova metrika je bitna zbog toga što uzima u obzir vizualnu percepciju ljudi, budući je ona efikasna u izvlačenju strukturalnih informacija unutar slike, za razliku od PSNR metrike koja ne uzima u obzir strukturalnu kompoziciju slike. Ona uspoređuje kontrast, osvjetljenje i strukturalne detalje unutar referentne slike.

Neka imamo sliku  $I_r$  s ukupnim brojem piksela  $P$ , kontrast  $C_I$  i osvjetljenje  $L_I$  mogu se zabilježiti kao standardna devijacija i srednja vrijednost intenziteta slike izraženi sljedećim izrazima

$$L_I = \frac{1}{M} \sum_{i=1}^M I_r(i) \quad (9)$$

$$C_I = \sqrt{\left( \frac{1}{M-1} \sum_{i=1}^M (I_r(i) - L_I)^2 \right)} \quad (10)$$

gdje se  $i$ -ti piksel referentne slike označava s  $I_r(i)$ . Usporedbe koje se zatim baziraju na kontrastu i osvjetljenju između referentne slike  $I_r$  i pretpostavljene slike  $I$ , odnosno slike dobivene SR metodom, mogu se zapisati

$$Com_l(I_r, I) = \frac{2L_{I_r}L_I + \mu_1}{L_{I_r}^2 + L_I^2 + \mu_1} \quad (11)$$

$$Com_c(I_r, I) = \frac{2C_{I_r}C_I + \mu_2}{C_{I_r}^2 + C_I^2 + \mu_2} \quad (12)$$

gdje su  $\mu_1 = (k_1S)^2$  i  $\mu_2 = (k_2S)^2$  konstante koje služe da osiguraju stabilnost, tako da je  $k_1 \ll 1$  i  $k_2 \ll 1$ .

Normalizirane vrijednosti piksela  $I_r - L_{I_r}/C_{I_r}$  predstavljaju strukturu slike, dokle njihov unutarnji (skalarni) umnožak je ekvivalentan strukturalnoj sličnosti između referentne slike  $I_r$  i one pretpostavljene slike  $I$ . Kovarijanca  $\sigma_{I_r, I}$  je dana izrazom

$$\sigma_{I_r, I} = \frac{1}{M-1} \sum_{i=1}^M (I_r(i) - L_{I_r}) (I(i) - L_I) \quad (13)$$

Funkcija za strukturalnu usporedbu  $Com_s(I_r, I)$  je dana izrazom

$$Com_s(I_r, I) = \frac{\sigma_{I_r, I} + \mu_3}{C_{I_r} C_I + \mu_3} \quad (14)$$

gdje  $\mu_3$  ponovo predstavlja konstantu za stabilnost.

Uzimajući u obzir dosad prikazane funkcije za osvjetljenje, kontrast i strukturalnu sličnost, na kraju za sami indeks strukturalne sličnosti SSIM imamo sljedeći izraz

$$SSIM(I_r, I) = \{Com_l(I_r, I)\}^\alpha \{Com_c(I_r, I)\}^\beta \{Com_s(I_r, I)\}^\gamma \quad (15)$$

Gdje se kontrolni parametri  $\alpha$ ,  $\beta$  i  $\gamma$  mogu podešavati u ovisnosti o tome što želimo od osvjetljenost, kontrast ili strukturalna sličnost staviti na naglasak pri proračunu SSIM-a.

Inače, PSNR se tradicionalno bio koristio u računalnome vidu za evaluaciju zadataka toga područja, no kako je SSIM baziran na biologiji ljudskog oka i samoj ljudskoj percepciji strukture neke slike, ona se je češće koristila za evaluaciju AI metoda za povećanje slike. Vrijedno je isto napomenuti kako SSIM metrika nije bila jako precizna u medicinskim snimkama gdje bi osvjetljenja bila niska, te bi rezultirali onda s lažnim pozitivnim rezultatima. To je ipak jedan od specijalnih slučajeva, jer na svu sreću za prirodne slike to nije bio slučaj i tu je ova metrika se pokazala dobrom.

### **Bodovanje mišljenja metrika**

Za razliku od PSNR i SSIM metrike, ova spada pod subjektivne metrike, što bi trebalo biti i potpuno jasno iz samog naziva. Nadalje, nije kvantitativna već kvalitativna metoda. U ovoj metodi, slično uobičajenim upitnicima i skalama za ocjenjivanje, od ispitanika se traži da procijeni kvalitetu slike ovisno o specifičnim kriterijima, npr. izoštenost, prirodnost, kvaliteta boja i slično. Na kraju bi konačni rezultat ocjenjivanja bio rezultat zbroja srednjih vrijednosti kriterija koji su se ocjenjivali.

Iako su ocjenjivači ljudi, i iako je ova metoda subjektivna, što znači da može biti popraćena nekakvim oblikom pristranosti ili uobičajenim ljudskim greškama, ipak je u pojedinim slučajevima jako bitna. Jedan od konkretnih primjera bi bio prepoznavanje lica, te drugih specifičnih dijelova ljudskog tijela, kao što bi to bile ruke i prsti. U ovim slučajevima objektivne metrike su znale imati dobre rezultate, no subjektivne metode, kao ova, dosta lošije. Zapravo i logično jer znamo jako dobro koja ograničenja umjetna inteligencija ima u tome pogledu, pogotovo što se tiče slika ljudi. Tako da se ova metrika i dan danas i dalje koristi, i barem još neko duže vrijeme neće izaći iz trenda, ako i uopće.

### **Percepcijska kvaliteta - DeepQA**

Metoda koja se nadovezuje na prethodno spomenutu metriku bodovanja mišljenja. U metodi bodovanja mišljenja ljudi su bili ti koji su ručno evaluirali kvalitete slika po danim kriterijima, i dokle je ona davala dobre rezultate, pogotovo za one specifične slučajeve kako je i spomenuto u toj metodi, problem je bilo vrijeme i ostali resursi. Zamislite, prihvatljivo je da čovjek procjenjuje desetke slika, pa donekle i stotine, no što kada je riječ o tisućama, desetinama tisuća itd. Tako da, ovisno o danome skupu podataka, odnosno slika, na kojima se je radilo, procjena od strane ljudi metrikom bodovanja mišljenja bi jednostavno postala neprihvatljiva. Ono što se je pokušalo napraviti je da se zapravo nauči računalo ljudskome ponašanju ocjenjivanja kvalitete slika, pa da se to onda provede na tim velikim skupovima podataka. Za izvođenje ovoga predložen je model procjena kvalitete pune referentne slike (FR-IQA, eng. *Full Reference Image Quality Assessment*) koji za osnovu koristi konvolucijske neuronske mreže (CNN, eng. *Convolution Neural Networks*) s kojima uči ljudsko ponašanje iz IQA baze podataka koja je sadržavala izobličene slike, subjektivne bodove i mape grešaka. Ova metoda je postala poznata pod nazivom DeepQA. [33] Dalje smo samo imali izvedenice ove metode koji su drugi ljudi prilagodili potrebama svojih modela, odnosno skupova podataka za slike koji su oni koristili, s drugim strategijama učenja, arhitekturom neuronskih mreža i slično. Neki od primjera su dipQA [32] MEON [34], NIMA [35], RankIQA itd. [36]

Inače, glavni problem ove metode je njena pouzdanost, jer u sklopu SR već govorimo o tome kako želimo naučiti računalo da poveća sliku, a već smo vidjeli kako ovo nije nimalo lagan proces, mi tu još dodatno želimo naučiti računalo prije povećanja slike da može samo procijeniti kvalitetu toga proces. Ukratko, ovo je skoro kao da nepouzdanost dižemo eksponencijalno na nepouzdanost. Iako je istina da ovom metodom uštedimo na vremenu

potrebne i resursima, zbog problema nepouzdanosti u cijelom pogledu, i dalje će SSIM i PSNR biti glavne metode metrike za SR i računalni vid. [2, 30]

### **Evaluacija temeljena na zadatku**

Kako smo već rekli, AI metode ne rade samo na povećanju slike, već pokrivaju i niz drugih zadataka. Istina je da je glavni zadatak SR povećati sliku fokusirajući se na postizanje što bolje rezolucije, radom na tome procesu istovremeno obavlja i niz drugih zadataka, te zapravo i rezultati tih drugih zadataka mogu poslužiti dijelom kao metrika za kvalitetu metoda povećanja slike. Pogotovo za ta posebna područja medicine i računalnog vida, gdje su nam jako bitni zadaci kao klasifikacija, prepoznavanje lica, segmentacija objekata i slično. Možemo imati AI metode koje bi postizale i ne bas izvrsne rezultate u pogledu povećanja slike i same rezolucije, ali ako bi one postigle jako dobre rezultate u pogledu npr. klasifikacije i detekcije objekata, značilo bi da ipak one imaju neku konkretnu upotrebu, odnosno da nisu beskorisne ili propali projekt. Malo specifičan slučaj, ne toliko zastupljen, ali kako ipak postoji vrijedan je spomena.

Priloženo ispod imamo sažeti prikaz prethodno navedenih metoda metrike s njihovim značajnostima i nedostacima. Dakle, najčešće su korištene PSNR i SSIM metrike, no ne smijemo zaboraviti ni na one specijalne slučajeve gdje bi nam i neke druge bile potrebne, tipa metrika bodovanja mišljenja.

Metode metrike:

- PSNR
  - Značajnosti
    - Najkorištenija IQA metrika, jednostavna za usporedbu rezultata drugih SR metoda
    - Kvantitativna i koristi MSE
  - Nedostaci
    - Budući je bazirana na pikselu, ukupan rezultat može zavarati, npr. dvije slike vizualno drukčije izgledaju, a PSNR rezultat dobar
    - Ne uzima u obzir strukturalnu informaciju slike
- SSIM
  - Značajnosti
    - Nakon PSNR druga najkorištenija IQA metrika, isto jednostavna za usporedbu drugih SR metoda

- Kvantitativno procjenjuje strukturalnu sličnost dobivene slike s originalom, s mogućnošću podešavanja težinskih vrijednosti osvjetljenosti, kontrasta i strukturalne sličnosti
  - Nedostaci
    - Nepouzdana gdje je osvjetljenost na slikama niska, npr. medicinske snimke
- Bodovanje mišljenja
  - Značajnosti
    - Subjektivna kvalitativna metrika gdje ljudi ocjenjuju kvalitetu slike prema specifičnim kriterijima, npr. izoštrenost, boja, prirodnost itd.
    - Pogodna metoda za prepoznavanje lica ljudi i sličnog
  - Nedostaci
    - Ograničena nelinearnosti bodovanja među ocjenjivačima, ljudskom greškom i promjenama u testnim parametrima
    - Potrebno jako puno vremena, pogotovo za veće skupove podataka
- Evaluacija temeljena na zadatku
  - Značajnosti
    - Metoda prikladna kada se SR slike koriste i za druge zadatke kao detekcija objekata i klasifikacija u računalnom vidu
    - Pomaže izmjeriti cjelokupnu performansu metode SR
  - Nedostaci
    - Ovisi o performansi svakog zadatka
    - Slike iste kvalitete za SR će dati drukčije rezultate ovisno o promjeni parametara pojedinih zadataka

### **Upravljanje kanalima boja**

U sklopu SR i većine skupa podataka koji se koriste, koristi se RGB prostor boja zbog toga što je većinom riječ o RGB slikama. Ipak, ima i nekih metoda koje koriste YCbCr prostor boja, gdje Y predstavlja intenzitet svjetla, a Cb i Cr su komponente boja, redom plavo diferencirani i crveno diferencirani kanali. Bitno je isto napomenuti da ovo ispada iz prakse i ostaje se samo na RGB danas. Isto tako, općenito određene metrike ne bi radile podjednako dobro na slikama različitih prostora boja, npr. PSNR bi dosta varirao. [2]

## 2.2. Referentni skupovi podataka

Skupovi podataka dolaze s raznim specifikacijama, te se razlikuju po broju slika, formatu slika, rezoluciji, tipu slika, odnosno da li su sparirane slike LR-HR ili ne, kvaliteti i postavljanju postavki hardvera. Jako bitna karakteristika skupa podataka je da li je riječ o tome da li su sparirane slike LR i HR za uvježbavanje modela, ili je dana samo HR slika, a onda se odgovarajuća LR slika generira korištenjem bi-kubične interpolacije s *antialiasing*-om. Ovo direktno utječe na resurse i arhitekturu mreže modela. Većina skupova podataka u SR je ipak bez parova HR i LR slika.

Primjeri popularnih referentnih skupova podataka su:

- BSD100 [10]
- BSD300 [10]
- BSDS500 [11]
- CelebaA [12]
- DIV2K [13]
- Manga109 [14]
- MS-COCO [15]
- OutdoorScene [16]
- PIRM [17]
- Set14 [18]
- Set5 [19]
- T91 [20]
- Urban100 [21]
- VOC2012 [22]

Ima ih još mnogo, dapače neprestano se i rade novi ovisno o tome za koju primjenu se radi model kojega želimo uvježbavati, no ovi su ustaljeni u području SR i dobra startna točka za većinu istraživača.

### **Izazovi SR**

Ovdje se više misli u kontekstu natjecanja i natječaja, postavljaju se određeni izazovi na razini globale ili pak regije, gdje se onda znanstvenici natječu u tim izazovima s svojim metodama, tražeći onu koja postiže najbolje rezultate u nekoj kategoriji. Primjeri izazova su bojanje,

uklanjanje šuma, SR itd. Jedan od tako poznatih izazova je *The New Trends in Image Restoration and Enhancement* (NTIRE). U sklopu ovog rada nas više zanima oni koji su fokusirani na SR, i po mogućnosti bliži regiji Europe. Tako da za taj primjer imamo PIRM (eng. *Perceptual Image Restoration and Manipulation*) izazov, koji je nastao u kolaboraciji Europske konferencije za računalni vid (ECCV, eng. *European Conference on Computer Vision*). U posebnom poglavlju ovoga rada biti će predstavljena tako jedna od metoda koja je dobila prvo mjesto u kategoriji koja se odnosila na SR, u jednom od tih PIRM izazova. Konkretno biti će prikazana novija verzija, budući je ipak prošlo par godina od onda (natjecanje je bilo 2018. godine). Kako je bilo puno izvedenica zbog velike popularnosti, postoji velika mogućnost da će i neposredno nakon ovoga rada doći nešto novije jako brzo, no ipak detaljnije o ovome svemu u za to predviđenom poglavlju. [2]



### 3. Metode super-rezolucije

Kod klasifikacije AI metoda u prvome poglavlju je bila spomenuta brojnost metoda super-rezolucije, te karakteristike po kojima bi se one mogle dalje dijeliti, odnosno po kojima se razlikuju. U ovome poglavlju će biti detaljnije prikazani osnovni pojmovi i tehnike koje definiraju AI metode, odnosno karakteristike metoda super-rezolucije s dubokim učenjem, dalje dijeljenih na one s nadziranom učenjem i one bez. Kratko se samo sjetimo tih karakteristika, što su zapravo ništa drugo nego komponente AI metoda, a to su: metoda povećanja uzorkovanja, duboke neuronske mreže (njihova arhitektura), strategija (algoritam) učenja i okviri modela.

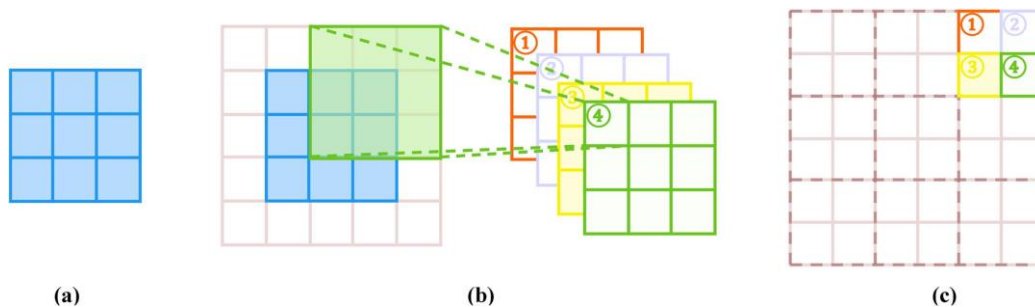
#### 3.1. Metode povećanja uzorkovanja – US metode

Povećanje uzorkovanja je ključno u metodama SR koje se baziraju na dubokom učenju, a pogotovo što se tiče pozicioniranja i metode koja se koristi da bi se izvršilo povećanje uzorkovanja, jer ovo ima značajnog utjecaja na uvježbavanje i testiranje same performanse modela.

Kako smo spomenuli kod tradicionalnih metoda, one su koristile interpolaciju, i to su bile metode bez sposobnosti učenja. Sada ćemo vidjeti kakve tehnike koriste današnje AI metode za povećanje uzorkovanja.

##### Pod-piksel sloj

Sloj učenja od kraja do kraja, koji se još naziva sloj pod-piksela, izvršava povećanje uzorkovanja u dva bitna koraka. Prvo generira nekoliko dodatnih kanala korištenjem konvolucije, i zatim preoblikuje te iste kanale da bi izvršio povećanje uzorkovanja. Vidljivo bolje na vizualnom primjeru na Slici 7. Prikazano izrazima, konvolucija u ovome sloju se primjenjuje na  $s^2_f$  koji predstavlja faktor skaliranja. Unos je slika dimenzija  $h \times w \times c$  gdje  $h$  stoji za visinu,  $w$  za širinu, a  $c$  za kanale boja. Rezultat konvolucije je  $h \times w \times cs^2_f$ , a da bi dobili finalnu sliku, potrebno je to preoblikovati (preslagati) tako da imamo izlaznu sliku prikazanu s  $s_f h \times s_f w \times c$ . Ovakav tip sloja se često koristi u SR modelima.



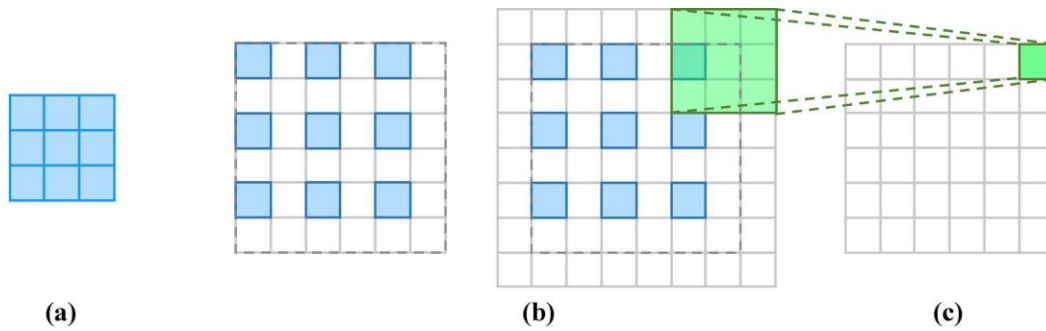
Slika 7. Princip rada pod-piksel sloja; (a) ulaz, (b) konvolucija, (c) preoblikovanje [2]

Značajka ovog sloja je da ima široko receptivno polje, koje pomaže da bi se naučilo više sadržajne informacije, da bi se onda mogli generirati realističniji detalji na slici. No, treba pripaziti jer ima i manu da može generirati lažne artefakte na granicama kompleksnih uzoraka zbog svoje neravnomjerne podjele dotičnog polja. Nadalje, predviđanje susjednih piksela u regijama blokovskih oblika može dati izlaze koji nisu glatki i ne izgledaju realistično u usporedbi s originalnom HR slikom. Način suočavanja s ovime je bilo korištenje dodatnog sloja za međuovisno predviđanje, kako je predloženo s PixelTCL. [37] Taj dodatni sloj bi koristio informaciju tih međusobno povezanih piksela usred povećanja uzorkovanja. Rezultati su dodavanjem tog dodatnog sloja bili realističniji i bolji, ali možete odmah i uzeti u obzir da su performanse bile zahtjevnije.

### Dekonvolucijski sloj

Transponirani konvolucijski sloj, još zvan dekonvolucijski sloj, predstavlja ono suprotno od sloja konvolucije. On predviđa moguću ulaznu sliku HR baziranu prema mapi značajki od slike LR. Kako radi, na način da se dodaju nule da bi povećali rezoluciju, i onda radimo na tome konvoluciju. Primjer na Slici 8. za SR sliku veličine konvolucijskog kernela od 3x3 i faktora skaliranja 2, možemo vidjeti kako se slika prvo proširila dodavajući nule, a zatim izvršila konvolucija.

Isto jako često korišten sloj u metodama SR, budući ne da samo generira sliku HR s kraja na kraj, već je i kompatibilan s originalnom konvolucijom. Mana bi bila nejednako preklapanje unutar generirane slike u nekim slučajevima, i budući je uzorak bio šahovskog oblika, dobili bi nerealističnu sliku kao rezultat.



Slika 8. Princip rada dekonvolucijskog sloja; (a) ulaz, (b) ekspanzija, (c) konvolucija [2]

### Meta povećanje veličine

U prethodnim primjerima faktor skaliranja je uvijek bio predefiniiran, pa bi zbog toga morali uvježbavati više različitih modela s različitim vrijednostima toga faktora, što i nije baš efikasan način niti pravi zahtjev metoda SR. Naprotiv, cilj je uvijek bio adaptacija i da neovisno o vrsti slike postignemo dobre rezultate. Zbog toga imamo modul za meta povećanje veličine, koji koristi proizvoljne faktore skaliranja za generiranje slike SR temeljene na meta-učenju.

Radi na način da projicira svaku poziciju u traženoj slici HR na maleni uzorak kojega dobije iz mape značajki oblika  $j \times j \times c_i$ , gdje je  $j$  proizvoljan a  $c_i$  ukupan broj kanala izvučenih iz mape značajki. Dodatno bi i generirao težine konvolucije ( $j \times j \times (c_i \times c_o)$ ), gdje  $c_o$  predstavlja kanale izlazne slike, a to bi uglavnom bio broj 3. Vidljivo je da za izvesti ovo potreban je jako veliki skup podataka za uvježbavanje, tako da je set za uvježbavanje ovog modula ogroman, i istovremeno se uvježbava veliki broj faktora. Ovisno dalje o veličini i težinama, ovaj modul može postati nestabilan, jer ipak radi za svaki piksel pojedinačno. [2]

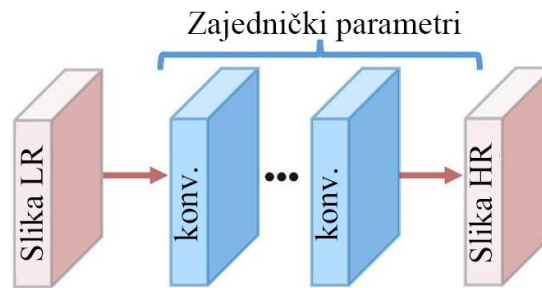
## 3.2. Mreže dubokog učenja

Dizajn neuronskih mreža i njihov napredak u arhitekturi su trenutno trend u dubokome učenju. U sklopu područja SR, istraživači su isprobali nekoliko implikacija dizajna zajedno s okvirima SR za dizajniranje cjelokupne mreže SR. Neki od osnovnih i najčešće korištenih dizajna u mrežama SR, te i neki od novijih, će biti predstavljeni u nastavku.

### Rekurzivno učenje

Jedna od osnovnih strategija učenja temeljenih na mreži je korištenje istog modula za rekurzivno učenje značajki visoke razine. Ovom metodom se minimaliziraju parametri

budući je strategija bazirana na istom modulu koji se nadograđuje rekurzivno, primjer prikazan na Slici 9.



