

UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Petar Đerković

# Konvolucione mreže super rezolucije

SPECIJALISTIČKI RAD

Podgorica, 2018.

UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

# Konvolucione mreže super rezolucije

SPECIJALISTIČKI RAD

Vještačka inteligencija

Mentor: dr Goran Šuković

Petar Đerković

Studijski program Računarske nauke

Podgorica, Septembar 2018.

## Apstrakt

U radu je predstavljen važan metod mašinskog učenja za problem super rezolucije (SR) slika. Metod uvodi konvolucione mreže kao rješenje problema i poznat je kao SRCNN. Oslanjajući se na rezultate tehnika rijetkog kodiranja, duboka konvoluciona mreža uči preslikavanje između slike manje rezolucije i slike veće rezolucije, tako da kada joj se preda slika manjeg kvaliteta, mreža daje kvalitetniji izlaz. Konvolucione mreže super rezolucije, za razliku od tradicionalnih metoda, optimizuju sve slojeve istovremeno, vodeći računa i o brzini i kvalitetu. SRCNN mogu da rade sa 3 kanala boje istovremeno, i iako imaju jednostavnu strukturu pokazuju vrhunske rezultate restauracije slike kao i najbolje već korišćene tehnike.

## **Abstract**

Here is presented a landmark work, a deep learning method for single image super-resolution (SR), that directly learns an end-to-end mapping between the low and high-resolution images. The mapping is represented as a deep convolutional neural network (CNN) that takes the low-resolution image as the input and outputs the high-resolution one. Furthermore, it has been shown that traditional sparse-coding-based SR methods can also be viewed as a deep convolutional network. Super resolution convolutional network (SRCNN) jointly optimizes all layers demonstrating state-of-the-art restoration quality and achieves fast speed for practical on-line usage. Moreover, it can cope with three color channels simultaneously, and show better overall reconstruction quality.

# Sadržaj

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Uvod</b>                                  | <b>8</b>  |
| <b>2</b> | <b>Osnovni koncepti</b>                      | <b>11</b> |
| 2.1      | Algoritmi SR-a                               | 11        |
| 2.2      | Konvolucione mreže                           | 13        |
| 2.3      | Tehnike dubokog učenja za restauraciju slike | 15        |
| <b>3</b> | <b>Konvolucione mreže super rezolucije</b>   | <b>16</b> |
| 3.1      | Definicija                                   | 16        |
| 3.1.1    | Izvlačenje i reprezentacija segmenata        | 17        |
| 3.1.2    | Nelinearno mapiranje                         | 18        |
| 3.1.3    | Rekonstrukcija                               | 18        |
| 3.2      | Analogija sa tehnikama SC-a                  | 19        |
| 3.3      | Treniranje mreže                             | 22        |
| <b>4</b> | <b>Ogledi</b>                                | <b>25</b> |
| 4.1      | Skup trening-uzoraka                         | 25        |
| 4.2      | Postavke modela                              | 26        |
| 4.2.1    | Broj filtera                                 | 27        |
| 4.2.2    | Veličina filtera                             | 27        |
| 4.2.3    | Dubina mreže                                 | 28        |

|          |   |           |
|----------|---|-----------|
| 4.3      | Poređenja sa drugim tehnikama . . . . . | 30        |
| 4.3.1    | Analiza i ocjene . . . . .              | 31        |
| 4.3.2    | Vrijeme izvršavanja . . . . .           | 32        |
| 4.4      | Kolor-kanali . . . . .                  | 32        |
| <b>5</b> | <b>Poboljšanje modela . . . . .</b>     | <b>36</b> |
| <b>6</b> | <b>Programska realizacija . . . . .</b> | <b>38</b> |
| <b>7</b> | <b>Zaključak . . . . .</b>              | <b>41</b> |
| <b>8</b> | <b>Prilozi . . . . .</b>                | <b>42</b> |
|          | <b>Bibliografija . . . . .</b>          | <b>45</b> |

