Super rezolucija slika tehnikama dubokih neuronskih mreža

Mavračić, Tin

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Tin Mavračić

SUPER REZOLUCIJA SLIKA TEHNIKAMA DUBOKIH NEURONSKIH MREŽA

Diplomski rad

Voditelj rada:
dr.sc. Tomislav Šmuc

Zagreb, studeni, 2018.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Zahvaljujem se obitelji na bezuvjetnoj potpori koju mi je pružila tokom svih godina studija.
Također, zahvaljujem se tvrtki MicroBlink na ustupljenim resursima bez kojih ovaj rad ne bi bio moguć, a posebno zaposlenicima odjela za istraživanje i razvoj na savjetima i podršci.

Sadržaj

Uvod

Super rezolucija postupak je kojim iz jedne ili više slika niske rezolucije dobivamo sliku veće prostorne rezolucije. Motivacija za promatranje problema super rezolucije u području računalnog vida je mnogo. Primjerice, u području medicinskih snimki gdje se javljaju skupi i/ili dugotrajni pregledi koji mogu rezultirati snimkama/slikama slabije kvalitete. Postupkom super rezolucije moguće je povećati kvalitetu takvih snimaka/slika te time olakšati dijagnostiku. Nadalje, česte su primjene u području satelitskih snimki gdje postoje fizička ograničenja na prostornu razlučivost prikupljenih slika. Super rezolucija općenito se može koristiti kao postupak predobrade drugih postupaka računalnog vida kao što su algoritmi prepoznavanja lica, čitanja teksta, detekcija i prepoznavanja prometnih znakova kako bi se povećala njihova točnost.

Računalni vid jedno je od standardnih područja strojnog učenja i umjetne inteligencije, a bavi se problemima obrade, analize i razumijevanja slika. Mnogi zadaci računalnog vida doživjeli su značajan napredak početkom ere dubokog učenja, pa tako i super rezolucija slika. Duboki konvolucijski modeli sastavni su dio svih *state-of-the-art* pristupa.

Cilj diplomskog rada je istražiti i ostvariti rješenje super rezolucije uz pomoć dubokih neuronskih mreža i napraviti usporedbu s klasičnim pristupima. Svrha ostvarenog rješenja jest povećati broj točnih čitanja barkodova aplikacije *PDF417 Barcode Scanner* tvrtke MicroBlink.

Diplomski rad strukturiran je kroz šest poglavlja. U ovom uvodnom dijelu predstavljena je motivacija za istraživanje super razlučivosti slika u širem skupu primjera te je zadan cilj diplomskog rada. U prvom poglavlju opisane su teorijske osnove strojnog učenja s naglaskom na duboko učenje, te je dana definicija problema super rezolucije. U drugom poglavlju opisan je širok pregled područja, navedene su glavne motivacije te osnovni principi u pozadini korištenih pristupa. U trećem poglavlju opisan je pristup za rješavanje problema super rezolucije barkodova. U četvrtom poglavlju opisan je postupak generiranja skupa za učenje kao i pribavljeni skup slika za testiranje, te korištene evaluacijske metrike. U petom poglavlju dani su detalji arhitekture i učenja modela uz analizu dobivenih rezultata. U šestom, i posljednjem poglavlju dan je zaključak ovog rada.

Poglavlje 1

Duboko učenje i super rezolucija slika

1.1 Teorija učenja

Jedna ne sasvim formalna definicija strojnog učenja mogla bi glasiti: Strojno učenje je način programiranja računala da optimiziraju zadani kriteriji uspješnosti koristeći primjere podataka ili prošla iskustva. Duboko učenje predstavlja granu strojnog učenja u kojoj su modeli učenja (umjetne) višeslojne neuronske mreže.

Iako postoje mnoge podjele algoritama strojnog učenja spomenut ćemo tek neke. Obzirom na dostupnost dodatnih informacija o primjerima za učenje razlikujemo nadzirano i nenadzirano učenje. Kod nadziranog učenja zahtjevamo da uz svaki podatak postoji i vrijednost njegove ciljne varijable. Tipični zadaci nadziranog učenja su klasifikacija i regresija. Kažemo da je promatrani algoritam regresijski ako su njegove predikcije neprekidne. Ako su predikcije promatranog algoritma diskretne, odnosno pripadaju unaprijed poznatom konačnom skupu oznaka, tada kažemo da se radi o klasifikacijskom algoritmu. S druge strane nenadzirano učenje podrazumijeva da nam nisu poznate nikakve dodatne ili povratne informacije o primjerima za učenje, a glavni cilj je otkrivanje strukturne pravilsnoti u podacima. U ovu kategoriju spadaju algoritmi grupiranja, otkrivanja iznimaka, kompresije podataka i dr.

Navedimo još podjelu klasifikacijskih algoritama na diskriminativne i generativne, ovisno o tome modelira li algoritam združenu vjerojatnost podataka i pripadne ciljne varijable ili ne. Generativni modeli uče distribuciju pojedinih klasa modelirajući združenu vjerojatnost podataka i pripadne ciljne varijable, te na temelju te vjerojatnosti provode klasifikaciju. Štoviše, generativne modele moguće je koristiti za generiranje novih sintetičkih podataka a neke potkategorije i za rekonstrukciju postojećih oštećenih podataka. S druge strane diskriminativni modeli eksplicitno modeliraju *aposteriori* vjerojatnost ciljne varijable uz dani (opaženi) ulazni podatak, odnosno direktno uče mapiranje ulaznih podataka na ciljne varijable.

Korištene oznake:

x	skalar
$\boldsymbol{x}$	vektor, slikovni isječak
$\boldsymbol{X}$	matrica, linearni operator
X	višedimenzionalno polje, tenzor ili slika
θ, $\boldsymbol{\theta}$, $\boldsymbol{\Theta}$	slobodni parametri modela
$\boldsymbol{x}_i$, $\boldsymbol{X}_{ij}$, X_{ijk}	indeksiranje elemenata vektora, matrice, tenzora
$\boldsymbol{x}^{(i)}$, $\boldsymbol{X}^{(i)}$, $\mathsf{X}^{(i)}$	indeksiranje različitih instanci istog objekta (npr. primjera za učenje)
X	slika niske rezolucije
Y	slika visoke rezolucije
p_{podaci}	vjerojatnosna distribucija podataka
p_{model}	vjerojatnosna distribucija modela

Prema [14], svaki algoritam strojnog učenja sastoji se od tri osnovne komponente:

1. *Model* koji koristimo za učenje s nepoznatim vrijednostima slobodnih parametara. Model predstavlja klasu mogućih hipoteza određenih slobodnim parametrima. Za svaki izbor parametara dobivamo jednu hipotezu.

$$H = \{h(\mathsf{x} \mid \boldsymbol{\theta})\}_\theta \tag{1.1}$$

   pri čemu smo s H i h označili model i hipotezu respektivno. Algoritmom učenja na temelju podataka u prostoru hipoteza traži hipotezu koja je u nekom smislu optimalna.

2. *Funkcija gubitka*, označimo je s L, mjeri razliku između željene (ciljne) vrijednosti i naše aproksimacije obzirom na trenutne parametre modela. *Funkcija cilja*, označena s J, je očekivana vrijednost funkcije greške koju aproksimiramo sumom funkcija greški po pojedinim primjerima:

$$J(h \mid \mathsf{x}) = \frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, h(\mathsf{x} \mid \boldsymbol{\theta})) \tag{1.2}$$

   te nam govori koliko dobro hipoteza h uz dane podatke x pogađa ciljne vrijednosti.

3. *Optimizacijski postupak* kojim tražimo parametre modela (u prostoru hipoteza) koji minimiziraju funkciju cilja. Odnosno tražimo optimalnu hipotezu u skladu s definiranom funkcijom cilja:

$$\min_h J(h \mid \boldsymbol{x}) = \min_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, h(\boldsymbol{x} \mid \boldsymbol{\theta})) \tag{1.3}$$

Ovaj postupak se još naziva i postupak učenja.

U području dubokog učenja model je obično duboka neuronska mreža, a postupak učenja najčešće se provodi (kada je to moguće) metodom gradijentnog spusta. Gradijentni spust je iterativna metoda minimizacije funkcije, koja ažurira parametre modela u suprotnom smjeru gradijenta funkcije J:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} J(h_{\boldsymbol{\theta}^{(t)}} | \boldsymbol{x}) \tag{1.4}$$

pri čemu skalar η kontrolira magnitudu promjene parametara, a naziva se stopa učenja (eng. *learning rate*).

1.2 Duboke neuronske mreže

Osnovni model svakog algoritma dubokog učenja je duboka neuronska mreža, a najjednostavniji oblik takve mreže jest unaprijedna neuronska mreža (eng. *feedforward neural network*) koju možemo reprezentirati acikličkim usmjerenim grafom koji je stablo s jednim listom za koji postoji točno jedan put od korijena do lista. Štoviše, duboke mreže općenito možemo shvatiti kao komputacijske grafove s ulazima i izlazima. Unaprijedna duboka mreža je tada komputacijski graf s ulazom u korijenu, te izlazom na listu.

Unaprijedna mreža sastoji se od najmanje tri sloja, to su ulazni i izlazni, te jedan ili više slojeva između, koje nazivamo skriveni slojevi (eng. *hidden layer*). Tipično se skriveni sloj sastoji od dvije komponente, afinog modela i nelinearne funkcije koju još nazivamo aktivacijska funkcija (eng. *activation function*. Za ulazni vektor $\boldsymbol{x} \in \mathbb{R}^n$ je tada izlaz skrivenog sloja zadan s:

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}) = g(\boldsymbol{W}^\mathsf{T} \boldsymbol{x} + \boldsymbol{b}) \tag{1.5}$$

pri čemu je g nelinearna funkcija, a $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ i $\boldsymbol{b} \in \mathbb{R}^m$ parametri mreže. U literaturi se obično aktivacijska funkcija definira kao realna funkcija jedne varijable a njeno proširenje na vektore se definira primjenom po koordinatama. Uobičajeno je koristiti istu oznaku za obje funkcije podrazumijevajući proširenje. Tako na primjer možemo definirati funkciju $g : \mathbb{R} \to \mathbb{R}$ s:

$$g(x) := \max\{0, x\} \tag{1.6}$$

ukoliko tada imamo $\boldsymbol{x} \in \mathbb{R}^n$ podrazumijevamo:

$$g(\boldsymbol{x}) = (\max\{0, \boldsymbol{x}_1\}, \ldots, \max\{0, \boldsymbol{x}_n\}) \in \mathbb{R}^n \tag{1.7}$$

Funkcija g definirana s 1.6 naziva se *ReLU* ili *rectified linear unit*. *ReLU* je opće prihvaćena aktivacijska funkcija u modernim neurnskim mrežama.

Općenito možemo imati više skrivenih slojeva $f^{(1)}, \ldots, f^{(k)}$, s pripadajućim parametrima $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \ldots, \boldsymbol{W}^{(k)}, \boldsymbol{b}^{(k)}$. Za dani $\boldsymbol{x}_{ulaz} \in \mathbb{R}^n$ je tada unaprijedna mreža zadana s:

$$F(\boldsymbol{x}_{ulaz}; \boldsymbol{\theta}) = f^{(k)}(\cdots f^{(1)}(\boldsymbol{x}_{ulaz}; \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}) \cdots ; \boldsymbol{W}^{(k)}, \boldsymbol{b}^{(k)}) \tag{1.8}$$

Za primjene u području računalnog vida najznačajnije su konvolucijske neuronske mreže koje je opisao LeCun, [37]. Osnova konvolucijskih neuronskih mreža su konvolucijski slojevi. Ime dolazi od matematičke operacije konvolucije funkcija. Za funkcije $f, g : \mathbb{R} \to \mathbb{R}$ definiramo funkciju h kao:

$$h(t) := \int f(a)g(t-a)da \tag{1.9}$$

Funkciju h zovemo konvolucija funkcija f i g, što obično označavamo zvjezdicom te pišemo:

$$h(t) = (f * g)(t) \tag{1.10}$$

U kontektsu konvolucijskih neuronskih mreža funkciju f zovemo ulaz konvolucije, a funkciju g jezgra konvolucije, ili kraće jezgra (eng. *kernel*). Ako pretpostavimo da funkcije f i g mogu poprimit samo diskretne vrijednosti, tada možemo definirati diskretnu konvoluciju h_D:

$$h_D(t) = (f * g) := \sum_{a=-\infty}^{+\infty} f(a)g(t-a) \tag{1.11}$$

U primjenama ulaz i jezgra konvolucije su višedimenzionalna polja koja se još nazivaju tenzori (eng. *tensor*), te sadrže diskretne vrijednosti. Posebice, za primjene u računalnom vidu tipičan primjer ulaznog tenzora je slika za koju želimo primjeniti konvoluciju po dvije koordinate. Za ulaznu sliku $\boldsymbol{I}$ i pripadnu dvodimenzionalnu jezgru $\boldsymbol{K}$ definiramo konvoluciju:

$$\boldsymbol{S}(i,j) = (\boldsymbol{I} * \boldsymbol{K})(i,j) := \sum_m \sum_n \boldsymbol{I}(m,n)\boldsymbol{K}(i-m, j-n) \tag{1.12}$$

Izlaz konvolucije naziva se mapa značajki. Gornja definicija analogno se proširuje za tenzore dimenzije 3 i 4 koji se tipično koriste u primjenama. U konvolucijskom sloju

obično imamo više konvolucijskih jezgri. Da bi u potpunosti odredili konvolucijski sloj potrebno je definirati broj i veličinu jezgri u sloju, korak konvolucije (eng. *stride*) i nadopunjavanje rubova (eng. *padding*). Veličina jezgre određuje susjedstvo vrijednosti ulaza koje utječe na izlaz, korak konvolucije određuje pomak jezgre, a nadopunjavanje određuje način nadomještanja nedostajućih vrijednosti. Da bi smanjili rezoluciju izlazne mape značajki u odnosu na ulaz potrebno je postavit korak $s > 1$. Ako pak želimo povećati rezoluciju tada postavljamo $s < 1$, što se postiže nadopunjavanjem vrijednosti između postojećih, pri čemu takav sloj nazivamo dekonvolucija ili transponirana konvolucija (eng. *deconvolution*, *transposed convolution*). Nakon konvolucijskog sloja također obično slijedi aktivacijska funkcija. Duboki konvolucijski modeli sasatoje se od jednog ili više konvolucijskih slojeva.

1.3 Generativni modeli

U pozadini mnogih *state-of-the-art* metoda super rezolucije nalaze se generativne suparničke mreže (eng. *generative adversarial networks* - GAN) koje spadaju u kategoriju generativnih modela, stoga ćemo radi boljeg razumijevanja ovog pristupa promotriti taksonomiju dubokih generativnih metoda predloženu u [21], gdje su pod generativnim modelima shavćeni svi modeli koji za dani skup primjera za učenje uzorkovanih iz distribucije p_{podaci}, uče reprezentirati aproksimaciju p_{model} te distribucije. Spomenuta taksonomija orijentirana je samo na metode temeljene na principu maksimalne vjerodostojnosti, ili pak metode koje je moguće formulirati u tom obliku, što predstavlja određenu restrikciju međutim omogućuje lakšu usporedbu različitih metoda.

Osnovna ideja metoda baziranih na principu maksimalne vjerodostojnosti jest definirati model parametriziran nekim parametrima $\boldsymbol{\theta}$ koji će aproksimirati vjerojatnosnu distribuciju prirodnih podataka p_{podaci} . Vjerodostojnost tada definiramo kao vjerojatnost koju model pridružuje podacima za učenje, a formalno možemo zapisati kao:

$$\prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \text{ gdje je } m \text{ broj primjera za učenje} \tag{1.13}$$

Princip maksimalne vjerodostojnosti tada kaže da odaberemo parametre modela θ^* za koje će vjerodostojnost biti maksimalna, odnosno parametre za koje će vjerojatnost podataka za učenje biti najveća moguća:

$$\theta^* = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \tag{1.14}$$

Parametar θ^* zovemo procjeniteljem maksimalne vjerodostojnosti (eng. *maximum likelihood estimation* - MLE). Često koristimo logaritam gornjeg izraza jer suma pojednostav-

ljuje izračun gradijenata te u ovom slučaju daje numerički stabilniji izraz:

$$\theta^* = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \tag{1.15}$$

Prema načinu reprezentacije funkcije gustoće, razlikujemo modele koji eksplicitno definiraju funkciju gustoće, te one koji implicitno zadaju funkciju gustoće. Kod modela koji eksplicitno definiraju funkciju gustoće odrediti parametre maksimalne vjerodostojnosti zapravo znači uvrstiti definiciju modela u izraz za vjerodostojnost i provesti optimizaciju prateći gradijente uzlazno. Glavni izazov ovog pristupa jest definirati model koji će biti dovoljno složen da obuhvati svu kompleksnost podataka uz uvjet traktabilnosti izraza vjerodostojnosti te njegovih gradijenata. Postoje dva pristupa za rješavanje ovog problema. Jedan pristup podrazumijeva definiranje modela na način da iz definicije slijedi traktabilnost spomenutih izraza, dok drugi pristup obuhvaća modele koji dozvoljavaju traktabilne aproksimacije izraza vjerodostojnosti i njegovih gradijenata.

Familija modela koji svojom konstrukcijom garantiraju traktabilnost izraza vjerodostojnosti i njegovih gradijenata omogućuje direktnu primjenu optimizacijskog postupka na izraz vjerodostojnosti, odnosno log-vjerodostojnosti zbog čega su ovi modeli uspješni u generiranju podataka, međutim imaju određena ograničenja. U ovu familiju spadaju potpuno vidljive mreže vjerovanja (eng. *Fully visible belief networks* ili kraće FVBNs, [18]) i nelinearna anliza nezavisnih komponenata (eng. *Nonlinear independent components analysis*). Potpuno vidljive mreže vjerovanja koristeći lančano pravilo vjerojatnosti faktoriziraju združenu distribuciju višedimenzionalnog vektora u produkt jednodimenzionalnih distribucija. Krenuvši od distribucije n-dimenzionalnog vektora:

$$p(\boldsymbol{x}) = p(\boldsymbol{x}_n, \ldots, \boldsymbol{x}_1)$$

iz definicije uvjetne vjerojatnosti:

$$p(x \mid y) := \frac{p(x, y)}{p(y)}, \text{ uz uvjet } p(y) \neq 0 \tag{1.16}$$

dobivamo pravilo:

$$p(x, y) = p(x \mid y)p(y). \tag{1.17}$$

Iterativnom primjenom pravila na n-dimenzinalnom vektoru dobivamo:

$$\begin{aligned} p(\boldsymbol{x}) &= p(\boldsymbol{x}_n, \ldots, \boldsymbol{x}_1) \\ &= p(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1} \ldots \boldsymbol{x}_1) p(\boldsymbol{x}_{n-1} \ldots \boldsymbol{x}_1) \\ &\ldots \\ &= \prod_{i=1}^{n} p(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1} \ldots \boldsymbol{x}_1) \end{aligned}$$

Faktorizacija distribucije reprezentirana potpuno vidljivom mrežom vjerovanja tada se može zapisati kao:

$$p_{model}(\boldsymbol{x}) = \prod_{i=1}^{n} p_{model}(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1} \dots \boldsymbol{x}_1) \tag{1.18}$$

Unutar familije FVBNs moguće su restrikcije na broj uvjetnih varijabli koje dopuštamo u faktorizaciji, jednu takvu restrikciju čini i tzv. PixelCNN model [65] kojeg ćemo detaljnije obraditi u 3.2., a familiju još spadaju i NADE [63], MADE [19]. Najveći nedostatak potupno vidljivih mreža vjerovanja jest postupak generiranja novih primjera koji traje $O(n)$ vremena, gdje je n duljina ulaznog vektora. Ako se uzme u obzir da se u primjenama za računanje uvjetnih vjerojatnosti $p(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1} \dots \boldsymbol{x}_1)$ uglavnom koriste duboki modeli, tada ukupno vrijeme generiranja postaje usko grlo za primjene u stvarnom vremenu.

Nadalje, imamo modele koji također eksplicitno modeliraju funkciju gustoće ali sami izrazi nisu traktabilni ali dopuštaju traktabilne aproksimacije. Spomenute modele moguće je podijeliti u dvije potkategorije. U prvu kategoriju pripadaju modeli koji koriste determinističke aproksimacije, dok drugoj kategoriji pripadaju oni koji koriste stohastičke aproksimacije. Determinističke aproksimacije uglavnom podrazumijevaju varijacijske metode čija je ideja definirati donju ogradu za log-vjerodostojnost modela koja je traktabilna:

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}) \leq \log p_{model}(\boldsymbol{x}; \boldsymbol{\theta}) \tag{1.19}$$

tada algoritam koji maksimizira donju ogradu $\mathcal{L}$ postiže log-vjerodostojnost barem jednaku $\mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta})$. Međutim, ukoliko je donja ograda preslaba tada bez obzira na dostupnost podataka za učenje i odabranog optimizacijkog algoritma nećemo uspjeti naučiti p_{podaci}. Glavni predstavnik ove klase je varijacijski autoenkoder (eng. *variational autoencoder* - VAE) [49]. Za razliku od FVBN varijacijski autonkoderi se smatraju puno težima za optimizaciju ali zahtjevaju konstanstno vrijeme za generiranje novih primjera. Stohastičke aproksimacije podrazumijevaju *Markov chain Monte Carlo* metode. U ovom kontekstu, Markovljev lanac (eng. *Markov chain*) je slučanji proces koji mijenja stanja pri čemu vjerojatnost slijedećeg stanja ovisi samo o trenutnom stanju. Odnosno možemo govoriti o procesu generiranja primjera gdje iterativno uzorkujemo $\hat{\boldsymbol{x}}$ iz $q(\hat{\boldsymbol{x}} \mid \boldsymbol{x})$. Uz dobro definiran operator prijelaza q moguće je garantirati da će $\hat{\boldsymbol{x}}$ konvergirati prema primjeru iz $p_{model}(\boldsymbol{x})$. Međutim, nemoguće je odrediti kada je uzorkovanje konvergiralo. Glavni predstavnici ovih metoda su Boltzmannovi strojevi (eng. *Boltzmann machines*) [15], a glavni nedostatak je loša skalabilnost na visokodimenzionalne prostore poput HD slika, te vrijeme potrebno za generiranje novih primjera.