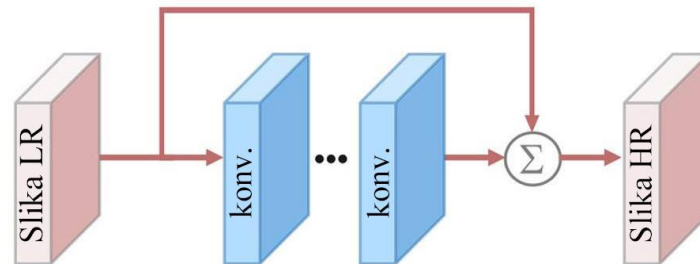
Slika 9. Struktura neuronske mreže za rekurzivno učenje prema [2]

Primjerak jedne od najčešće korištenih neuronskih mreža ovog oblika je DRCN (eng. *Deeply-recursive Convolutional Network*). [23] Korištenjem samo jednog konvolucijskog sloja DRCN mreža postiže repetitivno polje do veličine 41 x 41 bez zahtijevanja dodatnih parametara, što je jako duboko u usporedbi s SRCNN (eng. *Super-resolution Convolution Neural Network*) mrežom, koja to isto polje ima na 13 x 13. [24] Dalje primjer imamo DRRN (eng. *Deep Recursive Residual Network*) mrežu koja koristi ResBlock kao dio rekurzivnog modula za do 25 izvršavanja rekurzije, jer su istraživanja pokazala da ovo daje bolje rezultate nego sami osnovni ResBlock. [25, 26] Korištenjem koncepta DRCN mreže, imamo izvedenu i metodu MemNet koja je sadržavala šest rekurzivnih ResBlock-ova, dokle je CARN (eng. *Cascading Residual Network*) mreža isto koristila ResBlock ali kao rekurzivne jedinice. [27] U ovoj izvedbi mreže su dijelile težine globalno u rekurziji koristeći iterativni *up-and-down* pristup uzorkovanja. Pored *end-to-end* rekurzija koji je koristila većina izvedbi ove metode, imamo i DSRN (eng. *Dual-state Recurrent Network*) mreže u kojima se signal između slika LR i generiranih stanja slika HR dijelio unutar mreže. [28]

Iako mreže ovog tipa mogu naučiti kompleksno reprezentirane podatke, što su u ovom slučaju informacije slike, problem je hardverska izvedba i resursi, budući je ova metoda jako zahtjevna u računalnim performansama. Nadalje zbog ovih jako zahtjevnih računalnih performansi možemo imati i problema s eksplodirajućim ili nestajućim gradijentom. Zbog ovoga se metoda rekurzivnog učenja često koristi u kombinaciji s višestrukim nadzorom ili rezidualnim učenjem da mi se minimizirao rizik navedenog gradijenta.

## Rezidualno učenje

Ova metoda se je koristila jako puno u području SR, dokle nije nastao ResNet s pomoću kojega su se dalje bili učili ostaci (*reziduali*). Primjer imamo na Slici 10. Inače, imamo dva pristupa u ovoj metodi, lokalno i globalno rezidualno učenje.



Slika 10. Struktura neuronske mreže za rezidualno učenje prema [2]

Lokalni pristup rezidualnog učenja ublažava problem degradacije uzrokovane povećanom dubinom mreže. Nadalje ovaj pristup je i poboljšao stopu učenja i smanjio zahtjevnost uvježbavanja. Zbog ovih dobrih svojstava ovaj pristup se često koristi u području SR.

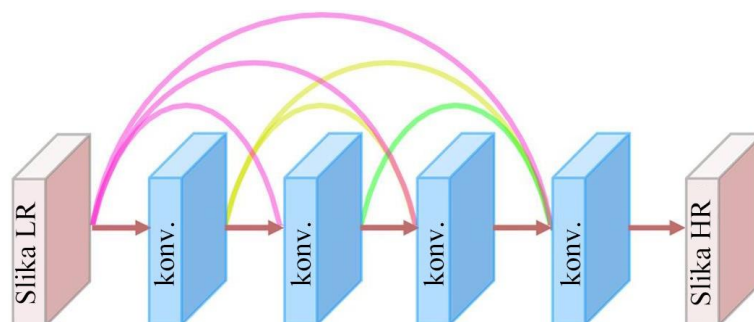
S druge strane imamo globalni pristup u kojem su i ulaz i izlaz korelirani, a u sklopu SR, izlazna slika HR je veoma korelirana s ulaznom slikom LR, a posljedično tome globalni reziduali između slike LR i slike HR su jako bitni za SR. U ovom pristupu model uči samo rezidualne mape koje pretvaraju sliku LR u sliku HR na način da generiraju visokofrekvencijske detalje koji nedostaju na slici LR. Reziduali ovdje su minimalni, te su posljedično tome kompleksnost modela i težina učenja smanjeni. Ovaj pristup se isto često koristi u metodama SR.

Uglavnom, oba pristupa spajaju ulaznu sliku s izlaznom slikom HR, s glavnom razlikom da u globalnome pristupu ta povezanost je direktna, dokle u lokalnom imamo više slojeva različitih dubina koji ih spajaju.

### Učenje temeljeno na gustim vezama

Ova metoda koristi guste blokove za SR, kao npr. DenseNet. [29] Gusti blokovi koriste sve mape značajki generirane prethodnim slojevima za ulaze i za svoje ulaze značajki, dovodeći do  $l(l-1)/2$  broja veza u  $l$ -slojevima ( $l \geq 2$ ) gustog bloka. Uporabom gustog bloka povećava se ponovno korištenje značajki, dokle istovremeno se rješava problem nestajanja gradijenta. Guste veze isto tako minimiziraju veličinu modela korištenjem male stope rasta i otvaranja kanala uporabom spojenih ulaznih značajki.

Ove guste veze se koriste u SR da bi povezale mape značajki niske razine i visoke razine za rekonstrukciju slike HR visoke kvalitete i finih detalja, a primjerak strukture toga imamo na Slici 11.



Slika 11. Struktura neuronske mreže za učenje temeljeno na gustim vezama prema [2]

Konkretni primjer bi bio SRDenseNet, koji koristi 69-slojnu mrežu koja sadrži guste veze unutar gustih blokova. [38] U ovome primjeru mape značajki prethodnih blokova i mape značajki su bile korištene kao ulazi svih prethodnih blokova. Primjeri mreža koji su koristili istu i sličnu ideju su još RDN [39], CARN [27], MemNet [40] i ESRGAN [41], dokle je recimo DBPN koristila samo guste veze između jedinica za povećanje i smanjivanje uzorkovanja. [16]

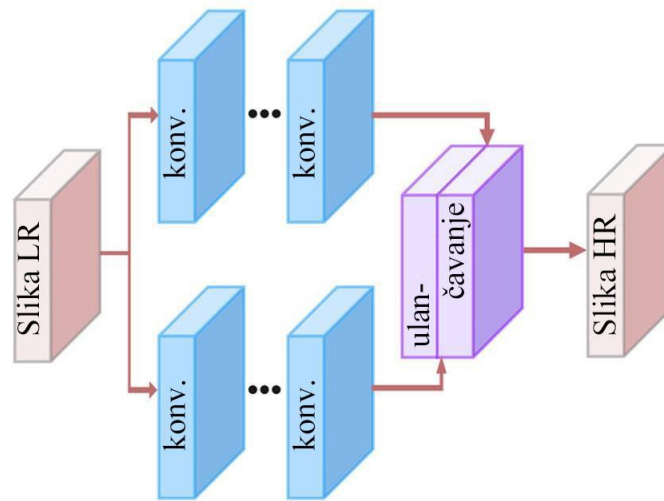
### Višesmjerno učenje

Kod ove metode značajke su prebačene na više puteva za različito predstavljanje, i rezultat toga se onda kasnije kombinira, a s time se dobije povećanje performansi. Postoje tri glavna oblika: lokalni, globalni i ljestvično specifični.

Ovisno o ljestvicama(skalama), modeli SR bi koristili drugačije načine izvlačenja značajki, a za takve slučajeve onda imamo jednu mrežu koja je bazirana na višesmjernom učenju za više skala. Srednji slojevi od modela su bili dijeljeni za ekstrakciju značajki, dokle ljestvično specifični putevi, uključujući pred obradu i povećanje uzorkovanja, bi se nalazili na krajevima modela, odnosno na početku i na kraju mreže. Tokom uvježbavanja putevi koji su ovisi o skali su se po potrebi uključivali i ažurirali shodno tome. Primjerak mreže bi bio MDSR, gdje je model bio još manje veličine zbog toga što su se parametri dijelili kroz sve skale. Još par primjera mreža s višesmjernim učenjem su ProSR i CARN.

Lokalna izvedba višesmjernog učenja je kreirana korištenjem novog bloka, koji se naziva modul započinjanja, koji se koristio u izvlačenju značajki na više skala. A za primjer toga imamo mrežu MSRN, i na Slici 12. možemo vidjeti njenu strukturu. Dodatni blokovi se

sastoje od konvolucijskog sloja s kernelima veličina od  $3 \times 3$  i  $5 \times 5$ , koji istovremeno izvlače značajke. Nakon kombiniranja izlaza dvaju konvolucijskih slojeva, finalni izlaz prolazi kroz konvoluciju  $5 \times 5$  kernela.



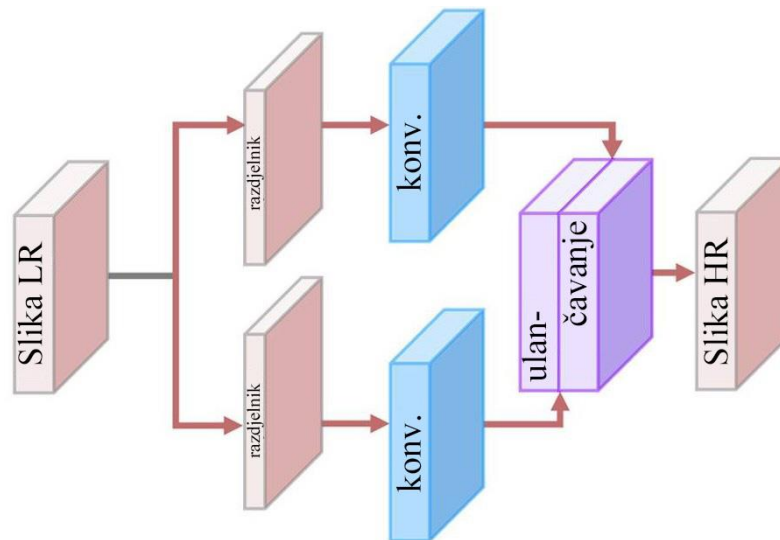
Slika 12. Struktura neuronske mreže za višesmjerno učenje prema [2]

Još imamo tako varijaciju s globalnim višesmjernim učenjem, ovdje razne izvučene značajke iz više smjerova mogu međusobno komunicirati. Primjerak bi bila mreža DSRN, gdje imamo dva puta za izvlačenje niskih i visokih razina informacije, i gdje imamo neprekidno dijeljenje značajki za poboljšano učenje.

### Učenje temeljeno na naprednoj konvoluciji

Općenito za metode SR vrijedi da se razlikuju prema tome na koji način vrše operaciju konvolucije, te su tako mnogi istraživači i znanstvenici pokušali modificirati tu istu operaciju da bi postigli bolje performanse i/ili rezultate. U posljednjim godinama za pokušaj izvedbe toga imamo grupnu konvoluciju, prikazana na Slici 13., gdje se je smanjio ukupan broj parametara s troškom manjih petlji u performansi.

Konkretni primjeri su CARN-M [27] i IDN mreža [42], gdje se je koristila prethodno spomenuta grupna konvolucija umjesto uobičajene konvolucije.



Slika 13. Struktura neuronske mreže za učenje temeljeno na naprednoj konvoluciji prema [2]

Još jedan primjer izvedbe ove ideje je proširena konvolucija u kojoj se informacije o sadržaju koriste za generiranje realističnih slike SR. Ova vrsta konvolucije se koristila da bi se poduplalo receptivno polje i s time poboljšali rezultati.

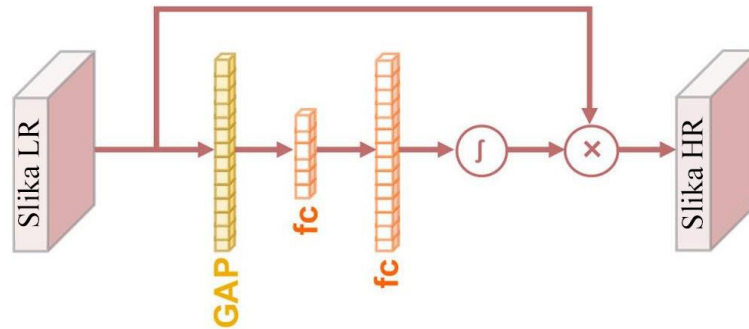
Dosad navedeni primjeri izvedbe ove ideje su svi imali više pozitivnih strana nego negativnih, odnosno bili su iskoristivi, ali zato imamo i jedan specifičan slučaj gdje je oblik konvolucije pod nazivom dubinski odvojiva konvolucija, unatoč smanjenju ukupnog broja parametara, rezultirala s znatnim pogoršanjem sveukupnih performansi.

### Učenje temeljeno na pažnji

U dubokom učenju, učenje pažnjom podrazumijeva ideju gdje se pojedini faktori gledaju bitnijim naspram drugih, te na veći značaj dan način kojim će se oni procesirati i njihovi podaci pored drugih. U sklopu SR, bit će predstavljena dva tipa ovog učenja.

Tako imamo verziju gdje je pažnja posvećena kanalu, tu se u model dodaje posebni blok gdje se sa globalnim prosječnim grupiranjem (GAP, eng. *Global Average Pooling*) stisnu unosi kanala, a dva potpuno povezana sloja dalje obrađuju te konstante da bi generirali ostatke ovisne o kanalima. Na Slici 14. imamo primjer strukture jedne takve mreže koja koristi ovu ideju. Konkretni primjer je RCAN mreža, koja je zahvaljujući ovoj ideju poboljšala svoje performanse.





Slika 14. Struktura neuronske mreže za učenje temeljeno na pažnji prema [2]

### Ostale metode mreža

Kako smo već spomenuli više puta, ima ih stvarno puno i dosad su navedene one najčešće korištene, no čisto da se da na uvid i ostale ideje koje su istraživači imali, priložiti ćemo par dodatnih metoda, koje smatramo vrijednima spomena.

### Učenje temeljeno na transformiranju valića

Transformacija valića (WT, eng. *Wavelet transform*), predstavlja uporabu visoko frekventnih pod pojaseva za tekture i globalnom strukturom informacija nisko frekventnih pod pojaseva na veoma efikasan način. U sklopu SR ova metoda se je koristila da bi se generirali ostaci pod-pojaseva HR koristeći pod pojaseve interpolirane valićem LR. Ukratko, koristeći WT bi prvo razgradili (dekomponirali) sliku LR, dokle bi onda radili inverz od WT da bi rekonstruirali sliku HR. Primjerak modela koji su koristili ovu ideju su WRAN [43] i MWCNN [44].

### Učenje temeljeno na rekurzivnim regijama

Većina metoda koja se koriste za ostvarivanje SR ide uvijek s pretpostavkom o procesima koji se fokusiraju na individualne piksele, odnosno sami piksel, bez stavljanja na prioritet moguću međuovisnost tog piksela o drugim generiranim pikselima, odnosno regijama. Korištenjem koncepta kojeg smo imali u PixelCNN, nastala je ova metoda, gdje su se koristile dvije neuronske mreže za generiranje piksela po piksel. [45] Ideja je bila uzeti u obzir informaciju o ovisnosti piksela zasebno i u globalnoj slici, s metodom rekurzije. S obzirom na metriku bodovanja mišljenja ova metoda je postizala dobre rezultate, no možemo zamisliti kakva je bila s performansama uzevši u obzir riječ o dvama mrežama i rekurziji. Ukratko, zahtjevnost performansi je bila znatno veća, tako da unatoč boljim rezultatima teško ju je prihvatiti.

### 3.3. Funkcije gubitka

Doslovno za bilo koju primjenu unutar dubokog učenja, izbor funkcije gubitka je jedan od najvažnijih dijelova. U sklopu SR ove funkcije se koriste da bi izračunale grešku pri rekonstrukciji slike HR, što dalje pomaže optimizirati model iterativno. Budući je osnovni element slike piksel, prvotno su sva istraživanja koristila gubitak piksela, no kasnije se je shvatilo da korištenjem samo gubitka piksela ne možemo predstaviti kvalitetu rekonstrukcije u potpunosti. Zbog toga se u kontekstu SR za funkciju gubitka uzima gubitak sadržaja ili kontradiktorni gubitak, pomoću kojih se mjeri greška u generiranju tih funkcija gubitka. Notacije koje su se dosad koristile u radu koristiti će se i ovdje u izrazima nastavku i označavati će isto, osim ako drugačije nije naglašeno. [2, 30]

#### Gubitak sadržaja

Perceptivna kvaliteta ili kvaliteta prikaza, kako je već bilo i spomenuto, je jedna od najnužnijih oblika evaluacije za model SR, i jednako tako je mjera gubitka ovoga korištena u SR na način da bi se mjerila razlika između generiranih i istinitih (originalnih) slika korištenjem mreže (N) za klasifikaciju slike. Neka su podaci visoke razine prikazani na  $l$ -tom sloju iz  $r^l(I)$ , tada je gubitak sadržaja definiran kao euklidski među prikazima dvaju slika  $I$  i  $I_g$ , gdje je  $I$  originalna slika a  $I_g$  generirana slika SR. Sve skupa prikazano sljedećim izrazom

$$\mathcal{L}(I, I_g; N_C, l) = \frac{1}{h_l w_l c_l} \sqrt{\sum_{i,j,k} \left( r_{i,j,k}^l(I_g) - r_{i,j,k}^l(I) \right)^2} \quad (16)$$

gdje  $h_l$ ,  $w_l$  i  $c_l$  redom označavaju visinu, širinu i nekoliko kanala prikaza slike u sloju  $l$ .

Cilj funkcije gubitka sadržaja je da dijeli informacije o značajkama slike iz mreže za klasifikaciju slike  $N_C$  s samom mrežom SR. Ova funkcija osigurava vizualnu sličnost između originalne slike i generirane slike zbog toga što ne uspoređuje individualne piksele već sami sadržaj. Tako da zbog navedenog ova funkcija pomaže dobivanju vizualno zamjetnih i realističnih slika u području SR, te je bila često korištena, a za primjer imamo mreže koje su koristile već uvježbane CNN mreže kao ResNet i VGG [46].

#### Kontradiktorni gubitak

U posljednjim godinama, razvojem GAN-ova (eng. *Generative Adversarial Networks*) i zahvaljujući njihovoj sposobnosti učenja i samo-nadziranosti, postale su sve popularnije pa

tako i u području SR. GAN-ovi kombiniraju dvostruke mreže za izvršavanje zadataka generiranja i diskriminiranja. Konkretno, izlaz generativne mreže se evaluira sa diskriminativnom mrežom. Zbog ovoga dokle se uvježbavaju GAN-ovi konstantno se izvršavaju dva ažuriranja, a to su: (i) prilagodba generatora da daje bolje rezultate i diskriminatora da diskriminira efikasnije, (ii) prilagodba diskriminatora dokle se uvježbava generator. Ovo je rekurzivno uvježbavanje mreže, i tek nakon jako mnogo iteracija uvježbavanja i evaluacije, generator može generirati izlaz koji odgovara distribuciji stvarne slike. Diskriminator ne može razlikovati stvarne i generirane informacije.

U sklopu SR, svrha generativne mreže je da generira sliku HR, dokle se druga diskriminativna mreža koristi da bi evaluirali da je ta generirana slika jednake distribucije kao ulazni podaci, odnosno stvarna slika. Ova metoda je prvi put bila predstavljena u SR kao SRGAN [47], a kontradiktorni gubitak je tu bio prikazan sljedećim izrazom

$$\mathcal{L}_{GAN\_CE\_g}(I_g; D) = -\log D(I_g) \quad (17)$$

$$\mathcal{L}_{GAN\_CE\_d}(I_g, I_s; D) = -\left\{\log D(I_s) + \log\left(1 - D(I_g)\right)\right\} \quad (18)$$

gdje je  $\mathcal{L}_{GAN\_CE\_g}$  funkcija kontradiktornog gubitka od generatora modela SR, a  $\mathcal{L}_{GAN\_CE\_d}$  funkcija kontradiktornog gubitka od diskriminatora  $D$ , koji je zapravo binarni klasifikator, a  $I_s$  označava pravu istinitu sliku dobivenu nasumično.

Pored primjera iznad s binarnom klasifikacijom greške, imamo još i s srednjom kvadratnom pogreškom koja je dala poboljšano uvježbavanje i bolje rezultate nego sama binarna klasifikacija, a izraz za tu funkciju je

$$\mathcal{L}_{GAN\_LS\_g}(I_g; D) = (D(I_g) - 1)^2 \quad (19)$$

$$\mathcal{L}_{GAN\_LS\_d}(I_g, I_s; D) = -\left\{\left(D(I_g)\right)^2 + (D(I_s) - 1)^2\right\} \quad (20)$$

Primjedba ovoj izvedbi je ipak pronađena u pojedinim primjerima, gdje su ti diskriminatori piksel razine generirali šum visoke frekvencije, tako da se je onda odlučilo dodati još jednu diskriminativnu mrežu koja je služila da evaluira onu prvu za te visoke frekvencije da bi se izbjegao navedeni problem. S ovime su se zatim uspjeli zahvatiti svi atributi točno.

### **Gubitak piksela**

Funkcija gubitka koja računa usporedbu(različitost i sličnost) piksela između generirane i referentne slike, i imamo dva izvedena tipa: L1 i L2 funkcije gubitka. Funkcija gubitka L1

se još naziva i srednja apsolutna greška (MAE, eng. *Mean Absolute Error*), a funkcija gubitka L2 se još naziva i srednja kvadratna greška, koju smo već spomenuli skraćenicom MSE. Izrazi koji se tiču njih su navedeni ispod

$$L_{PIX\_L1}(I, I_g) = \frac{1}{hwc} \sum_{i,j,k} |I_{i,j,k} - I_{g_{i,j,k}}| \quad (21)$$

$$L_{PIX\_L2}(I, I_g) = \frac{1}{hwc} \sum_{i,j,k} |I_{i,j,k} - I_{g_{i,j,k}}|^2 \quad (22)$$

Bitno je napomenuti da funkcija gubitka L1 u nekim slučajevima nije moguća za izračunati pa se koristi varijanta od L1 funkcije koja se zove Charbonnier gubitak [48] dana izrazom

$$L_{PIX\_CH}(I, I_g) = \frac{1}{hwc} \sum_{i,j,k} \sqrt{|I_{i,j,k} - I_{g_{i,j,k}}|^2 + e^2} \quad (23)$$

gdje je  $e$  konstanta koja osigurava numeričku stabilnost.

Ova metoda inače osigurava da slika HR koja se generira ima iste vrijednosti piksela kao i stvarna slika HR. Nedostaci, pored već spomenutog, bi još bili da L2 gubitak daje veći naglasak na težine većih razlika nego manjih, što rezultira onda s ili izraženijim rubovima ili mekanijim, ukratko ekstremnim slučajevima u ta dva smjera. Tako da se L1 više koristi nego L2, a nadalje isto bitno za spomenuti da kako je gubitak L1 povezan s metrikom PSNR i njenim izrazom, kada pokušavamo minimizirati L1 zapravo povećavamo PSNR, a tu se sada možemo sjetiti koji su bili nedostaci PSNR metrike. Uglavnom, mreže SR koje koriste funkcije gubitka piksela imaju općenito problema s detaljima visokih frekvencija te su slike, iako glatke, ne baš realistične.