S druge strane imamo modele koji ne definiraju funkciju gustoće eksplicitno. Ti modeli obično uzorkuju primjere iz p_{model}, te uče poboljšavati generiranje uspoređujući uzorkovane primjere s primjerima iz p_{podaci}. Ovdje možemo razlikovati metode koje generiraju nove primere u jednom koraku (GAN) i one koje to čine u više koraka poput generativnih

stohastičkih mreža (eng. *generative stochastic network* - GSN) []. GSN na temelju uzorkovanja iz p_{model} definiraju operator prijelaza za Markovljev lanac pa stoga imaju slične probleme kao i ranije spomenuti pristupi.

Preostaju nam modeli koji implicitno reprezentiraju funkciju gustoće te generiraju primjere u jednom koraku, a upravo ovdje pripadaju generativne suparničke mreže (GAN). Ideja iza GAN-ova je definirati igru za dva natjecatelja. Jednog natjecatelja zovemo generator i njegova zadaća je generirati primjere koji su što je moguće sličniji primjerima iz prave distribucije podataka p_{podaci}. Zadaća drugog natjecatelja kojeg zovemo diskriminator je razlikovati primjere koji dolaze iz p_{podaci} od onih koji dolaze od generatora što je problem binarne klasifikacije. Dvoje igrača reprezentirano je diferencijabilnim funkcijama G i D koje odgovaraju generatoru i diskriminatoru, a koje ovise o parametrima $\boldsymbol{\theta}_g$ i $\boldsymbol{\theta}_d$ respektivno. Funkcija G mapira skrivenu varijablu $\boldsymbol{z}$ uzorkovanu iz neke unaprijed definirane distribucije $p(\boldsymbol{z})$ u prostor podataka, dok funkcija D kao ulaz prima $\boldsymbol{x}$ iz prostora podataka te daje procjenu vjerojatnosti da ulazni podatak nije umjetno generiran. Distribucija $p(z)$ predstavlja slučajni šum omogućujući nam generiranje novih primjera.

Diskriminator učimo da maksimizira vjerojatnost pridruživanja točne oznake pravim podacima i onim koje mu dolaze od generatora, odnosno maksimiziramo unakrsnu entropiju oznaka i predikcija. Generator istovremeno učimo da minimizira izraz $\log(1 - D(G(\boldsymbol{z})))$, odnosno da minimizira log-vjerojatnost da je diskriminator dodijelio točnu oznaku umjetnim podacima. Drugim riječima, D i G igraju minimax igru za dva igrača s funkcijom vrijednosti V:

$$\min_G \max_D V(D, G) = \min_G \max_D \mathbb{E}_{\boldsymbol{x} \sim p_{podaci}(\boldsymbol{x})} \log D(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \log(1 - D(G(\boldsymbol{z}))) \quad (1.20)$$

Tražimo ekvilibrij igre u sedlu, odnosno Nash-ev ekvilibrij. Ekvilibrij minimax igre nije lokalni minimum funkcije V, nego njena sedlasta točka što znači da je to točka koja predstavlja lokalni minimum u odnosu na jednog igrača, te lokalni maksimum u odnosu na drugog igrača. Ekvivalentno problem možemo izrazit u terminima parametara funkcija D i G:

$$\min_{\boldsymbol{\theta}_g} \max_{\boldsymbol{\theta}_d} \mathbb{E}_{\boldsymbol{x} \sim p_{podaci}} \log D(\boldsymbol{x}; \boldsymbol{\theta}_d) + \mathbb{E}_{\boldsymbol{z}} \log(1 - D(G(\boldsymbol{z}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d)) \quad (1.21)$$

Kako provoditi učenje? Potrebno je definirati funkcije cilja za generator i diskriminator koje ćemo označavati s J_g i J_d respektivno. Navedene funkcije ovise o oba parametra $\boldsymbol{\theta}_g$ i $\boldsymbol{\theta}_d$ međutim svaki igrač smije ažurirati samo svoje parametre. U proizvoljnom koraku uzimamo uzorak pravih podataka $\boldsymbol{x}$ i uzorkujemo $\boldsymbol{z}$ iz $p(\boldsymbol{z})$. Provodimo dva koraka gradijentnog spusta pri čemu jedan ažurira parametre $\boldsymbol{\theta}_d$ kako bi minimizirao funkciju cilja J_d, dok drugi ažurira parametre $\boldsymbol{\theta}_g$ minimizirajući J_g.

Postoji nekoliko korištenih funkcija cilja, pri čemu se razlike očituju u funkciji cilja za generator, dok je svima zajednička funkcija cilja za diskriminator definirana s:

$$J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{podaci}} \log D(\boldsymbol{x}; \boldsymbol{\theta}_d) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log(1 - D(G(\boldsymbol{z}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d)) \tag{1.22}$$

Možemo se pitati kako izgleda optimalni diskriminator za fiksirani generator G:

$$\begin{aligned} J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) &= -\frac{1}{2}\int p_{podaci}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx - \frac{1}{2}\int p_z(\boldsymbol{z})(1 - D(G(\boldsymbol{z})))dz \\ &= -\frac{1}{2}\int p_{podaci}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_G(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \\ &\geq -\frac{1}{2}\int \max_y (p_{podaci}(\boldsymbol{x}) \log(y)) + p_G(\boldsymbol{x}) \log(1 - y))dx \end{aligned}$$

Funkcija pod integralom je oblika $a\log(y) + b\log(1 - y)$ za neke $(a, b) \in \mathbb{R}^2$. Uz pretpostavku da su p_{podaci} i $p_{model} = p_G$ pozitivni svugdje imamo $(a, b) \in \mathbb{R}^2 \setminus \{(0, 0)\}$, pa funkcija $y \mapsto a\log(y) + b\log(1 - y)$ postiže maksimum u točki $\frac{a}{a+b}$. Iz čega slijedi da je optimalni diskriminator uz fiksni generator G dan s:

$$D^*_G(\boldsymbol{x}) = \frac{p_{podaci}(\boldsymbol{x})}{p_{podaci}(\boldsymbol{x}) + p_{model}(\boldsymbol{x})} \tag{1.23}$$

Nadalje, optimalni generator je onaj koji uspješno generira primjere koji odgovaraju distribuciji p_{podaci}, odnosno vrijedi $p_{model} = p_{podaci}$. Ako pretpostavimo da je generator optimalan tada zbog 1.23 za optimalni diskriminator vrijedi $D_G(\boldsymbol{x}) = \frac{1}{2}$, odnosno optimalni generator svim primjerima pridružuje vjerojatnost $\frac{1}{2}$ da dolaze iz distribucije pravih podataka ne razlikujući ih.

Preostaje definirati funkciju cilja za generator. U najjednostavnijoj varijanti promatramo igru zbroja nula u kojoj je funkcija cilja generatora definirana s:

$$J_g(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) := -J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) \tag{1.24}$$

Odnosno generator maksimizira unakrsnu entropiju (u odnosu na parametre $\boldsymbol{\theta}_g$) koju diskriminator minimizira (u odnosu na $\boldsymbol{\theta}_d$). Međutim, pokazuje se da ova formulacija u postupku učenja nije stabilna. Naime, kada diskriminator s velikim vjerojatnostima odbacuje primjere generatora tada gradijent funkcije J_g teži ka nul-vektoru, što je poznato pod nazivom problem nestajućeg gradijenta. Stoga redefiniramo funkciju cilja generatora tako da zahtjevamo od generatora da maksimizira log-vjerojatnost da je diskriminator napravio pogrešku umjesto da minimizira log-vjerojatnost da je diskriminator točno označio primjer:

$$J_g := -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D(G(\boldsymbol{z})) \tag{1.25}$$

pri čemu podrazumijevamo $J_g = J_g(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g)$, $D = D(\cdot; \boldsymbol{\theta}_d)$, te $G = G(\cdot; \boldsymbol{\theta}_g)$.

Također kako bi se poboljšala stabilnost učenja GAN-ova javljaju se i druge modifikacije. Tako se u [2] predlaže varijanta diskriminatora koja uz pretpostavku o Lipshitz glatkosti vodi na minimiziranje Wasserstein udaljenosti između distribucije modela $p_{model} = p_G$ i distribucije podataka p_{podaci}. Wasserstein udaljenost poznata i pod nazivaom *Earth-Mover* udaljenost između distribucija p i q definirana je s:

$$W(p, q) := \inf_{\gamma \in \prod(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{1.26}$$

a pripadne funkcije cilja za diskriminator i generator dane su jednadžbama:

$$J_d^{WGAN} = -\mathbb{E}_{\boldsymbol{x}\sim p_{podaci}} + \mathbb{E}_{\boldsymbol{z}}\, D(G(\boldsymbol{z})) \tag{1.27}$$

$$J_g^{WGAN} = -J_d^{WGAN} \tag{1.28}$$

U [24] predlažu dodatno penalizaciju norme gradijenta diskriminatora kako bi se osiguralo Lipshitz-ovo svojstvo. Gradijent norme evaluirali na interpolaciji između podataka iz p_{podaci} i generiranog primjera iz p_{model} gdje bi gradijent optimalnog diskriminatora trebao imat jediničnu normu. Pripadne funkcije cilja uz modifikacije su dane s:

$$J_d^{WGAN-GP} = J_d^{WGAN} + \lambda\, \mathbb{E}_{\boldsymbol{z}}(\|\nabla D(\alpha \boldsymbol{x} + (1-\alpha)G(\boldsymbol{z}))\| - 1)^2 \tag{1.29}$$

$$J_g^{WGAN-GP} = -\mathbb{E}_{\boldsymbol{z}}\, D(G(\boldsymbol{z})) \tag{1.30}$$

Postoje razne modifikacije bazirane na drugim divergencijama poput Jensen-Shannon divergencije, f-divergencije te *maximum mean discrepancy*. Široka usporedba funkcija ciljeva za GAN-ove napravljena je u [43]. Kako GAN-ovi ne reprezentiraju funkciju distribucije eksplicitno ne možemo direktno izračunati $p_{model}(\boldsymbol{x})$, odnosno $p_G(\boldsymbol{x})$ zbog čega ne možemo niti koristiti klasične evaluacijske metrike poput log-vjerodostojnosti na test skupu. Za evaluaciju se stoga koriste *Inception Score* (IS) te *Fréchet Inception Distance* (FID). IS se temelji na ideji da bi za dobro generirane primjere trebalo vrijediti da distribucija klasa evaluiranog prednaučenog klasifikatora ima nisku entropiju. FID promatra udaljenost u prostoru značajki, kao udaljenost između neprekidnih višedimenzionalnih Gaussovih distribucija. Autori zaključuju kako ne postoje značajne razlike u rezultatima dobivenim korištenjem uspoređenih funkcija ciljeva, već glavne razlike proizlaze iz dostupnosti resursa za pretraživanje hiperparametara.

Zanimaju nas GAN-ovi kod kojih su funkcije G i D reprezentirane dubokim neuronskim mrežama. Tu možemo navesti DCGAN (*deep convolutional GAN*) arhitekturu [47] koja se u literaturi često spominje kao referentna arhitektura. DCGAN se sastoji od konvolucijskih i dekonvolucijskih slojeva, pri čemu se po potrebi koriste koraci veći od 1 kako bi se povećala odnosno smanjila rezolucija. Također, koristi se normalizacija po grupi u svim

slojevima osim zadnjeg sloja generatora i prvog sloja diskriminatora kako bi se omogućilo modelu da nauči točno očekivanje i varijancu podataka iz p_{podaci}.

GAN-ovi imaju puno prednosti u odnosu na ranije spomenute modele. Naime zahtijevaju konstantno vrijeme za generiranje primjera za razliku od FVBN, ne koriste Markovljeve lance kao Boltzmannovi strojevi ili GSN, ne zahtijevaju varijacijske ograde kao VAE, te se smatra da generiraju najprirodnije primjere od svih spomenutih metoda, iako je to subjektivna procjena. Unatoč spomenutim prednostima, nedostatak GAN-ova je nestabilnost učenja što predstavlja otvoren problem za daljnja istraživanja, kao i problem evaluacije generatora.

1.4 Super rezolucija slika

Problemi rekonstrukcije slika poput super rezolucije, uklanjanja zamućenja (eng. *deblurring*), šuma (eng. denoising), dodanih čestica ili elemenata (eng. *dehazing*), te rekonstrukcija uklonjenih i oštećenih dijelova (eng. *image inpainting*), ili pak općenito poboljšanje kvalitete slika (eng. *image enhancement*) u literaturi se često naziva jednim imenom *image-to-image* problemi. Spomenuti problemi su zapravo problemi traženja inverza. Potpuno općenito, na raspolaganju nam je slika X za koju pretpostavljamo da je dobivena iz neke ciljne slike $\boldsymbol{Y}$ na neki od gore opisanih načina te je zadatak odrediti nepoznatu sliku $\boldsymbol{Y}$.

Problem super rezolucije, te uklanjanja zamućenja i šuma možemo opisati jednim modelom:

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S}(\boldsymbol{Y}) + \boldsymbol{N} \tag{1.31}$$

pri čemu su $\boldsymbol{H}$ i $\boldsymbol{S}$ operatori degradacije, a $\boldsymbol{N}$ je aditivan šum. Ako su operatori $\boldsymbol{H}$ i $\boldsymbol{S}$ jednaki identiteti $\boldsymbol{I}$ tada se radi o problemu uklanjanja šuma, ukoliko je $\boldsymbol{H} = \boldsymbol{I}$ i $\boldsymbol{S}$ je operator blura tada se radi o problemu uklanjanja zamućenja, te u konačnici ako je $\boldsymbol{H}$ operator smanjenja rezolucije i $\boldsymbol{S}$ operator zamućenja tada govorimo o problemu super rezolucije.

U literaturi se spominju i druge formulacije, tako na primjer problem super rezolucije slika možemo modelirati koristeći jezgru zamućenja $\boldsymbol{S}_{blur}$ i nepoznati postupak smanjenja rezolucije $\downarrow$:

$$\boldsymbol{X} = (\boldsymbol{Y} * \boldsymbol{S}_{blur}) \downarrow \tag{1.32}$$

Super rezolucija je tipičan primjer loše postavljenog problema (eng. *ill-posed problem*). Francuski matematičar J. Hadamard uvodi pojam dobro postavljenog problema (eng. *well-posed problem*). Prema Hadamardu [48], problem je dobro postavljen ako zadovoljava sljedeća svojstva:

1. egzistencijalnost - postoji rješenje problema

2. jedinstvenost - rješenje problema je jedinstveno
3. stabilnost - male promjene u ulaznih varijablama uzrokuju male promjene u rješenju

Ako problem nije dobro postavljen, kažemo da je loše postavljen. Problem super rezolucije je loše postavljen problem jer svojstva 2. i 3. općenito nisu zadovoljena.

Poglavlje 2

Pregled postojećih metoda

Cilj ovog poglavlja je dati širok pregled postojećih metoda super rezolucije slika stavljajući naglasak na metode koje su važne zbog svog povijesnog utjecaja, motiviranja novog pogleda na modeliranje problema ili daju najbolje rezultate u vrijeme svog razvijanja. Svrha ovog poglavlja nije u potpunosti opisati spomenute metode, već istaknuti motivirajuće ideje i principe u pozadini danih pristupa.

Razlog pojave pristupa koje ćemo promatrati u sljedećim poglavljima jest nadomjestiti nedostatke interpolacijskih metoda koje predstavljaju efikasan način povećanja rezolucije slika ali uzrokuju zamućenje i gubitak detalja.

2.1 Prve metode učenjem

Zbog ranije spomenutih problema interpolacije početkom 2000-ih javljaju se pristupi [17] [16] [7] koji problem uvećavanja slike pokušavaju riješiti učenjem. Njihova ideja temelji se na učenju veze između slika niske i visoke rezolucije. Funkcija koja bi direktno mapirala slike niske rezolucije u slike visoke rezolucije bila bi suviše kompleksna, a njezino modeliranje suviše zahtjevno za tada poznate metode. Stoga se u prvim pristupima koji su bazirani na učenju problem pokušava riješiti na razini slikovnih isječaka (eng. *image patch*). Smislenost učenja veze između slikovnih isječaka niske i visoke rezolucije opravdava se opservacijom postojanja puno manje varijacije u vrijednostima piksela u slikovnom isječku uzetom iz prirodne slike u odnosu na slikovni isječak konstruiran nezavisnim slučajnim odabirom vrijednosti piksela. To nam zapravo govori da funkciju koja mapira isječke niske rezolucije u isječke visoke rezolucije nije potrebno promatrati, odnosno modelirati na skupu svih mogućih slikovnih isječaka nego na skupu slikovnih isječaka koji dolaze iz prirodnih slika. (A to je intuitivno lakši problem jer u prostoru svih slika ($\{0, 1, ..., 255\}^{nm}$) prirodne slike čine mnogostrukost.)

Autori u [17][16] koriste bazu od 100000 parova slikovnih isječaka niske i visoke rezolucije za rekonstrukciju tražene slike. Ideja je za svaki isječak polazne slike $\boldsymbol{X}$ niske rezolucije u skupu za učenje pronaći najsličnije isječke niske rezolucije te njima pripadne isječke visoke rezolucije iskoristiti za rekonstrukciju ciljnog isječka visoke rezolucije. Međutim autori primjećuju kako sama sličnost nije dovoljna, jer uzima u obzir samo lokalnu informaciju ignorirajući okruženje isječka. Za modeliranje prostorne povezanosti isječaka koriste Markovljevu mrežu, čime postižu veću kompatibilnost preklapajućih isječaka u ciljnoj slici $\boldsymbol{Y}$.

Metoda predložena u [7] zasniva se na pretpostavci da parovi slikovnih isječaka niske i visoke rezolucije u pripadnim prostorima značajki imaju slične lokalne geometrije. Ovdje je lokalna geometrija karakterizirana načinom na koji je vektor značajki pripadnog slikovnog isječka moguće rekonstruirati koristeći njegove susjede.

Neka je $D = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{m}$ skup parova slikovnih isječka za učenje, D_{LR} skup slikovnih isječka niske rezolucije, te D_{HR} skup slikovnih isječka visoke rezolucije. Odnosno, $D = \{(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{x} \in D_{LR}, \boldsymbol{y} \in D_{HR}\}$. Spomenuti ideja oblikovana je u algoritam na sljedeći način. Za svaki slikovni isječak $\boldsymbol{x}$ ulazne slike $\boldsymbol{X}$ niske rezolucije radimo sljedeće:

- U skupu D_{LR} tražimo K najbližih susjeda $\{\boldsymbol{x}^{(i_1)}, \dots \boldsymbol{x}^{(i_K)}\}$ u smislu Euklidske udaljenosti
- Tražimo parametre $\mathbf{w} \in \mathbb{R}^K$ koji minimiziraju grešku rekonstrukcije slikovnog isječka njegovim susjedima:

$$\min \left\| \boldsymbol{x} - \sum_{k=1}^{K} \mathbf{w}_k \boldsymbol{x}^{(i_k)} \right\|^2 \tag{2.1}$$

  uz ograničenje $\sum_{k=1}^{K} \mathbf{w}_k = 1$
- Koristeći K pripadnih isječaka visoke rezolucije $\{\boldsymbol{y}^{(i_1)}, \dots \boldsymbol{y}^{(i_K)}\}$ te pronađene težine $\mathbf{w} \in \mathbb{R}^K$ rekonstruiramo slikovni isječak visoke rezolucije:

$$\boldsymbol{y} = \sum_{k=1}^{K} \mathbf{w}_k \boldsymbol{y}^{(i_k)} \tag{2.2}$$

Nakon rekonstrukcije preklapajući dijelovi slikovnih isječaka se usrednjavaju kako bi se formirala konačna slika visoke rezolucije. Primijetimo kako smo pretpostavku o lokalnoj geometriji iskoristili u trećem koraku.

Važno je istaknuti nekoliko bitnih komponenata koje zahtijevaju suptilniju razradu. Reprezentacija slikovnih isječaka u prostoru značajki je trivijalna te se svodi na vektorizaciju, dok bi reprezentacija spomenutih isječaka u pažljivo odabranom prostoru značajki mogla olakšati problem rekonstrukcije. Nadalje, za promatrani slikovni isječak rekonstrukcija se

vrši najbližim susjedima u skupu za učenje, što pak nameće pitanje postojanja optimalnih susjeda koji bi bolje rekonstruirali dani isječak. U konačnici tu je i problem spajanja preklapajućih slikovnih isječaka usrednjavanjem što može rezultirati zamućenim regijama. Ovaj pristup uvelike motivira metode temeljene na rijetkim reprezentacijama koje pokušavaju riješiti neke od spomenutih problema, a kojima ćemo se baviti u poglavlju 2.2.

2.2 Metode temeljene na rijetkim reprezentacijama

Pristupi iz 2.1. pokazali su da možemo dobro reprezentirati slikovne isječke kao linearne kombinacije slikovnih isječaka iz skupa za učenje te potom koristeći dobivene koeficijente i pripadajuće slikovne isječke visoke rezolucije rekonstruirati traženi izlaz. Međutim, kako bi dobili kvalitetne reprezentacije potrebni su nam veliki skupovi primjera za učenje koji bi pokrili cjelokupnu domenu što pak uvelike utječe na složenost algoritma jer je potrebno izvršiti pretragu najbližih susjeda nad cijelim skupom. Ideja ovih metoda [71][72][13][46] je zamijeniti preveliki skup za učenje kompaktnijim skupom unoseći pretpostavku o rijetkim reprezentacijama.

Dakle, cilj nam je pronaći kompaktnije reprezentacije skupova D_{LR} i D_{HR} u vektoriziranom obliku. Zapravo tražimo matrice $\boldsymbol{D_l} \in \mathbb{R}^{n_l \times K}$ i $\boldsymbol{D_h} \in \mathbb{R}^{n_h \times K}$ (tipično $K > n_l, n_h$), čiji stupci predstavljaju tražene vektore. Spomenute matrice se u literaturi često nazivaju rječnicima (eng. *dictionaries*) dok se pripadni vektori nazivaju atomima (eng. *atoms*). Pretpostavka o rijetkim reprezentacijama tada kaže da se slikovni isječak visoke rezolucije $\boldsymbol{y}$ može prikazati kao rijetka linearna kombinacija atoma u $\boldsymbol{D_h}$ učenog iz skupa primjera. Odnosno postoji $\mathbf{w} \in \mathbb{R}^K$ takav da vrijedi:

$$\boldsymbol{y} \approx \boldsymbol{D_h}\mathbf{w}, \text{ uz uvjet } \|\mathbf{w}\|_0 \ll K \tag{2.3}$$

gdje smo s $\|\cdot\|_0$ označili l_0 pseudo-normu koja daje broj nenul komponenata vektora. Da bi osigurali jedinstvenog rijetkih reprezentacija između prostora niske i visoke rezolucije potrebno je uvest dodatna ograničenja u postupak učenja rječnika. Pretpostavimo li da smo konstruirali $\boldsymbol{D_l}$ i $\boldsymbol{D_h}$, za dani slikovni isječak niske rezolucije $\boldsymbol{x}$ tražimo pripadni isječak visoke rezolucije $\boldsymbol{y}$ na sljedeći način.

- Određujemo rijetku reprezentaciju od $\boldsymbol{x}$ u prostoru značajki niske rezolucije rješavajući problem minimizacije:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\boldsymbol{F}\boldsymbol{D_l}\mathbf{w} - \boldsymbol{F}\boldsymbol{x}\|_2^2 + \lambda \|\mathbf{w}\|_0 \tag{2.4}$$

  gdje je $\boldsymbol{F}$ operator značajki. Tipično, $\boldsymbol{F}$ je visokopropusni filtar ili je izostavljen, odnosno $\boldsymbol{F} = \boldsymbol{I}$. Konstanta $\lambda \in \mathbb{R}^+$ kontrolira odnos preciznosti rekonstrukcije i rijetkosti reprezentacije.

- Rekonstruiramo isječak visoke rezolucije:

$$\boldsymbol{y} = \boldsymbol{D_h}\mathbf{w}^*$$

Kako minimizacija $\lambda \|\mathbf{w}\|_0$ pripada klasi NP-teških problema, obično se taj izraz zamjenjuje s relaksiranom varijantom koja koristi l_1 normu.

Preostaje problem određivanja rječnika kojeg ćemo spomenuti u sljedećem pregledu pojedinih pristupa kao i druge specifičnosti vezane uz reprezentaciju u prostoru značajki. Autori u [71] proširuju (2.4) izrazom koji uvodi ograničenje na rekonstrukciju preklapajućih dijelova slikovnih isječaka u rezultantnoj slici visoke rezolucije pokušavajući tako osigurati kompatibilnost između preklapajućih isječaka, te primjenjuju relaksiranu varijantu s l_1 normom. Prošireni izraz tada možemo zapisati kao:

$$\arg\min_{\mathbf{w}} \left\| \begin{bmatrix} \boldsymbol{F}\boldsymbol{D_l}\mathbf{w} - \boldsymbol{F}\boldsymbol{x} \\ \boldsymbol{P}\boldsymbol{D_h}\mathbf{w} - \bar{\boldsymbol{y}} \end{bmatrix} \right\|_2^2 + \lambda \|\mathbf{w}\|_1$$

pri čemu je $\boldsymbol{P}$ operator projekcije na preklapajući dio između prethodno rekonsturiranog isječka visoke rezolucije i ciljnog isječka, te je $\bar{\boldsymbol{y}}$ preklapajući dio prethodno rekonstruiranog isječka. Nadalje, učenje rječnika postavljaju kao problem minimizacije u prostoru niske i visoke rezolucije koji traži vektore koji najbolje rekonstruiraju slikovne isječke iz skupa za učenje uz pretpostavku rijetkih reprezentacija. Ako s $\boldsymbol{X}$ označimo matricu čiji stupci predstavljaju vektorizirane slikovne isječke niske rezolucije iz skupa za učenje, te analogno s $\boldsymbol{Y}$ vektorizirane isječke visoke rezolucije problem možemo postaviti kao:

$$\min_{\boldsymbol{D_l},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{D_l}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_1 \tag{2.5}$$

$$\min_{\boldsymbol{D_h},\boldsymbol{Z}} \|\boldsymbol{Y} - \boldsymbol{D_h}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_1 \tag{2.6}$$

Kako bi osigurali podudaranje rijetkih reprezentacija među prostorima niske i visoke rezolucije autori objedinjuju (2.5) i (2.6) u jedan optimizacijski problem čije rješenje su traženi rječnici $\boldsymbol{D_l}$ i $\boldsymbol{D_h}$:

$$\min_{\boldsymbol{D_l},\boldsymbol{D_h},\boldsymbol{Z}} \frac{1}{n_l}\|\boldsymbol{X} - \boldsymbol{D_l}\boldsymbol{Z}\|_F^2 + \frac{1}{n_h}\|\boldsymbol{Y} - \boldsymbol{D_h}\boldsymbol{Z}\|_F^2 + \lambda(\frac{1}{n_l} + \frac{1}{n_h})\|\boldsymbol{Z}\|_1 \tag{2.7}$$

pri čemu su n_l, n_h dimenzije slikovnih isječaka niske i visoke rezolucije respektivno.

U [72] autori na danu sliku niske rezolucije $\boldsymbol{X}$ primjenjuju bikubnu interpolaciju kako bi dobili početnu verziju $\boldsymbol{Y}_0$ ciljne slike visoke rezolucije $\boldsymbol{Y}$. Tada je zadaća odrediti rezidual između ciljne slike i dobivene aproksimacije $\boldsymbol{Y} - \boldsymbol{Y}_0$. Na ovaj način prebačen je

fokus učenja na karakteristične rubove i teksturu koja nedostaje slikama dobivenim bikubnom interpolacijom. Na uvečanu sliku $\boldsymbol{Y}_0$ autori primjenjuju nekoliko visokopropusnih filtara, ekstrakciju slikovnih isječaka te redukciju njihove dimenzionalnosti koristeći metodu glavnih komponenata. Tako dobivaju skup slikovnih isječaka $\{\boldsymbol{y}_0^{(1)}, \ldots, \boldsymbol{y}_0^{(m)}\}$ čiju pripadnu matricu označavamo s $\boldsymbol{Y}_0 \in \mathbb{R}^{n_l \times m}$. Za razliku od [71], u ovom radu učenje rječnika odvija se u dvije odvojene faze. Prvo se modelira problem za prostor niske rezolucije:

$$\min_{\boldsymbol{D_l}, \boldsymbol{Z}} \|\boldsymbol{Y}_0 - \boldsymbol{D_l}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_0 \tag{2.8}$$

rješenje se pronalazi K-SVD algoritmom za rijetke reprezentacije opisanom u [1], a rezultat su rječnik $\boldsymbol{D_l} \in \mathbb{R}^{n_l \times K}$ i matrica koeficijenata $\boldsymbol{Z} \in \mathbb{R}^{K \times m}$. Sada bi mogli direktno dobiti $\boldsymbol{D_h}$ zahtijevajući podudarnost reprezentacija za pripadne isječke visoke rezolucije:

$$\boldsymbol{D_h} = \arg\max_{\boldsymbol{D_h}} \|\boldsymbol{E} - \boldsymbol{D_h}\boldsymbol{Z}\|_F^2 \tag{2.9}$$

gdje smo s $\boldsymbol{E}$ označili matricu slikovnih isječaka iz rezidualne slike $\boldsymbol{Y} - \boldsymbol{Y}_0$. Međutim, tako direktna rekonstrukcija ne uzima u obzir kompatibilnost preklapajućih dijelova isječaka. Kako se ciljna slika dobiva usrednjavanjem slikovnih isječaka trebalo bi unjeti tu informaciju u postupak učenja rječnika. Definirajmo s $\boldsymbol{R}_k$ operator koji izvlači isječak dimenzije $n \times n$ s dane slike visoke rezolucije iz lokacije k, te neka je $\boldsymbol{Z} = [\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}]$. Tada bi želji da je ciljna slika dobivena kao:

$$\hat{\boldsymbol{Y}} = \boldsymbol{Y}_0 + \left[\sum_k \boldsymbol{R}_k^\mathsf{T}\boldsymbol{R}_k\right]^{-1}\left[\sum_k \boldsymbol{R}_k^\mathsf{T}\boldsymbol{D_h}\boldsymbol{z}^{(k)}\right] \tag{2.10}$$

pri čemu izraz $\boldsymbol{R}_k^\mathsf{T}\boldsymbol{D_h}\boldsymbol{z}^{(k)}$ konstruira isječak visoke rezolucije $\boldsymbol{D_h}\boldsymbol{z}^{(k)}$, pozicionirajući ga u slici visoke rezolucije, dok izraz $\sum_k \boldsymbol{R}_k^\mathsf{T}\boldsymbol{R}_k$ predstavlja dijagonalnu matricu koja penalizira rezultantne vrijednosti slike prema broju doprinosa iz preklapajućih dijelova isječaka. Sada je prirodno definirati $\boldsymbol{D_h}$ kao problem minimizacije:

$$\begin{aligned}\boldsymbol{D_h} &= \arg\min_{\boldsymbol{D_h}} \left\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\right\|_2^2 \\ &= \arg\min_{\boldsymbol{D_h}} \left\|\boldsymbol{Y} - \boldsymbol{Y}_0 - \left[\sum_k \boldsymbol{R}_k^\mathsf{T}\boldsymbol{R}_k\right]^{-1}\left[\sum_k \boldsymbol{R}_k^\mathsf{T}\boldsymbol{D_h}\boldsymbol{z}^{(k)}\right]\right\|_2^2\end{aligned}$$

Na taj način se provodi učenje rječnika $\boldsymbol{D_l}$ i $\boldsymbol{D_h}$. U postupku primjene algoritma, za danu ulaznu sliku niske rezolucije $\boldsymbol{X}$ provodimo isti postupak pretprocesiranja, ekstrakcije isječaka, te traženje reprezentacije u $\boldsymbol{D_l}$ te potom vršimo rekonstrukciju koristeći dobivene koeficijente i rječnik $\boldsymbol{D_h}$.

Autori u [13] dodatno nastoje smanjiti veličinu rječnika i vrijeme primjene algoritma podjelom rječnika u pod-rječnike s manje atoma. Manji pod-rječnici smanjuju vrijeme traženja rijetkih reprezentacija te tako ubrzavaju algoritam. Ovo nije nužno restrikcija jer očekujemo da će reprezentacija biti rijetka, te da postoji samo manji broj vektora u rječniku koji su relevantni za promatrani isječak. Također, ne koriste parove rječnika za kodiranje isječaka niske i visoke rezolucije, već svakom danom isječku niske rezolucije adaptivno dodjeljuju pod-rječnik samo za prostor visoke rezolucije. To je moguće uz restrikciju da nam je poznat operator zamućenja $\boldsymbol{S}$ te operator smanjivanja $\boldsymbol{H}$.

Pretpostavimo da smo konstruirali $S \in \mathbb{N}$ pod-rječnika $\boldsymbol{D}^1, \ldots, \boldsymbol{D}^S$, te neka je ponovno $\boldsymbol{R}_k$ operator koji izvlači isječak dimenzije $n \times n$ sa slike iz lokacije k. Za proizvoljni isječak niske rezolucije $\boldsymbol{x} \in \mathbb{R}^{n_l}$ tada adaptivno određujemo pripadni pod-rječnik $\boldsymbol{D}^{assign(\boldsymbol{x})}$, gdje je $assign\colon \mathbb{R}^{n_l} \to \{1, \ldots, S\}$ funkcija koja danom isječku pridružuje pod-rječnik. Da bi pronašli rijetku reprezentaciju koja će rekonstruirati isječak visoke rezolucije koristeći rječnik $\boldsymbol{D}^{assign(\boldsymbol{x})}$, rješavamo optimizacijski problem koji istovremeno zahtijeva rekonstruiranje isječka visoke rezolucije rijetkim reprezentacijama i kompatibilnost s opserviranim isječkom niske rezolucije $\boldsymbol{x}$ a dan je sljedećim izrazom:

$$\min_{\boldsymbol{z}} \left\| \boldsymbol{x} - \boldsymbol{H}\boldsymbol{S}\boldsymbol{D}^{assign(\boldsymbol{x})}\boldsymbol{z} \right\|_2^2 + \lambda \left\| \boldsymbol{z} \right\|_1 \tag{2.11}$$

Koristeći rješenje $\boldsymbol{z}$, rekonstruiramo isječak visoke rezolucije $\boldsymbol{y} = \boldsymbol{D}^{assign(\boldsymbol{x})}\boldsymbol{z}$. Ponavljajući postupak za svaki isječak slično kao u [72] rekonstruiramo sliku visoke rezolucije $\boldsymbol{Y}$ težinski usrednjavajući preklapajuće dijelove koristeći operator izvlačenja isječaka $\boldsymbol{R}_k$:

$$\boldsymbol{Y} = \left[\sum_k \boldsymbol{R}_k^\top \boldsymbol{R}_k \right]^{-1} \left[\sum_k \boldsymbol{R}_k^\top \boldsymbol{D}^{assign(\boldsymbol{x}^{(k)})} \boldsymbol{z}^{(k)} \right] \tag{2.12}$$

Preostaje pitanje konstrukcije pod-rječnika i definiranja funkcije $assign$. Ideja je jednostavna, klasteriramo skup primjera isječaka za učenje u $S \in \mathbb{N}$ kategorija, te učimo zasebno rječnik za svaku kategoriju. Sada je moguće definirati funkciju $assign$ na način da svakom isječku pridruži onu kategoriju, odnosno rječnik, do čijeg centroida jest najmanje udaljen. Autori u radu primjenjuju sofisticiranije metode za adaptivno dodjeljivanje rječnika, kao i posebne regularizacijske uvjete koji poboljšavaju reprezentabilnost pod-rječnika, te znatiželjnog čitatelja upućujemo na originalni članak [13]. Jedan nedostatak ove metode je određivanje optimalnog broja pod-rječnika. Naime, premali broj pod-rječnika produljuje vrijeme izvršavanja algoritma i smanjuje izražajnost rječnika, dok prevelik broj klasa smanjuje reprezentativnost rječnika.

Pristupi opisani u ovom poglavlju pokazuju značajniji napredak u odnosu na interpolacijske metode, prvenstveno u pogledu robusnosti na šum, što je posljedica rijetkih reprezentacija.

2.3 Metode temeljene na samoreferirajućim primjerima

Dosad spomenute metode koriste primjere parova slikovnih isječaka iz unaprijed pripremljenog skupa za učenje kako bi odredile povezanost slika niske i visoke rezolucije. S druge strane, metode u ovom poglavlju [20][28] ne zahtijevaju takav skup za učenje nego generiraju sliku visoke rezolucije samo na temelju dane slike niske rezolucije bez potrebe za prethodnim učenjem. Spomenute metode temlje se na opservaciji redundantnosti slikovnih isječaka unutar iste prirodne slike. Naime, statistički se pokazuje [20] da u prirodnim slikama postoji znatan broj malih isječaka (5×5, 7×7 piksela) koji su međusobno slični, štoviše pokazuje se da takve sličnosti postoje i među isječcima različitih skala. Ideja ovih pristupa je iskoristiti te sličnosti za generiranje slike visoke rezolucije.

Obzirom na spomenuta dva tipa sličnosti, onaj između isječaka istih skala, te između isječaka različitih skala možemo promatrati dva pristupa rješavanja problema super rezolucije. Kada imamo isječke istih skala, tada se radi o problemu generiranja slike visoke rezolucije iz nekoliko primjera niske rezolucije koji su eventualno malo pomaknuti. Tipično se taj problem rješava postavljanjem sustava linearnih jednadžbi, pri čemu svaka od slika niske rezolucije zadaje nekoliko jednadžbi koje postavljaju ograničenja na sliku visoke rezolucije. Ukoliko je takvih slika niske rezolucije dovoljno tada postoji rješenje sustava, a to je tražena slika visoke rezolucije. S druge strane, kada imamo sličnost između isječaka različitih skala, tada ih možemo promatrati kao primjere parova niske i visoke rezolucije iz kojih je moguće učiti povezanost, te se zapravo radi o problemu promatranom u prethodnim poglavljima. Metode ovog poglavlja kombiniraju navedena dva pristupa oslanjajući se na spomenute sličnosti isječaka.

Tipičan problem super rezolucije iz više slika niske rezolucije pretpostavlja dostupnost skupa niske rezolucije $\{\boldsymbol{X}^1, \ldots, \boldsymbol{X}^K\}$. Nadalje, pretpostavlja se da je navedeni skup dobiven iz iste slike visoke rezolucije $\boldsymbol{Y}$ uz eventualno različite faktore smanjenja s_k ili pak jezgre zamućenja $\boldsymbol{S}_{blur}$ prema modelu:

$$\boldsymbol{X}^k = (\boldsymbol{Y} * \boldsymbol{S}^k_{blur}) \downarrow_{s_k} \tag{2.13}$$

Tada svaki piksel (i, j) slike $\boldsymbol{X}^k$ postavlja jedno linearno ograničenje na nepoznate vrijednosti slikovnog isječka visoke rezolucije iz kojeg je generiran na temelju jezgre zamućenja i faktora smanjenja:

$$\boldsymbol{X}^k(i, j) = \sum_{(p,q)\in\mathcal{N}\left((i,j);\boldsymbol{S}^k_{blur},s_k\right)} \boldsymbol{Y}(p, q)\boldsymbol{S}^k_{blur}(p - i, q - j) \tag{2.14}$$

pri čemu s $\mathcal{N}\left((i, j); \boldsymbol{S}^k_{blur}, s_k\right)$ označavamo skup indeksa koji pripadaju slikovnom isječku u slici visoke rezolucije iz kojeg je dobivena vrijednost piksela (i, j) u pripadnoj slici niske

rezolucije. Kao što je ranije spomenuto, ukoliko postoji dovoljan broj nezavisnih jednadžbi tada dobivamo sliku visoke rezolucije, a pokazuje se [41] da je ovakav pristup relativno stabilan za faktor uvećanja 2. Budući da se bavimo problemom super rezolucije iz jedne slike niske rezolucije moramo samostalno generirati skup $\{\boldsymbol{X}^1, \ldots, \boldsymbol{X}^K\}$. Autori u [20] predlažu spušanje problema na razinu slikovnih isječaka. Dakle, za nepoznati slikovni isječak visoke rezolucije potrebno nam je nekoliko isječaka niske rezolucije koji su dobiveni zamućenjem i smanjenjem rezolucije iz traženog isječka visoke rezolucije. Uz pretpostavku redundantnosti isječaka unutar iste skale autori za slikovni isječak niske rezolucije algoritmom najbližih susjeda traže $K = 9$ najbližih isječaka koji čine traženi skup niske rezolucije. Dobiveni skup generira linearna ograničenja na slikovni isječak visoke rezolucije kako je opisano ranije, pri čemu autori skaliraju jednadžbe prema sličnosti isječka koji je generirao jednadžbu s polaznim isječkom niske rezolucije.