### **Gubitak rekonstrukcije stila**

Idealna bi rekonstruirana slika HR trebala imati usporedne stilove stvarnoj slici HR, a to se misli na boje, teksture, gradijent, kontrast i ostalo. Tako imamo ovu funkciju gubitka rekonstrukcije koja želi podudariti detalje tekstura referentne slike s onom generiranom. Ta korelacija detalja tekstura i značajki mapa različitih kanala je dana s Gram matricom  $G^{(l)}$  [49].

$G_{i,j}^{(l)}$  je točkasti produkt (skalarni umnožak) značajki  $i$  i  $j$  s slojem  $l$ , danih izrazom

$$G_{i,j}^{(l)} = \text{vec} \left( ch_i^{(l)}(I) \right) \text{vec} \left( G_j^{(l)}(I) \right) \quad (24)$$

gdje  $vec$  predstavlja operaciju vektorizacije odgovarajućih kanala na  $i$ -tom kanalu od mape značajki u sloju  $l$ . A gubitak tekstura je dan izrazom

$$L_{TEX}(I, I_g; ch, l) = \frac{1}{c_l^2} \sqrt{\sum_{i,j} \left( G_{i,j}^{(l)}(I) - G_{i,j}^{(l)}(I_g) \right)^2} \quad (25)$$

Primjer uporabe ove funkcije gubitka imamo u EnhanceNet [61].

### Gubitak ukupne varijacije

Gubitak ukupne varijacije je koristio vrijednosti piksela od susjednih piksela, te je bio definiran kao suma apsolutne razlike između vrijednosti tih susjednih piksela, prikazana izrazom

$$L_{TV}(I_g) = \frac{1}{hwc} \sum_{i,j,k} \sqrt{\left( I_{g_{i+1,j,k}} - I_{g_{i,j,k}} \right)^2 - \left( I_{g_{i,j+1,k}} - I_{g_{i,j,k}} \right)^2} \quad (26)$$

### Gubitak dosljednosti ciklusa

Primjerak ove izvedbe imamo u CycleGAN [50], gdje se je koristio gubitak dosljednosti ciklusa. Ideja je bila jednostavna, da se generira jedna dodatna slika LR iz generirane slike HR, koja bi se uspoređivala s originalnom slikom LR, tako da smo imali dvije usporedbe HR-HR i LR-LR, originalnih i generiranih.

U samoj praksi su se zapravo često koristile kombinacije više različitih funkcija gubitaka ovisno o tome kakva je primjena bila same metode SR. Izbor prikladnih težina i samih funkcija gubitaka nije nimalo lagan zadatak, i dapače ovo je jedan cijeli dio problema učenja mreža koji može biti potpuno odvojena tema sama za sebe, no kako to nije cilj ovoga rada dublje u to zasad nemamo namjere ulaziti.

## 3.4. Strategije učenja

Strategije učenja imaju direktnog utjecaja na performanse bilo kojeg algoritma SR, a i sama evaluacija ovisi o izabranoj strategiji učenja. U ovom dijelu će biti kratko predstavljene neke od osnovnih strategija učenja koje se koriste u sklopu SR, direktno se nadovezujući na prethodno predstavljene funkcije gubitaka.

## **Učenje kurikuluma**

Metoda koja se prilagođava promjenljivoj težini zadataka, primjer za SR, kretanjem od jednostavnih slika s minimalnim sumom, do kompleksnih slika. Kako područje SR uvijek pati od nepovoljnih uvjeta, jer znamo kako na slikama možemo imati niz problema, od osvijetljenja pa do samih boja, zamućenosti, izoštrenosti itd., ovaj pristup s učenjem kurikuluma je prikladan u SR uglavnom za težinu učenja i veličinu mreže. Konkretno, za smanjenje težine uvježbavanja mreže SR koristimo manji faktor skaliranja; prvo se uvježbava s povećanim uzorkovanjem od 2x, te postepeno dolazi do faktora 4x, zatim 8x itd. Primjeri ove primjene i strategija učenja kurikuluma bi bili ProSR [51], ADRSR [52], i CARN [53].

Druga alternativa, odnosno pristup ideji, je da krećemo od većeg problema i rješavamo ga tako da razbijemo na više manjih pod-skupova, npr. želimo povećati sliku s povećanjem uzorkovanja 8x, i taj problem onda razbijemo na 3 manja problema, a to su: 1x do 2x, 2x do 4x, i onda 4x do 8x. Uglavnom, s kurikulumom učenja postizemo da smanjimo računalne troškove, te je zato ono bitno uzeti u obzir.

## **Normalizacija serije**

Normalizacija *batch*-a (BN, eng. *Batch normalization*) stabilizira i ubrzava duboke mreže CNN na način da smanjuje unutarnji kovarijantni pomak mreže. U sklopu SR ona je odgovorna za rad na među-srednjim mapama značajki, i tako potpomaže uklanjanju problema nestajućeg gradijenta dokle istovremeno dopušta veliku stopu učenja. Korištena je bila u jako mnogo modela, no u nekim modelima je bila uklonjena i umjesto nje je bila korištena dodatna memorija da bi se dizajnirao jedan veliki model koji je imao čak bolje performanse.

## **Višestruki nadzor**

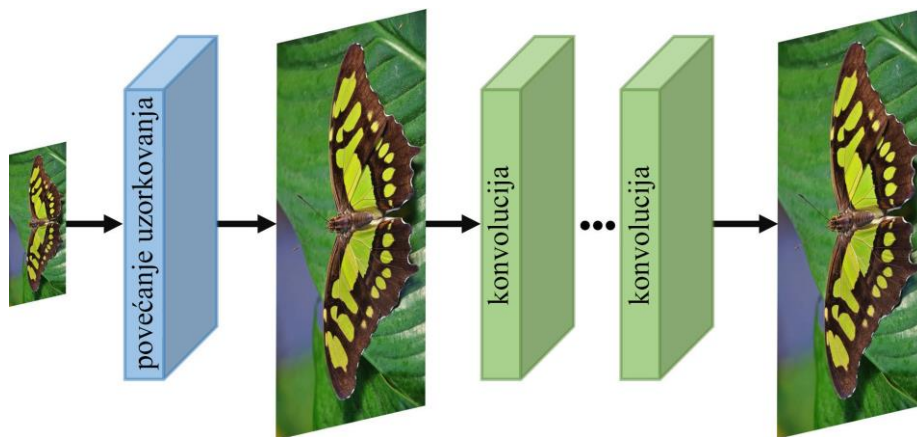
Jedan drugi način za poboljšavanjem gradijenta propagacije i izbjegavanjem problema onog nestajućeg gradijenta imamo višestruki nadzor. To su napravili uključivanjem višestrukog nadzora u rekurzivne jedinice da bi se izbjegao taj problem. U sklopu SR, višestruki nadzor se implementira tako da se na pažnju stave i podese pojedini faktori funkcije gubitka, što poboljšava onda put propagacije unatrag i smanjuje težinu uvježbavanja samog modela.

### 3.5. Okviri SR

Kako smo spomenuli, povećanje uzorkovanja je jedna od ključnih metoda za performanse samih metoda SR. Ovisno o strategijama učenja, metodama povećanja uzorkovanja, vrstama mreža, imamo nekoliko bitnih okvira unutar SR, a u nastavku će detaljnije biti predstavljeno njih četiri, s naglaskom na metodu povećanja uzorkovanja u njima.

#### Prethodno povećanje uzorkovanja SR

Ovim okvirom se uče funkcije za povećanje uzorkovanja za direktno preslikavanje slike LR na sliku HR, gdje se slika LR već povećano uzorkuje na početku, a onda korištenjem raznih konvolucijskih slojeva izvuče prikaz slike na iterativan način korištenjem dubokih neuronskih mreža. Korištenjem ovog koncepta nastao je SRCNN [54], a izvedbu te ideje imamo prikazano na Slici 15.



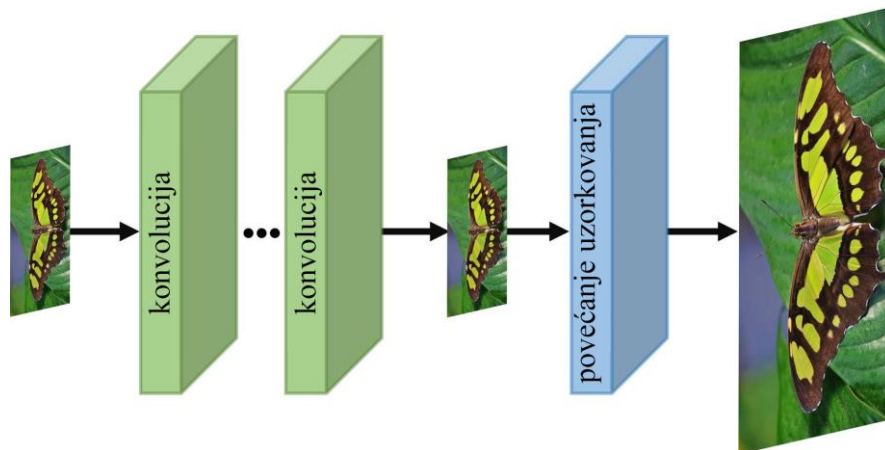
Slika 15. Cjevovod mreže SR temeljene na prethodnom povećanju uzorkovanju prema [2]

SRCNN se koristi da se nauči mapiranje slika LR-HR s kraja na kraj, koje su nastale konverzijom od strane mreža CNN. Prvo se znači dobije slika LR iz slike HR, a onda bi se koristile te mreže CNN da se nauči mapiranje prikaza slike HR.

Zbog toga što se u ovome okviru već izvede pretvorba stvarnog piksela s slojem za prethodno povećanje uzorkovanja, mreža mora te rezultate prepraviti korištenjem mreža CNN, a ovim načinom se je uspjela smanjiti težina učenja. Nedostatak korištenja ovih okvira je isti kao kod već na početku spomenutih klasičnih metoda, a to bi bile zamućenost i pojačavanje suma u povećanim slikama, što je na kraju normalno rezultiralo i na zaključnoj slici HR. Jedino novo vrijedno spomena je da su se računalni troškovi i zahtjevi memorije i ostalih resursa povećali, jer već na samom početku povećavamo sliku, pa je jasno zašto nam treba i više resursa.

## Naknadno povećanje uzorkovanja SR

Metoda koja je nastala s ciljem da se smanje zahtjevi memorije i poveća računalna efikasnost. Koristi duboko učenje da bi mapirala funkcije u prostoru manjih dimenzija, a time uštedila na navedenom. Na Slici 16. imamo prikaz te izvedbe. Zbog očiglednih prednosti, niskih računalnih troškova, ovaj model je vidio široku primjenu u SR jer je zahvaljujući ovoj strukturi kompleksnost modela općenito reducirana.



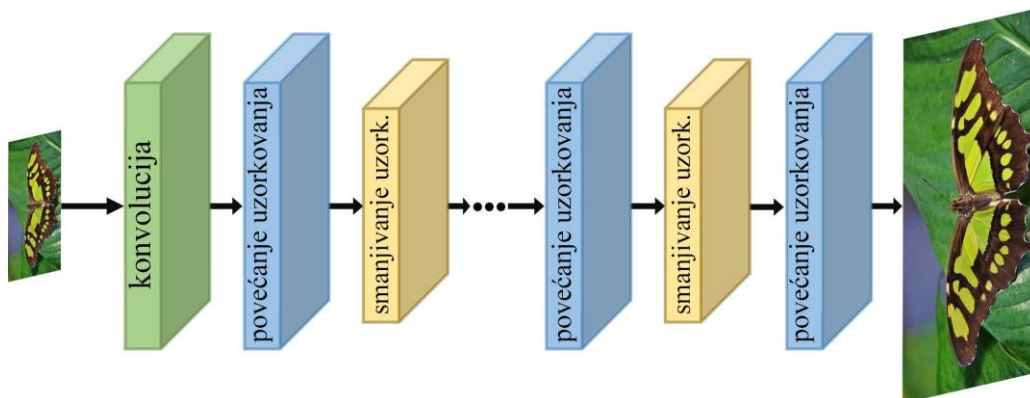
Slika 16. Cjevovod mreže SR temeljene na naknadnom povećanju uzorkovanju prema [2]

## Iterativno povećanje-i-smanjivanje uzorkovanja SR

Tip okvira gdje se je pokušao riješiti problem mapiranja slika LR-HR na efikasan način učenjem sparenih slika LR-HR uz pomoć propagacije unatrag. Izvedbu toga imamo za primjer na Slici 17.

Ovaj model prepravlja slike korištenjem rekurzivne propagacije unatrag, a to radi konstantnim mjerenjem greške i prepravljajanjem modela temeljenog na greški rekonstrukcije. Primjerak imamo s DBPN metodom [55], koja je koristila ovaj koncept, te je finalna slika dobivena ovom metodom bila konstruirana korištenjem među-generacija od slike HR. Još imamo par varijacija za primjer, kao SRFBN [56] i RBPN [55].

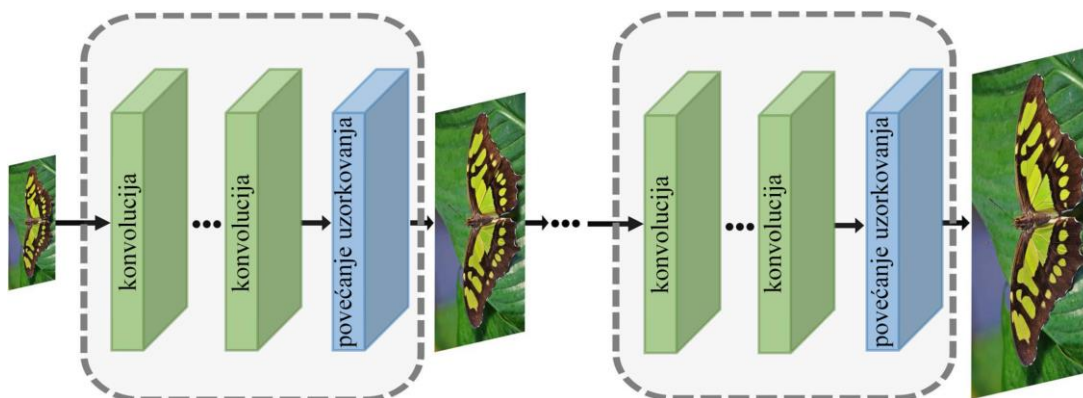




Slika 17. Cjevovod mreže SR temeljene na iterativnom povećanju-i-smanjivanju uzorkovanja prema [2]

### Progresivno povećanje uzorkovanja SR

Kod okvira s naknadnim povećanjem uzorkovanja, imali smo bili jedan sloj za povećanje uzorkovanja na kraju i učenje je bilo fiksirano za faktore skaliranja. Sada, kada bi htjeli povećati skalu, odnosno da ovo radi za više različitih skala, korištenjem tog oblika i njegovom prilagodbom dalje bi drastično skočili računalni troškovi. To se je pokušalo riješiti izvedbom ovog okvira za progresivno povećanje uzorkovanja, koja bi postepeno dizala skaliranje, onoliko koliko je bilo potrebno. Primjer izvedbe ovog okvira imamo na Slici 18.



Slika 18. Cjevovod mreže SR temeljene na progresivnom uzorkovanju prema [2]

Za primjerak samog modela imamo LapSRN [57], koji je koristio module temeljene na stepenastim mrežama CNN, koji su bili odgovorni za mapiranje jednog faktora skaliranja, a izlaz jednog tog modula bi bio ulaz slike LR u druge module koji su slijedili nakon. Ovaj okvir je isto tako bio i korišten u ProSR [51] i MS-LapSRN [57] modelima.

Ovaj model postiže veće stope učenja kako je težina učenja manja, budući je operacija SR podijeljena na više manjih zadataka povećanja uzorkovanja, što je jasnije i direktnije za učiti što se tiče mreža CNN.

### 3.6. Primjeri najsuvremenijih metoda SR

U zadnje vrijeme zastupljenost metoda koje primjenjuju istovremeno veći broj dosad predstavljenih metoda i tehnika je sve prisutniji, tako da imamo metode SR koje koriste po više različitih strategija učenja i pojedinih tehnika istovremeno. U nastavku će biti predstavljene one najznačajnije metode zadnjih deset godina, i njihove najosnovnije karakteristike, a u poglavlju koje slijedi i detaljnije jedna od novih metoda koja spaja više predstavljenih metoda u jednu cjelinu.

Karakteristike koje će biti navedene za predstavnike najsuvremenijih metoda SR su redom: naziv metode, godina objave i skraćenica konferencije na kojoj je bila predstavlja ova metoda, metoda povećanja uzorkovanja koju koristi (US metoda), mreža, okvir SR, funkcija gubitka, i najosnovnija znamenitost te metode.

Tako da, slijedi lista tih najsuvremenijih metoda SR:

- SRCNN [54] – 2014.godine na ECCV-u; bi-kubična US; CNN mreža; prethodno povećanje uzorkovanja;  $L_{L2}$ ; prva metoda SR temeljena na dubokom učenju
- DRCN [23] – 2016.godine na CVPR-u; bi-kubična US; mreža rekurzivnog i rezidualnog učenja; prethodno povećanje uzorkovanja;  $L_{L2}$ ; rekurzivni slojevi
- FSRCNN [58] – 2016.godine na ECCV-u; dekonvolucijski sloj US; naknadno povećanje uzorkovanja;  $L_{L2}$ ; lake težine
- ESPCN [59] – 2017.godine na CVPR-u; pod-piksel US; prethodno povećanje uzorkovanja;  $L_{L2}$ ; pod-pikseli
- LapSRN [57] – 2017.godine na CVPR-u; bi-kubična US; rezidualna mreža; progresivno povećanje uzorkovanja;  $L_{LI}$  i  $L_{PIX\_CH}$ ; kaskadna CNN
- DRRN [40] – 2017.godine na CVPR-u; bi-kubična US; rezidualna i rekurzivna mreža; prethodno povećanje uzorkovanja;  $L_{L2}$ ; blokovi rekurzivnih slojeva
- SRResNet [47] – 2017.godine na CVPR-u; pod-piksel US; rezidualna mreža; naknadno povećanje uzorkovanja;  $L_{L2}$ ; gubitak sadržaja
- SRGAN [47] – 2017.godine na CVPR-u; pod-piksel US; rezidualna mreža; naknadno povećanje uzorkovanja;  $L_{GAN}$ ; temeljena na GAN gubitku
- EDSR [60] – 2017.godine na CVPR-u; pod-piksel US; rezidualna mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; kompaktni dizajn

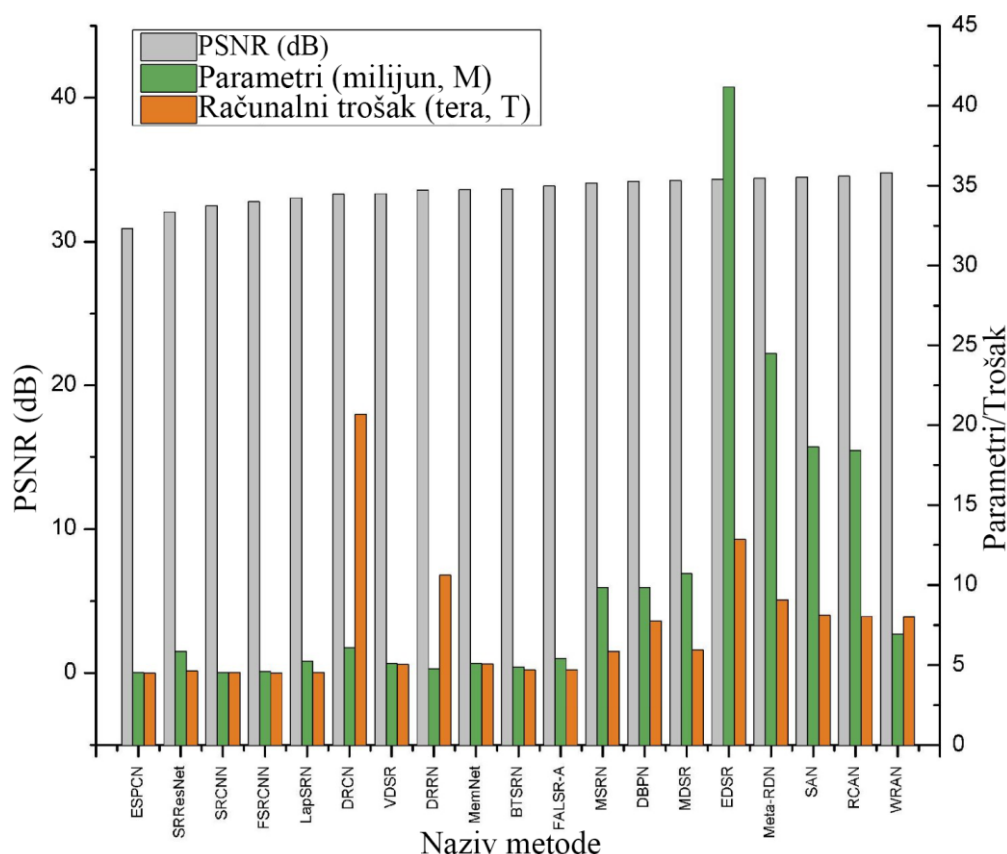
- EnhanceNet [61]– 2017.godine na ICCV-u; bi-kubična US; rezidualna mreža; prethodno povećanje uzorkovanja;  $L_{GAN}$ ; temeljena na GAN gubitku
- MemNet [40]– 2017.godine na ICCV-u; bi-kubična US; kombinacija rezidualne, rekurzivne i guste mreže; prethodno povećanje uzorkovanja;  $L_{L2}$ ; blokovi memorijskih slojeva
- SRDenseNet [38] – 2017.godine na ICCV-u; dekonvolucijski sloj US; rezidualna i gusta mreža; naknadno povećanje uzorkovanja;  $L_{L2}$ ; potpuno povezani slojevi
- DBPN [55] – 2018.godine na CVPR-u; dekonvolucijski sloj US; rezidualna i gusta mreža; iterativno povećanje-i-smanjivanje uzorkovanja;  $L_{L2}$ ; temeljeno na propagaciji unatrag
- DSRN [28] – 2018.godine na CVPR-u; dekonvolucijski sloj US; rezidualna i gusta mreža; prethodno povećanje uzorkovanja;  $L_{L2}$ ; mreža dvostrukog stanja
- ProSR, ProGanSR [51] – 2018.godine na CVPRW-u; progresivno povećanje US; rezidualna i gusta mreža; progresivno povećanje uzorkovanja;  $L_{LS}$ ; gubitak najmanjeg kvadrata
- MSRN [62] – 2018.godine na ECCV-u; pod-piksel US; rezidualna mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; višestruki put
- RCAN [63] – 2018.godine na ECCV-u; pod-piksel US; rezidualna i temeljena na pažnji mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; gubitak temeljen na pažnji
- ESRGAN [41] – 2018.godine na ECCV-u; pod-piksel US; rezidualna i gusta mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; temeljena na GAN gubitku
- Meta-RDN [64] – 2019.godine na CVPR-u; meta povećanje US; rezidualna i gusta mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; više-skalni model
- Meta-SR [64] – 2019.godine na CVPR-u; meta povećanje US; rezidualna i gusta mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; arbitraran faktor skaliranja kao ulaz
- RBPN [65] – 2019.godine na CVPR-u; pod-piksel US; rekurzivna mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; koristi SISR i MISR zajedno
- SAN [66] – 2019.godine na CVPR-u; pod-piksel US; rezidualna i temeljena na pažnji mreža; naknadno povećanje uzorkovanja;  $L_{LI}$ ; drugi-po-redu poredak pažnje
- SRFBN [56] – 2019.godine na CVPR-u; dekonvolucijski sloj US; kombinacija rezidualne, rekurzivne i guste mreže; naknadno povećanje uzorkovanja;  $L_{LI}$ ; put za *feedback*

- WRAN [43] – 2020.godine na *Neuro-computing*, bi-kubična US; rezidualna i temeljena na pažnji; prethodno povećanje uzorkovanja;  $L_{LI}$ ; temeljena na valiću

Kao što se može vidjeti iz priložene liste, modela i metoda ima jako puno, i bitno je napomenuti da je ovo lista njih najboljih do 2020. godine, dokle imamo naravno već objavljeno i nekoliko novijih kojih se mogu ubrajati o ovu kategoriju najboljih metoda. Zato će u sljedećem poglavlju biti predstavljen Real-ESRGAN, koji je nasljednik poznatog ESRGAN modela. [67]

Inače, u ovoj iznad prikazanoj listi najsvremenijih metoda SR, govorilo se o njihovim najosnovnijim značajkama, tj. karakteristikama strukture. Pored toga znamo da je još jako bitno kakve su njihove performanse, koje su računalno zahtjevnije, koje manje, koje su brže, koje preciznije i niz drugih usporedbi. U svrhu toga na Slici 19. imamo prikazano nekolicinu tih metoda, međusobno uspoređenih metrikom PSNR (predstavljenu s nekoliko parametara) i s računalnim troškom. Testirani su na skupovima podataka Set14, B100 i Urban100, a ukupni rezultat je dan prosjekom bodova postignutim na ta tri skupa podataka. Napomena da je faktor skaliranja za sve te metode bio postavljen na x2, i ovo je bitno napomenuti jer neki modeli su bili dizajnirani za veće faktore. I još za računalni trošak, mjerio je broj tera množenja-zbrajanja.

Kako je vidljivo na samoj Slici 19., top pet metoda na odabranim skupovima podataka uzevši u obzir PSNR metriku su WRAN s 34.790 dB, RCAN s 34.540 dB, SAN s 34.480 dB, Meta-RDN 34.400 dB i EDSR s 34.330 dB. Dalje gledajući na parametre imamo WRAN sa 2.710 milijuna parametara, dokle je EDSR pokazao čak 40.74 milijuna. Uzimaju se sve mjere u obzir, na način da je bolje što veći rezultat PSNR metrike, a što manji računalni trošak i što manji broj parametara, budući oni predstavljaju bolju računalnu efikasnost. Tako da na samome kraju najbolje metode bi bile WRAN i RCAN koji su pokazali dobre performanse na PSNR metrici s obzirom na računalni trošak i parametre. Jako bitno je napomenuti da je riječ ipak o specifičnom skupu podataka i da određene metode rade bolje na drugim skupovima podataka i slično, te da u pravilu biranje metode ovisi o mnogo toga. Sjetimo se isto nedostataka PSNR metrike koji se ne smiju nikada izostaviti.



Slika 19. Benchmark nekoliko selektiranih modela SR prema [2]

### 3.7. Super-rezolucija bez nadgledanja

Dosad su uglavnom bile predstavljene metode i modeli koji spadaju pod kategoriju metoda dubokog učenja s nadgledanjem, izuzev naravno onih općih pojmova koji se tiču svih modela SR. U ovom dijelu će kratko biti predstavljene karakteristike i vrste metoda bez nadgledanja.

Općenito, karakteristika metoda bez nadgledanja je da nema potrebe za parovima slika LR-HR da bi radila, što je ujedno bilo i glavno ograničenje metoda s nadgledanjem budući su se tamo slike LR generirale korištenjem poznatih degradacija. U ovom obliku učenja bez nadgledanja, model uči kako napraviti reverznu transformacijsku funkciju od degradacijske funkcije, konvertiranjem slike LR u sliku HR. Ovaj oblik modela je noviji, ali interes raste, a glavni razlog tome bi bio da se uče slike iz stvarnoga svijeta za SR, bez sparenih slika LR i HR, i na taj način odmah imamo uštedu na resursima.

## **Slabo nadzirana super-rezolucija**

Metoda koja koristi ne-sparirane slike LR i slike HR za uvježbavanje modela. Iako ova metoda i dalje koristi slike oba tipa, znači manje LR i veće HR, povezanosti među njima nisu definirane.

Imaju se dva pristupa:

1. Prvo se uči degradacijska funkcija, koja može generirati degradirane slike LR da bi uvježbali model da generira slike HR,
2. Iskoristiti degradacijsku funkciju za učenje i mapiranje LR-HR ciklički, potvrđujući rezultate same sa sobom.

## **Ciklička slabo nadzirana super-rezolucija**

Korištenjem slika LR i slika HR koje nisu sparane, i na koje se odnosi kao dva potpuno nepovezana skupa podataka, ova metoda koristi pristup ciklus-u-ciklusu da predvidi mapiranje funkcija tih dvaju skupova podataka. Konkretno, od LR do HR i od HR do LR slika. Ovo je rekurzivan proces gdje funkcije mapiranja generiraju slike s jednakom distribucijom, i onda se te slike ciklički nadodaju u drugo predviđanje.

Primjerak primjene ovoga imamo u CycleGAN-u, i ovo je okvir koji koristi četiri generatora i dva diskriminatora. Dva GAN-a koja se koriste uče prikaze od degradirane slike LR u sliku LR, i slike LR u sliku HR, te njihova mapiranja. Prvi generator je element koji uklanja šum i generira slike LR slične skale s tim uklonjenim sumom, a onda te slike služe kao ulaz drugog generatora koji obnavlja slike HR, kojega dalje potvrđuje kontradiktorna mreža, odnosno diskriminator.

## **Učenjem degradacijske funkcije**

Sličan koncept cikličkoj SR, ali s dvije mreže, jedna za učenje degradacijske mreže a druga za mapiranje LR-HR, i one se uvježbavaju odvojeno. Dvofazna metoda, gdje u prvoj fazi jedan GAN uči transformaciju slike HR u sliku LR, a u drugoj fazi taj izlaz prve transformacije služi kao ulaz drugome GAN-u da uči mapiranje prikaza transformacije LR slike u sliku HR.

## **Zero-shot super-rezolucija**

Ova metoda koristi jednu sliku da uvježba duboku mrežu učenja korištenjem tehnika augmentacije slike da nauči degradacijsku funkciju. Prvo se je predvidio degradacijski kernel, koji se dalje koristio za generiranje skaliranih slika, a u zadnjem koraku se je

uvježbavala SRCNN mreža koja je učila prikaz skupa podataka. Glavni problem ovog modela je vrijeme uvježbavanja, kako se je uvježbavala jedna slika kroz cijeli taj proces, vrijeme da se zaključi rezultat je bilo značajno dugo.

### **Slika prije u SR**

Sitni detalji se u bilo kojem problemu učenja mogu mapirati korištenjem mreža CNN, i tako smo imali primjer gdje se je isto koristilo u SR. Mreža nije uvježbana već bi koristila nasumični vektor  $v$  kao ulaz u model, i s time generira slike HR. Ova metoda želi odrediti sliku HR, koja kada se smanji vraća natrag sliku LR koja je slična ulaznoj slici LR.

## **3.8. Primjene metoda SR u specifičnim područjima**

U ovom dijelu će biti navedene neke od konkretnih primjena SR u specifičnim područjima, no njih naravno ima još mnogo više.

### **Super-rezolucija za slike lica**

Postoji jedno područje koje se naziva halucinacija lica (FH, eng. *Face Hallucination*), koji je dio super-rezolucije, orijentiranog na zadatak prepoznavanja lica. Primjena bi trebala biti očigledna, npr. u prometu ili na ulicama, imamo kamere koje uzimaju slike ljudi, često su te slike udaljene i trebali bi ih povećati da bi bolje vidjeti lica ljudi i prepoznali ih. Tako da je jako bitno imati modele koji mogu jako dobro rekonstruirati ljudsko lice s svim njegovim značajkama, što je poprilično težak zadatak. Konkretno primjere modela koji se bave ovakvim tipom zadatka imamo CBN [69], FSRNet [70], SICNN [71], Super-FAN [72], TDAE [73], Attention-FH [74], MLCMR [75] itd.

### **Super-rezolucija za slike iz stvarnog svijeta**

Posljedica senzora s kojima kreiramo slike iz stvarnoga svijeta je da ovisno o nizu parametara možemo rezultirati s rezultatom koji nam nije nimalo prihvatljiv, a nažalost nemamo ponovnih prilika da slikamo tu sliku, npr. slike utrka automobila, slike skokova životinja i slično. Zapravo već od samog početka, s činjenicom da *raw* sliku, koja je uglavnom prostora boja od 14-bitova pa i naviše, moramo konvertirati i prikazati kao finalnu 8-bitnu sliku RGB prostora boja, već nalaže da imamo dosta velikog gubitka informacija na slici. Ovo je i opći problem uvježbavanja modela za povećanje slika iz stvarnog svijeta, jer kako su slike koje njima dajemo na uvježbavanje već s velikim gubitkom informacija, pitanje je kako uopće izvesti to da uvježbamo model (mrežu) da može kreirati slike koje bi

odgovarale onima iz stvarnog svijeta. Zbog toga je ovo još u procesu razvoja i moglo bi se reći da se svi modeli najviše muče s ovim dijelom. Uglavnom, za rješavanje ovog problema prvo bi trebali kreirati potrebne skupove podataka za daljnje uvježbavanje modela, a za primjer toga imamo skup podataka SR-RAW koji je bio korišten da se kreira skup podataka City100 koji predstavlja slike iz stvarnog svijeta. Ipak, još se puno treba poraditi na svemu ovome da to budu rezultati kojima stvarno možemo reći da smo zadovoljili prave slike iz stvarnog svijeta. [2]

### **Karte dubine u super-rezoluciji**

U području računalnog vida, zadaci kao segmentacije slike i procjena poze, su se pokušali riješiti s mapama dubine. One služe da bi sačuvali udaljenost informacije s scene(slike) i promatrača, i kako su one uglavnom malih rezolucija zbog ograničenja hardvera u sustavima kamera i sličnom, potrebno je imati nešto da bi povećali rezoluciju tih slika da budu jasnije.

### **Detekcija na daljinu i snimanje satelitima**

Jako široka primjena ovih tehnika danas u svijetu, a kako vrijeme prolazi sve i učestalije se koristi detekcija objekata na daljinu, snimanja površine zemlje ili svemira satelitima, detekcije objekata na nebu i slično. Dovoljno je jasno zašto bi SR imala primjene za ovako nešto. Konkretni primjer imamo s YOLO algoritmom za detekciju objekata i modelom SRCGAN-RFA-YOLO [76], gdje se je koristi združivanje rezidualnih značajki i ciklički GAN da bi se poboljšala rezolucija usred daljinskog detektiranja objekata na slikama.

### **Video super-rezolucija**

Odnosi se na podatke oblika video, što je ništa drugo nego više sličica povezanih i prikazanih u vremenskim intervalima. Ipak, u sklopu SR ovo je već dosta zahtjevnije budući želimo povećati veliki broj sličica, a dosad smo vidjeli kako je zapravo bilo teško povećati i samo jednu sliku. Ukratko, izvedivo je, samo hardverski znatno zahtjevnije, a i vremenski. Primjera imamo no ipak je manje istraživano nego li i sama slika. Naravno, imamo i specifičnih slučajeva kao u uvodu spomenuti DLSS od NVIDIA-e, koji se ne mogu baš uklopiti u ovu kategoriju, ali čisto je vrijedno spomenuti da kako je to bilo izvedivo, kako vrijeme prolazi biti će i puno više pristupačnijih alata i modela za video SR. Neki od konkretnih primjera modela bi bili CVSRnet [77], VESPCN [59], STCN [78], i nešto novije, kao STMN [79].



## **Super-rezolucija u medicinskim snimkama**

Spomenuto isto kratko pri početku ovoga rada, s naglaskom na to kako je ovdje riječ o slikama specifičnog oblika i specifičnih karakteristika, potrebni su ipak malo specifičniji modeli. Primjena bi inače trebala biti isto očigledna, u svrhu bolje i jasnije detekcije raznoraznih bolesti i sličnog, jer zbog raznih uvjeta slike ne budu uvijek potpuno jasno vidljive, i onda doktori moraju na temelju iskustva nagađati ili tražiti novo snimanje čak. Uglavnom, za primjer modela imamo DeepResolve [80], gdje se je uvježbavalo na slikama magnetske rezonancije (MR, eng. *Magnetic resonance*).

## **Ostala područja**

Daljnji primjeri su u stereo slikama, optičkoj mikroskopiji, općenito medicini, računalnom vidu itd. Ne bi bilo pogrešno reći da će se koristiti gotovo u svemu što koristi slike u nekom obliku, odnosno digitalne prikaze, kako smo već i pokušali ilustrirati pričom u uvodu i poglavlju koje je slijedilo.

## 4. Real-ESRGAN

Za primjerak jedne od *state-of-the-art* (najsuvremenijih) metoda SR, odabran je Real-ESRGAN. Odabran je iz više razloga, no glavni su:

- nadograđuje i nadovezuje se na ESRGAN koji je pokazao impresivne rezultate,
- dostupan je svima za otvoreno istraživanje i korištenje u osobnim potrebama, te tako od strane zajednice i korisnika ima veliki broj otvorenih vodiča online,
- jednako tako zbog toga što ima korisna grafička sučelja (GUI, eng. *Graphical User Interface*) za laku upotrebu lokalno, i zbog toga što je dostupan čak i na Google Colab-u u slučaju da nemate uvjete za obradu lokalno,
- osobnom povijesti s projektima i alatima kao waifu2x [93], koji koriste ovaj model za povećanje slika i sličnog.

U nastavku će kratko biti predstavljeni najbitniji dijelovi Real-ESRGAN-a, kao što su to arhitektura mreže, skupovi podataka, dodatne funkcije itd. Uglavnom, biti će dana na uvid teorijska osnova i karakteristike ovog modela, dokle u poglavlju iza s eksperimentom će biti preispitana valjanost i kvaliteta ovog modela na različitim slikama, u usporedbi s nekoliko drugih modela.

### 4.1. Općenito

Spomenuli smo u više navrata kako je teško uvježbavati modele da rade za sve tipove slika, iz zapravo veoma logičnih razloga, dapače i s naglaskom na zašto su stvarne slike najzahtjevnije, te unatoč brojnim pokušajima suočavanja s tim izazovima i kreiranja modela koji bi zadovolji stvarne slike i uvjete slijepe SR, mi dan danas još nemamo model koji bi zadovoljio to. Tvorci ESRGAN-a su se tako pokušali na svoj način suočiti s time i izvesti model koji bi to zadovoljio, ili barem pokušao zadovoljiti donekle, tako da su nadogradili ESRGAN s primjenom praktične restauracije i uvježbavali ga na potpuno sintetičkim podacima. Taj novi model su nazvali Real-ESRGAN, jer je on zapravo predstavljao proces modeliranja degradacije visokog reda, kreiranog da bolje simulira kompleksne degradacije slika stvarnog svijeta.

Prije govora o konkretnim detaljima strukture Real-ESRGAN-a, prvo se prisjetimo malo osnovnih pojmova super-rezolucije. Imamo super-rezoluciju i dio nje koji se zove super

rezolucija jedne slike (SISR, eng. *Single Image Super-resolution*), u kojoj je kako smo do sada sve govorili, cilj bio rekonstruirati sliku HR iz njenog suprotnog dijela, a to je bila slika LR. Tako smo često bili imali one modele s sparenim LR-HR slikama, ili one koji su bili ne-sparirani, odnosno koji su sami kreirali sliku LR iz slike HR za daljnje uvježbavanje itd. Nadalje, sjetimo se i prvog modela SRCNN s kojim je započeo fokus na duboko učenje i duboke neuronske mreže u sklopu SR sa svim onim modelima koji su koristili mreže CNN nakon ovog modela. Njihov glavni nedostatak je bio u tome da svi imaju pristup gdje se pretpostavio idealni bi-kubični kernel smanjivanja uzorkovanja, a on je općenito drukčiji od degradacija prisutnim u stvarnim slikama, što je logično onda bili nepraktično za primjenu na slikama iz stvarnog svijeta. S druge strane od SISR, imamo pojam slijepe super-rezolucije, koja je predstavljala oblik SR gdje želimo povećati slike LR koje su patile od nepoznatih i kompleksnih degradacija. Postojeći pristupi rješavanja problema slijepe SR se mogu ugrubo kategorizirati na eksplicitno modeliranje i implicitno modeliranje. Klasični degradacijski model koji uključuje i zamucenje, smanjenje uzorkovanja, šum i posljedice JPEG kompresije je u širokoj primjeni u metodama eksplicitnog modeliranja. Degradacije slika stvarnog svijeta su uglavnom previše kompleksne da bi bile modelirane jednostavnom kombinacijom višestrukih degradacija, te će zato ove spomenute metode veoma lako pogriješiti na uzorcima iz stvarnog svijeta. Još imamo i metode implicitnog modeliranja, koje koriste distribuciju podataka s mrežama GAN da bi dobile degradacijski model, no one su isto imaju svoja ograničenja, a to su degradacije unutar skupa podataka za uvježbavanje, te zato onda ne mogu dobro generalizirati slike koje su im bile van distribucije.

Prethodno spomenute izazove su znači tvorci ESRGAN-a probali riješiti novim modelom kojeg su nazvali Real-ESRGAN. Te stvarne kompleksne degradacije uglavnom dolaze od kompliciranih kombinacija različitih degradacijskih procesa. Za primjer imamo kada slikate sliku kamerom na mobitelima, te slike zbog ograničenja senzora i njihovih karakteristika, imaju nekoliko degradacija dodatno, kao što su to zamucenost, šum senzora, artefakti izoštravanja i JPEG kompresija. Onda dalje te slike još uređujemo i objavljujemo na stranicama ili aplikacijama socijalnih medija, što još dalje uzrokuje dodatnu kompresiju i nepredvidive šumove. Sa svakim daljnjim dijeljenjem i prebacivanjem slike preko interneta mogu se dodatno oštetiti te slike. Ovi problemi su naveli tvorce da naprave prijelaz s klasičnog degradacijskog „prvi-poredak“ model na degradacijski „visoki-poredak“ model da bi se suočili s tim degradacijama iz stvarnog svijeta. Napravili su to ponavljanjem nekoliko degradacijskih procesa, od kojih je svaki taj proces zapravo bio klasični degradacijski model.

Isto tako su empirički iskoristili i degradacijski „drugi-poredak“ proces da daju ravnotežu između jednostavnosti i efikasnosti modela. Dodatno su još koristili i *sinc* filtere u procesu sinteze da simuliraju uobičajeno odzvanjanje i artefakte koji prelijetaju van dometa. Prostor degradacije je puno veći nego kod ESRGAN-a te je ovo učinilo uvježbavanje puno izazovnijim. Konkretno, diskriminator traži puno moćnije sposobnosti da može diskriminirati stvarnost kompleksnih izlaza uvježbavanja, dokle povratna veza gradijenta od diskriminatora treba biti više precizna za pojačavanje lokalnih detalja. Tako da su tu unaprijedili VGG tip diskriminatora s U-net dizajnom. Pored toga da bi se izbjegla posljedica nestabilnosti usred uvježbavanja tih kompliciranih degradacija, iskoristili su spektralnu normalizaciju (SN, eng. *Spectral normalization*) regulacije da stabiliziraju dinamiku uvježbavanja. S ovim dodacima uvježbavanje modela Real-ESRGAN je bilo znatno olakšano i ostvarena je dobra ravnoteža između pojačavanja lokalnih detalja i supresije artefakta. Na Slici 20. možemo vidjeti primjere rezultata koje daje Real-ESRGAN u usporedbi s par drugih metoda (modela), kako je spomenuto cilj mu je bio na pojačavanju detalja i supresiji artefakta, no vezano za ovaj zadnji dio, odmah možemo pretpostaviti kako na pojedinim slikama to može biti i negativno, jer će ukloniti teksturu ili gradijent koji je zapravo bio bitan detalj a on ga je odlučio suprimirati. Detaljnije o ovome u nastavku.



Slika 20. Usporedba Real-ERSGAN-a s nekoliko drugih metoda na dvije različite slike [67]

## 4.2. Metodologija

### Klasični degradacijski model

Kako smo već spomenuli, slijepa SR želi restaurirati slike HR iz slika LR koje imaju nepoznate i kompleksne degradacije, odnosno većinski te stvarne slike. U svrhu toga uglavnom se koristi klasični degradacijski model da bi sintetizirao ulaz niske rezolucije. Ono

što se generalno radi je, da istinitu sliku prvo stavljamo na proces konvolucije s kernelom zamućenja. Zatim se obavlja operacija smanjivanja uzorkovanja s odgovarajućim faktorom skaliranja, i onda se dobije slika LR dodavanjem šuma. Ovim ukupnim procesom se pokušava izvesti kompresija JPEG koja je u širokoj primjeni kod stvarnih slika.

### **Zamućenje**

Zamućenje se uglavnom modelira kao konvolucija s linearnim filterom (kernelom) zamućenja. Česti izbori su izotropni i anizotropni Gaussian filteri. Odmah i napomena da ti filteri iako često korišteni ne mogu baš u potpunosti zadovoljiti zamućenost prave kamere. Zbog toga su u Real-ESRGAN model uključili više različitih oblika tih filtera, sa poravnom distribucijom. Zahvaljujući ovome su se postigli izoštriji rezultati.

### **Šum**

U obzir su uzeta dva često korištena oblika šuma: aditivni Gaussian šum i Poisson-ov šum. Aditivni Gaussian šum ima funkciju vjerojatnosti gustoće jednaku normalnoj (Gaussovoj) distribuciji. Intenzitet šuma je kontroliran standardnom devijacijom, konkretno sa sigma vrijednosti, ove normalne distribucije. Kada svaki kanal od RGB slike ima nezavisni uzorak šuma, tada je sintetički šum onda šum boje. Na taj način se sintetizira i sivi šum, korištenjem istog uzorka šuma na sva tri kanala boja.

Poisson šum prati Poisson distribuciji, i uglavnom se koristi da bi približili model šumu senzora uzrokovanog statističkim kvantnim fluktuacijama, odnosno varijaciji u broju fotona koji osjete senzori na određenoj ekspoziciji. Ovaj šum ima intenzitet proporcionalan intenzitetu slike, a šumovi na različitim pikselima su nezavisni.

### **Promjena veličine – smanjivanje uzorkovanja**

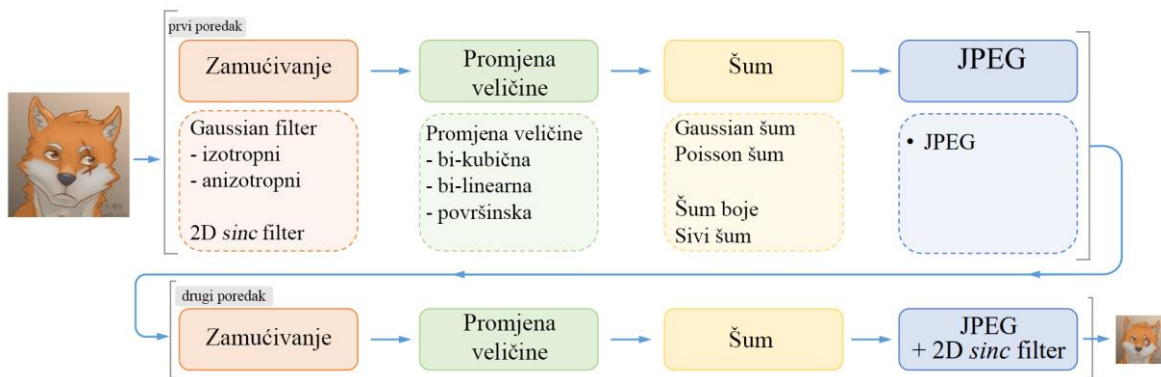
Smanjivanje uzorkovanja je osnovna operacija u sintetiziranju slika LR u sklopu SR, o čemu smo dosad već dosta bili pričali, kao i o suprotnome povećanju uzorkovanja. Već smo pričali detaljno o metodama za izvođenje ovog procesa, tako da ćemo samo kratko spomenuti što se tiče dijela za Real-ESRGAN. Tako da bi oni ovdje postigli različite i kompleksnije efekte promjene veličine, koristili su nasumično odabranu metodu od onih o kojima smo dosad bili govorili, a to su: interpolacija najbližeg susjeda, bi-linearna interpolacija, bi-kubična interpolacija itd. Ipak, kako je interpolacija najbližeg susjeda stvarala problem s neusklađenosti, bila je izbačena iz izbora i koristili su se same preostale navedene.