Prestaje iskoristiti redundantnost između različitih skala. Autori u [20] predlažu sljedeći postupak. Neka je kao i ranije $\boldsymbol{X}$ slika niske rezolucije, te $\boldsymbol{Y}$ tražena slika visoke rezolucije pri čemu pretpostavljamo model:

$$\boldsymbol{X} = (\boldsymbol{Y} * \boldsymbol{S}_{blur}) \downarrow_s \tag{2.15}$$

Neka je $\boldsymbol{I}_0, \boldsymbol{I}_1 \ldots, \boldsymbol{I}_n$ niz slika rastuće rezolucije takvih da vrijedi $\boldsymbol{I}_0 = \boldsymbol{X}$ i $\boldsymbol{I}_n = \boldsymbol{Y}$. Neka su $\boldsymbol{S}^0_{blur}, \boldsymbol{S}^1_{blur}, \ldots, \boldsymbol{S}^n_{blur}$, te $s_0, s_1, \ldots, s_n$ pripadni nizovi jezgri zamućenja i faktora smanjenja takvi da vrijedi:

$$\boldsymbol{X} = (\boldsymbol{I}_l * \boldsymbol{S}^l_{blur}) \downarrow_{s_l} \tag{2.16}$$

Nadalje, neka je $\boldsymbol{I}_0, \boldsymbol{I}_{-1} \ldots, \boldsymbol{I}_{-m}$ niz slika padajuće rezolucije uz $\boldsymbol{I}_0 = \boldsymbol{X}$, određenih prema jednadžbi:

$$\boldsymbol{I}_{-l} = (\boldsymbol{X} * \boldsymbol{S}^l_{blur}) \downarrow_{s_l} \tag{2.17}$$

tako da koristimo jednake jezgre zamućenja kao u 2.16 uz pripadne faktore smanjenja. U primjeni spomenuti niz jezgri zamućenja možemo aproksimirati dvodimenzionalnim Guassovim funkcijama pri čemu varijance određujemo sukladno faktorima smanjenja. Dakle, cilj nam je odrediti nepoznati niz slika $\boldsymbol{I}_1 \ldots, \boldsymbol{I}_n$, gdje nam je dakako najzanimljivija slika $\boldsymbol{I}_n = \boldsymbol{Y}$.

Označimo s $\mathcal{P}_l(i, j)$ slikovni isječak sa slike $\boldsymbol{I}_l$ na lokaciji (i, j). Za svaki isječak $\mathcal{P}_0(i, j)$ polazne slike $\boldsymbol{I}_0 = \boldsymbol{X}$ možemo tražiti slične isječke u padajućem nizu slika $\{\boldsymbol{I}_{-l}\}_{l=1}^m$. Neka je $\mathcal{P}_{-l}(u, v)$ jedan takav slični isječak pronađen u slici I_{-l}. Tada on i njemu pripadni isječak $Q_0(s_l * u, s_l * v)$ u polaznoj slici $\boldsymbol{I}_0 = \boldsymbol{X}$ čine par niske i visoke rezolucije koji nam daje informaciju kako bi trebao izgledati isječak $\mathcal{P}_0(i, j)$ u slici visoke rezolucije $\boldsymbol{I}_l$, stoga autori predlažu kopiranje isječka $Q_0(s_l * u, s_l * v)$ na mjesto isječka $Q_l(s_l * i, s_l * j)$ u slici $\boldsymbol{I}_l$. Za isječak u slici $\boldsymbol{I}_0$ osnovni korak možemo kraće opisati:

$$\mathcal{P}_0(i, j) \xrightarrow{\textit{pronađi sl.}} \mathcal{P}_{-l}(u, v) \xrightarrow{\textit{odredi roditelj}} Q_0(s_l * u, s_l * v) \xrightarrow{\textit{kopiraj}} Q_l(s_l * i, s_l * j)$$

Spajanjem uvjeta dobivenih iz dvaju pristupa autori dolaze do konačnog rješenja $\boldsymbol{I}_n = \boldsymbol{Y}$. Primjećuju kako najveće poboljšanje dobivaju zahvaljujući primjerima između različitih skala dok su primjeri unutar istih skala važni za sprječavanje halucinacija i artefakata koje može proizvesti kopiranje isječaka iz različitih skala.

Autori u [28] primjećuju da prilikom traženja sličnih isječaka u različitim skalama dolazi do velikih grešaka kada se uparuju isječci s različitih ploha unutar slike. Kako bi uzeli u obzir informaciju o plohi autori dodatno traže transformacijsku matricu koja će prebaciti isječak iz polazne slike u najbolji mogući isječak u padajućem nizu isječaka. Na taj način i dodatno povećavaju prostor pretraživanja unutar slike $\boldsymbol{X}$. Ovim pristupom postižu usporedive rezultate s tada postojećim *state-of-the-art* metodama učenim na parovima niske i visoke rezolucije.

2.4 Regresijske metode

Metode ovog poglavlja nastoje direktno naučiti funkciju koja preslikava isječak niske rezolucije $\boldsymbol{x}$ u njemu pripadni isječak visoke rezolucije $\boldsymbol{y}$, postavljajući tako problem super rezolucije kao problem regresije na razini isječaka:

$$\boldsymbol{y} = f(\boldsymbol{x}) \tag{2.18}$$

Općenita prednost regresijskih pristupa u odnosu na ranije spomenute proizlazi iz činjenice da ne zahtjevaju rješavanje minimizacijskih problema u trenutku primjene algoritma na neviđenim slikama. Naime, nakon predobrade te eventualno pretraživanja najbližih susjeda potrebno je samo primjeniti funkciju f na dani isječak $\boldsymbol{x}$ kako bi rekonstruirali isječak visoke rezolucije $\boldsymbol{y}$ pri čemu je funkcija f određena u postupku učenja.

Pristup predložen u [58] (*Anchored Neighborhood Regression*-ANR) polazi od formulacije problema prema metodama rijetkih reprezentacija, relaksirajući l_0 pseudo-normu, odnosno u primjenama l_1 normu u l_2 normu:

$$\min_{\boldsymbol{w}} \|\boldsymbol{D}_l \mathbf{w} - \boldsymbol{F}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{w}\|_2 \tag{2.19}$$

gdje je $\boldsymbol{D}_l$ rječnik niske rezolucije, a $\boldsymbol{F}$ operator izvlačenja značajki. Takva reformulacija daje rješenje minimizacijskog problema 2.19 u zatvorenoj formi:

$$\boldsymbol{w} = (\boldsymbol{D}_l{}^{\mathsf{T}}\boldsymbol{D}_l + \lambda \boldsymbol{I})\boldsymbol{D}_l{}^{\mathsf{T}}\boldsymbol{F}\boldsymbol{x} \tag{2.20}$$

nakon čega možemo rekonstruirati isječak visoke rezolucije $\boldsymbol{y}$ primjenom dobivenih koeficijenata u rječniku visoke rezolucije $\boldsymbol{D}_h$:

$$\boldsymbol{y} = \boldsymbol{D}_h \boldsymbol{w} \tag{2.21}$$

Neka je $P_G := \boldsymbol{D_h}(\boldsymbol{D_l}^\mathsf{T}\boldsymbol{D_l} + \lambda\boldsymbol{I})\boldsymbol{D_l}^\mathsf{T}$ tzv. matrica projekcije na prostor visoke rezolucije, tada vrijedi:

$$\boldsymbol{y} = P_G\boldsymbol{F}\boldsymbol{x} \tag{2.22}$$

pri čemu je važno primijetiti da P_G određujemo u postupku učenja. Time smo dakle rekonstrukciju isječka $\boldsymbol{y}$ sveli na primjenu operatora izvlačenja značajki te množenje matricom P_G. Ovaj pristup autori nazivaju globalna regresija jer koristi sve atome rječnika. Međutim, kao što je ranije pokazalo samo manji broj atoma je relevantan, odnosno trebao bi biti relevantan za rekonstrukciju pojedinog isječka stoga autori grupiraju atome rječnika koristeći algoritam k-sredina u podrječnike $(\boldsymbol{N_l}^{(i)}, \boldsymbol{N_h}^{(i)})$, koje nazivaju susjedstva. Sada za svaki par podrječnika određuju matricu projekcije:

$$P_G^{(i)} = \boldsymbol{N_h}^{(i)}((\boldsymbol{D_l}^{(i)})^\mathsf{T}\boldsymbol{N_l}^{(i)} + \lambda\boldsymbol{I})(\boldsymbol{N_l}^{(i)})^\mathsf{T} \tag{2.23}$$

U postupku primjene dani isječak niske rezolucije $\boldsymbol{x}$ nakon primjene operatora $\boldsymbol{F}$ algoritmom najbližeg susjeda obzirom na centroide smještamo u jedno od susjedstva te primjenjujemo pripadnu matricu projekcije. Kako što je ranije spomenuto ovakva reformulacija značajno ubrzava primjenu algoritma.

Autori u [70] ne koriste rječnike nego primjenjuju algoritam k-sredina na cijeli skup isječaka za učenje. Pretpostavimo da se isječci $\boldsymbol{x}^{(1)} \dots \boldsymbol{x}^{(l)}$ nalaze u i-toj grupi. Označimo s $\boldsymbol{U} := [\boldsymbol{F}\boldsymbol{x}^{(1)}, \dots, \boldsymbol{F}\boldsymbol{x}^{(l)}] \in \mathbb{R}^{n\times l}$ matricu pripadnih vektora značajki, te za pripadne isječke visoke rezolucije $\boldsymbol{V} := [\boldsymbol{F}\boldsymbol{y}^{(1)}, \dots, \boldsymbol{F}\boldsymbol{y}^{(l)}] \in \mathbb{R}^{m\times l}$. Cilj je naučiti m linearnih regresija koje predviđaju m komponenti vektora značajki za prostor visoke rezolucije. Koeficijente regresija određujemo rješavajući:

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \left\| \boldsymbol{V} - \boldsymbol{W} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{1} \end{bmatrix} \right\|_2^2 \tag{2.24}$$

pri čemu je $\boldsymbol{1} \in \mathbb{R}^{1\times l}$. Slično kao prije danom isječku $\boldsymbol{x}$ algoritmom najbližeg susjeda pridružujemo grupu, te potom primjenjujemo naučeni regresor:

$$\boldsymbol{y} = \boldsymbol{W}^*\boldsymbol{F}\boldsymbol{x} \tag{2.25}$$

Nastavno na upravo opisani pristup, autori u [59] predlažu modifikaciju ANR pristupa na način da umjesto učenih rječnika koristi cijeli skup isječaka za učenje, prijavljujući *state-of-the-art* rezultate i najkraće vrijeme izvođenja.

Pristup u [51] promatra rekonstrukciju u jednadžbi 2.22 kao:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} \tag{2.26}$$

gdje je $\boldsymbol{W}$ nepoznata matrica rekonstrukcije. Autori konstatiraju da $\boldsymbol{W}$ ovisi od isječku $\boldsymbol{x}$ pa bi problem određivanja $\boldsymbol{W}$ mogli formulirati kao:

$$\min_{\boldsymbol{W}} \sum_i \|\boldsymbol{y} - \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x}\|_2^2 \tag{2.27}$$

međutim, promatraju generalizaciju ovog modela koristeći $\gamma + 1 \in \mathbb{N}$ baznih funkcija:

$$\min_{\boldsymbol{W}} \sum_i \left\| \boldsymbol{y} - \sum_{j=0}^{\gamma} \boldsymbol{W}_j(\boldsymbol{x})\phi_j(\boldsymbol{x}) \right\|_2^2 \tag{2.28}$$

Za bazne funkcije promatraju identitetu $\phi_j(\boldsymbol{x}) = \boldsymbol{x}$, polinomne funkcije $\phi_j(\boldsymbol{x}) = \boldsymbol{x}^{[j]} := (\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_n^j)$, te *radial basis function* $\phi_j(\boldsymbol{x}) = exp\left(\frac{\|x-\mu_j\|^2}{\sigma_j}\right)$. Važno je primijetiti kao je problem još uvijek linearan u parametrima koje tražimo. Da bi odredili $\boldsymbol{W}_j$ autori predlažu korištenje slučajnih šuma, prijavljujući *state-of-the-art* rezultate.

Metode ovog poglavlja uče funkcije koje direktno mapiraju isječke niske rezolucije u isječke visoke rezolucije, uklanjajući tako potrebu za rješavanjem rekonstrukcijskih problema minimizacije prilikom primjene algoritma. Spomenuti modeli su relativno jednostavne, u sljedećem poglavlju promotrit ćemo složenije modele koji direktno preslikavaju slike niske rezolucije u slike visoke rezolucije.

2.5 Metode dubokih neuronskih mreža

Motivacija za metode u ovom poglavlju dolazi iz raznih izvora. Neki pristupi su motivirani ranije spomenutim metodama, te ih nastoje interpretirati u okviru dubokog učenja, dok su drugi potaknuti uspjehom dubokih arhitektura kao što su *ResNet*, *DenseNet*, *U-Net* u drugim problemima računalnog vida. Treći su pak vođeni posebno osmišljenom predobradom ili postobradom, optimizacijskim funkcijama greške i domenskim znanjem. Sukladno tome, mogli bismo promatrati razne podjele ovih metoda. Oslanjajući se na pregled dubokih modela za super rezoluciju slika opisan u [69], promatrat ćemo metode u kontekstu dviju kategorija: arhitektura dubokih modela te optimizacijskih funkcija greški.

Neka je kao i ranije s $\boldsymbol{X}$ označena ulazna slika niske rezolucije. Zadatak je odrediti pripadnu sliku visoke rezolucije $\boldsymbol{Y}$. Zapravo, htjeli bi smo pronaći funkciju F, takvu da vrijedi:

$$\boldsymbol{Y} = F(\boldsymbol{X}) \tag{2.29}$$

U ovom poglavlju će F biti duboka neuronska mreža određena parametrima $\boldsymbol{\theta}$. Odnosno:

$$\boldsymbol{Y} = F(\boldsymbol{X}; \boldsymbol{\theta}) \tag{2.30}$$

pri čemu će ako nije drugačije navedeno $\boldsymbol{\theta}$ biti određeno gradijentnim spustom na temelju skupa primjera za učenje $\{(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)})\}_{i=1}^m$.

Duboke arhitekture

Uobičajena praksa je imenovati pristup radi lakše komunikacije i referenciranja. Tako na primjer, prvu primjenu dubokih neuronski mreža na problem super rezolucije slika autori nazivaju *SRCNN* [11], što je skraćenica za *super resolution convolutional neural network*. Motivacija za *SRCNN* dolazi od metoda temeljenih na rijetkim reprezentacijama. Naime, *SRCNN* se sastoji od tri konvolucijska sloja, koja intuitivno odgovaraju trima koracima metoda rijetkih reprezentacija: izvlačenje slikovnih isječaka i značajki, kodiranje u rječniku niske rezolucije, te dekodiranje u rječniku visoke rezolucije odnosno rekonstrukcija slike visoke rezolucije. Ulaz za *SRCNN* je bikubična interpolacija dane slike niske rezolucije $\boldsymbol{X}$, zbog čega je dovoljan relativno mali broj filtra za dobre rezultate. Štoviše, spomenuti pristup pokazuje superiorne rezultate u odnosu na sve prijašnje metode.

Međutim, *SRCNN* ima mnoge nedostatke. Kao prvo, ulaz za mrežu je bikubična interpolacija slike $\boldsymbol{X}$ koja predstavlja aproksimaciju slike visoke rezolucije. S jedne strane, problem je što takva aproksimacija ne mora biti dobra, a kako se pokazuje u [], davanje pogrešne početne procjene degradira konačni rezultat, dok s druge strane, bikubična aproksimacija znači da smo odmah prebacili svu obradu u prostor visoke rezolucije te tako povećali broj operacija u konvolucijama za faktor četiri. Nadalje, nameće se pitanje kompleksnosti *SRCNN* i upotrebe većih modela. Također, *SRCNN* je samo troslojna konvolucijska mreža bez domenskog znanja o problemu super rezolucije, nadalje učena je gradijentnim spustom minimizirajući l_2 normu između izlaza mreže i ciljne slike $\boldsymbol{Y}$ za koju se pokazuje da rezultira zamućenim slikama. Ista grupa autora u [10] koristi veći broj primjera za učenje i više filtra prijavljujući bolje rezultate u odnosu na *SRCNN* iako zadržavaju početnu dubinu modela, dajući naslutiti daljnju ekspanziju dubokih modela.

Potaknuti uspjehom 20-slojne mreže za klasifikaciju objekata *VGG-net* [54] javlja se *VDSR* [35]. Osim dublje arhitekture, *VDSR* uvodi još dvije novosti. Također kao u *SRCNN* ulaz za mrežu je bikubična interpolacija slike niske rezolucije, međutim mreža ne uči direktno mapiranje između danog ulaza i slike visoke rezolucije, već se kao predikcija očekuje rezidual između ciljne slike visoke rezolucije i bikubične aproksimacije. Autori prijavljuju da te modifikacije ubrzavaju konvergenciju modela i poboljšavaju rezultate. Druga novost se odnosi na korištenje jednog modela za različite faktore uvećanja, što se temelji na opservaciji jake povezanosti između učenih slika različitih faktora.

Vezano uz ranije spomenutu bikubičnu interpolaciju kao ulaz za duboku mrežu, autori [33] kritiziraju takav pristup jer se 16 parametara koji određuju bikubični filtar ne optimizira, stoga predlažu učenje uvećanja, promatrajući samo uvećanje faktora dva. Za takvo uvećanje moguće je podijeliti ciljnu sliku $\boldsymbol{Y}$ visoke rezolucije u 2×2 disjunktne isječke u kojima razlikujemo 4 tipa piksela: A, B, C i D kao što je prikazano tablicom 2.1. Sukladno takvoj distinkciji predlaže se tri načina uvećanja slike. U prvom načinu uče se 4 konvolucijske mreže koje ne dijele parametre, tako da svaka mreža kao ulaz prima sliku niske rezolucije $\boldsymbol{X}$ dimenzije $H \times W$, te kao izlaz daje predikciju iste rezolucije koja je zadužena

A	B
C	D

Tablica 2.1: Četiri tipa piksela A, B, C, D u pristupu [33].

za točno jednog od četiri tipa piksela. Označimo izlaze tih mreža s $\boldsymbol{X}^A$, $\boldsymbol{X}^B$, $\boldsymbol{X}^C$ i $\boldsymbol{X}^D$. Tada sliku $\boldsymbol{Y}$ rekonstruiramo preslagivanjem izlaza četiriju mreža prema formuli:

$$\boldsymbol{Y}_{ij} = \begin{cases} \boldsymbol{X}^A_{\lfloor i/s \rfloor, \lfloor j/s \rfloor} & mod(i,2) = 1, mod(j,2) = 1 \\ \boldsymbol{X}^B_{\lfloor i/s \rfloor, \lfloor j/s \rfloor} & mod(i,2) = 1, mod(j,2) = 0 \\ \boldsymbol{X}^C_{\lfloor i/s \rfloor, \lfloor j/s \rfloor} & mod(i,2) = 0, mod(j,2) = 1 \\ \boldsymbol{X}^D_{\lfloor i/s \rfloor, \lfloor j/s \rfloor} & mod(i,2) = 0, mod(j,2) = 0 \end{cases} \tag{2.31}$$

za neke $l, k \in \mathbb{N}$. Drugi način koristi samo jednu konvolucijsku mrežu koja kao izlaz daje predikciju iste rezolucije kao $\boldsymbol{X}$ ali dubine 4, pri čemu kanali odgovaraju mapama $\boldsymbol{X}^A$, $\boldsymbol{X}^B$, $\boldsymbol{X}^C$ i $\boldsymbol{X}^D$, a rekonstrukcija se provodi kao i prije prema formuli 2.31. Treći i ujedno najbolji pristup je varijanta drugog načina pri čemu se dodatno zahtjeva konzistentnost rekonstrukcija za rotacije od 0, 90, 180 i 270 stupnjeva. Konačna slika visoke rezolucije se dobiva usrednjavanjem predikcija svih rotacija.

S druge strane autori u [12] nastoje ubrzati *SRCNN* prebacujući komputaciju u prostor niske rezolucije. Točnije, kao ulaz u mrežu uzimaju sliku niske rezolucije, zatim nizom konvolucija simuliraju *SRCNN* u prosotru niske rezolucije te kao posljednji sloj koriste dekonvoluciju za mapiranje u prostor visoke rezolucije.

Primjećujući artefakte prouzročene nejednolikim doprinosima veza u dekonvolucijskim slojevima, te nastavno na učenje uvećanja [33] autori u [52] predlažu ESPCN. ESPCN je potpuno konvolucijska mreža koja kao posljednji sloj sadrži tzv. *pixel shift* operator, u oznaci $\mathcal{PS}$, koji predstavlja generalizaciju pristupa predloženog u [33]. Za ulaznu sliku $\boldsymbol{X}$ dimenzije $H \times W \times C$, te željeni faktor uvećanja s, posljednji konvolucijski sloj mora generirati izlaz $\boldsymbol{X}^{izlaz}$ dimenzije $H \times W \times C \cdot s^2$. Tada dobivamo sliku visoke rezolucije primjenom $\mathcal{PS}$ operatora na dani izlaz konvolucije:

$$\boldsymbol{Y}_{ijk} = \mathcal{PS}(\boldsymbol{X}^{izlaz})_{i,j,k} = \boldsymbol{X}^{izlaz}_{\lfloor i/s \rfloor, \lfloor j/s \rfloor, k \cdot s \cdot mod(j,s) + k \cdot mod(i,s)} \tag{2.32}$$

čija je dimenzija $s \cdot H \times s \cdot W \times C$. $\mathcal{PS}$ operator je moguće koristiti i između konvolucijskih slojeva za postepeno uvećanje rezolucije.

S ciljem smanjenja broja parametara i ubrzavanja konvergencije autori u [34] predlažu dijeljenje težina među konvolucijskim slojevima. Mreža se sastoji od niza klasičnih konvolucijskih slojeva nakon čega slijedi 16 konvolucijskih slojeva koji dijele težine. Nadalje,

izlaz iz svake od 16 dijeljenih konvolucija promatra se kao aproksimacija ciljne slike visoke rezolucija. Konačan izlaz mreže definiran je kao težinska suma 16 aproksimacija, pri čemu se koeficijenti sumacije određuju u postupku učenja. Ovakva topologija olakšava protok gradijenata prilikom učenje te tako ubrzava postupak učenja. Ideju dijeljenja parametara također koriste i [56]. Autori dijele težine rezidualnih blokova složenih u rekurzivnu topologiju. Svaki od rezidualnih blokova sastoji se od dvije konvolucije s nelinarnostima i rezidualne veze, kao što je predloženo u [26].

Rezidualne veze i njihova uspješna primjena u *ResNet* [26] arhitekturi za klasifikaciju objekata motivira konstrukciju *SRResNet* [38] modela koji se sastoji od 16 rezidualnih blokova, te koristi normalizaciju po grupi (eng. *batch normalization*) [30] kako bi stabilizirao postupak učenja i ubrzao konvergenciju. Još jedan model temeljen na rezidualnim vezama predložen je u [44], a sastoji se od niza konvolucijskih slojeva s pomakom dva koji dodatno smanjuju rezoluciju, nakon čega slijedi niz dekonvolucijskih slojeva koji podižu rezoluciju. Razine istih rezolucije povezane su rezidualnim vezama. Ovakva arhitektura povećava receptivno polje mreže dok rezidualne veze olakšavaju protok gradijenata. Trenutni *state-of-the-art* model *EDSR* [40] također je baziran na rezidualnim vezama. Međutim donosi određene promjene. Za razliku od *SRResNet* arhitekture *EDSR* izbacuje normalizaciju po grupi. Ovaj korak opravdavaju empirijski, dok intuitivno objašnjavaju kako je normalizacija po grupi pogodna za klasifikacijski problem jer klasifikacija zahtjeva izrazito apstraktne reprezentacije koje su neosjetljive na pomake koje uzrokuje takva normalizacija. S druge strane kod problema super rezolucije ulaz i izlaz mreže su izrazito korelirani pa stoga i osjetljiviji na velike pomake u unutarnjim reprezentacijama. Nadalje, *EDSR* uz povećanje dubine također značajno povećava i broj filtra po konovoluciji, iako to uz uklanjanje normalizacije po grupi dodatno usporava konvergenciju modela. Treća promjena koju donosi *EDSR* odnosi se na korištenje naučenog modela uvećanja faktora dva kao inicijalizacije za model uvećanja faktora tri i četiri, što se temelji na opservaciji jake povezanosti problema uvećanja za različite faktore.

Vođeni drugom uspješnom klasifikacijskom arhitekturom *DenseNet* [27], autori u [61] predlažu *SR-DenseNet* koja se sastoji od niza gustih blokova (eng. *dense blocks*). Gusti blokovi su konstruirani od nekoliko konvolucijskih slojeva pri čemu se za ulaz svake od konvolucija uzima konkatenacija izlaza svih prethodnih konvolucija u bloku. Takve veze nazivamo preskočnim vezama (eng. *skip connection*). Nakon niza gustih blokova slijedi *bottleneck* sloj [55] te dekonvolucija za povećanje rezolucije. Nastavno na *DenseNet* autori u [57] zamjenjuju konvolucije u gustim blokovima reziudalnim blokovima [26], te dodaju guste veze između različitih blokova. Rekonstrukciju slike visoke rezolucije rade kombinirajući izlaze svih blokova. Objašnjavaju kako intuitivno veze unutar istog bloka odgovaraju kratkotrajnom pamćenju dok veze između različitih blokova odgovaraju dugotrajnom pamćenju. Sličnu arhitekturu predlažu i u [73] gdje na krajevima gustih blokova dodaju *bottleneck* sloj za smanjenje broja parametara, te rezidualne veze koje spajaju početak s

krajem bloka. Dodatno, blokove povezuju rezidualnim vezama.

Dosad spomenute metode uglavnom nisu unosile domensko znanje i pretpostavke o problemu super rezolucije. Za razliku od spomenutih, autori u [68] nastoje spojiti pretpostavke o rijetkim reprezentacijama, odnosno pristupe temeljene na rijetkim reprezentacijama s dubokim neuronskim mrežama pod nazivom *SCN*. Koristeći notaciju kao u poglavlju 2.2. prisjetimo se da slikovni isječak visoke rezolucije $\boldsymbol{y}$ možemo rekonstruirati koristeći rječnik visoke rezolucije $\boldsymbol{D_h}$ kao:

$$\boldsymbol{y} = \boldsymbol{D_h}\mathbf{w} \tag{2.33}$$

pri čemu je $\mathbf{w}$ rješenje minimizacijskog problema za dani isječak niske rezolucije te pripadni rječnik $\boldsymbol{D_l}$:

$$\min_{\mathbf{w}} \|\boldsymbol{D_l}\mathbf{w} - \boldsymbol{x}\|_2^2 + \lambda \|\mathbf{w}\|_0 \tag{2.34}$$

Autori predlažu neuronsku mrežu koja za dani isječak niske rezolucije aproksirmira kod $\mathbf{w}$ kao što bi se dobio rješavanjem jednadžbe 2.34 uz dani rječnik $\boldsymbol{D_l}$. Mreža se sastoji od niza rekurentnih koraka pri čemu svaki od koraka popravlja aproksimaciju koda prema jednadžbi:

$$\mathbf{w}_{k+1} = h_{\boldsymbol{\theta}}(\boldsymbol{V}\boldsymbol{x} + \boldsymbol{U}\mathbf{w}) \tag{2.35}$$

gdje su $\boldsymbol{U}$, $\boldsymbol{V}$ matrice težina, a $h_{\boldsymbol{\theta}}$ funkcija definirana po elementima sa:

$$[h_{\boldsymbol{\theta}}(\boldsymbol{z})]_i := sign(\boldsymbol{z}_i)(|\boldsymbol{z}_i| - \boldsymbol{\theta}_i)_+ \text{ za vektor } \boldsymbol{z} \in \mathbb{R}^n, \text{ te } \boldsymbol{\theta}_i > 0 \tag{2.36}$$

Autori koriste varijantu iterativnog algoritma spuštanja i ograničavanja [22] (eng. *iterative shrinkage and thresholding algorithm*) temeljenu na gradijentnom spustu kojim se određuju $\boldsymbol{U}$, $\boldsymbol{V}$, te $\boldsymbol{\theta}$. Također u [42] definiraju kaskadnu varijantu *CSCN* koja koristi nekoliko *SCN* modela te daje bolje rezultate.

Kako bi iskoristili već ranije spomenutu povezanost problema super rezolucije slika za različite faktore uvećanja u [36] se predlaže LapSRN pristup koji istovremeno uči uvećanje za faktore 2, 4, i 8. Faktori 2 i 4 se promatraju kao potproblemi, stoga autori smanjuju ciljnu sliku $\boldsymbol{Y}$ te uz funkciju greške koja penalizira rekonstrukciju slike za faktor 8 dodaju funkcije greške koje penaliziraju rekonstrukciju za faktore 2 i 4. Arhitektura mreže sastoji se od dvije grane: grana za izvlačenje značajki i grana za rekonstrukciju. Rekonstrukcijska grana daje početnu aproksimaciju uvećane slike za faktor 2 dok grana za izvlačenje značajki uči rezidual između aproksimacije i ciljne slike. Ovaj uzorak se ponavlja tri puta dobivajući tako uvećanja za faktore 2, 4 i 8.

Metoda *ZSSR* iz [53] prvi je pokušaj povezivanja dubokog učenja i pristupa temeljenih na samoreferirajućim primjerima. Kao u poglavlju 2.3. osim ulazne slike niske rezolucije $\boldsymbol{X}$, metoda ne zahtjeva nikakav dodatan skup parova za učenje. *ZSSR* smanjuju rezolucije dane slike $\boldsymbol{X}$, te uči mapiranje između umanjene slike i $\boldsymbol{X}$. Naučeni model zatim primjenjuju na $\boldsymbol{X}$ kako bi generirali sliku visoke rezolucije $\boldsymbol{Y}$. Za razliku od velikih skupova

za učenje unutar jedne slike ne postoje velike varijacije pa autori koriste relativno malu konvolucijsku mrežu. Problem ovog pristupa je velika komputacijska složenost u trenutku primjene algoritma budući da se upravo tada odvija učenje dubokog modela.

Autori u [62] pokazuju kako struktura dubokih neuronskih mreža na pogodan način regularizira rekonsturkcijske probleme slika poput super rezolucije, uklanjanja šuma ili zamućenja, također ne koristeći dodatan skup primjera za učenje. Za danu ulaznu sliku niske rezolucije $\boldsymbol{X}$ fiksiraju proizvoljno odabrani vektor $\boldsymbol{z}$ te uzimaju duboku konvolucijsku neuronsku mrežu F sa slučajno inicijaliziranim težinama tražeći rješenje optimizacijskog problema:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{H}(F(\boldsymbol{z};\boldsymbol{\theta})) - \boldsymbol{X}\|_2^2 \tag{2.37}$$

pri čemu je $\boldsymbol{H}$ diferencijabilan operator smanjenja. Traženje optimalnog rješenja se prekida kad norma razlike padne ispod unaprijed definiranog praga. Za optimalno rješenje $\boldsymbol{\theta}^*$ rekonstruirana slika visoke rezolucije se tada dobiva kao:

$$\boldsymbol{Y} = F(\boldsymbol{z};\boldsymbol{\theta}^*) \tag{2.38}$$

Takva optimizacija je smislena jer autori empirijski pokazuju da između slika različitih statističkih karakteristika duboke mreže najbrže konvergiraju za slike iz prirodne distribucije, dok im za izrazito nekorelirani ulaz poput slučajnog šuma treba znatno veći broj iteracija. Iako daje zanimljiv uvid u rekonsrukcijske probleme slika spomenuti rad se kvalitetom samih rekonstrukcija ne može mjeriti s metodama nadziranog učenja.

Alogoritam iterativne unazadne projekcije (eng. *iterative back projection*, *IBP*) [31] spominje se kao postupak postobrade metoda super rezolucije [60] koji poboljšava kvalitetu uvećane slike. U iterativnom koraku aproksimacija slike visoke rezolucije se smanjuje a greška između tako dobivene i početne slike niske rezolucije propagira se za ispravak aproksimirane slike visoke rezolucije. Autori u [25] predlažu duboku mrežu koja se sastoji od naizmjence poredanih jedinica za dizanje i spuštanje rezolucije alternirajući između niske i visoke rezolucije, simulirajući tako iterativne korake ispravljanja u IBP algoritmu. Spomenute jedinice sastoje se od konvolucijskih i dekonvolucijskih slojeva te rezidualnih veza. Ovom metodom autori postižu *state-of-the-art* za faktor uvećanja 8.

Spomenimo na kraju ovog dijela pregleda metodu [8] baziranu na generativnom modelu *PixelCNN* [64], [65] koja generira piksele u slici visoke rezolucije na temelju prethodno generiranih piksela, pri čemu svaki piksel u slici visoke rezolucije ovisi o svim pikselima lijevo i iznad njega koji su veći bili generirani. Specifičnost ovog pristupa je u autoregresijskoj grani koju čini *PixelCNN*, te koja ovom modelu omogućuje stabilne rezultate uvećanja za velike faktore.

Optimizacijske funkcije greške

Većina dosad promatranih modela nastavno na SRCNN minimizira l_2 normu razlike predikcije i ciljne slike visoke rezolucije po svim primjerima iz skupa za učenje, odnosno miniziraju srednju kvadratnu grešku (eng. *mean squared error* - MSE):

$$\min_{\boldsymbol{\theta}} \sum_i \left\| F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}) - \boldsymbol{Y}^{(i)} \right\|_2^2 \tag{2.39}$$

što možemo promatrati kao regresijski problem, a traženo rješenje je točkovni procjenitelj. Ako pretpostavimo da postoji Gaussov bijeli šum $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ koji je nezavisan od ciljne slike tada prema regresijskom modelu:

$$\boldsymbol{Y} \approx F(\boldsymbol{X}; \boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{2.40}$$

imamo probabilističku interpretaciju u kojoj je uvjetna distribucija slike visoke rezolucije $\boldsymbol{Y}$ uz uvjet slike niske rezolucije $\boldsymbol{X}$ zapravo normalna distribucija (Gaussova) s vektorom očekivanja $F(\boldsymbol{X}; \boldsymbol{\theta})$ i dijagonalnom kovarijacijskom matricom $\sigma^2 \boldsymbol{I}$:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{Y}; F(\boldsymbol{X}; \boldsymbol{\theta}), \sigma^2 \boldsymbol{I}) \tag{2.41}$$

Sada primjenom principa maksimalne vjerodostojnosti tražimo MLE procjenitelj kao u poglavlju 1.3 i pretpostavljenog parametarskog modela 2.41 imamo:

$$\begin{aligned}
\max_{\boldsymbol{\theta}} \ln \prod_i p(\boldsymbol{Y}^{(i)}|\boldsymbol{X}^{(i)}) &= \max_{\boldsymbol{\theta}} \sum_i \ln \frac{1}{\sqrt{(2\pi)^k \det(\sigma^2 \boldsymbol{I})}} e^{-\frac{(\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)};\boldsymbol{\theta}))^\mathsf{T} (\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)};\boldsymbol{\theta}))}{2\sigma^2}} \\
&= \max_{\boldsymbol{\theta}} \sum_i - \ln \sqrt{(2\pi)^k \det(\sigma^2 \boldsymbol{I})} - \frac{(\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}))^\mathsf{T} (\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}))}{2\sigma^2} \\
&= \max_{\boldsymbol{\theta}} - \sum_i \frac{(\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}))^\mathsf{T} (\boldsymbol{Y}^{(i)} - F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}))}{2\sigma^2} \\
&= \max_{\boldsymbol{\theta}} - \sum_i \frac{\left\| F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}) - \boldsymbol{Y}^{(i)} \right\|_2^2}{2\sigma^2} \\
&= \min_{\boldsymbol{\theta}} \sum_i \left\| F(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}) - \boldsymbol{Y}^{(i)} \right\|_2^2
\end{aligned}$$

Pa smo time pokazali da je MSE poseban slučaj MLE uz pretpostavljeni Guassov model. Međutim kao što je ranije spomenuto minimizacija MSE ne daje vizualno zadovoljavajuće rezultate. Kao alternativu, autori u [74] predlažu da se pretpostavka o Gaussovom bijelom šumu zamijeni Laplaceovim bijelim šumom, $\boldsymbol{\epsilon} \sim \mathit{Laplace}(\mathbf{0}, \boldsymbol{b})$. Slično kao gore pokazuje se da možemo svesti MLE na minimiziranje srednje apsolutne pogreške, odnosno na miniziranje l_1 norme razlike između predikcije i ciljne slike $\boldsymbol{Y}$. Spomenuti autori na temelju

usporedbe rezultata ovih dvaju formulacija zaključuju da minimizacija srednje apsolutne greške producira slike koje su oštrije u odnosu na minimizaciju MSE.

Pretpostavka o nezavisnom aditivnom šumu u slici neovisno o odabranoj distribuciji ne mora vrijediti, stoga autori [4] predlažu traženje mapiranja koje će prebaciti slike u prostor u kojem će pretpostavka o šumu ipak vrijediti. Za ulaznu sliku niske rezolucije $\boldsymbol{X}$ određuju MSE procjenitelj visoke rezolucije $\boldsymbol{Y}_{MSE}$, te im je cilj naučiti rezidual $\boldsymbol{R}$ između procjenitelja ciljne slike $\boldsymbol{Y}$ i $\boldsymbol{Y}_{MSE}$, pri čemu od $\boldsymbol{R}$ očekujemo da aproksimira detalje visoke frekvencije izgubljene u MSE procjeni. Vjerojatnost od $\boldsymbol{R}$ uz uvjet $\boldsymbol{X}$ modeliraju Gibbsovom distribucijom:

$$p(\boldsymbol{R}|\boldsymbol{X}) = e^{-\|\Phi(\boldsymbol{X})-\Psi(\boldsymbol{R})\|^2-\log Z} \tag{2.42}$$

gdje su Φ, Ψ duboki konvolucijski modeli koji mapiraju slike u prostor značajki, a Z je funkcija particije. Ψ je obično neki prednaučeni model, a Φ određujemo na temelju skupa za učenje tražeći rješenje minimizacijskog problema:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \|\Phi(\boldsymbol{X};\boldsymbol{\theta}) - \Psi(\boldsymbol{R})\|^2 \tag{2.43}$$

Tada za proizvoljnu sliku $\boldsymbol{X}$ određujemo ciljni rezidual rješavajući:

$$\boldsymbol{R}^* = \arg\min_{\boldsymbol{R}} \|\Phi(\boldsymbol{X};\boldsymbol{\theta}^*) - \Psi(\boldsymbol{R})\|^2 \tag{2.44}$$

nakon čega sliku visoke rezolucije dobivamo direktno kao $\boldsymbol{Y} = \boldsymbol{Y}_{MSE} + \boldsymbol{R}$. Autori predlažu korištenje VGG-net modela za Ψ, ali tada u jednadžbama 2.43 i 2.44 direktno koriste $\boldsymbol{Y}$. Ova modifikacija je potrebna jer je VGG-net učen na prirodnim slikama, pa nije jasno da li bi reprezentacija dobivena za rezidual bila smislena. Dodatno predlažu prilagođavanje modela Φ i Ψ korištenim podacima uz pretpostavljenu distribuciju 2.42 što zahtjeva aproksimacije gradijenta funkcije particije. Grešku $\|\Phi(\boldsymbol{X};\boldsymbol{\theta}) - \Psi(\boldsymbol{R})\|^2$ nazivaju perceptualnom greškom jer penalizira razliku u protosru značajki zahtjevajući jednake reprezentacije koje su zbog korištenja dubokih modela invarijante na male pomake u polaznim slikama. Drugim riječima, umjesto da zahtjevamo poklapanje rekonstrukcija po pikselima, dozvoljavamo male promjene u pikselima zahtjevajući globalno poklapanje sadržaja. Nedostatak ovog pristupa je što prilikom primjene modela na neviđenoj slici $\boldsymbol{X}$ zahtjeva rješavanje minimizacijskog problema 2.44 što je vremenski zahtjevno, stoga je u [32] predložena modifikacija perceptualne greške tako da se koristi ista funkcija za mapiranje u prostor značajki a dodatno se uvodi model $F(\cdot;\boldsymbol{\theta})$ koji uči super rezoluciju postavljajući problem minimizacije:

$$\min_{\boldsymbol{\theta}} \|\Psi(F(\boldsymbol{X};\boldsymbol{\theta})) - \Psi(\boldsymbol{Y})\|^2 \tag{2.45}$$

pri čemu za preslikavanje Ψ koriste prednaučeni VGG-net. Primjena na neviđenoj slici sada zahtjeva samo prolaz kroz mrežu F. Općenito se pokazuje da perceptualne funkcije greške daju vizualno kvalitetnije rezultate u odnosu na direktno minimiziranje MSE.