## **JPEG kompresija**

Prisjetimo se ponovo kratko rasterske grafike i pojmova vezanih za nju, pa tako znamo kako se koristi kompresija s gubitkom za digitalne slike. Ne zaboravimo tako i na konvertiranje boja prostora, jer nama ipak treba RGB prostor boja. Najveći problem ove kompresije su oni artefakti. Općenito, kvaliteta komprimiranih slika ovisi o faktoru kvalitete  $q$ , koji se kreće od 0 do 100, gdje što manja vrijednost označava veći omjer kompresije i lošiju kvalitetu. U sklopu modela Real-ESRGAN, koristi se implementacija DiffJPEG iz PyTorch-a.

## **Model degradacije visokog poretka**

Korištenjem samo klasičnog degradacijskog modela, uvježbani model bi već mogao srediti i neke prave primjere, ali još ne dobro one iz stvarnog svijeta s kompliciranim degradacijama, a pogotovo ne one s nepoznatim šumovima i kompleksnim artefaktima. Zbog toga je predložen ovi model degradacije visokog poretka da bi se sredile te praktične degradacije i smanjio razmjer razlike sintetičkih slika LR, dobivene klasičnim modelom, s onima iz stvarnog svijeta. Tako da, ovaj model degradacije visokog reda je dodan u Real-ESRGAN da bi se suočili s svim onim prethodno navedenim problemima koji prate JPEG kompresiju i slično. Klasični degradacijski model se još i poznaje pod nazivom model prvog poretka („prvi poredak model“), a ovaj model degradacije visokog poretka, predstavlja zapravo model  $n$ -tog poretka, gdje je zapravo to ništa drugo nego  $n$  ponavljanih procesa koji su ekvivalentni klasičnom degradacijskom modelu, tj. model visokog poretka je zapravo ništa drugo nego više klasičnih degradacijskih modela na više razina. Tako da je ovdje iskorišten taj model degradacije visokog poretka, na način da su kombinirani degradacijski model prvog poretka i drugog poretka, odnosno dva klasična degradacijska modela uzastopno, i jedino je bitno napomenuti da taj drugi poredak koristi drukčije hiper-parametre u odnosu na prvi poredak. Na Slici 21. možemo vidjeti pregled generiranja sintetičkih podataka u modelu Real-ESRGAN, nastalog implementiranjem ovog modela degradacije visokog reda.



Slika 21. Pregled generiranja sintetičkih podataka u modelu Real-ESRGAN implementiranjem modela degradacije visokog reda prema [67]

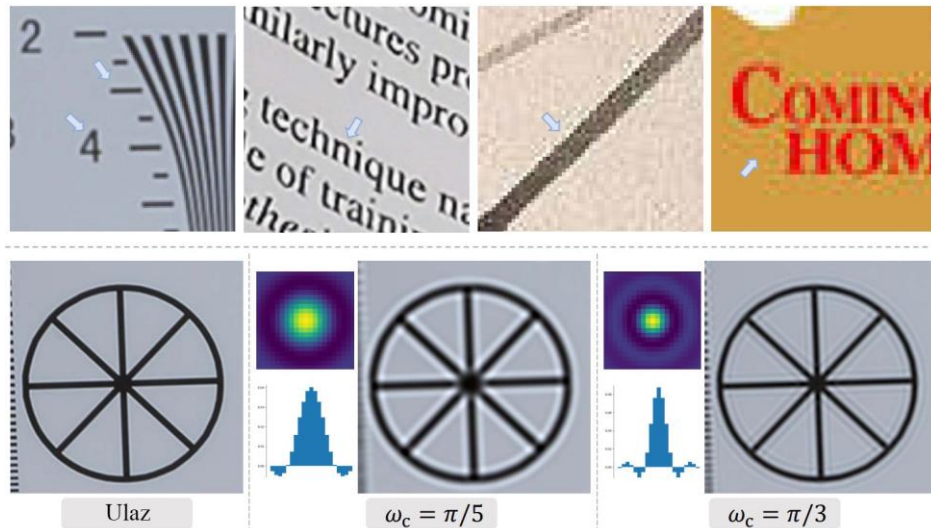
Napomena da ovaj model degradacije visokog reda je drukčiji od onih koji se koriste u matematičkim funkcijama, odnosno ima drukčije značenje u sklopu ovog područja SR, gdje uglavnom označava vrijeme implementacije iste operacije, jer možemo vidjeti u prikazanome modelu kako imamo više istih operacija međusobno povezanih u niz(poredak). Inače, jedna od nuspojava biranja strategije za povećanje uzorkovanja, kako se ovdje ponavlja taj proces degradacije, može dati rezultate na slikama s duplim zamučivanjem, što nije poželjno. Tako da ukratko, ovo i dalje nije savršeno rješenje problematike slijepe SR, ali barem daje na uvid mogućnost približavanja eventualnom pravom rješenju.

### Artefakti zvonjave i prekoračenja

Bilo tko tko je vidio barem jednu sliku rasterske grafike, imao je prilike suočiti se s ovim pojmovima, često i bez potrebe za zumiranjem, jer je već to jednostavno toliko zastupljeno na slikama rasterske grafike. Uglavnom, artefakti zvonjave su ti mutni i lažni rubovi koje vidimo okolo oštih prijelaza rubova nekog objekta na slici, gdje mi znamo da u stvarnosti toga tu nema, ali na slici vidimo kako taj dio slike ne daje nam punu liniju već ona nekako „odzvanja“. Artefakti prekoračenja često dolaze u kombinaciji s onima od zvonjave, gdje jednostavno čine taj prijelaz na slici još gorim. Glavni uzrok ovog problema s artefaktima je ograničenje signala koji ne može podnijeti određene visoke frekvencije koje su karakteristične za stvarne slike i slično. Općenito, znamo kako su artefakti posljedica prirode rasterske grafike(JPEG kompresije) i ograničenja signala. Oni su jako učestali problem na slikama i nešto o čemu se treba uvijek voditi računa u sklopu SR.

Način na koji su ovaj problem pokušali riješiti u modelu Real-ESRGAN je korištenjem *sinc* filtera, a ovdje se možemo odmah lagano prisjetiti *sinc* funkcije iz 1.1 poglavlja o Lanczos

interpolaciji. Na Slici 22. možemo vidjeti primjer kako su iskoristili taj *sinc* kernel (filter) za filtriranje spomenutih artefakata, odnosno na donjem dijelu slike možemo vidjeti same *sinc* kernele i odgovarajuću sliku pored njih, koja je filtrirana tim kernelima za danu frekvenciju, a na gornjem dijelu možemo vidjeti primjere isječaka slika na kojima se vidi problem tih artefakata.

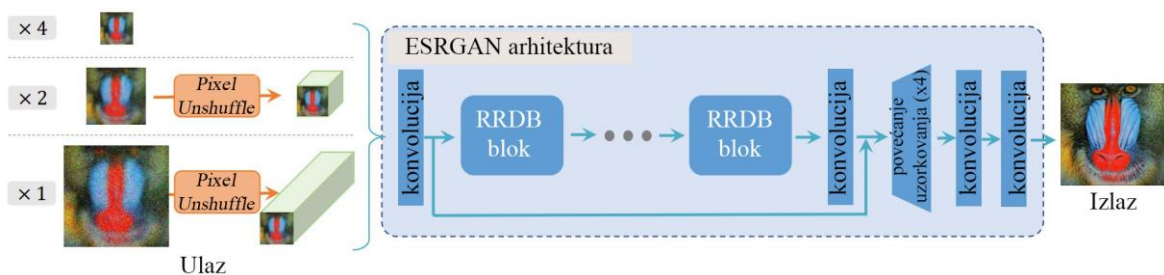


Slika 22. Gornji dio: artefakti na isječcima slika; Donji dio: Primjer filtriranja slike s *sinc* kernelima različitih frekvencija prema [67]

## Mreže i uvježbavanje

### ESRGAN generator

Koristio se isti ESRGAN generator kao za njihov ESRGAN model, koji je predstavljao duboku mrežu s nekoliko rezidualnih gustih blokova unutar rezidualnih gustih blokova, odnosno strukture RRDB (eng. *Residual-in-residual dense blocks*). Inače, proširili su originalnu x4 ESRGAN strukturu da može imati i x2 i x1 opcije za izvršavanje SR. Sve spomenuto je vidljivo na Slici 23. gdje je prikazana ta proširena struktura ESRGAN generatora.



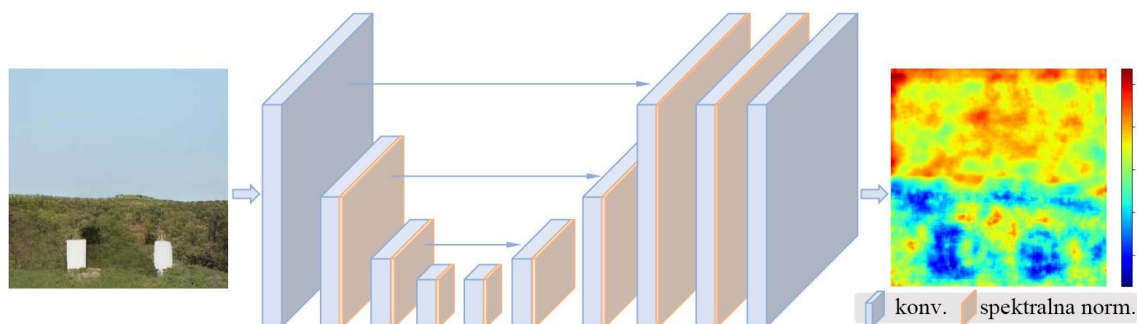
Slika 23. Arhitektura proširenog ESRGAN generatora korištenog u modelu Real-ESRGAN prema [67]



Bitno je spomenuti kako je ESRGAN sam po sebi dosta težak, odnosno teška mreža s puno slojeva, da bi smanjili prostornu veličinu i povećali veličinu kanala prije nego li počnemo dodavati ulaze u mrežu, izvodi se nešto što se zove poništavanje miješanja piksela (eng. *pixel unshuffle*), jasno prikazano na Slici 22. kao sami ulaz u ESRGAN mrežu, odnosno ta pred-obrađena *pixel unshuffle*. Tako da se većina računanja izvodi u manjem rezolucijskom prostoru, što može smanjiti računalne troškove potrebne za izvršavanje i GPU memoriju.

### U-Net diskriminator s spektralnom normalizacijom(SN)

Kako Real-ESRGAN cilja riješiti puno veći degradacijski prostor nego ESRGAN, originalni diskriminator nije više bio pogodan i trebao se je dizajnirati novi. Specifičnije, bio je potreban puno snažniji diskriminator s većom moći diskriminacije za uvježbavanje kompleksnih unosa. Nadalje ne smije diskriminirati na temelju globalnih stilova, a mora i proizvesti preciznu povratnu vezu za gradijent od lokalnih tekstura. Tako da je onda nastao novi diskriminator dizajna U-Net s vezama za preskakanje. Na Slici 24. možemo vidjeti strukturu tog diskriminatora.



Slika 24. Arhitektura U-Net diskriminatora s spektralnom normalizacijom prema [67]

Ovaj U-Net diskriminator izbacuje stvarnije vrijednosti za svaki piksel, i može dati detaljnu povratnu vezu po pikselu samome generatoru. Naravno, struktura i komplicirane degradacije isto povećavaju nestabilnost uvježbavanja. Zbog toga je bila iskorištena spektralna normalizacija regulacije da bi stabilizirali dinamiku uvježbavanja. Bilo je zamijećeno da ovaj oblik regulacije je pomogao i smanjenju prethodno spomenutih artefakta, koji se pojavljuju kao nuspojava uvježbavanja mreže GAN.

Sami proces uvježbavanja je inače podijeljen u dvije faze:

1. Uvježbava se model orijentiran PSNR metrikom s L1 gubitkom, i model koji se dobije tom fazom se zove Real-ESRNet,

2. Model dobiven 1.-om fazom se koristi kao inicijalizacija generatora, gdje se onda uvježbava pravi Real-ESRGAN u kombinaciji s L1 gubitkom, perceptivnim gubitkom i GAN gubitkom.

## 4.3. Eksperimenti

### Skupovi podataka i implementacija

#### Detalji uvježbavanja

Slično ESRGAN modelu, koristili su se DIV2K [81], Flickr2K [82] i OutdoorSceneTraining [16] skupovi podataka za uvježbavanje. Veličina uzorka uvježbavanih slika HR je postavljena na 256. Modeli su uvježbavani kombinacijom četiri NVIDIA V100 GPU-a s ukupnom veličinom serije (*batch*) od 48. Koristio se Adam optimizator. Real-ESRNet se fino podešava od Real-ESRGAN modela za postizanjem brže konvergencije. Real-ESRNet se uvježbava sa 1000K iteracija i sa stopom učenja od  $2 \times 10^{-4}$ , dokle Real-ESRGAN se uvježbava sa 400K iteracija i sa stopom učenja od  $1 \times 10^{-4}$ . Za stabilnije uvježbavanje i bolje performanse se iskoristilo eksponencijalno kretanje prosjeka (EMA, eng. *Exponential Moving Average*). Kako je već rečeno, uvježbavan je kombinacijom L1 gubitka, perceptivnog i GAN gubitka, s redom težinama  $\{1, 1, 0.1\}$ . Koristile su se  $\{\text{conv1}, \dots, \text{conv5}\}$  mape značajki (s težinama  $\{0.1, 0.1, 1, 1, 1\}$ ) prije aktivacije prethodno uvježbane VGG19 mreže kao perceptivni gubitak. Implementacija je općenito bazirana na BasicSR [83].

#### Detalji degradacije

Koristio se znači taj degradacijski model drugog poretka za postizanje dobre ravnoteže jednostavnosti i efikasnosti. Ako drugačije nije spomenuto, dva degradacijska procesa imaju iste postavke. Koristili su se Gaussian kerneli, generalizirani Gaussian kerneli i zaravnati kerneli, s redom vjerojatnostima  $\{0.7, 0.15, 0.15\}$ . Kernel zamučivanja se birao nasumično iz skupa  $\{7, 9, \dots, 21\}$ . Standardna devijacija zamučivanja je uzorkovana iz  $[0.2, 3]$  intervala, odnosno  $[0.2, 1.5]$  za drugi degradacijski proces. Parametri oblikovanja za Gaussian i zaravnane kernele su redom uzet iz intervala  $[0.5, 4]$  i  $[1, 2]$ . Spomenuti *sinc* kernel je radio s vjerojatnosti od 0.1, a za preskakanje zamučivanja druge degradacije vrijedila je vjerojatnost od 0.2.

U vezi šuma, Gaussian šum i Poisson šum su uzeti s vjerojatnosti  $\{0.5, 0.5\}$ . Sigma raspon šuma i skala Poisson šuma su redom postavljeni na  $[1, 30]$  i  $[0.05, 3]$  za prvi proces, a  $[1,$

25] i [0.05, 2.5] za drugi proces. Vjerojatnost sivog šuma je postavljena na 0.4, a faktor JPEG kompresije na [30, 95]. Finalni sinc filter je primijenjen s vjerojatnosti od 0.8.

### **Bazen za parove uvježbavanja**

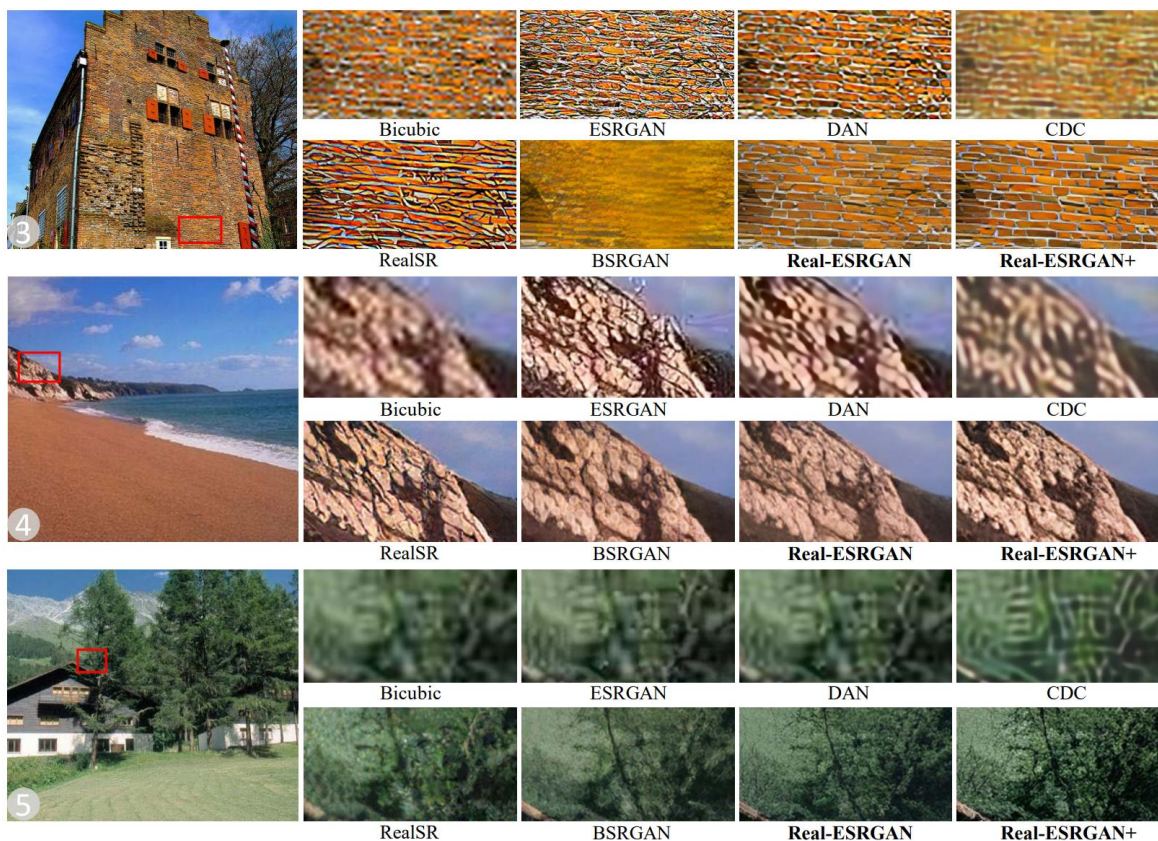
Svi procesi degradacije su implementirani u PyTorch-u s CUDA ubrzavanjem, tako da bi se mogli sintetizirati parovi za uvježbavanje u hodu. Nažalost, procesiranje serije ograničava raznolikost sintetičkih degradacija u seriji. Zbog toga je bio implementiran bazen parova za uvježbavanje da bi povećali raznolikost degradacije u seriji. Pri svakoj iteraciji uzorci koji bi se uvježbavali su uzeti nasumično iz tog bazena i tako bi se formirala serija za uvježbavanje. Veličina bazena je stavljena na 180 u njihovoj implementaciji.

### **Izoštavanje istinitih slika usred uvježbavanja**

Trik uvježbavanja da vizualno poboljša izoštrenost, a bez da donese nove vidljive artefakte na slici. Način na koji se uglavnom ovo radi je korištenjem nekakvog algoritma za naknadno procesiranje, kao što je to USM (eng. *Un-Sharp Making*). No, kako ovaj model zna biti popraćen problemima s artefaktima, oni su primijenili novi model uvježbavan s izoštrim istinitim slikama kojeg su nazvali Real-ESRGAN+. Usporedbe ovog modela s Real-ESRGAN i nekoliko drugih može se vidjeti na Slici 25.

### **Usporedbe s prethodnim djelima**

Model Real-ESRGAN je bio uspoređen s nekolicinom drugih state-of-the-art metoda, koje uključuju ESRGAN [41], DAN [84], CDC [86], RealSR [83] i BSRGAN [86]. Testirano je na nekoliko različitih skupova podataka koji sadrže slike iz stvarnog svijeta, kao što su to RealSR [87], DRealSR [85], OST300 [16], DPED [88], ADE20K [89] i nekim slikama s interneta. Kako nažalost većina postojećih metrika ne može dobro reflektirati perceptivnu kvalitetu u usporedbi s stvarnom ljudskom perceptivnom procjenom i samim doživljajem ljudskog oka da prepozna slike stvarnog svijeta, na Slici 25. priloženo je par reprezentativnih uzoraka u svrhu usporedbe ljudskom percepcijom. Ostatak primjera u prilogu od službene dokumentacije Real-ESRGAN modela. [67]



Slika 25. Kvalitativna usporedba nekoliko modela na reprezentativnim uzorcima stvarnih slika povećanih faktorom skaliranja x4 [67]

### Ablacija istraživanja i ograničenja

Model degradacije visokog poretka, odnosno drugog poretka, se je pokazao dobrim jer je s njim lakše kontrolirati degradacije i može općenito dati bolji uvid na utjecaj samih degradacija na slici.

Bez *sinc* filtera usred uvježbavanja ne bi se mogli izbjeći povećani artefakti na slikama, a pogotovo ne na onima oko teksta i linija, tako da su se ovi filteri pokazali dobrima u tu svrhu.

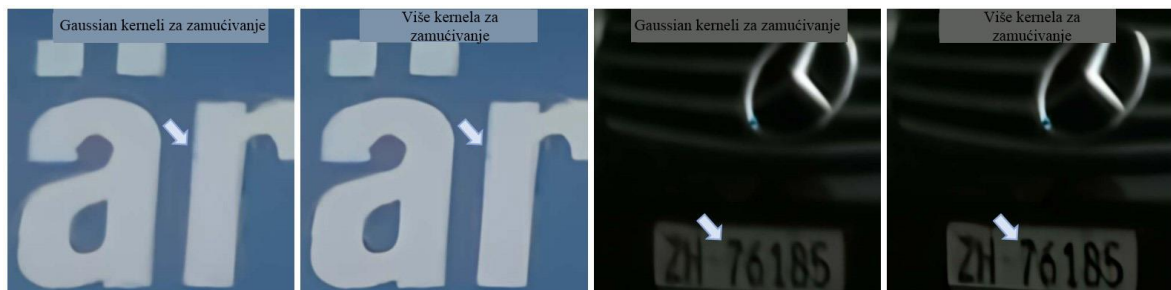
Dokle je U-Net diskriminator sa SN regulacijom bio potreban zbog prethodno navedenih razloga (diskriminacija i stabilnost kao glavni razlozi), i dokle on može poboljšati lokalne detalje, nažalost ima i slučajeva gdje stvara neprirodne teksture, a to možemo bolje vidjeti na Slici 26. na konkretnom primjeru.