Problem super rezolucije možemo pokušati riješiti GAN-ovima. Tada je u minimax igri generator G uvjetovan slikom niske rezolucije $\boldsymbol{X}$ te mora rekonstruiranom slikom visoke rezolucije $G(\boldsymbol{X})$ zavarati diskriminator, koji pokušava razlikovati $G(\boldsymbol{X})$ od prirodnih slika visoke rezolucije $\boldsymbol{Y}$:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{Y}\sim p_{podaci}(\boldsymbol{Y})} \log D(\boldsymbol{Y}) + \mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log(1 - D(G(\boldsymbol{X}))) \tag{2.46}$$

pri čemu je $p_{podaci}(\boldsymbol{Y})$ prirodna distribucija slika visoke rezolucije, a $p_{podaci}(\boldsymbol{X})$ prirodna distribucija slika niske rezolucije. Ekvivalentna formulacija u terminima distribucije generatora glasi:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{Y}\sim p_{podaci}(\boldsymbol{Y})} \log D(\boldsymbol{Y}) + \mathbb{E}_{\hat{\boldsymbol{X}}\sim p_{model}(\hat{\boldsymbol{X}})} \log(1 - D(\hat{\boldsymbol{X}})) \tag{2.47}$$

gdje je $p_{model} = p_G$. Značajan rad koji opisuje primjenu GAN-a na problem super rezolucije je [10]. Autori predlažu *SRGAN* čiji je generator ranije opisan model *SRResNet*. Funkcija cilja generatora J_g sastoji se od dvije komponente J_{VGG} i J_{GAN}. Prva komponenta je motivirana perceptulanom greškom opisanom u [32], a penalizira razliku u aktivacijama prednaučene VGG-net mreže za ciljnu sliku visoke rezolucije i danu predikciju generatora:

$$J^{i,j}_{VGG} = \frac{1}{const} \left\| \Psi_{i,j}(\boldsymbol{Y}) - \Psi_{i,j}(G(\boldsymbol{X})) \right\|^2 \tag{2.48}$$

pri čemu je s $\Psi_{i,j}$ označena mapa značajki dobivena iz j-te konvolucije prije i-tog sloja sažimanja. Minimiziranje perceptualne greške pospješuje konvergenciju GAN-a. J_{GAN} je zapravo tipična funkcija cilja generatora koja se odnosi na log-vjerodostojnost diskriminatora:

$$J_{GAN} = -\mathbb{E} \log D(G(\boldsymbol{X})) \tag{2.49}$$

Ukupna funkcija cilja dana je s:

$$J_g = J_{VGG} + \lambda J_{GAN} \tag{2.50}$$

Funkcija cilja diskriminatora je klasična unakrsna entropija pa je stoga izostavljamo iz daljnih razmatranja. Pristup predložen u [67] nastoji spojiti ideju o povezanosti uvećanja slike za različite faktore s GAN-ovima, primjenjujući J_{VGG} i J_{GAN} u različitim skalama uvećanja. Nadalje, autori u [5] primjenjuju GAN na problem super rezolucije slika zamjenjujući J_{GAN} s J_{WGAB} Wasserstein varijantom uz penalizaciju gradijenta diskriminatora:

$$J_{WGAN} = -\mathbb{E}_{\boldsymbol{Y}\sim p_{podaci}(\boldsymbol{Y})} D(\boldsymbol{Y}) + \mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} D(G(\boldsymbol{X})) + \lambda \mathbb{E}_{\hat{\boldsymbol{X}}\sim p_{\hat{X}}} \left[\left(\left\| \nabla D(\hat{\boldsymbol{X}}) \right\| - 1 \right)^2 \right] \tag{2.51}$$

gdje je distribucija $p_{\hat{X}}$ dobivena uniformnim uzorkovanjem linija između parova iz $p_{podaci}(\boldsymbol{Y})$ i $p_G(\boldsymbol{X})$. Zanimljivo je da autori dodaju i MSE grešku (J_{MSE}) između predikcije i ciljne

slike prijavljujući bolje rezultate. Nastavno na *SRGAN* autori u [50] predlažu dodavanje izraza u J_g čija će minimizacija zahtjevati poklapanje tekstura u ciljnoj slici i predikciji. Spomenuto postižu promatranjem korelacije između aktivacija VGG mreže nad parovima slikovnih isječaka niske i visoke rezolucije:

$$J_{tekstura} = \sum_{(\boldsymbol{X}_{isjecak}, \boldsymbol{Y}_{isjecak})} \left\| Gram(\Psi(\boldsymbol{X}_{isjecak})) - Gram(\Psi(\boldsymbol{Y}_{isjecak})) \right\|_2^2 \tag{2.52}$$

gdje je $Gram(\boldsymbol{W}) := \boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}$. Autori u [6] sugeriraju kako je za učenje super rezolucije važno učiti smanjenje rezolucije, stoga predlažu GAN s dva generatora i dva diskriminatora. Jedan generator uči super rezoluciju slike, dok drugi uči smanjenje rezolucije. Analogno, jedan od diskriminatora uči razlikovati slike u prostoru visoke rezolucije, dok drugi to čini u prostoru niske rezolucije. Funkcija cilja generatora je tzv. *hinge loss* varijanta opisana u [45] definirana s:

$$J_{GAN} = \mathbb{E}_{\boldsymbol{Y} \sim p_{podaci}(\boldsymbol{Y})}[\min(0, -1 + D(\boldsymbol{Y}))] + \mathbb{E}_{\boldsymbol{X} \sim p_{podaci}(\boldsymbol{X})}[\min(0, -1 - D(G(\boldsymbol{X})))] \tag{2.53}$$

a dodatno koriste i J_{MSE}. Dosada spomenuti GAN pristupi su zahtijevali parove slika niske i visoke rezolucije u postupku učenja, bilo za perceptualnu grešku J_{VGG} ili za piksel rekonstrukciju J_{MSE}. To nije ograničavajuće jer ako posjedujemo slike visoke rezolucije tada lako možemo generirati pripadne slike niske rezolucije primjenjujući operator zamućenja i spuštanja rezolucije na polazne slike. Odabirom tih operatora uvodimo određenu pristranost. Pristup predložen [29] ne zahtijeva parove niske i visoke rezolucije već samo dva skupa podataka, pri čemu se jedan skup sastoji od slika niske rezolucije a drugi od slika visoke rezolucije. Zapravo se problem koji autori rješavaju može postaviti potpuno općenito kao problem učenja preslikavanja iz ishodišne domene u ciljnu domenu. Da bi postigli kvalitetnu rekonstrukciju slike visoke rezolucije uz spomenute domene također koriste dva generatora G i G_s, dva diskriminatora D_b i D_t, te nekoliko funkcija cilja: $J_{sadrzaj}$, J_{boja}, $J_{tekstura}$ i J_{tv}. Generator G mapira slike niske rezolucije u slike visoke rezolucije, a generator G_s obratno. Funkcija cilja $J_{sadrzaj}$ motivirana je ranije spomenutim perceptualnim greškama [4],[32] a definirana je u ishodišnoj domeni zahtijevajući da preslikavanje $G_s \circ G$ uspješno rekonstruira sadržaj polazne slike niske rezolucije:

$$J_{sadrzaj} = \frac{1}{const} \left\| \Psi(\boldsymbol{X}) - \Psi(G_s(G(\boldsymbol{X}))) \right\| \tag{2.54}$$

pri čemu je Ψ prednaučeni VGG-net model. Nadalje, da bi osigurali kvalitetnu rekonstrukciju boja, autori uče diskriminator D_b da razlikuje zamućenu verziju pravih $\boldsymbol{Y}$ i generiranih $G(\boldsymbol{X})$ slika. Zamućene slike dobivaju konvolucijom slike s Gaussovom jezgrom $K(\mu_x, \mu_y, \sigma_x, \sigma_y)$. Zapravo želimo da diskriminator nauči razlike u svjetlini, kontrastu te velike razlike u boji bez da pokušava razlikovati teksturu i sadržaj. J_{boja} je definirana kao

tipična funkcija cilja generatora uz dodano zamućivanje:

$$J_{boja} = -\mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log D_b(G(\boldsymbol{X}) * K) \tag{2.55}$$

pri čemu su parametri jezgre K određeni empirijski. Slično kao za boju, diskriminator D_t uči razlikovati teksture. Da bi se uklonio utjecaj boje promatra se pripadna siva (eng. *grayscale*) slika. Za RGB sliku $\boldsymbol{Y}$ s $Gray(Y)$ označimo pripadnu sivu sliku. Tada je $J_{tekstura}$ definirana s:

$$J_{tekstura} = -\mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log D_t(Gray(G(\boldsymbol{X})) \tag{2.56}$$

Da bi osigurali prostornu glatkoću rekonstruirane slike autor dodaju tzv. *total variation* funkciju cilja koja penalizira normu gradijenta rekonstruirane slike $G(\boldsymbol{X})$:

$$J_{tv} = \frac{1}{const} \left\| \nabla_x G(\boldsymbol{X}) + \nabla_y G(\boldsymbol{X}) \right\| \tag{2.57}$$

Konačna funkcija cilja generatora dana je linearnom kombinacijom spomenutih funkcija ciljeva, pri čemu su koeficijenti određeni empirijski. Iako ovaj pristup ne zahtjeva primjere parova niske i visoke rezolucije već manje nadzirani oblik domena, autori uspješno rekonstruiraju tražene slike visoke rezolucije.

Metode bazirane na GAN-ovima daju vizualno ugodne rezultate, što pokazuju i ispitivanja ljudskog mišljenja [3], ipak nedostaju prave metrike koje bi omogućile široku usporedbu pristupa.

Poglavlje 3

Duboke neuronske mreže za super rezoluciju slika

3.1 Duboki konvolucijski modeli

U ovom poglavlju opisujemo i analiziramo duboke konvolucijske modele za problem super rezolucije slika barkodova. Polazimo od potpuno konvolucijskih modela poput [11], [12], [34], [35], [44] razmatrajući poboljšanja u dva smjera. Jedan se odnosi na funkciju cilja, a drugi se odnosi na operaciju uvećanja rezolucije. Kao referentna arhitektura koristi se [40]. Cilj je odgovoriti na pitanje, koja su poboljšanja moguća neovisno o korištenoj arhitekturi što podrazumijeva broj i veličinu konvolucijskih slojeva i filtra, rezidualne i preskočne veze.

Spomenuti polazni radovi koriste srednju kvadratnu grešku (MSE) kao funkciju cilja. Kao što je pokazano u poglavlju 2.5. to vodi na regresijski problem uz probabilističku interpretaciju s Gaussovim parametarskim modelom gdje je uvjetna vjerojatnost slike visoke rezolucije $\boldsymbol{Y}$ uz uvjet dane slike niske rezolucije $\boldsymbol{X}$ dana s:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{Y}; F(\boldsymbol{X}; \boldsymbol{\theta}), \sigma^2 \boldsymbol{I}) \tag{3.1}$$

Budući da je kovarijacijska matrica dijagonalna ovdje smo implicitno pretpostavili da su vrijednosti piksela ciljne slike visoke rezolucije uvjetno nezavisni uz danu sliku niske rezolucije, odnosno pretpostavili smo da vrijedi:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{i,j} p(\boldsymbol{Y}_{i,j}|\boldsymbol{X}) \tag{3.2}$$

Alternativa pretpostavci 3.2 je uvjetna zavisnost piksela ciljne slike što zahtijeva zadavanje uređaja među lokacijama piksela u slici. Ovim pristupom bave se autori u [8] koristeći autoregresijski model PixelCNN [65] za modeliranje uvjetnih zavisnosti. Ako

ipak pretpostavimo da vrijedi 3.2, i Gaussov parametarski model tada kao što je ranije spomenuto izlaz mreže aproksimira vektor očekivanja uz fiksnu kovarijacijsku matricu $\sigma^2 \boldsymbol{I}$, pri čemu se često uzima $\sigma = 1$. Takav Guassov model je unimodalan. To predstavlja problem ukoliko je stvarna distribucija multimodalna. Naime, najbolje što možemo prilikom aprokismacije multimodalne distribucije unimodalnom jest usrednjiti modove. Intuitivno bi to značilo da ukoliko želimo generirati slike zelenih i crvenih jabuka, koje predstavljaju dva moda, tada ne želimo generirati slike smeđih jabuka što bi bila unimodalna aproksimacija spomenutih. Pokušaj rješenja tog problema možemo promatrati neku multimodalnu distribuciju. Kao logični prijedlog nameće se multinomijalna distribucija. Svakom pikselu $\boldsymbol{Y}_{i,j} \in \{0, 1, \ldots, 255\}$ ciljne slike pridružimo vektor dimenzije 256, pri čemu svaka koordinata pridruženog vektora predstavlja vjerojatnost da pripadni piksel ima vrijednost pripadnog indeksa. Tada zahtijevamo da je izlazna mapa značajki $F(\boldsymbol{X}; \boldsymbol{\theta})$ dimenzije $H \times W \times 256$ gdje $H \times W$ dimenzija ciljne slike $\boldsymbol{Y}$. Vjerojatnost da piksel $\boldsymbol{Y}_{i,j}$ ima vrijednost upravo k modeliramo *softmax* funkcijom:

$$\begin{aligned} p(\boldsymbol{Y}_{i,j} = k|\boldsymbol{X}) &= softmax(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,:}) \\ &:= \frac{exp(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,k})}{\sum_{l=1}^{256} exp(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,l})} \end{aligned}$$

Predikciju za ciljni piksel $\boldsymbol{Y}_{i,j}$ dobivamo kao indeks najveće prediktane vjerojatnosti, odnosno kao $\arg\max(softmax(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,:}))$. Ukoliko je vrijednost ciljnog piksela $\boldsymbol{Y}_{i,j}$ jednaka k, tada bi u idealnom slučaju imali $softmax(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,:}) = e_k$, gdje je $e_k \in \mathbb{R}^{256}$ vektor kanonske baze, odnosno vektor koji samo na koordinati k ima 1 a na svim ostalima mjestima 0. Za piksel $\boldsymbol{Y}_{i,j} = k$ označimo s $\bar{\boldsymbol{Y}}(i, j) \in \mathbb{R}^{256}$ vektor koji ima sve 0 osim na k-tom mjestu gdje ima 1. Tada imamo:

$$p(\boldsymbol{Y}_{i,j}|\boldsymbol{X}) = \prod_{k=1}^{256} p(\boldsymbol{Y}_{i,j} = k|\boldsymbol{X})^{\bar{\boldsymbol{Y}}(i,j)_k} \tag{3.3}$$

Za pripadnu log-vjerodostojnost imamo:

$$\begin{aligned} \ln p(\boldsymbol{Y}|\boldsymbol{X}) &= \ln \prod_{i,j} p(\boldsymbol{Y}_{i,j}|\boldsymbol{X}) \\ &= \ln \prod_{i,j} \prod_{k=1}^{256} p(\boldsymbol{Y}_{i,j} = k|\boldsymbol{X})^{\bar{\boldsymbol{Y}}(i,j)_k} \\ &= \sum_{i,j} \sum_{k-1}^{256} \bar{\boldsymbol{Y}}(i, j)_k \ln p(\boldsymbol{Y}_{i,j} = k|\boldsymbol{X}) \\ &= \sum_{i,j} \sum_{k-1}^{256} \bar{\boldsymbol{Y}}(i, j)_k \ln softmax(F(\boldsymbol{X}; \boldsymbol{\theta})_{i,j,:})_k \end{aligned} \tag{3.4}$$

Maksimizacija log-vjerodostojnosti 3.4 ekvivalentna je minimizaciji unakrsne entropije po svim pikselime između ciljne anotacije $\bar{\boldsymbol{Y}}$ i *softmax* predikcije:

$$\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{Y}|\boldsymbol{X}) = \min_{\boldsymbol{\theta}} - \sum_{i,j} \sum_{k-1}^{256} \bar{\boldsymbol{Y}}(i,j)_k \ln softmax(F(\boldsymbol{X};\boldsymbol{\theta})_{i,j,:})_k \tag{3.5}$$

Upravo zadanu entropiju koristimo kao funkciju greške u kasnijim eksperimentima kod modela koji modeliraju multinomijalnu distribuciju.

Mogli bi se pitat jesu li zamućene slike dobivene modelom učenim da minimizira MSE (l_2 norma) posljedica unimodalnosti Gaussovog modela. Odgovor nije jednoznačan. Naime, kao argument za afirmativan odgovor možemo priloži gornju analizu. S druge strane kao empirijski argument protiv možemo promotriti eksperimente koji pokazuju da modeli učeni da miniziraju l_1 normu generiraju oštrije slike [74]. Pri čemu miniziranje l_1 norme vodi na Laplaceov model koji je također unimodalan, što pak navodi na zaključak kako je problem u samoj definiciji Gaussove funkcije.

Spomenuti polazni modeli [11],[12],[34],[35],[44] kao ulaz koriste bikubičnu aproksimaciju ciljne slike visoke rezolucije ili podižu rezolucije koristeći dekonvolucijske slojeve. Kao što je ranije argumentirano, bikubična aproksimacija nije poželjna iz dva razlog. Prvi razlog je praktični i odnosi se na povećanje vremenske složenosti izvođenja, a drugi je empirijski te pokazuje da loše aproksimacije mogu naškoditi kranjem rezultatu. Preostaju modeli koji koriste dekonvolucije, odnosno transponirane konvolucije za povećavanje rezolucije. Problem dekonvolucijskih slojeva su nejednoliki doprinosi ulaznih vrijednosti koji se događaju kada dimenzija jezgre nije djeljiva korakom, te se tada u generiranoj slici javljaju tzv. *checkerboard* artefakti [23]. Kao odgovor na ovaj problem javljaju se pristupi koji nastoje zamijeniti dekonvolucijski sloj [33],[52]. Predloženi $\mathcal{PS}$ [52] (*pixel shuffle*) operator koristi se stoga kao alternativa dekonvolucijskom sloju u mnogim pristupima [38], [40], [50]. Kao što je ranije opisano, za faktor uvećanja $s \in \mathbb{N}$ i ulaznu sliku $\boldsymbol{X}$ rezolucije $H \times W \times C$ potrebno je generirati mapu značajki $\boldsymbol{X}^{izlaz}$ dimenzije $H \times W \times C \cdot s^2$. Tada primjenom $\mathcal{PS}$ operatora na $\boldsymbol{X}^{izlaz}$ dobivamo mapu značajki rezolucije $s \cdot H \times s \cdot W \times C$ prema formuli:

$$\boldsymbol{Y}_{ijk} = \mathcal{PS}(\boldsymbol{X}^{izlaz})_{i,j,k} = \boldsymbol{X}^{izlaz}_{\lfloor i/s \rfloor, \lfloor j/s \rfloor, k \cdot s \cdot mod(j,s) + k \cdot mod(i,s)} \tag{3.6}$$

$\mathcal{PS}$ operator možemo promatrat kao "preslikavanje" iz dubine u prostor, budući da se zapravo radi o preslagivanju mapa značajki uz smanjenje dubine kanala i povećanje rezolucije. Kako ovdje ne koristimo konvolucije s korakom većim od 1 nemamo niti neravnomjernih doprinosa.

3.2 Generativne suparničke mreže

Generativne suparničke mreže za problem super rezolucije možemo opisati na sljedeći način. Slično kao ranije imamo dva igrača koje zovemo generator G i diskriminator D. Zadaća generatora je na temelju slika niske rezolucije generirati slike visoke rezolucije koje su što sličnije slikama koje dolaze iz prirodne distribucije $p_{podaci}(\boldsymbol{Y})$ slika visoke rezolucije. S druge strane diskriminator ima zadaću razlikovati slike koje dolaze iz dviju distribucija, $p_{podaci}(\boldsymbol{Y})$ i p_G. Spomenuti igrači igraju minimax igru:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{Y}\sim p_{podaci}(\boldsymbol{Y})} \log D(\boldsymbol{Y}) + \mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log(1 - D(G(\boldsymbol{X}))) \tag{3.7}$$

Rješenje minimax igre 3.7 je Nashev ekvilibrij. Kao što je pokazano u poglavlju 1.3 uz optimalni generator G koji je u potpunosti naučio generirati slike koje odgovaraju prirodnoj distribuciji, odnosno vrijedi $p_G = p_{podaci}(\boldsymbol{Y})$, optimalni generator svakom primjeru pridruže vjerojatnost 0.5 da dolazi iz prave distribucije. Drugim riječima ne radi razliku između pravih slika visoke rezolucije i generatorovih rekonstrukcija. Cilj name je ostvariti optimalni generator.

Polazne funkcije cilja za diskriminator i generator redom su dane s:

$$J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{Y}\sim p_{podaci}(\boldsymbol{Y})} \log D(\boldsymbol{Y}; \boldsymbol{\theta}_d) - \frac{1}{2}\mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log(1 - D(G(\boldsymbol{X}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d)) \tag{3.8}$$

$$J_g(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) := -\frac{1}{2}\mathbb{E}_{\boldsymbol{X}\sim p_{podaci}(\boldsymbol{X})} \log D(G(\boldsymbol{X}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d) \tag{3.9}$$

pri čemu očekivanje aproksimiramo primjerima za učenje:

$$J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = -\frac{1}{2}\sum_{i=1}^{m} \frac{1}{m} \log D(\boldsymbol{Y}^{(i)}; \boldsymbol{\theta}_d) - \frac{1}{2}\sum_{i=1}^{m} \frac{1}{m} \log(1 - D(G(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d)) \tag{3.10}$$

$$J_g(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) := -\frac{1}{2}\sum_{i=1}^{m} \frac{1}{m} \log D(G(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d) \tag{3.11}$$

Kao u [38] koristimo parove niske i visoke rezolucije $(\boldsymbol{X}, \boldsymbol{Y})$ kako bi definirali rekonstrukcijske funkcije cilja koje će ubrzati konvergenciju generatora te ustabiliti učenje. Promatramo dvije rekonstrukcijske funkcije cilja: MSE i perceptualnu funkciju cilja predloženu u [32]. Srednja kvadratna greška (MSE) između slika visoke rezolucije $\boldsymbol{Y}$ i generiranih slika $G(\boldsymbol{X})$ definirana je s:

$$J_{MSE}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{H \cdot W} \left\| \boldsymbol{Y} - G(\boldsymbol{X}; \boldsymbol{\theta}_g) \right\|_2^2 \tag{3.12}$$

Perceptualna funkcija cilja predložena u [32] bazira se na VGG modelu koji je učen za problem klasifikacije na ImageNet [9] skupu za učenje. Navedena funkcija cilja promatra srednju kvadratnu grešku mapa značajki ciljne slike $\boldsymbol{Y}$ i generirane slike $G(\boldsymbol{X})$ dobivenu iz VGG modela:

$$J_{VGG}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) \sum_{i=1}^{m} \left\| VGG(\boldsymbol{Y}) - VGG(G(\boldsymbol{X}; \boldsymbol{\theta}_g)) \right\|_2^2 \tag{3.13}$$

J_{VGG} penalizira rekonstrukcijcu sadržaja što je posebno bitno za problem super rezolucije barkodova, relaksirajući rekonstrukciju po pikselima, dok J_{MSE} zahtijeva preciznu rekonstrukciju po pikselima. U skladu s promatranima funkcijama testirat ćemo tri varijante funkcije cilja za generator:

$$J_g^{MSE} = J_g + J_{MSE} \tag{3.14}$$

$$J_g^{VGG} = J_g + J_{VGG} \tag{3.15}$$

$$J_g^{MSE+VGG} = J_g + J_{MSE} + J_{VGG} \tag{3.16}$$

Kao što i autori opisuju u [38] predikcije generatora nisu uvijek lokalno stabilne. Lokalana nestablinost zapravo podrazumijeva pojavljivanje artefakata koji zauzimaju samo manja područja u slici. Takvi artefakti se mogu javiti kod područja koja nisu česta, odnosno dovoljno zastupljena u distribuciji primjera za učenje. Prema [39], krivac za ovaj problem je i diskriminator koji daje globalnu predikciju za cijelu sliku, klasificirajući je ili kao pravu ili kao umjetnu, pa ukoliko je slika samo lokalno degradirana može zavarati diskriminator. Kao rješenje tog problema predlaže se PatchGAN [39], koji umjesto globalne klasifikacije radi lokalno klasifikaciju po slikovnim isječcima (eng. *patch*). Izlaz diskriminatora $D(\hat{\boldsymbol{Y}})$ tada nije procjena vjerojatnosti da ulazna slika $\hat{\boldsymbol{Y}} \in \mathbb{R}^{H \times W}$ nije umjetno generirana nego mapa značajki dimenzije $h \times w$ ($h < H$, $w < W$). Pretpostavimo da je $h = s \cdot H$, $w = s \cdot W$ za $s \in \mathbb{N}$, odnosno mapa predikcija dobivena je iz ulazne slike uz faktor smanjenja s. Tada $D(\hat{\boldsymbol{Y}})_{i,j}$ odgovara procjeni vjerojatnosti da je isječak s lokacijom gornjeg lijevog kuta $(k \cdot s, l \cdot s)$ (za neke $k, l \in \{1, \ldots, s\}$) u polaznoj slici isječak is prave slike, odnosno slike iz prirodne distribucije $p_{podaci}(\boldsymbol{Y})$. Veličina isječka ovisi o veličinama konvolucijskih jezgri te broju konvolucijskih slojeva. Sukladno predloženoj reformulaciji diskriminatora modificiramo funkciju cilja 3.10:

$$\begin{aligned} J_d(\boldsymbol{\theta}_d, \boldsymbol{\theta}_g) = &- \frac{1}{2} \sum_{i=1}^{m} \frac{1}{mhw} \sum_{k,l=1}^{h,w} \log D(\boldsymbol{Y}^{(i)}; \boldsymbol{\theta}_d)_{k,l} \\ &- \frac{1}{2} \sum_{i=1}^{m} \frac{1}{mhw} \sum_{k,l=1}^{h,w} \log(1 - D(G(\boldsymbol{X}^{(i)}; \boldsymbol{\theta}_g); \boldsymbol{\theta}_d)_{k,l}) \end{aligned} \tag{3.17}$$

Arhitekture generatora i diskriminatora motivirane su arhitekturama predloženim u [38], izbacujući normalizaciju po grupi kao što je predloženo u [40] budući da ne postoji

problem s konvergencijom mreža. Za povećanje rezolucije u generatoru koristi se ranije opisani $\mathcal{PS}$ (*pixel shuffle*) operator.

Poglavlje 4

Super rezolucija slika barkodova

4.1 Podatkovni skup

Generirani skup parova niske i visoke rezolucije sastoji se od 94639 primjera generiranih u programskom okviru za crtanje i renderiranje Processing. Inicijalno se na bijelu pozadinu postavlja potpuno crni barkod, čime se dobiva binarna slika (crno/bijela) te se potom primjenjuje niz transformacija. Crni barkod dobiven je koristeći Java biblioteku iText PDF za generiranje barkodova prema PDF417 specifikaciji, varirajući dimenzije barkodova i pravokutnika u barkodovima. Enkodirana informacija predstavlja slučajan niz znakova engleskog alfabeta, znamenaka, te interpunkcijskih znakova.

Na binarnu sliku primjenjuju se blage transformacije perspektive i rotacije barkoda. Zatim slijedi niz transformacija koristeći GLSL *shaders* koji zadaju promjene nad svim pikselima kako bi se rekreirale degradacije nastale prilikom slikanja barkodova kamerom mobilnog uređaja u stvarnim uvjetima. Prva transformacija dodaje bljesak (eng. *glare*). Druga transformacija odnosi se na zamućenje koje se rekreira primjenom Gaussove jezgre zamućenja varirajući standardnu devijaciju Gaussove funkcije. Povremeno se varira intenzitet zamućenja unutar iste slike kako bi se simuliralo neravnomjerno zamućenje nastalo zbog naglih pomaka mobilnog uređaja (eng. *motion blure*), te situacije u kojima krajevi barkodova nisu jednako udaljeni od kamere pa je jedan kraj izvan fokusa. U sljedećem koraku dodaje se šum koji simulira šum senzora kamere mobilnog uređaja, te se potom dodaju varijacije svjetline i kontrasta. U konačnici se slika transformira u YC_bC_r prostor boja u kojem je informacija o luminaciji izdvojena u zaseban kanal (Y), a informacije o boji su sadržane u kromatskim kanalima C_b i C_r, te se koristi samo Y kanal.

Slika niske rezolucije dobiva se primjenom bikubnog uzorkovanja uz faktor smanjenja 2, dok se pripadan slika visoke rezolucije dobiva primjenom istog niza transformacija isključujući zamućenje i smanjenje rezolucije. Primjer generirane slike niske rezolucije prikazan slikom 4.1, a pripadna ciljna slika visoke rezolucije prikazana je slikom 4.2.

U postupku učenja na sliku niske rezolucije na slučajan način dodaje se svjetlina u intervalu $[0, 5]$ te se potom slika skalira u interval $[-1, 1]$ što predstavlja ulaz za mrežu.

Skup pravih slika barkodova tvrtke MicroBlink sastoji se od 1023 primjera. Na slikama su vidljivi problemi zamućenja, šuma, te degradacije barkodova nastale zbog bljeska i niske rezolucije.

Slika 4.1: Primjer generirane slike barkoda niske rezolucije.

Slika 4.2: Primjer generirane slike barkoda visoke rezolucije.

4.2 Evaluacijske metrike

Osnovne metrike za evaluaciju rekonstrukcijskih problema slika, te posebno i za problem super rezolucije slika su PSNR (eng. *peak-signal-to-noise-ratio*) i SSIM (eng. *structural similarity index*). Obje metrike definirane su za sivu sliku (eng. *grayscale*).

Inženjerskim rječnikom, PSNR je omjer između maksimalne moguće vrijednosti signala i šuma u rekonstrukciji, izražen u logaritamskoj skali. Za ciljnu sliku $\boldsymbol{Y}$ i njenu aproksimaciju $\hat{\boldsymbol{Y}}$ definiramo PSNR s:

$$PSNR(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) := 10 \log_{10} \frac{MAX_I^2}{MSE(\boldsymbol{Y}, \hat{\boldsymbol{Y}})} \tag{4.1}$$

pri čemu smo s MAX_I označili maksimalnu moguću vrijednost piksela u slici. Ako je $\boldsymbol{Y} \in \{0, \ldots, 255\}^2$ tada je $MAX_I = 255$, a ukoliko je slika skalirana, obično $\boldsymbol{Y} \in [0, 1]^2$, tada imamo $MAX_I = 1$. Iz definicije slijedi da minimizacija MSE direktno maksimizira PSNR.

Međutim mnogi autori [38], [50], [66] primjećuju kako PSNR metrika ne uspjeva uhvatiti razlike u tekturalnim detaljima, što je posljedica definicije metrike po pikselima koja ne uključuje okolinu. Stoga se često koristi SSIM [66] koji je predložen kao poboljšanje PSNR metrike. Zapravo se kao mjera kvalitete aproksimacije $\hat{\boldsymbol{Y}}$ za danu ciljnu sliku $\boldsymbol{Y}$ promatra srednja vrijednost SSIM indeksa po slikovnim isječcima. SSIM za slikovne isječke dimenzije $n \times n$, $\boldsymbol{y}$ iz $\boldsymbol{Y}$ te $\boldsymbol{x}$ iz $\hat{\boldsymbol{Y}}$ definiran je s:

$$\mathrm{SSIM}(\boldsymbol{y}, \boldsymbol{x}) = \frac{(2\mu_{\boldsymbol{y}}\mu_{\boldsymbol{x}} + c_1)(2\sigma_{yx} + c_2)}{(\mu_{\boldsymbol{y}}^2 + \mu_{\boldsymbol{x}}^2 + c_1)(\sigma_{\boldsymbol{y}}^2\sigma_{\boldsymbol{x}}^2 + c_2)} \tag{4.2}$$

gdje su $\mu_{\boldsymbol{y}}$ i $\sigma_{\boldsymbol{y}}$ aritmetička sredina i standardna devijacija isječka $\boldsymbol{y}$ respektivno:

$$\mu_{\boldsymbol{y}} = \frac{1}{n^2}\sum_{i=1}^{n^2} \boldsymbol{y}_i$$

$$\sigma_{\boldsymbol{y}} = \frac{1}{n^2 - 1}\sum_{i=1}^{n^2} (\boldsymbol{y} - \mu_{\boldsymbol{y}})^2$$

uz analogne definicije za isječak $\boldsymbol{x}$. Nadalje s σ_{yx} označen koeficijent korelacije isječaka $\boldsymbol{y}$ i $\boldsymbol{x}$ definiran s:

$$\sigma_{yx} = \frac{1}{n^2}\sum_{i=1}^{n^2} (\boldsymbol{y} - \mu_{\boldsymbol{y}})(\boldsymbol{x} - \mu_{\boldsymbol{x}})$$

Presotaju konstane c_1 i c_2 koje služe za numeričku stabilizaciju izraza, a za koje se tipično uzima $c_1 = (0.01 \cdot \mathrm{MAX}_I)^2$ i $c_2 = (0.03 \cdot \mathrm{MAX}_I)^2$. SSIM mjeri tri komponente kompatibilnosti: intenzitet piksela (l), kontrast (c) i strukturu (s). A možemo ih izrazit kao:

$$l(\boldsymbol{y}, \boldsymbol{x}) = \frac{2\mu_{\boldsymbol{y}}\mu_{\boldsymbol{x}} + c_1}{\mu_{\boldsymbol{y}}^2 + \mu_{\boldsymbol{x}}^2 + c_1}$$

$$c(\boldsymbol{y}, \boldsymbol{x}) = \frac{2\sigma_{\boldsymbol{y}}\sigma_{\boldsymbol{x}} + c_2}{\sigma_{\boldsymbol{y}}^2 + \sigma_{\boldsymbol{x}}^2 + c_2}$$

$$s(\boldsymbol{y}, \boldsymbol{x}) = \frac{\sigma_{yx} + c_3}{\sigma_{\boldsymbol{y}}\sigma_{\boldsymbol{x}} + c_3}$$

gdje je konstanta $c_3 = c_2/2$. Tada SSIM možemo izraziti kao umnožak navedenih komponenti:

$$\mathrm{SSIM}(\boldsymbol{y}, \boldsymbol{x}) = l(\boldsymbol{y}, \boldsymbol{x})c(\boldsymbol{y}, \boldsymbol{x})s(\boldsymbol{y}, \boldsymbol{x})$$

Obično se za evaluciju uzima samo podskup isječaka kako bi se smanjilo vrijeme izvršavanja, te se promatraju isječci dimenzije 8×8 piksela. Vrijednost SSIM indeksa je u rasponu $[-1, 1]$, pri čemu je SSIM indeks jednak 1 ako i samo ako su slike jednake.

Unatoč spomenutim evaluacijskim metrikama kao glavna metrika u ovom radu promatra se broj uspješnih čitanja barkodova aplikacije *PDF417 Barcode Scanner*. Zapravo, ponajviše nas zanima broj slika barkodova koje su postale čitljive nakon primjene razvijenog rješenja super rezolucije.

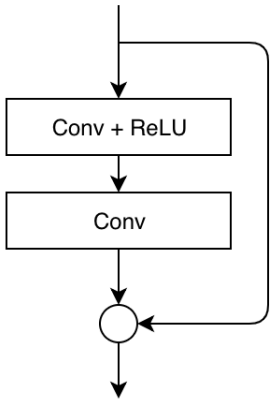
Poglavlje 5

Eksperimenti

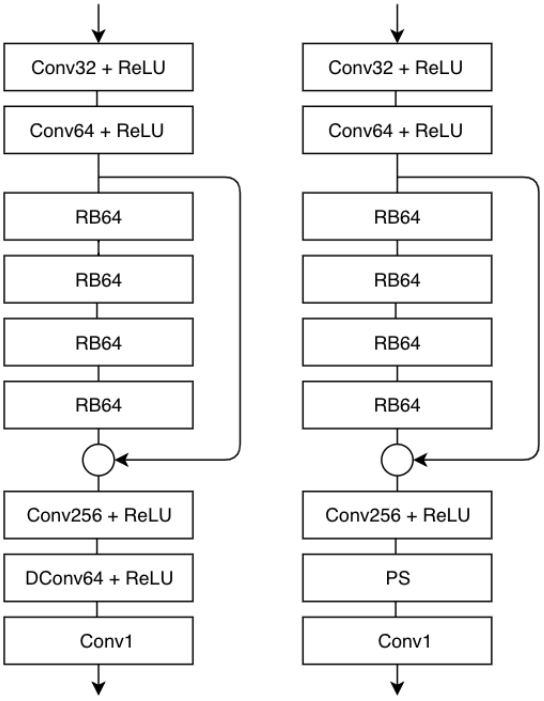
5.1 Arhitektura i učenje modela

U ovom poglavlju iznosimo detalje vezane uz arhitekture i učenja modela. Promatramo dvije konvolucijske arhitekture koje nazivamo *ResDeconv* i *ResSubpixel*, opisane slikom 5.2, a sastoje se od konvolucijskih slojeva (Conv) i rezidualnih veza složenih u rezidualne blokove (RB) [26], te uvećanja rezolucije u dvije varijante: dekonvolucijski sloj (DConv) u *ResDeconv*, te $\mathcal{PS}$ operator (PS) u *ResSubpixel*. Rezidualni blok sastoji se od dva konvolucijska sloja i jedne rezidualne veze kao što je ilustrirano slikom 5.1.

Za spomenute arhitekture promatramo dvije funkcije cilja: srednja kvadratna greška (MSE) i srednja apsolutna greška (MAP). Time dobivamo četiri varijante modela: *Re-*



Slika 5.1: Rezidualni blok predložen u [26].



Slika 5.2: Graf lijevo prikazuje predloženu *ResDeconv* arhitekturu, dok graf desno prikazuje *ResSubpixel* arhitekturu. Broj uz oznaku sloja označava broj filtra u sloju. Conv - konvolucijski sloj, RB - rezidualni blok, DConv - dekonvolucijski sloj, PS - *Pixel Shuffle* operator.

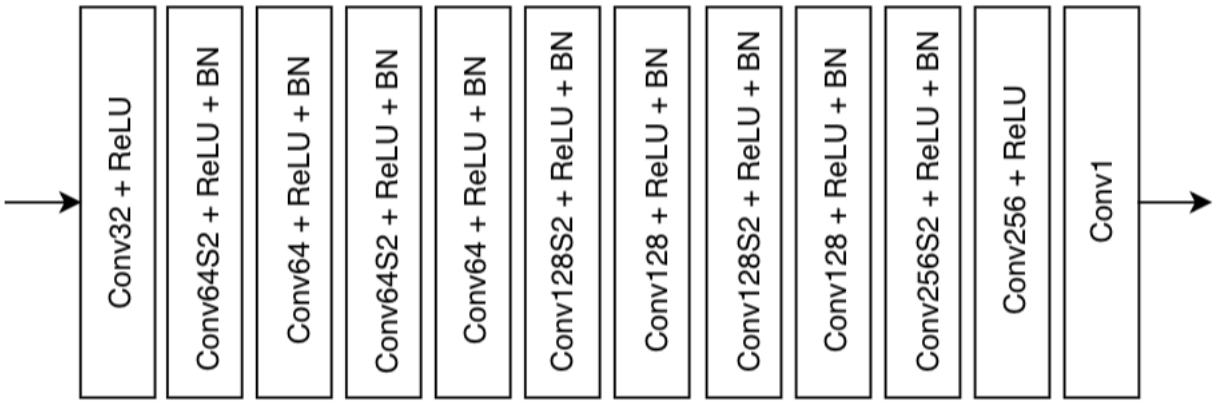
sDeconvMSE, *ResDeconvMAP*, *ResSubpixelMSE* i *ResSubpixelMAP*. Pri čemu su nazivi dobiveni konkatenacijom naziva arhitekture i korištene funkcije cilja.

Za modeliranje multinomialne distribucije koriste se modificirane *ResDeconv* i *ResSubpixel* arhitekture postavljajući izlazni broj kanala u zadnjem sloju na 256. Tako dobiveni modeli označeni su s *ResDeconvMultinomial* i *ResSubpixelMultinomial* respektivno, a učenje se provodi minimiziranjem unakrsne entropije između *softmax* predikcije i proširene anotacije izražene kanonskim vektorima.

Svi spomenuti modeli učeni su na 1500000 iteracija, uz inicijalnu stopu učenja 0.003, te smanjenje stope učenja svakih 300000 iteracija za faktor 0.1.

U GAN pristupu promatramo tri modela. Kao generator svih modela odabran je *ResSubpixel*, dok je arhitektura diskriminatora motivirana [38] [39], te prikazana slikom 5.3.

Funkcija cilja diskriminatora opisana je u poglavlju 3.2., a razlikuje tri funkcije generatora J_g^{MSE}, J_g^{VGG} i $J_g^{MSE+VGG}$ koje određuju redom tri modela: GAN-MSE, GAN-VGG i



Slika 5.3: Predložena arhitektura diskriminatora, motivirana arhitekturom [38], te pristupom [39]. Conv - konvolucijski sloj, ReLU - *rectified linear unit*, BN - normalizacij po grupi (eng. *batch norm*). Oznaka S u nazivima konvolucija označava korak konvolucije (eng. *stride*).

GAN-MSE-VGG. Pripadne funkcije cilja dane su s:

$$J_g^{MSE} = 0.01 \cdot J_g + \cdot J_{MSE}$$
$$J_g^{VGG} = 0.001 \cdot J_g + 10^{-8} \cdot J_{VGG}$$
$$J_g^{MSE+VGG} = 0.001 \cdot J_g + J_{MSE} + 10^{-8} \cdot J_{VGG}$$

pri čemu su koeficijeni u sumi empirijski određeni.

Prvi korak učenja sastoji se od 100000 iteracija generatora uz stopu učenja od 0.003. U sljedećem koraku se naizmjence optimiziraju funkcije cilja diskriminatora i generatora, uz inicijalnu stopu učenje od 0.0001. Kao i ranije stopa učenja se smanjuje svakih 300000 iteracija. Ukoliko je generator nedovoljno naučen tada diskriminator lako odbacuje generirane slike te učenje divergira. Uz dovoljno naučeni generator potrebno je pažljivo odabrati koeficijente za funkcije J_g, J_{MSE} i J_{VGG} kako nebi samo jedna od funkcija dominirala izrazom.

5.2 Rezultati

U ovom poglavlju prezentirani su rezultati evaluacije naučenih modela. Promatramo modele: *ResDeconvMSE*, *ResDeconvMAP*, *ResSubpixelMSE*, *ResSubpixelMAP ResDeconv-*

Multinomial, *ResSubpixelMultinomial*, GAN-MSE, GAN-VGG i GAN-MSE-VGG, kao što je opisano u poglavlju 5.1.

U tablici 5.1 dana je evaluacija promatranih modela prema PSNR i SSIM metrikama na 10000 primjera generiranih slika iz skupa za testiranje. Primjetno je da modeli koji minimiziraju MSE i MAP postižu najbolje rezultate prema PSNR metrici, iako možemo reći da su i rezultati GAN-MSE i GAN-MSE-VGG usporedivi sa spomenutima. Nadalje, i kod SSIM metrike je vidljiv isti odnos među modelima. Promatramo li modele prema njihovim arhitekturama *ResDeconv* i *ResSubpixel*, tada možemo reći da ne postoji značajna razlika između promatranih arhitektura. Primjer uvećanih slika barkodova generiranog skupa dan je slikom 5.4.