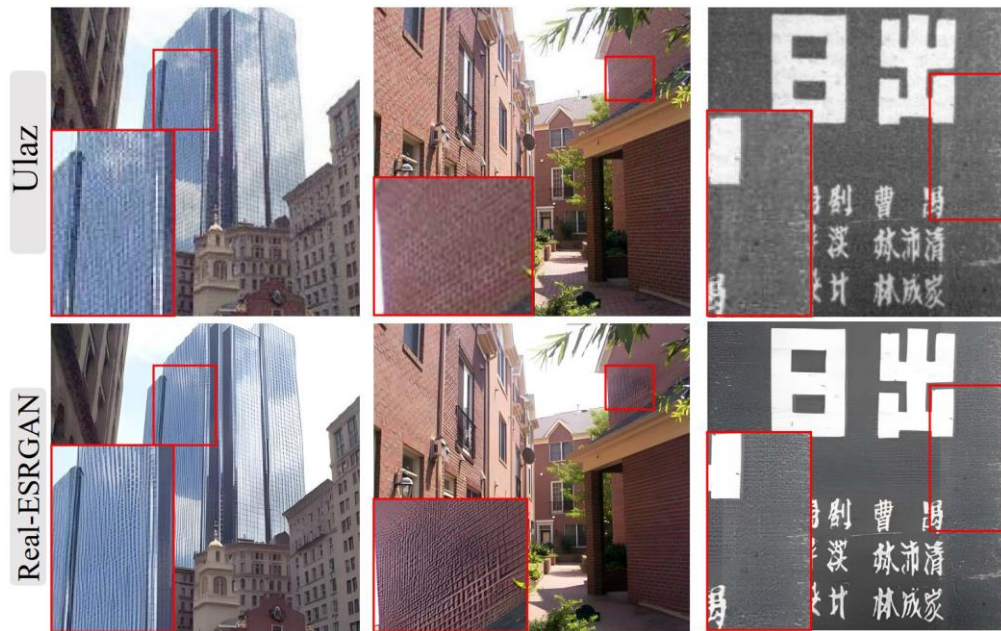
Slika 26. Ablacija na dizajn diskriminatora prema [67]

Više komplicirani kerneli za zamućenost koji su se koristili u ovom modelu su pokazali bolje rezultate nego dosad uobičajeno korišteni kerneli za zamućivanje, no razlika je u većini primjera neprimjetna, bolje vidljivo na Slici 27. Inače, zbog bolje performanse i činjenice kako ovi kompliciraniji kerneli mogu ipak bolje ukloniti zamućenosti, preporučaju se za daljinu upotrebu.



Slika 27. Ablacija na korištenje više kernela za zamućenost prema [67]

Iako se je Real-ESRGAN pokazao dobrim korakom naprijed u usporedbi s svojim prethodnikom, odnosno ESRGAN-om, još nažalost ne može u potpunosti dati prikaze koji bi odgovarali pravim stvarnim slikama. Na Slici 28. se mogu vidjeti primjeri učestalih problema ovog modela, što su uglavnom iskrivljene linije na teksturama zgrada i sličnome, neki čudni artefakti kao posljedica uvježbavanja GAN-a na nekim specifičnim uzorcima, i sama nemogućnost uklanjanja određenih kompliciranih degradacija izvan distribucije.



Slika 28. Primjerci ograničenja modela prema [67]

Ovi nedostaci imaju velikog utjecaja na praktičnu primjenu ovog modela na slikama stvarnog svijeta, odnosno u npr. stvarnim uvjetima rada, ali unatoč tome zbog drugih značajki, i pak poboljšanja u usporedbi s drugim modelima izdanima prije ovog modela, ne možemo negirati korisnost ovog modela za osnovne potrebe.

U sljedećem poglavlju će tako biti provjerena valjanost ovog modela i rezultata koji su istraživači prezentirali, i koji su dijelom bili prikazani u ovome poglavlju, te na samome kraju sve to kratko komentirano i da li je stvarno iskoristivo u osobnim potrebama.

## 5. Eksperiment

Dosad su u ovome radu bile analizirane metode i tehnike koje se koriste za obavljanje procesa povećanja slike, s naglaskom na onima koji se tiču dubokog učenja, odnosno super-rezolucije, koje ljudi danas često zovu i AI metodama budući ih karakterizira to što koriste umjetnu inteligenciju za izvršavanje tog procesa. Napravljen je i pregled metrika i drugih karakteristika koji se tiču tih metoda, sve u svrhu boljeg razumijevanja svega onog potrebnog za provođenje ovog eksperimenta, budući je glavna ideja ovog rada i bila istražiti AI metode i provjeriti njihovu valjanost i iskoristivost s trenutnim tehnologijama. Uglavnom, kako je već bilo najavljeno, u ovom poglavlju će biti napravljen eksperiment, u kojem će se na nizu različitih vrsta slika (skupova podataka), korištenjem jedne i/ili više metrika predstavljenih u radu, provjeriti kvaliteta procesa povećanja slike izvršena s nekoliko različitih metoda, odnosno modela. Da, to će i uključivati Real-ESRGAN model koji je bio malo detaljnije predstavljen u prethodnome poglavlju.

### 5.1. Izbor alata

Danas na internetu možete pronaći uistinu ogroman broj stranica koje promoviraju povećanje slike korištenjem AI, i dokle ih je nekolicina besplatna, veći dio ih je oblika *freemium* ili vrsta usluga koja se plaća. Istina da korisniku koji ne radi povećanje slike učestalo ovo može i biti prihvatljivo, no sama činjenica da morate kreirati dodatne račune na tim stranicama, često s još dodatnim troškovima, onima koji to rade svakodnevno i htjeli bi izbjeći raditi to online, to nije prihvatljiva opcija. Tako da u ovome radu, uzimajući takve razloge i slično u obzir, fokus će biti na izboru alata za rad lokalno, budući se ide s pretpostavkom da se ima zadovoljena potrebna hardverska podrška od samog početka. Naravno, u slučaju da se nema zadovoljena hardverska podrška, postoji mogućnost implementacije većine ovih metoda i putem bilježnica na *Google Colabs*, no s ograničenim opcijama.

Primjeri nekoliko tih alata za rad lokalno, većinom GUI-ova, su:

- *chaiNNer* – GUI za procesiranje slika koji je temeljen na čvorovima, s ciljem da učini spajanje nizova tih procesa u lanac jednostavnim i lakim za podesiti [90]
- *enhancer* – elegantan i lagan za upotrebu, GUI kojemu je glavna svrha da poveća video podatke [91]

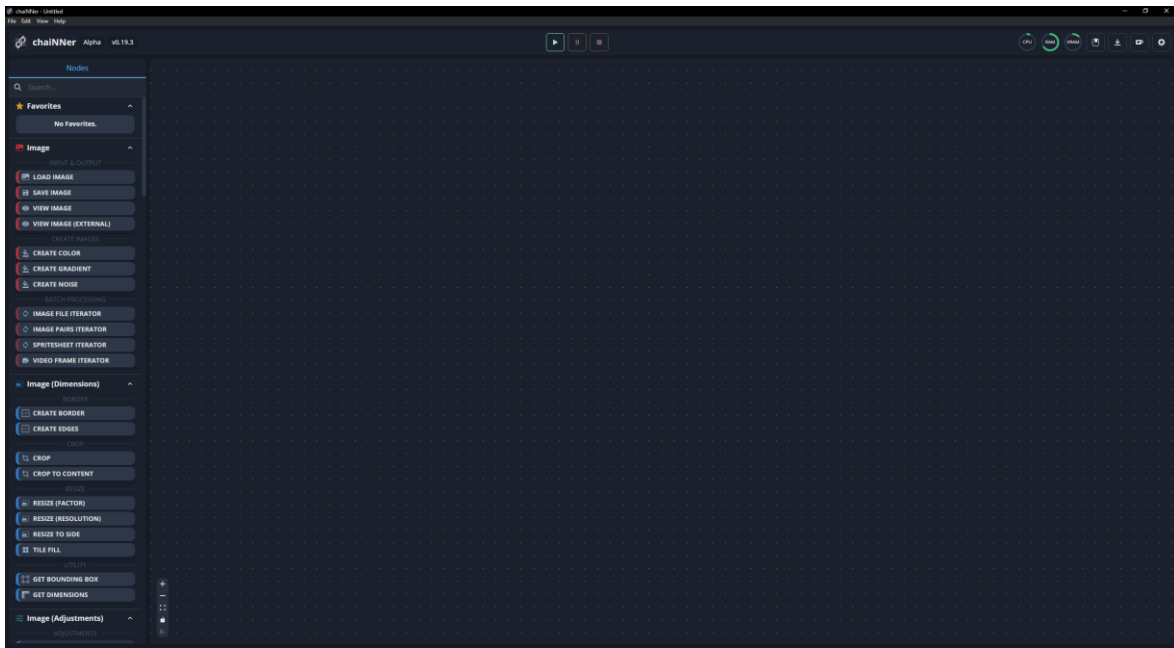
- Upscayl – aplikacija za povećanje veličine slika LR u slike HR korištenjem naprednih AI algoritama [92]
- ComfyUI – moćni i modularni GUI stabilne difuzije koji je i *backend* [94]
- Waifu2x-Extension-GUI – GUI za povećanje veličina slike, videa i gif-ova [95]
- Squirrel-RIFE – GUI za povećanje slike temeljen na RIFE algoritmu [96]

Svi navedeni alati koriste barem jedan oblik algoritama dubokog učenja i neuronskih mreža za izvršavanje svojih procesa, odnosno određeni omogućuju biranje vlastitog modela i slično, dokle su drugi ipak ograničeni s već postavljenim model ili biranjem jednog od par ponuđenih bez dodatnih modifikacija.

U sklopu ovog eksperimenta odabran je chaiNNer, zbog toga što je noviji (projekt od niti godinu dana starosti) i *open-source* GUI, jako jednostavan za upotrebu s velikom razinom modificiranja, uistinu fleksibilan s obzirom na opcije koje sve nudi i poprilično moćan. Još jedan veliki plus koji ima pored većine drugih alata je da ga možete pokrenuti na više platformi, odnosno na Windows, MacOS i Linux, dokle većina drugih alata je uglavnom ograničena samo na Windows i eventualno Linux. Nadalje, sama instalacija i postavljanje je stvarno jednostavno i zahtjeva gotovo minimalno poznavanje aplikacija (softvera) na računalu, te je rijetki primjer gdje potrebni *dependencies* za ostvarivanje funkcionalnosti ovog alata se svi mogu skinuti i instalirati direktno unutar postavki alata, tako da nema potrebe za pustim zipovima i paketima skidanja odvojeno s pustih stranica online ili igranja po terminalu. Isto jedna napomena, a to je da je projekt službeno u alfa verziji i označen s WIP (eng. *Work in progress*), tako da redovno ima nadogradnje s dodavanjem funkcionalnosti i popravljanjem ponekih *bugova*. Unatoč ovome, poprilično je robustan, i osobno u korištenju ovog alata već neko vrijeme i kroz nekoliko verzija, nisam uspio primijetiti nikakve probleme vrijedne spomena. Sada da li to ovisi o mojem računalu, konkretno hardveru i operacijskom sustavu, i eventualno o procesima i modelima koje sam ja koristio, odnosno da li sam jednostavno imao sreće da se ne susretnem s njima naspram nekih drugih, naravno da to uvijek vrijedi kao argument. Uglavnom, sve specifikacije računala i ostatka softvera i alata korištenih za vrijeme ovog eksperimenta će biti navedeni u nastavku, a prije detalja same implementacije i svega ostalog, na Slici 29. možemo vidjeti kako izgleda opće okruženje samog chaiNNer-a, gdje vidimo glavnu radnu površinu koja se lako može proširivati i smanjivati ovisno o veličini koja nam je potrebna, čak i preko *scrollanja*, no jednako tako i preko za to posvećenih botuna u donjem lijevom kutu te ravne površine. Slijeve strane vidimo čvorove (eng. *nodes*) koje dodajemo potezanjem (princip



*drag-and-drop*) na radnu površinu za kasnije spajanjem u jedan lanac koji predstavlja proces koji želimo izvesti, a konkretni primjer ovoga će biti prikazan u samoj implementaciji. Inače jednostavniji način dodavanja čvorova je zapravo čisto desnim klikom miša bilo gdje na radnoj površinu gdje se otvara minimalizirana lista s opcijom pretraživanja i automatskim obnavljanjem te liste, što čini filtriranjem pojedinih čvorova koji nam trebaju jako brzim.



Slika 29. Početno stanje alata chaiNer

Cijeli proces je uistinu intuitivan i podosta brz, s time da se mogu i spremati cijeli lanci kao *template* za brzo pokretanje kada vam zatreba taj lanac. Bitno je i spomenuti kako ovaj alat omogućuje iterativno povećanje slika, odnosno možete označiti i cijelu mapu slika na kojima želite izvršiti te lance (niz procesa obrade slike), te na vrlo lagan način kontrolirati cijeli tijek, prikaze i slično.

U gornjem desnom kutu imamo niz osnovnih parametara, gdje alat prikazuje opterećenost CPU-a, RAM-a i GPU što se tiče VRAM-a. Odmah do toga se nalazi i dokumentaciju čvorova, poprilično korisno i više nego dovoljno detaljno je objašnjen svaki čvor s njegovim svojstvima i funkcionalnosti. Dalje se nalazi botun za upravljanjem *dependencies* i osnovne postavke okruženja, odnosno gdje se može i odabrati CPU ili GPU za pokretanje lanca, pored ostalih osnovnih i nužnih postavki.

Tako da zbog jednostavnosti sučelja i modernog osjećaja koji ovaj GUI ima, a ponajviše zbog dobre funkcionalnosti, u sklopu ovog eksperimenta gdje se je htjela provjeriti valjanost

modela na većem broju slika (skupu podataka), ovaj alat se je pokazao najboljom opcijom za lokalnu obradu.

## 5.2. Dokumentacija

U ovome eksperimentu, primarni cilj je bio provjeriti valjanost i praktičnu primjenu modela za povećanje slike dostupnih javnosti, a sekundarni je bio provjeriti da li postoji kakav alat koji omogućuje efikasno povećanje slika lokalno, bilo korištenjem već ugrađenih modela, ili poželjnije s mogućnošću dodavanja modela po izboru. Sekundarni cilj se je dijelom već ispunio istraživanjem recenzija i komentara od strane zajednice na internetu koja se bavi općenito povećanjem slike i njihovom preporukom, no naravno ne i ispravnost toga, odnosno tek se je trebala provjeriti kvaliteta toga alata konkretnim izvođenjem niza zadataka lokalno da bi se uvjerali koliko je konkretni alat dobar za osobnu upotrebu. Naravno, pravilno bi bilo to usporediti na većem broju alata da se pokaže koji je uistinu bolji, no kako je to sekundarni cilj a rezultati testiranja su bili i više nego zadovoljavajući, ostalo se je pri tom jednom izboru alata, odnosno chaiNNer-u kako je prethodno bilo i navedeno.

Za provjeru primarnog cilja odlučio sam se provjeriti modele kojima je svrha bila općenito povećanje slike, odnosno onaj dio modela koji su spadali u kategoriju generalnog povećanja slike (sjetimo se pojma slijepe super-rezolucije), znači bez specificirane vrste slike za koju su oni bili namijenjeni već više da se mogu primijeniti na bilo kojim slikama. Razlog sužavanja izbora na ove modele je bio sljedeći, smatralo se da nije u redu usporediti model koji je specijalizirao određeni tip slike, jer bi bilo logično da onda on bude bolji u tome naspram nekog drugog koji nije možda uopće učio na tom tipu slika. Istina da je praksa znala pokazati nepovezanost ovih tvrdnji, gdje su se znali dobivati iznenađujući rezultati, u smislu da je model koji uopće nije bio učen za to raditi, zapravo to nešto napravio bolje nego model koji je to specijalizirao, barem u povećanju slika. Ipak, ovo je rijetkost.

Nadalje, odlučio sam usporediti modele s dvije vrste faktora skaliranja slike, a to su x2 i x4 faktori skaliranja. U pravilu je dovoljno bilo provjeriti samo x2 povećanje slike, ali kada su slike malih rezolucija u pitanju, npr. rezolucije 600 x 300, povećanje dva puta često ne zadovoljava cilj, tako da bi onda znali prvo napraviti povećanje dva puta, pa onda taj rezultat ponovo povećati dva puta. Tako da sam se uzevši to u obzir odlučio odmah i provjeriti modele za x4 povećanje slike. Inače, još jedan uvjet je bio da modeli budu različitih

arhitektura i veličina, s dodatnom opcijom da se uzme po dva modela iste arhitekture ali drukčijih skupova podataka, da se može provjeriti i to.

Uzevši navedeno u obzir, navedeni su modeli (s datumom objave) koji su izabrani za provjeru x2 skaliranja su:

- RealESRGAN\_x2Plus – 22.7.2021
- BSRGANx2 – 24.8.2021
- SwinIR-M-x2 (classicalSR-DF2K-s64w8) – 26.8.2021
- SwinIR-M-x2 (classicalSR-DIV2K-s64w8) – 26.8.2021
- OmniSR 2x DF2K – 19.4.2023
- OmniSR 2x DIV2K – 19.4.2023

Da se primijetiti odmah kako su neki navedeni modeli gotovo isti modeli, odnosno jedina im je razlika da su uvježbavani na različitim skupovima podataka. Smatrao sam vrijednim provjere i toga usput, da li će biti razlike u rezultatima i slično, te na kraju primijenio isto i za modele za provjeru od x4 skaliranja. Ti modeli za x4 faktor povećanja su:

- RealESRGAN\_x4Plus
- BSRGAN
- SwinIR-M-x4 (classicalSR-DF2K-s64w8)
- SwinIR-M-x4 (classicalSR-DIV2K-s64w8)
- OmniSR 4x DF2K
- OmniSR 4x DIV2K

Modeli su izabrani prethodno navedenim ograničenjima, poredanih po važnosti na bazi modela OpenModelDB [97], s jedinom iznimkom za x4 skaliranje gdje je namjerno odabran RealESRGAN, iako se nije nalazio pri vrhu poretka, zbog toga što je dodatno obrađen u radu. Uglavnom, s navedenim ograničenjima, i da bude model za generalno povećanje slike s uvjetom, i da ima i x2 i x4 skalu, na kraju imamo dva modela s arhitekturom ESRGAN, dva modela s SwinIR i dva modela s OmniSR. Ova dva modela temeljena na ESRGAN-u i dva od SwinIR su sličnih veličina, dokle OmniSR je znatno manje veličine i spada u kategoriju *lightweight* modela. Bitno je isto napomenuti kako je OmniSR znatno noviji, jer je ovo model koji je izašao na proljeće 2023.godine, dokle su ovi ostali iz 2021.godine.

U vezi skupova podataka, zbog specifikacija računala (detaljnije odmah ispod) sam se ograničio na osnovnije i manje skupove podataka za provjeru, nabavljenih s [68], a to su:

- Set14
- BSD100
- Urban100

Ideja je bila uzeti skupove podataka koji imaju sparirane slike, znači slike LR i slike HR, i to u primjercima s faktorima skaliranja  $\times 2$  i  $\times 4$ , odnosno SRF(eng. *Scale Reduction Factor*) 2 i SRF 4. Nadalje, da slike budu šarolike, odnosno da se ima više različitih objekata i više različitih scena. Prva dva skupa podataka su zadovoljili ove kriterije u potpunosti, dokle je Urban100 jedino diskutabilan u pogledu detalja na slikama, jer je bio skup podataka s uglavnom scenama na kojima su se nalazile zgrade i ostale urbane sredine, no prihvaćen je bio zbog velikog broja tih sitnih linija i detalja na slikama, koje su dobro poslužile u svrhu perceptivne procjene kvalitete povećanih slika.

Metrika koja će se koristiti za evaluaciju je PSNR i SSIM metrika, te perceptivna procjena kvaliteta slika, točnije evaluacija ljudskim okom na sličnosti i različitosti slika.

Prije same implementacije, još će biti navedene bitne specifikacije hardvera i softvera korištenih u sklopu ovog eksperimenta.

Specifikacije računala na kojemu se provodi eksperiment su:

- CPU: AMD Ryzen 5 3600
- RAM: 16GB DDR4-2400
- GPU: NVIDIA GeForce RTX 3060 12GB VRAM
- OS: Win10 64-bit
- SSD: 1TB M.2
- HDD: 2TB 7200RPM

Verzije *drivera*, korištenih alata i svega ostalog, koje su bitne za spomenuti su sljedeće:

- CUDA 12.2
- GeForce Game Ready Driver 537.13
- chaiNNer Alpha v0.19.3
- Python 3.10.7
- PyTorch 2.0.1
- OpenCV 4.8.0.76

- Numpy 1.23.4
- Scikit-image 0.21.0

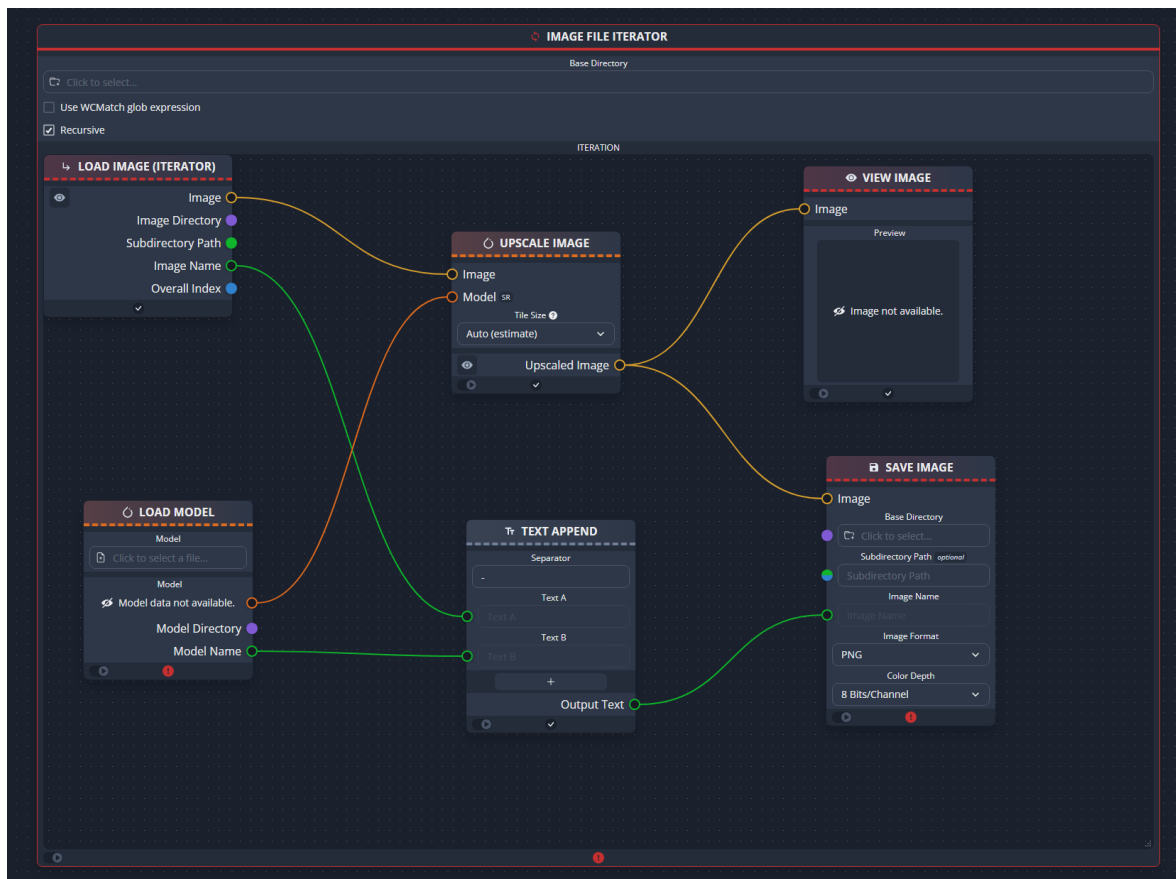
### 5.3. Implementacija

Kako smo već rekli, skupovi podataka su sa sparenim slikama, i ono što prvo radimo je da razdvojimo slike LR, koje želimo povećati, i slike HR koje računamo kao referentne slike s kojima ćemo usporediti povećane slike i na njima provesti navedenu metriku.

Prvo unutar `chaiNNer`-a kreiramo lanac koji će izvesti ovaj proces, odnosno procese, jer želimo više slika povećati zaredom. Tako da zapravo želimo napraviti lanac iteracije povećanja svih slika LR skupa podataka sa odabranim modelom željenog faktora skaliranja. U nastavku će kratko biti prikazana izrada tog lanca, odnosno *template*, koji se koristio za povećanje odabranih skupova podataka. Iđentičan lanac će se prvo koristiti na x2 modelima, a zatim na x4 modelima. Kako je rečeno, radimo iteraciju povećanja slika, a to radimo tako da odaberemo čvor *Image File Iterator* koji je centralna figura ovog lanca, odnosno okvir unutar čega ćemo dodati ostale čvorove. Zatim nam još trebaju sljedeći čvorovi:

- *Upscale Image* – čvor koji za ulaze ima sliku LR i model koji će se primijeniti na tu sliku, a izlaz je povećana slika
- *Save Image* – čvor koji prima izlaznu sliku i nazive datoteke, format slike i mjesto spremanja slike
- *Load Model* – čvor s kojim učitavamo model kojeg ćemo iskoristiti za proces povećanja slike
- *Text Append* – u pravilu opcionalni čvor, no kada radite s iteracijom zapravo je nužan, učini stvar puno praktičnijom jer preko njega odredite nazive datoteka, i generalna praksa je spojiti originalni naziv slike i modela kojim je ona povećana
- *View Image* – opcionalni čvor, cisto za prikaz slike unutar samog okruženja, ima i drugi oblik čvora za prikaz eksterno, odnosno preko nekog drugog alata van `chaiNNer`-a

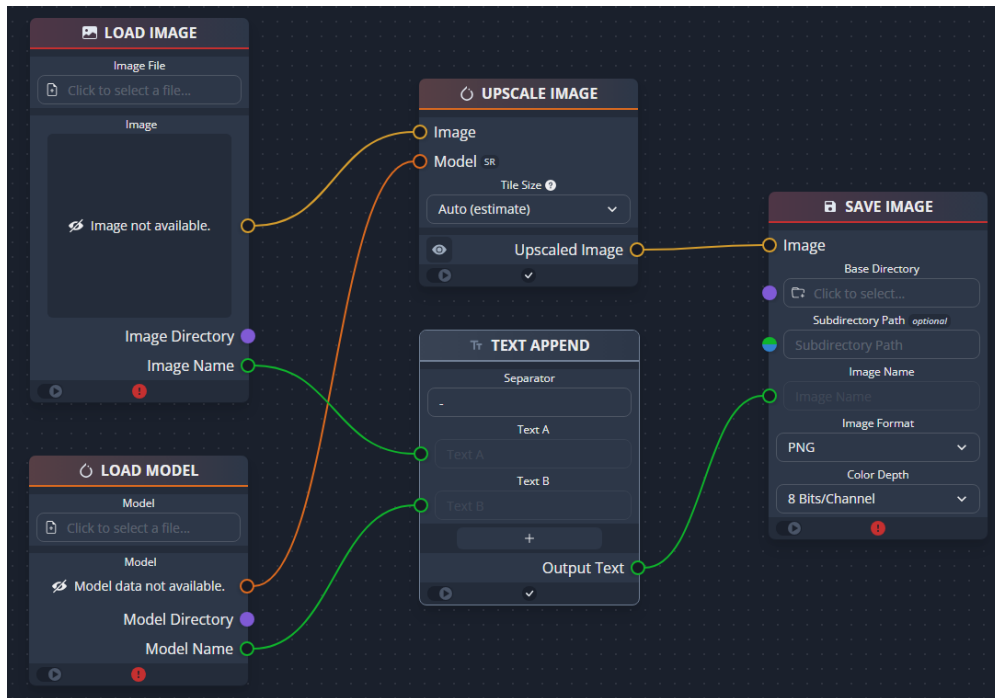
Na Slici 30. možemo vidjeti prikaz tog cijelog kalupa (*template*) koji je bio korišten za iterativno povećanje slika skupa podataka. Kako je spomenuto, isti će se koristiti na svim modelima i u svrhu svih faktora skaliranja.



Slika 30. *Template* za izvođenje iterativnog povećanja slike

Generalno, lanac za povećanje veličine jedne slike je jako sličan onome na Slici 30. s zapravo jedinom razlikom da se nema onog velikog vanjskog okvira za iteraciju, odnosno učitavanje slike je drukčije (samo jednom), i svi ti manji čvorovi su sada sami na radnoj površini. Konkretni primjer na Slici 31. za takav lanac.

Inače, postoji još jako mnogo opcija u chaiNner-u, gdje se mogu konkretno birati čak i pojedinačno procesi kao razni filteri za zamućivanje, izoštravanje i šum, razvlačiti slike, raditi konvolucije, mijenjati osnovne postavke kao kontrast, osvijetljenost i još mnogo toga. Ipak, jedna od opciju koju osobno smatram jako praktičnom je da možete automatski spojiti po dvije ili više slika, bilo vertikalno ili horizontalno, radi lakše usporedbe vizualno.



Slika 31. Lanac za povećanje jedne slike

## 5.4. Evaluacija

Sada kada je povećanje skupova podataka napravljeno, potrebno je evaluirati modele. U svrhu toga sam izradio jednu malenu skriptu koja se lako modificira ovisno o skupu podataka i modelu koji se koristi, tako da evaluacija modela s faktorom skaliranja x2 i x4 će biti napravljena istom skriptom, a cijela skripta će biti u prilogu rada.

Koristiti će se metrike PSNR i SSIM, izračunata kao prosječna vrijednost ukupnih vrijednosti na cijelome skupu podataka, i programski jezik Python zbog moćnih biblioteka koje on ima na raspolaganju. Potrebne biblioteke su:

- NumPy – za proračune i operacije s matricama
- OpenCV – trebati će nam cv2 za rad s slikama, konkretno učitavanje i pretvorba u *grayscale* za daljnji proračun
- Scikit-image – potrebna zbog SSIM metrike, ima ugrađenu funkciju za proračun te metrike

Prvo znači treba učitati te biblioteke u skriptu da se mogu koristiti njihove ugrađene funkcije, a to se radi uobičajenim načinom:

```
import numpy as np
import cv2
from skimage.metrics import structural_similarity as ssim
```

Srž skripte je sljedeća funkcija za evaluaciju:

```
def evaluate_dataset(dataset_hr, dataset_generated):
    num_images = len(dataset_hr)
    total_psnr = 0.0
    total_ssim = 0.0

    for i in range(num_images):
        hr_image = cv2.imread(dataset_hr[i])
        generated_image = cv2.imread(dataset_generated[i])

        hr_gray = cv2.cvtColor(hr_image, cv2.COLOR_BGR2GRAY)
        generated_gray = cv2.cvtColor(generated_image,
cv2.COLOR_BGR2GRAY)

        psnr = calculate_psnr(hr_image, generated_image)
        total_psnr += psnr

        ssim_value = calculate_ssim(hr_gray, generated_gray)
        total_ssim += ssim_value

    average_psnr = total_psnr / num_images
    average_ssim = total_ssim / num_images
    return average_psnr, average_ssim
```

Kod 1. – Funkcija za evaluaciju skupa podataka s PSNR i SSIM metrikom

Gdje se je funkcija `calculate_psnr` trebala malo detaljnije razraditi budući nije bila korištena ugrađena funkcija, a ona se računala ovom funkcijom:

```
def calculate_psnr(img1, img2):
    mse = np.mean((img1 - img2) ** 2)
    if mse == 0:
        return float('inf')
    max_pixel_value = 255.0 # 0-255
```



```
psnr = 20 * np.log10(max_pixel_value / np.sqrt(mse))
return psnr
```

## Kod 2. – Funkcija za proračun PSNR vrijednosti

Kako je SSIM metrika računata gotovom ugrađenom funkcijom, samo ju je trebalo pozvati unutar skripte, a za samu izvedbu može se detaljno pogledati unutar `scikit-image` dokumentacije, ili prisjetiti kako se je računala ta metrika prikazana u poglavlju za to u ovome radu. Uglavnom, pozvali smo je sljedećom funkcijom:

```
def calculate_ssim(img1, img2):
    return ssim(img1, img2, multichannel=True)
```

Još kako je bila riječ o skupu podataka, odnosno većem broju slika, da ne bi svaku sliku i njenu putanju ručno unosili, iskoristio sam sljedeći kod za jednostavnije generiranje liste koja sadrži putanje svih slika, bilo je jednostavno budući su nazivi slika isti s inkrementalnim indeksom, a to je ovaj dio skripte:

```
def generate_image_pathHR(base_path, num_images,
extension="_SRF_4_HR.png"):
    image_paths = [f"{base_path}{i:03d}{extension}" for i in
range(1, num_images + 1)]
    return image_paths
```

Slična funkcija je bila izrađena i za generiranje putanja za slike LR, gdje je samo još dodatno trebalo pripaziti na ekstenzije, jer se naziv slike promijenio ovisno o modelu koji je bio korišten za povećanje. Pogledati skriptu u prilogu za konkretnu izvedbu.

Na kraju se samo ispisivala vrijednost prosjeka PSNR i SSIM za pojedini model, npr. za Real-ESRGAN:

```
avg_psnr, avg_ssim = evaluate_dataset(dataset_hr,
dataset_generated1)
print(f"Average PSNR for RealESRGAN: {avg_psnr:.2f} dB")
print(f"Average SSIM for RealESRGAN: {avg_ssim:.2f}")
```

U prilogu će biti dana i pojednostavljena verzija skripte, koja provjerava samo metriku za dvije slike, jednostavno se stavi skripta u isti direktorij s tim dvama slikama za usporedbu i samo se promijene njihovi nazivi u za to predviđene varijable. Razlog dodavanja ove skripte je bio da se lakše može provjeriti razlika nekih konkretnih dvaju slika, budući je bilo rezultata PSNR i SSIM metrika gdje su modeli imali ukupni prosjek isti, no pojedine slike su naočigled ipak bile drukčije, što je sjetimo se ona mana ovih metrika koju smo već spomenuli u za to predviđenom poglavlju.

## 5.5. Rezultati i rasprava

U ovome dijelu će biti grupirani i predstavljeni rezultati odabranih modela i njihovih evaluacija na odabranim skupovima podataka, evaluiranih spomenutim metrikama. Naknadno sam odlučio dodati još jednu metriku, nije dosad bila spomenuta jer nije metrika koja se direktno tiče IQA-a, niti je nešto što se računalo u samoj skripti za usporedbu slika. Ipak, kako je osobno smatram bitnom, odlučio sam je dodati i kasnije prokomentirati s ostatkom rezultata metrika. Riječ je o vremenu izvršavanja procesa povećanja slika, gdje je ipak bilo zamjetne razlike od modela do modela. Vrijeme je bilo mjereno od strane alata chaiNNer-a s obzirom na to koliko mu je dugo trajao cijeli lanac procesa, uključujući i učitavanje i spremanje slike. Istina da je učitavanje i spremanje bilo reda veličine 0.01 sekundi, kako je riječ ipak o 100 slika po skupu podataka, dalo bi se to uzeti u obzir za oduzimanje od ukupnog mjerenog vremena, ali kako je ovo bilo jednako za sve modele, zbog jednostavnosti prikaza će biti nepromijenjeno.

Na Tablici 1. imamo prikazano rezultate metrika za modele s faktorom skaliranja x2.

	Set14			BSD100			Urban100		
	PSNR	SSIM	t	PSNR	SSIM	t	PSNR	SSIM	t
<b>RealESRGAN_x2plus</b>	31.50	0.81	11.6	31.60	0.78	27.5	30.67	0.79	31.3
<b>BSRGANx2</b>	31.75	0.85	11.0	32.01	0.82	56.1	31.04	0.85	67.0
<b>SwinIR-M-x2-DF2K</b>	34.83	0.93	19.6	34.35	0.91	89.0	34.65	0.95	109.0
<b>SwinIR-M-x2-DIV2K</b>	34.71	0.92	18.1	34.30	0.91	89.0	34.54	0.95	109.0
<b>OmniSR 2x DF2K</b>	34.54	0.92	3.9	34.21	0.90	20.7	34.06	0.94	23.8
<b>OmniSR 2x DIV2K</b>	34.49	0.92	3.9	34.18	0.90	21.0	34.04	0.94	23.9

Tablica 1. Rezultati metrika x2 modela

PSNR je izražena u mjernoj jedinici od dB (decibela), i bolje je što veći broj jer to indicira bolju kvalitetu slike. SSIM metrika nema mjerne jedinice, i ona predstavlja indeks vrijednosti koji se kreće od -1 do 1, s time da 1 znači potpunu identičnost, odnosno da su slike iste, a 0 da su skroz drukčije, a negativna vrijednost znači da je slika previše izobličena i prepuna artefaktima. Ukratko, za SSIM bolje je što bliže vrijednosti 1. Vrijeme

$t$  je izraženo u sekundama i označava ukupno vrijeme potrebno za povećanje svih slika tog skupa podataka, bolje je što manji iznos jer to znači da je model brži.

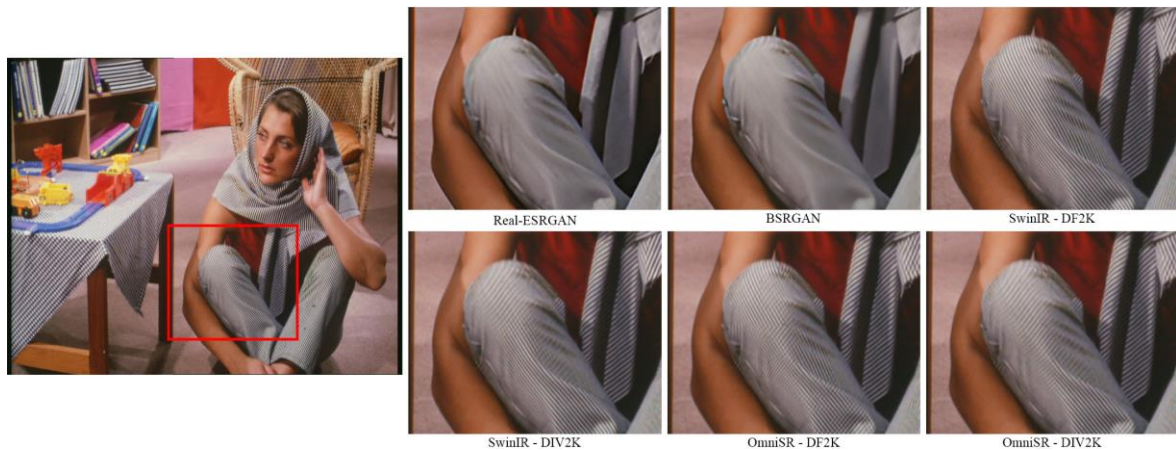
Odmah će se priložiti i Tablica 2. za rezultate metrika x4 modela.

	Set14			BSD100			Urban100		
	PSNR	SSIM	t	PSNR	SSIM	t	PSNR	SSIM	t
<b>RealESRGAN_x4plus</b>	30.50	0.68	5.0	30.71	0.64	27.6	30.39	0.67	80.0
<b>BSRGAN</b>	30.59	0.68	4.8	30.84	0.64	26.6	30.58	0.68	78.0
<b>SwinIR-M-x4-DF2K</b>	32.47	0.80	6.7	32.23	0.75	36.0	32.41	0.83	121.0
<b>SwinIR-M-x4-DIV2K</b>	32.39	0.79	6.5	32.18	0.75	36.1	32.28	0.82	120.0
<b>OmniSR 4x DF2K</b>	32.23	0.78	2.6	32.12	0.74	16.2	32.10	0.80	32.0
<b>OmniSR 4x DIV2K</b>	32.18	0.78	2.6	32.09	0.74	15.5	32.02	0.79	32.0

Tablica 2. Rezultati metrika x4 modela

Rezultati su u jednu ruku bili iznenađujući, a pogotovo je impresionirao OmniSR model, obje njegove verzije su bile i jako brze i davale jako dobre rezultate. S druge strane imamo RealESRGAN+ koji je na danim skupovima podataka davao najgore rezultate što se tiče kvalitete, te je na svega manji broj slika imao prihvatljivo ili nijansu prirodnije rezultate nego njegov konkurent BSRGAN, iste arhitekture, a općenito je znatno zaostajao za modelima arhitekture SwinIR i OmniSR. Istina da je ovo bio model verzije RealESRGAN+, i ne smijemo zaboraviti razliku njega i RealESRGAN-a te skupova podataka na kojima su uvježbavani, te s obzirom na skupove podataka na kojima je proveden ovaj eksperiment, mislim da bi RealESRGAN postigao možda malo bolje rezultate nego plus verzija. No ipak sam poprilično siguran da bi svejedno zamjetno kaskao za ovim drugim boljim modelima, jer sam primijetio veliki broj problema koje on ima na slikama s detaljima i teksturama. Najučestaliji tip problema koji sam primijetio je prikazan na Slici 32., gdje bi ESRGAN modeli najčešće gubili sitne detalje kao što su to neki uzorci, tipa ovaj tekstura (uzorak) hlača. Donekle razumljivo zbog prirode i svrhe tih modela, a to je da im je bio fokus na izoštravanju i uklanjanju šuma, no nuspojava je da tako u potpunosti izbrišu bitne detalje na slikama. Naravno, slični problemi su se dali zamijetiti i s drugim modelima na pojedinim slikama, malo drugačije prirode, no može se vidjeti na toj Slici 32. kako je i SwinIR model, mada samo DIV2K verzija, imao problema s time. Najveći,

odnosno najproblematičniji problem kod ESRGAN baziranih modela je bez sumnje bio u izobličenjima i deformacijama što se tiče obilježja živih bića, konkretno ljudskog lica i tipa detalja očiju, znao bi ih u potpunosti uništiti, a pogotovo na x4 modelu. Dodatni primjeri kvalitativne usporedbe slika se nalaze u privitku rada.



Slika 32. Usporedba na slici 2 iz Set14 SRF2

Sama metrika nalaže da je SwinIR model uvježbavan na DF2K skupu podataka dao općenito najbolji rezultat, i ovo je većinski istina, uistinu je na najvećem primjeru slika imao najbolje rezultate od konkurentnih modela s kojima se izveo ovaj eksperiment. Inače, ako bi veći naglasak dali na vrijeme potrebno za obavljanje povećanja slike i razliku u računalnoj moći, odnosno resursima potrebnim, OmniSR model uvježbavan na DF2K skupu podataka bi vjerojatno bio pobjednik, jer pokazao je odlične rezultate s obzirom na svoju veličinu. Sjetimo se kako on spada u kategoriju *lightweight* modela, što ga onda čini još impresivnijim, jer je prosječno bio barem 3 puta brži od SwinIR modela.

Rezultati u slučajevima povećanja slike s faktorom skaliranja x4 su za sve modele bili zamjetno gori nego oni od povećanja slika s faktorom skaliranja x2, a za neke na određenim slikama poprilično zastrašujući, gdje su se stvarale takve deformacije da, ugrubo rečeno, skoro liče na scene iz horor filma. U većini tih primjera slike dobivene ESRGAN modelima su više sličile na slike koje bi spadale direktno pod kategoriju AI umjetnosti, gubile su velik dio bitnih informacija na slikama, i istina da su gladile rubove i izoštravale sliku, ali su bitno narušavali prirodnost te slike.

Gledajući na rezultate od x2 i x4 faktora skaliranja, unatoč znatno bržem i većem povećanju slike s modelima x4, rezultati su toliko lošiji da se iskreno ne isplati to raditi, već bi bilo bolje ostati na povećanju slike s faktorom skaliranja x2 i tu operaciju izvesti dva puta zaredom, i dobili bi bolje rezultate nego korištenjem tih modela za povećanje od x4 puta.

# Zaključak

U ovome radu su tako bile analizirane metode koje se vežu za područje super-rezolucije, odnosno metode za izvođenje procesa povećanja slike. Krenulo se je od tradicionalnih metoda, vezanih za pojam interpolacije koji koriste matematičke algoritme da bi povećali sliku, pa do onih metoda koje koriste tehnike umjetne inteligencije za ostvarivanje tog procesa, koje su ujedno bile i glavni fokus ovoga rada. Isproban je niz modela, različitih arhitektura, za izvršavanje procesa povećanje slike, s relativno malenom razlikom u starosti tih modela. Najnoviji model OmniSR je pokazao veoma impresivne rezultate, pogotovo uzevši njegovu veličinu i vrijeme u obzir, te s obzirom na konkurenta SwinIR-DF2K koji je dao generalno najbolje rezultate (s obzirom na kvalitetu slike) u provedenom eksperimentu, kaskao je za gotovo ne zamjetnu razliku, dokazujući da je sama super-rezolucija u neprestanom razvoju i postaje sve pristupačnija. Naravno, kako je on i gotovo dvije godine mlađi nego konkurencija, da se argumentirati da nije bila korektna usporedba. Osobno smatram da je to zapravo bio izvrstan primjer za dokazati da se može postići bolje i optimizirati model, jer i da je davao iste rezultate kvalitete, samo zbog znatno manje veličine i veće brzine, već je bolja opcija kao metoda povećanja slike za većinu korisnika kojima nije potrebno povećanje slike u svrhu profesionalizma.

Nažalost, ni najnoviji model ni model s najboljim rezultatima u ovome eksperimentu, nisu bili savršeni na svim vrstama slika, i još uvijek imaju nekolicinu tipičnih problema super-rezolucije. Tako da još uvijek nismo na onoj razini gdje uistinu možemo reći da smo postigli željeno povećanje stvarne slike, ali napredak je očigledan tako da nade ima.

Sama činjenica da je dokazana primjena alata i modela gdje se za većinu vrsta slika postižu zapravo jako dobri rezultati, gdje izgleda i nisu potrebna super-računala za izvršavanje povećanja slika za lokalnu (osobnu) upotrebu, daje ohrabrujuće poglede na budućnost gdje će se možda jednog dana uistinu i ostvariti oni „primjeri fantazije“ koji su bili spomenuti u uvodu.

## Literatura

- [1] »NVIDIA DLSS,« [Mrežno]. Available: <https://developer.nvidia.com/rtx/dlss>.
- [2] W. Y. K. M. N. Y. Bashir SMA, »A comprehensive review of deep learning-based single image super-resolution,« *PeerJ Comput Sci*, 2021 Jul 13.
- [3] »Discover the differences between classical and AI upscaling methods, their benefits, and applications in enhancing image quality,« 2023. [Mrežno]. Available: <https://unimatrixz.com/topics/ai-upscaler/upscaling-methods/>.
- [4] »Image Processing – Bicubic Interpolation,« [Mrežno]. Available: <https://theailearner.com/2018/12/29/image-processing-bicubic-interpolation/>.
- [5] »Image Processing – Bilinear Interpolation,« [Mrežno]. Available: <https://theailearner.com/2018/12/29/image-processing-bilinear-interpolation/>.
- [6] »Image Processing – Nearest Neighbour Interpolation,« [Mrežno]. Available: <https://theailearner.com/2018/12/29/image-processing-nearest-neighbour-interpolation/>.
- [7] »Lánczos interpolation explained,« [Mrežno]. Available: <https://mazzo.li/posts/lanczos.html>.
- [8] G. RW, »Super-resolution through error energy reduction,« *Optica Acta: International Journal of Optics*, 1974.
- [9] Z. W. Z. L. Zhang K, »Learning a single convolutional super-resolution network for multiple degradations,« u *Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [10] F. C. T. D. M. J. Martin D, »A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2001.