Model	PSNR	SSIM
ResDeconvMSE	23.3829	0.8912
ResSubpixelMSE	23.7304	0.9013
ResDeconvMAP	23.6515	0.9031
ResSubpixelMAP	23.5771	0.9020
ResDeconvMultinomial	20.9385	0.8292
ResSubpixelMultinomial	20.4451	0.8184
GAN-MSE	23.2618	0.9045
GAN-VGG	20.8070	0.8240
GAN-MSE-VGG	23.1526	0.8840
Bikubna interpolacija	17.7694	0.5632

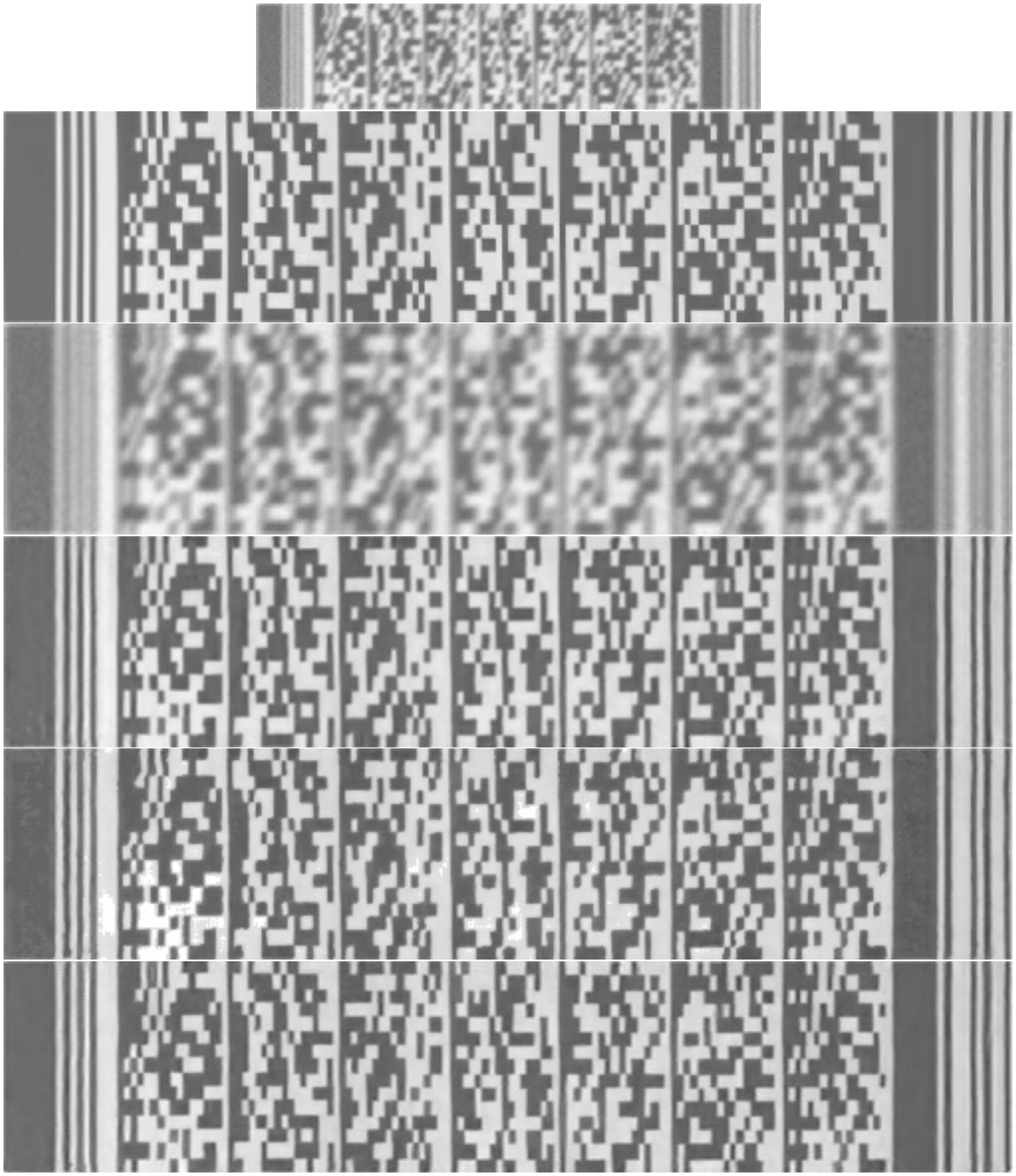
Tablica 5.1: Srednja vrijednost PSNR i SSIM metrikama na 10000 generiranih primjera za testiranje, po modelima.

U tablici 5.2 dana je evaluacija skupa pravih barkodova tvrtke *MicroBlink* koristeći aplikaciju za čitanje barkodova *PDF417 Barcode Scanner*. Tablica prikazuje broj slika barkodova koji su postale čitljive nakon primjene odgovarajućeg modela za super rezoluciju barkodova. Skup se sastoji od 1023 primjera, pri čemu su u velikom broju slika prisutne značajne degradacije prouzročene zamućenjem, šumom i bljeskalicom. Iz tablice je vidljivo da su modeli potpuno usporedivi, iako možemo izdvojit modele koji modeliraju multinomijalnu distribuciju kao nešto uspješnije, odnosno posebno *ResSubpixelMultinomial*. Možemo primjetiti da su rezultati modela koji minimiziraju MSE nešto bolje u odnosu na modele koji minimiziraju MAP, što upučuje na to da je *PDF417 Barcode Scanner* osjetljivi na oštre i nagle skokve vrijednosti piksela u slici koji su posljedica MAP minimizacije. Također, kao i ranije u GAN pristupu se pokazuje da je MSE komponenta

izrazito bitna budući da GAN-VGG model bez te komponente postiže najlošije rezultate mjereći PSNR i SSIM, ali i uspješan broj čitanja na pravim slikama barkodova. U obje evaluacije vidljiva je superiornost pristupa temeljenih na dubokim mrežama u odnosu na bikubnu interpolaciju.

Model	Uspješan broj čitanja
ResDeconvMSE	130
ResSubpixelMSE	142
ResDeconvMAP	114
ResSubpixelMAP	129
ResDeconvMultinomial	135
ResSubpixelMultinomial	149
GAN-MSE	131
GAN-VGG	105
GAN-MSE-VGG	133
Bikubna interpolacija	21

Tablica 5.2: Uspješan broj čitanja barkodova koji nisu bili čitljivi prije primjene odgovarajućeg modela za super rezolucija slika.



Slika 5.4: Redom su dane: slika niske rezolucije, ciljna slika visoke rezolucije, slika uvećana bikubnom interpolacijom, te slike dobivene modelima *ResSubpixelMSE*, *ResSubpixelMultinomial*, GAN-MSE-VGG.

Poglavlje 6

Zaključak

U sklopu ovog diplomskog rada istraženi su pristupi za super rezolucija slika, te je dan širok pregled područja koji uključuje klasične pristupe i pristupe temeljene na dubokim neuronskim mrežama, s posebnim naglaskom na pristupe bazirane na generativnim suparničkim mrežama (GAN).

Kao konkretan problem super rezolucije promotren je problem super rezolucije slika barkodova na skupu slika tvrtke Microblink. Kao što je pokazano evaluacijom, taj skup primjera iznimno je težak zbog značajnog utjecaja degradacijskih procesa kao što su zamućenje, šum i bljesak. Nadalje, poseban izazov predstavljaju različite domene učenja i primjene modela. Naime, potrebno je uložiti veliki napor kako bi se s jedne strane, vjerno simulirali degradacijski procesi koji nastaju u raznim situacijama primjene kamere u stvarnim uvjetima, te s druge strane pokušali razviti pristupi koji su dovoljno robusni na razlike između domena.

Kao moguća poboljšanja predloženog rješenja nameću se dva smjera. U jednom smjeru možemo iterirati kroz novo generiranje parova niske i visoke rezolucije koji bi bolje simulirali degradacijske postupke koji se javljaju u pravim slikama. Takav pristup zahtijeva pažljivo vizualno analiziranje degradacija koje nisu uspješno naučene kako bi se u novoj generaciji skupa za učenje nadomjestile upravo te degradacije. Nadalje potrebne su posebne vještine i alati kako bi se vjerno imitirali spomenuti procesi. Drugi smjer bi nastojao prebaciti učenje modela u skup pravih slika. To je moguće ako su nam dostupna dva skupa primjera, jedan koji uključuje samo prave slike niske rezolucije, te drugi koji sadrži samo prave slike visoke rezolucije. Tada bi mogli koristeći generativne suparničke mreže pokušati naučit model (generator) koji rješava problem super rezolucije. Naravno, nije dovoljno odvojiti primjere samo po rezoluciji već je potrebno zahtijevati da u skupu visoke rezolucije imamo samo primjere koji ne sadrže degradacije. Takva selekcija primjera predstavlja izazov budući da nije uvijek jasno je li slika degradirana ili ne.

U ovom radu demonstrirana je superirornost pristupa temeljenih na dubokim neuron-

skim mrežama u odnosu na interpolacijske metode, ali i ograničenja istih u rekonstrukciji složenih degradacijski postupaka koji se javljaju u svakodnevnoj upotrebi kamere mobilnog uređaja.

Bibliografija

[1] M. Aharon, M. Elad i A. Bruckstein, *K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation*, (2006), https://ieeexplore.ieee.org/document/1710377.

[2] M. Arjovsky, S. Chintala i L. Bottou, *Wasserstein Generative Adversarial Networks*, (2017), http://proceedings.mlr.press/v70/arjovsky17a.html.

[3] Y. Blau i T. Michaeli, *The Perception-Distortion Tradeoff*, (2018), https://arxiv.org/pdf/1711.06077.pdf.

[4] J. Bruna, P. Sprechmann i Y. LeCun, *Super-Resolution with Deep Convolutional Sufficient Statistics*, (2015), https://arxiv.org/abs/1511.05666.

[5] A. Bulat i G. Tzimiropoulos, *Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs*, (2017), https://arxiv.org/abs/1712.02765.

[6] A. Bulat, J. Yang i G. Tzimiropoulos, *To learn image super-resolution, use a GAN to learn how to do image degradation first*, (2018), https://arxiv.org/abs/1807.11458.

[7] H. Chang, D. Yeung i Y. Xiong, *Super-resolution through neighbor embedding*, (2004), http://repository.ust.hk/ir/bitstream/1783.1-2284/1/yeung.cvpr2004.pdf.

[8] R. Dahl, M. Norouzi i J. Shlens, *Pixel Recursive Super Resolution*, (2017), https://arxiv.org/abs/1702.00783.

[9] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li i L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*, (2009), https://ieeexplore.ieee.org/document/5206848.

[10] C. Dong i C. Chage Loy, *Image Super-Resolution Using Deep Convolutional Networks*, (2015), https://arxiv.org/pdf/1501.00092.pdf.

[11] C. Dong, C. Chage Loy, K. He i X. Tang, *Learning a deep convolutional network for image super-resolution*, (2014), https://link.springer.com/chapter/10.1007/978-3-319-10593-2_13.

[12] C. Dong, C. Chage Loy i X. Tang, *Accelerating the Super-Resolution Convolutional Neural Network*, (2016), https://arxiv.org/abs/1608.00367.

[13] W. Dong, L. Zhang, G. Shi i X. Wu, *Image Deblurring and Super-resolution by Adaptive Sparse Domain Selection and Adaptive Regularization*, (2011), https://ieeexplore.ieee.org/document/5701777.

[14] Alpaydin E., *Introduction to Machine Learning: Supervised Learning*, London: The MIT Press, 2004.

[15] S. E. Fahlman, G. E. Hinton i Sejnowski T. J., *Massively Parallel Architectures for Al: NETL, Thistle, and Boltzmann Machines*, (1983), http://www.csri.utoronto.ca/~hinton/absps/fahlmanBM.pdf.

[16] W. T. Freeman, T. R. Jones i E. C. Pasztor, *Example-based super-resolution*, (2002), http://www.merl.com/publications/docs/TR2001-30.pdf.

[17] W. T. Freeman, E. C. Pasztor i O. T. Carmichael, *Learning low-level vision*, (2000), http://people.csail.mit.edu/billf/papers/TR2000-05.pdf.

[18] B. J. Frey, G. E. Hinton i P. Dayan, *Does the wake-sleep algorithm learn good density estimators?*, (1996), https://papers.nips.cc/paper/1153-does-the-wake-sleep-algorithm-produce-good-density-estimators.pdf.

[19] M. Germain, K. Gregor, I. Murray i H. Larochelle, *MADE: Masked Autoencoder for Distribution Estimation*, (2015), https://arxiv.org/abs/1502.03509.

[20] D. Glasner, S. Bagon i M. Irani, *Super-Resolution from a Single Image*, (2009), http://www.wisdom.weizmann.ac.il/~vision/single_image_SR/files/single_image_SR.pdf.

[21] I. Goodfellow, *NIPS 2016 Tutorial: Generative Adversarial Networks*, (2017), https://arxiv.org/pdf/1701.00160.pdf.

[22] K. Gregor i Y. LeCun, *Learning Fast Approximations of Sparse Coding*, (2010), http://yann.lecun.com/exdb/publis/pdf/gregor-icml-10.pdf.

[23] J. Guathier, *Conditional generative adversarial nets for convolutional face generation*, (2015), https://www.foldl.me/uploads/2015/conditional-gans-face-generation/paper.pdf.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin i A. Courville, *Improved Training of Wasserstein GANs*, (2017), https://arxiv.org/abs/1704.00028.

[25] M. Haris, G. Shakhnarovich i N. Ukita, *Deep Back-Projection Networks For Super-Resolution*, (2018), https://arxiv.org/abs/1803.02735.

[26] K. He, X. Zhang, S. Ren i J. Sun, *Deep Residual Learning for Image Recognition*, (2015), https://arxiv.org/abs/1512.03385.

[27] G. Huang, Z. Liu, L. van der Maaten i K. Q. Weinberger, *Densely Connected Convolutional Networks*, (2016), https://arxiv.org/abs/1608.06993.

[28] J. B. Huang, A. Singh i N. Ahuja, *Single Image Super-Resolution from Transformed Self-Exemplars*, (2015), https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Huang_Single_Image_Super-Resolution_2015_CVPR_paper.pdf.

[29] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey i L. Van Gool, *WESPE: Weakly Supervised Photo Enhancer for Digital Cameras*, (2017), https://arxiv.org/abs/1709.01118.

[30] S. Ioffe i C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, (2015), https://arxiv.org/abs/1502.03167.

[31] M. Irani i S. Peleg, *Improving Resolution by Image Registration*, (1991), https://www.cse.unr.edu/~bebis/CS791E/FinalPresPapers/ResolutionIrani.pdf.

[32] J. Johnson i L. Alahi, A. and. Fei-Fei, *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*, (2016), https://arxiv.org/abs/1603.08155.

[33] Y. Kato, S. Ohtani, N. Kuroki, T. Hirose i M Numa, *Image super-resolution with multi-channel convolutional neural networks*, (2016), https://ieeexplore.ieee.org/document/7604756.

[34] J. Kim, J. K. Lee i K. M. Lee, *Deeply-Recursive Convolutional Network for Image Super-Resolution*, (2015), https://arxiv.org/abs/1511.04491.

[35] J. Kim, J. K. Lee i M. K. Lee, *Accurate Image Super-Resolution Using Very Deep Convolutional Networks*, (2016), https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Kim_Accurate_Image_Super-Resolution_CVPR_2016_paper.pdf.

[36] W. S. Lai, J. B. Huang, N. Ahuja i M. H. Yang, *Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution*, (2017), https://arxiv.org/abs/1704.03915.

[37] Y. LeCun, *Generalization and Network Design Strategies*, (1989), http://yann.lecun.com/exdb/publis/pdf/lecun-89.pdf.

[38] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang i W. Shi, *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, (2016), https://arxiv.org/abs/1609.04802.

[39] C. Li i M. Wand, *Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks*, (2016), https://arxiv.org/abs/1604.04382.

[40] B. Lim, S. Son, H. Kim, S. Nah i K. M. Lee, *Enhanced Deep Residual Networks for Single Image Super-Resolution*, (2017), https://arxiv.org/abs/1707.02921.

[41] Z. Lin i Heung Yeung Shum, *Fundamental limits of reconstruction-based superresolution algorithms under local translation*, (2003), https://ieeexplore.ieee.org/document/1261081.

[42] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han i T. S. Huang, *Robust Single Image Super-Resolution via Deep Networks With Sparse Prior*, (2016), https://ieeexplore.ieee.org/document/7466062.

[43] M. Lucic, K. Kurach, M. Michalski, S. Gelly i O. Bousquet, *Are GANs Created Equal? A Large-Scale Study*, (2017), https://arxiv.org/abs/1711.10337.

[44] X. J. Mao, C. Shen i Y. B. Yang, *Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections*, (2016), https://arxiv.org/abs/1606.08921.

[45] T. Miyato, T. Kataoka, M. Koyama i Y. Yoshida, *Spectral Normalization for Generative Adversarial Networks*, (2018), https://arxiv.org/abs/1802.05957.

[46] T. Peleg i M. Elad, *A statistical prediction model based on sparse representations for single image super-resolution*, (2014), https://ieeexplore.ieee.org/document/6739068.

[47] A. Radford, L. Metz i S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, (2015), https://arxiv.org/abs/1511.06434.

[48] M. Renardy i R. C. Rogers, *An Introduction to Partial Differential Equations*, Springer-Verlag New York, 2004.

[49] D. J. Rezende, S. Mohamed i D. Wierstra, *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, (2014), https://arxiv.org/abs/1401.4082.

[50] M. S. M. Sajjadi, B. Scholkopf i M. Hirsch, *EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis*, (2016), https://arxiv.org/abs/1612.07919.

[51] S. Schulter, C. Leistner i H. Bischof, *Fast and accurate image upscaling with super-resolution forests*, (2015), https://ieeexplore.ieee.org/document/7299003.

[52] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert i Z. Wang, *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*, (2016), https://arxiv.org/abs/1609.05158.

[53] A. Shocher, N. Cohen i M. Irani, *"Zero-Shot" Super-Resolution using Deep Internal Learning*, (2017), https://arxiv.org/abs/1712.06087.

[54] K. Simonyan i A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, (2014), https://arxiv.org/abs/1409.1556.

[55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke i A. Rabinovich, *Going Deeper with Convolutions*, (2014), https://arxiv.org/abs/1409.4842.

[56] Y. Tai, J. Yang i X. Liu, *Image Super-Resolution via Deep Recursive Residual Network*, (2017), https://ieeexplore.ieee.org/abstract/document/8099781.

[57] Y. Tai, J. Yang, X. Liu i C. Xu, *MemNet: A Persistent Memory Network for Image Restoration*, (2017), https://arxiv.org/abs/1708.02209.

[58] R. Timofte, V. De i L. Van Gool, *Anchored Neighborhood Regression for Fast Example-Based Super-Resolution*, (2013), https://ieeexplore.ieee.org/document/6751349.

[59] R. Timofte, V. De Smet i L. Van Gool, *A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution*, (2014), https://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_01165.pdf.

[60] R. Timofte, R. Rothe i L. Van Gool, *Seven ways to improve example-based single image super resolution*, (2015), https://arxiv.org/abs/1511.02228.

[61] T. Tong, G. Li, X. Liu i Q. Gao, *Enhanced Deep Residual Networks for Single Image Super-Resolution*, (2017), ImageSuper-ResolutionUsingDenseSkipConnections.

[62] D. Ulyanov, A. Vedaldi i V. Lempitsky, *Deep Image Prior*, (2017), https://arxiv.org/abs/1711.10925.

[63] B. Uria, M. A. Coté, K Gregor, I. Murray i H. Larochelle, *The Neural Autoregressive Distribution Estimator*, (2011), https://arxiv.org/abs/1605.02226.

[64] A. van den Oord, N. Kalchbrenner i K Kavukcuoglu, *Pixel Recurrent Neural Networks*, (2016), https://arxiv.org/abs/1601.06759.

[65] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves i K Kavukcuoglu, *Conditional Image Generation with PixelCNN Decoders*, (2016), https://arxiv.org/abs/1606.05328.

[66] Z. Wand, A. C. Bovik, H. R. Sheikh i E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, (2004), https://ieeexplore.ieee.org/document/1284395.

[67] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung i C. Schroers, *A Fully Progressive Approach to Single-Image Super-Resolution*, (2018), https://arxiv.org/abs/1804.02900.

[68] Z. Wang, D. Liu, J. Yang, W. Han i T. Huang, *Deep Networks for Image Super-Resolution with Sparse Prior*, (2015), https://ieeexplore.ieee.org/document/7410407.

[69] Y. Wenming, Z. Xuechen, T. Yapeng, W. Wei i X. Jing-Hao, *Deep Learning for Single Image Super-Resolution: A Brief Review*, (2018), https://arxiv.org/abs/1808.03344.

[70] C. Y. Yang i M. H. Yang, *AFast Direct Super-Resolution by Simple Functions*, (2013), https://ieeexplore.ieee.org/document/6751179.

[71] J. Yang, J.and Wright, T. S. Huang i Y. Ma, *Image super-resolution via sparse representation*, (2010), https://ieeexplore.ieee.org/document/5466111?arnumber=5466111.

[72] R. Zeyde, M. Elad i M. Protter, *On single image scale-up using sparse-representations*, (2010), https://link.springer.com/chapter/10.1007/978-3-642-27413-8_47.

[73] Y. Zhang, Y. Tian, Y. Kong, B. Zhong i Y. Fu, *Residual Dense Network for Image Super-Resolution*, (2018), https://arxiv.org/abs/1802.08797.

[74] H. Zhao, O. Gallo, I. Frosio i J. Kautz, *Loss Functions for Neural Networks for Image Processing*, (2015), https://arxiv.org/abs/1511.08861.

Sažetak

Računalni vid jedno je od standardnih područja strojnog učenja i umjetne inteligencije. Tipični zadaci računalnog vida mogu se svrstati u dohvat, obradu, analizu i razumijevanje slika. Super rezolucija slika jedan je od postupaka predobrade, kojim se nastoji povećati postojeća rezolucija slike. Često se koristi u obradi satelitskih i medicinskih slika, te kao postupak u različitim multimedijskim aplikacijama, kako bi se riješili problemi zbog prisutnosti šuma, zamagljenih i degradiranih područja. U okviru ovog rada istraženi su pristupi za rješavanje problema super rezolucije uz širok pregled područja, te je implementirano rješenje super rezolucije pomoću dubokih neuronskih mreža.

Summary

Computer vision is one of the standard areas of machine learning and artificial intelligence. Typical computer vision tasks include extraction, processing, analysing and understanding images. Image super resolution is one of the preprocessing steps by which one wishes to enlarge existing image resolution. It is often used in satellite and medical images, but also as processing step in multimedia applications to resolve problems caused by noise, blur, glare and other image degrading processes. Wide area of super resolution methods have been examined within this master's thesis, and an approach based on deep neural networks was implemented.

Životopis

Rođen sam 3. travnja 1994. godine u Zagrebu. Nakon osnovne škole upisujem Srednju školu Jastrebarsko te je završavam sa zvanjem Ekonomist. Potaknut željom za dubljim razumijevanjem matematike 2013. godine upisujem preddiplomski sveučilišni studij Matematike, na Prirodoslovno-matematičkom fakultetu u Zagrebu. Godine 2016. upisujem studij Računarstva i matematike. Tijekom druge polovice diplomskog studija zapošljavam se u tvrtki MicroBlink gdje i danas radim kao softverski inženjer.