- [11] M. M. F. C. M. J. Arbeláez P, »Contour detection and hierarchical image segmentation,« *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- [12] L. P. W. X. T. X. Liu Z, »Deep learning face attributes in the wild,« u *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [13] T. R. Agustsson E, »NTIRE 2017 challenge on single image super-resolution: dataset and study,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2017.
- [14] O. T. Y. K. M. Y. Y. T. A. K. Fujimoto A, »Manga109 dataset and creation of metadata,« u *CM International Conference Proceeding Series*, New York, 2016.
- [15] M. M. B. S. H. J. P. P. R. D. D. P. Z. C. Lin TY, »Microsoft COCO: common objects in context,« u *European Conference on Computer Vision*, 2014.
- [16] Y. K. D. C. C. L. C. Wang X, »Recovering realistic texture in image super-resolution by deep spatial feature transform,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [17] M. R. T. R. M. T. Z.-M. L. Blau Y, »The 2018 PIRM challenge on perceptual image super-resolution,« u *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [18] E. M. P. M. Zeyde R, »On single image scale-up using sparse-representations,« u *International Conference on Curves and Surfaces*, 2012.
- [19] R. A. G. C. M. M. Bevilacqua M, »Low-complexity single-image super-resolution based on nonnegative neighbor embedding,« u *Electronic Proceedings of the British Machine Vision Conference*, 2012.
- [20] W. J. H. T. M. Y. Yang J, »Image super-resolution via sparse representation,« u *IEEE Transactions on Image Processing*, 2010.

- [21] S. A. A. N. Huang JB, »Single image super-resolution from transformed self-exemplars,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2015.
- [22] E. S. V. G. L. W. C. W. J. Z. A. Everingham M, »The Pascal Visual Object Classes Challenge: A Retrospective,« *International Journal of Computer Vision*, 2014.
- [23] L. J. L. K. Kim J, »Deeply-recursive convolutional network for image super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [24] R. K. B. W. L. V. Thapa D, »A performance comparison among different super-resolution techniques,« *Computers & Electrical Engineering*, 2016.
- [25] Y. J. L. X. Tai Y, »Image super-resolution via deep recursive residual network,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Piscataway, 2017.
- [26] Z. X. R. S. S. J. He K, »Deep residual learning for image recognition,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2016.
- [27] K. B. S. K. Ahn N, »Fast, accurate, and lightweight super-resolution with cascading residual network,« u *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2018.
- [28] C. S. L. D. Y. M. W. M. H. T. Han W, »Image super-resolution via dual-state recurrent networks,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [29] L. Z. V. D. M. L. W. K. Huang G, »Densely connected convolutional networks,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, 2017.
- [30] A. S. a. B. P. Shrivastava, »Different Techniques of Image SR Using Deep Learning: A Review,« *IEEE Sensors Journal*, svez. 23, br. 3, pp. 1724-1733, 2023.



- [31] »Image Scaling Encyclopedia on Academic Accelerator,« [Mrežno]. Available: <https://academic-accelerator.com/encyclopedia/image-scaling>. [Pokušaj pristupa 2023].
- [32] L. W. L. T. W. Z. T. D. Ma K, »DipIQ: blind image quality assessment by learning-to-rank discriminable image pairs,« *IEEE Transactions on Image Processing*, 2017.
- [33] L. S. Kim J, »Deep learning of human visual sensitivity in image quality assessment framework,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, 2017.
- [34] L. W. Z. K. D. Z. W. Z. Z. W. Ma K, »End-to-end blind image quality assessment using deep neural networks,« *IEEE Transactions on Image Processing*, 2018.
- [35] M. P. Talebi H, »NIMA: neural image assessment,« *IEEE Transactions on Image Processing*, 2018.
- [36] V. D. W. J. B. A. Liu X, »RankIQA: learning from rankings for no-reference image quality assessment,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2017.
- [37] Y. H. W. Z. J. S. Gao H, »Pixel transposed convolutional networks,« *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [38] L. G. L. X. G. Q. Tong T, »Image super-resolution using dense skip connections,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2017.
- [39] T. Y. K. Y. Z. B. F. Y. Zhang Y, »Residual dense network for image super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [40] Y. J. L. X. X. C. Tai Y, »MemNet: a persistent memory network for image restoration,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2017.

- [41] Y. K. W. S. G. J. L. Y. Wang X, »ESRGAN: enhanced super-resolution generative adversarial networks,« u *Computer Vision—ECCV, 2018 Workshops*, Springer, 2019, pp. 63-79.
- [42] W. X. G. X. Hui Z, »Fast and accurate single image super-resolution via information distillation network,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [43] Q. W. L. F. J. X. Xue S, »Wavelet-based residual attention network for image super-resolution,« *Neurocomputing*, 2020.
- [44] Z. H. Z. K. L. L. Z. W. Liu P, »Multi-level wavelet-CNN for image restoration,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2018.
- [45] K. N. V. O. E. L. G. A. K. K. Van Den Oord A, »Conditional image generation with PixelCNN decoders,« u *Advances in Neural Information Processing Systems*.
- [46] Z. A. Simonyan K, »Very deep convolutional networks for large-scale image recognition,« u *3rd International Conference on Learning Representations*, 2015.
- [47] T. L. H. F. C. J. C. A. A. A. A. A. T. A. T. J. W. Z. S. W. Ledig C, »Photo-realistic single image super-resolution using a generative adversarial network,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, 2017.
- [48] B. JT, »A General and Adaptive Robust Loss Function,« 2017. [Mrežno]. Available: <https://arxiv.org/abs/1701.03077>. [Pokušaj pristupa 2023].
- [49] G. Y. Levy O, »Neural word embedding as implicit matrix factorization,« *Advances in Neural Information Processing Systems*, 2014.
- [50] P. T. I. P. E. A. Zhu JY, »Unpaired image-to-image translation using cycle-consistent adversarial networks,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2017.

- [51] P. F. M. B. S.-H. A. S.-H. O. S. C. Wang Y, »A fully progressive approach to single-image super-resolution,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2018.
- [52] D. A. H. S. M. S. R. N. R. C. Bei Y, »New techniques for preserving global structure and denoising with low information loss in single-image super-resolution,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2018.
- [53] K. B. S. K. Ahn N, »Image super-resolution via progressive cascading residual network,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2018.
- [54] L. C. H. K. T. X. Dong C, »Learning a deep convolutional network for image superresolution,« u *Computer Vision—ECCV 2014*, Springer, 2014.
- [55] S. G. U. N. Haris M, »Deep back-projection networks for super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [56] Y. J. L. Z. Y. X. J. G. W. W. Li Z, »Feedback network for image super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2019.
- [57] B. H. A. N. Y. M. Lai WS, »Deep laplacian pyramid networks for fast and accurate super-resolution,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*,,, Piscataway, 2017.
- [58] L. C. H. K. T. X. Dong C, »Image super-resolution using deep convolutional networks,« *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [59] L. C. A. A. A. A. T. J. W. Z. S. W. Caballero J, »Real-time video super-resolution with spatio-temporal networks and motion compensation,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, 2017.

- [60] S. S. K. H. N. S. L. K. Lim B, »Enhanced deep residual networks for single image super-resolution,« u *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Piscataway, 2017.
- [61] S. B. H. M. Sajjadi MSM, »EnhanceNet: single image super-resolution through automated texture synthesis,« u *Proceedings of the IEEE International Conference on Computer Vision*, Piscataway, 2017.
- [62] F. F. M. K. Z. G. Li J, »Multi-scale residual network for image super-resolution,« u *Computer Vision–ECCV 2018*, Springer, 2018.
- [63] L. K. L. K. W. L. Z. B. F. Y. Zhang Y, »Image super-resolution using very deep residual channel attention networks,« u *Computer Vision–ECCV 2018*, Springer, 2018.
- [64] M. H. Z. X. W. Z. T. T. S. J. Hu X, »Meta-SR: a magnification-arbitrary network for super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2019.
- [65] S. G. U. N. Haris M, »Recurrent back-projection network for video super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2019.
- [66] C. J. Z. Y. X. S. Z. L. Dai T, »Second-order attention network for single image super-resolution,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2019.
- [67] L. X. C. D. Y. S. Xintao Wang, »Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data,« 2021. [Mrežno]. Available: <https://arxiv.org/abs/2107.10833>. [Pokušaj pristupa 2023].
- [68] J.-B. a. S. A. a. A. N. Huang, »Single Image Super-Resolution from Transformed Self-Exemplars (CVPR 2015),« 2015. [Mrežno]. Available: <https://github.com/jbhuan0604/SelfExSR>. [Pokušaj pristupa 2023].
- [69] L. S. L. C. T. X. Zhu S, »Deep cascaded Bi-network for face hallucination,« u *Proceedings of the European Conference on Computer Vision*, 2016.

- [70] T. Y. L. X. S. C. Y. J. Chen Y, »FSRNet: end-to-end learning face super-resolution with facial priors,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [71] Z. Z. C. C. H. W. Q. Y. L. W. Z. T. Zhang K, »Super-identity convolutional neural network for face hallucination,« u *Proceedings of the European Conference on Computer Vision*, 2018.
- [72] T. G. Bulat A, »Super-FAN: integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2018.
- [73] Y. K. D. C. T. X. L. C. Wang X, »Deep network interpolation for continuous imagery effect transition,« u *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Piscataway, 2019.
- [74] L. L. S. Y. L. X. L. G. Cao Q, »Attention-aware face hallucination via deep reinforcement learning,« u *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, 2017.
- [75] Y. Y. X. J. Y. J. Y. M. Z. J. Gao G, »Constructing multilayer locality-constrained matrix regression framework for noise robust face super-resolution,« *Pattern Recognition*.
- [76] W. Y. Bashir SMA, »Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network,« *Remote Sensing*, 2021.
- [77] Y. S. D. Q. K. A. Kappeler A, »Super-resolution of compressed videos using convolutional neural networks,« u *Proceedings—International Conference on Image Processing*, 2016.
- [78] C. H. Guo J, »Building an end-to-end spatial-temporal convolutional network for video super-resolution,« u *31st AAAI Conference on Artificial Intelligence*, 2017.

- [79] L. Z. L. J. S. Q. Zhu X, »Video super-resolution based on a spatio-temporal matching network,« *Pattern Recognition*, 2021.
- [80] F. Z. K. F. W. J. S. K. G. E. L. J. G. G. H. B. Chaudhari AS, »Super-resolution musculoskeletal MRI using deep learning,« *Magnetic Resonance in Medicine*, 2018.
- [81] E. A. a. R. Timofte, »Ntire 2017 challenge on single image super-resolution: Dataset and study,« u *CVPRW*, 2017.
- [82] »Ntire 2017 challenge on single image super-resolution: Methods and results,« u *CVPRW*, 2017.
- [83] K. Y. K. C. C. C. D. a. Xintao Wang, »Basicsr,« 2020. [Mrežno]. Available: <https://github.com/XPixelGroup/BasicSR>. [Pokušaj pristupa 2023].
- [84] Y. H. S. L. L. W. a. T. T. Zhengxiong Luo, »Unfolding the alternating optimization for blind super resolution,« *NeurIPS*, 2020.
- [85] Z. X. H. L. Z. Z. Q. Y. W. Z. L. L. P. Wei, »Component divide-and-conquer for real-world image super-resolution,« u *ECCV*, 2020.
- [86] J. L. L. G. R. K. Zhang, »Designing a practical degradation model for deep blind image super-resolution,« u *arXiv:2103.14006*, 2021.
- [87] H. Z. H. Y. Z. C. L. Z. J. Cai, »Toward real-world single image super-resolution: A new benchmark and a new model,« u *ICCV*, 2019.
- [88] K. N. T. R. V. K. G. L. Ignatov A, »Dslr-quality photos on mobile devices with deep convolutional networks,« u *ICCV*, 2017.
- [89] Z. H. P. X. X. T. F. S. B. A. T. A. Zhou B, »Semantic understanding of scenes through the ade20k dataset,« *International Journal of Computer Vision*, 2019.
- [90] »chaiNer,« [Mrežno]. Available: <https://github.com/chaiNer-org/chaiNer>. [Pokušaj pristupa 2023].
- [91] »enhancr,« [Mrežno]. Available: <https://github.com/mafiosnik777/enhancr>. [Pokušaj pristupa 2023].

- [92] »Upscayl,« [Mrežno]. Available: <https://github.com/upscayl/upscayl>. [Pokušaj pristupa 2023].
- [93] »waifu2x,« [Mrežno]. Available: <https://github.com/nagadomi/waifu2x>. [Pokušaj pristupa 2023].
- [94] »ComfyUI,« [Mrežno]. Available: <https://github.com/comfyanonymous/ComfyUI>. [Pokušaj pristupa 2023].
- [95] »Waifu2x-Extension-GUI,« [Mrežno]. Available: <https://github.com/AaronFeng753/Waifu2x-Extension-GUI>. [Pokušaj pristupa 2023].
- [96] »Squirrel-RIFE,« [Mrežno]. Available: <https://github.com/Justin62628/Squirrel-RIFE>. [Pokušaj pristupa 2023].
- [97] »OpenModelDB,« [Mrežno]. Available: <https://openmodeldb.info/>. [Pokušaj pristupa 2023].

## Skraćenice

AI	Artificial Intelligence	umjetna inteligencija
BN	Batch normalization	normalizacija serije ( <i>batch-a</i> )
CNN	Convolution Neural Networks	konvolucijska neuronske mreže
DI	Directional Interpolation	usmjerena interpolacija
DRCN mreža	Deeply-recursive Convolutional Network mreža	duboko rekurzivna konvolucijska mreža
DS	Down-sampling	smanjenje uzorkovanja
ECCV	European Conference on Computer Vision	Europska konferencija za računalni vid
EMA	Exponential Moving Average	eksponencijalno kretanje prosjeka
FR-IQA	Full Reference Image Quality Assesment	procjena kvalitete pune referentne slike
GAN	Generative Adversial Networks	generativne suparničke mreže
GAP	Global Average Pooling	globalno prosječno grupiranje
GUI	Graphical User Interface	grafičko korisničko sučelje
HR	High-resolution	visoka rezolucija
IQA	Image Quality Assesment	procjena kvalitete slike
LR	Low-resolution	niska rezolucija
MAE	Mean Absolute Error	srednja apsolutna greška
MISR	Multi Image Super Resolution	super-rezolucija više slika
MR	Magnetic resonance	magnetska rezonancija
MSE	Mean Square Error	srednja kvadratna pogreška
PSNR	Peak signal-to-noise ratio	omjer vršnog signala i šuma
RRDB blokova	Residual-in-residual dense blocks blokova	gusti blokovi unutar rezidualnih gustih blokova
SISR	<i>Single Image Super Resolution</i>	super-rezolucija jednom slikom
SN	Spectral normalization	spektralna normalizacija
SR	Super-resolution	super-rezolucija
SRCNN rezolucije	Super-resolution CNN rezolucije	konvolucijska neuronska mreža super-rezolucije
SSIM	Structural Similarity Index Metrics	metrika za indeks strukturalne sličnosti
US	Up-sampling	povećanje uzorkovanja
USM	Un-Sharp Making	neoštro maskiranje
WIP	Work in progress	rad u tijeku
WT	Wavelet transform	transformacija valića



# Privitak

## Skripta za evaluaciju metrika

Skripta je dostupna na linku ispod, i jedino treba pripaziti modificirati putanje za slike u slučaju da se skripta pokreće izvan mape gdje se slike nalaze, i jednako tako nazive modela i SRF da odgovaraju slikama.

[https://drive.google.com/file/d/1e\\_9IytP0tYEGyc3JugVR9whii0S9F9DO/view?usp=sharing](https://drive.google.com/file/d/1e_9IytP0tYEGyc3JugVR9whii0S9F9DO/view?usp=sharing)

## Pojednostavljena skripta za evaluaciju samo dvaju slika

Kako je bilo i spomenuto u poglavlju za implementaciju provedenog eksperimenta, priložena je i pojednostavljena skripta koja je računala PSNR i SSIM metriku samo za dvije slike, u usporedbi s referentnom slikom. Skripta je dostupna na linku ispod, a pravilo modificiranja vrijedi kao i za prethodno spomenutu glavnu skriptu evaluacije, samo što je sada dovoljno promijeniti samo nazive dvaju slika dobivenih različitim metodama i originalne slike za evaluaciju.

<https://drive.google.com/file/d/1dsnNeMncpW2vKMOqCzZrSQUr28relHkl/view?usp=sharing>

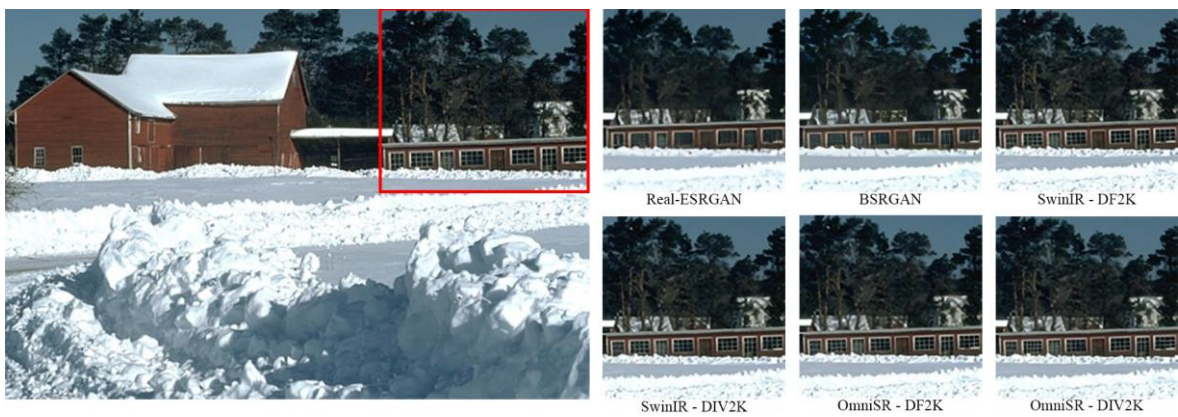
## Dodatne kvalitativne usporedbe slika eksperimenta



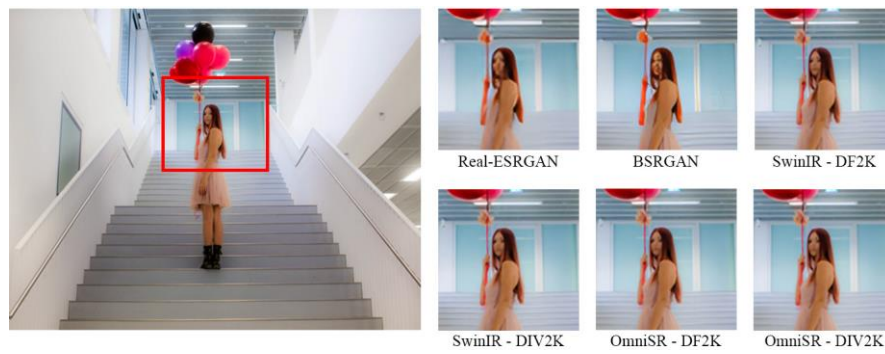
Slika 33. Usporedba na slici 13 iz Set14 SRF2



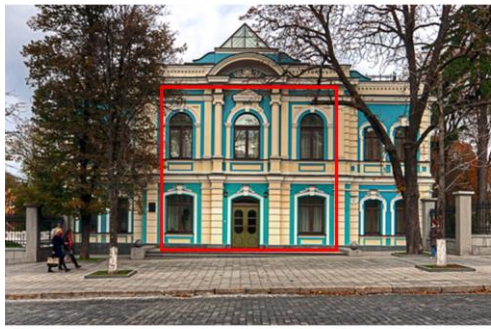
Slika 34. Usporedba na slici 96 iz BSD100 SRF2



Slika 35. Usporedba na slici 100 iz BSD100 SRF2



Slika 36. Usporedba na slici 9 iz Urban100 SRF2



Slika 37. Usporedba na slici 88 iz Urban100 SRF2

## Popis slika

Slika 1. Primjer povećanja slike i vrijednosti njenih piksela tehnikom najbližeg susjeda [6] .....	5
Slika 2. Primjer povećanja slike i vrijednosti njenih piksela tehnikom bi-linearne interpolacije [5].....	7
Slika 3. Primjer povećanja slike i vrijednosti njenih piksela tehnikom bi-kubične interpolacije [4].....	8
Slika 4. Primjerak grafa <i>sinc</i> funkcije [7].....	9
Slika 5. Klasifikacija metoda za povećanje slike - SR .....	11
Slika 6. Smanjivanje i povećanje uzorkovanja u SR prema [2] .....	14
Slika 7. Princip rada pod-piksel sloja; (a) ulaz, (b) konvolucija, (c) preoblikovanje [2] ....	24
Slika 8. Princip rada dekonvolucijskog sloja; (a) ulaz, (b) ekspanzija, (c) konvolucija [2]	25
Slika 9. Struktura neuronske mreže za rekurzivno učenje prema [2] .....	26
Slika 10. Struktura neuronske mreže za rezidualno učenje prema [2] .....	27
Slika 11. Struktura neuronske mreže za učenje temeljeno na gustim vezama prema [2]....	28
Slika 12. Struktura neuronske mreže za višesmjerno učenje prema [2].....	29
Slika 13. Struktura neuronske mreže za učenje temeljeno na naprednoj konvoluciji prema [2] .....	30
Slika 14. Struktura neuronske mreže za učenje temeljeno na pažnji prema [2].....	31
Slika 15. Cjevovod mreže SR temeljene na prethodnom povećanju uzorkovanju prema [2] .....	37
Slika 16. Cjevovod mreže SR temeljene na naknadnom povećanju uzorkovanju prema [2] .....	38
Slika 17. Cjevovod mreže SR temeljene na iterativnom povećanju-i-smanjivanju uzorkovanja prema [2].....	39
Slika 18. Cjevovod mreže SR temeljene na progresivnom uzorkovanju prema [2].....	39
Slika 19. <i>Benchmark</i> nekoliko selektiranih modela SR prema [2].....	43

Slika 20. Usporedba Real-ERSGAN-a s nekoliko drugih metoda na dvije različite slike [67] .....	50
Slika 21. Pregled generiranja sintetičkih podataka u modelu Real-ESRGAN implementiranjem modela degradacije visokog reda prema [67] .....	53
Slika 22. Gornji dio: artefakti na isječcima slika; Donji dio: Primjer filtriranja slike s <i>sinc</i> kernelima različitih frekvencija prema [67] .....	54
Slika 23. Arhitektura proširenog ESRGAN generatora korištenog u modelu Real-ESRGAN prema [67].....	54
Slika 24. Arhitektura U-Net diskriminatora s spektralnom normalizacijom prema [67] ....	55
Slika 25. Kvalitativna usporedba nekoliko modela na reprezentativnim uzorcima stvarnih slika povećanih faktorom skaliranja x4 [67] .....	58
Slika 26. Ablacija na dizajn diskriminatora prema [67].....	59
Slika 27. Ablacija na korištenje više kernela za zamućenost prema [67].....	59
Slika 28. Primjerci ograničenja modela prema [67] .....	60
Slika 29. Početno stanje alata chaiNer.....	63
Slika 30. <i>Template</i> za izvođenje iterativnog povećanja slike.....	68
Slika 31. Lanac za povećanje jedne slike .....	69
Slika 32. Usporedba na slici 2 iz Set14 SRF2 .....	74
Slika 33. Usporedba na slici 13 iz Set14 SRF2 .....	87
Slika 34. Usporedba na slici 96 iz BSD100 SRF2 .....	88
Slika 35. Usporedba na slici 100 iz BSD100 SRF2 .....	88
Slika 36. Usporedba na slici 9 iz Urban100 SRF2 .....	88
Slika 37. Usporedba na slici 88 iz Urban100 SRF2 .....	89