# Slike

|     |  |    |
|-----|--|----|
| 2.1 | Primjer konvolucije: Na monohromatsku sliku se primjenjuje filter, pri čemu je korak $K = 1$ . . . . . | 13 |
| 2.2 | (Plavi) neuroni KS povezani sa (crvenim) RP: veze putuju kroz dubinu                                   | 14 |
| 3.1 | Vizuelna reprezentacija SRCNN mreže [1] . . . . .  | 19 |
| 3.2 | SC tehnike iz ugla CNN mreže [1] . . . . .   | 20 |
| 3.3 | Rezultati treniranja i testiranja: primjer . . . . .   | 24 |
| 4.1 | Treniranje na većoj bazi popravlja performanse . . . . .   | 26 |
| 4.2 | Veći filteri, bolji rezultati . . . . .  | 27 |
| 4.3 | 3-slojna nasuprot 4-slojne arhitekture . . . . .   | 29 |

|     |  |    |
|-----|--|----|
| 4.4 | Dublje strukture sa većim filterima . . . . .  | 30 |
| 4.5 | Krive konvergencije za testni Set5 . . . . .   | 31 |
| 4.6 | Grafik konvergencije SRCNN-a i DNC-a na Set5 skupu . . . . .   | 31 |
| 4.7 | SRCNN je brz i veoma kvalitetan metod (testni skup: Set14) . . . . .   | 33 |
| 4.8 | Filteri prvog sloja Cb i Cr kanala kod strategije Y-prije: uporediti sa<br>3.3a, filterima Y kanala kod osnovne strategije . . . . . | 35 |
| 5.1 | Vizualizacija prelaza sa SRCNN na FSCRNN . . . . .   | 37 |
| 5.2 | Grafik poboljšanja, test Set14 . . . . .   | 37 |
| 8.1 | Primjer: <i>Lena</i> . SRCNN daje mnogo oštrije i čistije ivice, bez vidnih<br>artefakata po slici. . . . .                          | 44 |

## Tabele

|     |  |    |
|-----|--|----|
| 8.1 | Rezultati treniranja mreže sa različitim brojem filtera . . . . .                                      | 42 |
| 8.2 | Prosječna PSNR vrijednost kolor-kanala pri različitim strategijama<br>(Set5 test.) . . . . .           | 42 |
| 8.3 | Prosječne ocjene metrika - testiranje metoda na BSD200 skupu. Po-<br>debljane najveće ocjene . . . . . | 43 |

# Glava 1

## Uvod

Problem super rezolucije (SR) je jedan od klasičnih problema kompjuterske vizije. To je problem dobijanja visokokvalitetne<sup>1</sup> (engl. high-resolution, skr. HR) slike od slike manje rezolucije (engl. low resolution, skr. LR). Rješavanje ovog problema nije lako zato što za bilo koji piksel početne, niskokvalitetne slike postoji veći broj mogućih ishoda - odnosno, njegovo restauriranje nije jedinstveno i jasno definisano. Težina te nedefinisanosti se prevazilazi tako što se skup mogućih ishoda ograniči nekom ranijom *prior* informacijom. Za učenje te informacije, najmodernije tehnike pribjegavaju strategiji učenja iz uzoraka. One, u zavisnosti od svojih trening-uzoraka, mogu biti definisane za opšti problem super rezolucije, ili dizajnirane da zadovolje neke usko specifične zahtjeve, npr. prepoznavanje lica.

Konvolucione mreže super rezolucije (SRCNN) prvi put su predstavljene u radu [2], i kasnije unaprijeđene u verziji [1]. Motivacija za ovaj metod došla je iz tehnika rijetkog kodiranja (engl. sparse coding, skr. SC). SC tehnike [7] predstavljaju klasu nenadgledanog učenja koje traži što "kraću" reprezentaciju ulaznog vektora  $x$  - odnosno, nalaze (nadkompletni<sup>2</sup>) skup baznih vektora tako da se  $x$  može predstaviti

---

<sup>1</sup>Pojam *kvalitet slike* u ovom radu će imati značenje *rezolucija*

<sup>2</sup>Podskup vektora  $\{\phi_i\}_{i \in J}$  prostora  $X$  predstavlja kompletan skup ako se svaki element prostora  $X$  može predstaviti linearnom kombinacijom vektora  $\{\phi_i\}$ . Takav skup je nadkompletan ako, uklanjanjem  $\phi_j$ , skup  $\{\phi_i\}_{i \in J \setminus \{j\}}$  je i dalje kompletan.



kao linearna kombinacija tih vektora, sa što manje koeficijenata različitih od nule<sup>3</sup>. Pomenute SC tehnike kod SR problema [11] sastoje se od nekoliko koraka obrade (u literaturi, kanal SR obrade): (1) Ulazna slika se dijeli na mnoge, preklapajuće segmente<sup>4</sup> i ti segmenti se preprocesiraju. Faza preprocesiranja je primjena neke funkcije na piskele, obično je oduzimanje srednje vrijednosti i normalizacija. (2) Zatim se ti segmenti kodiraju *rječnicima male rezolucije*. (3) Rijetki koeficijenti se pretvaraju u *rječnike velike rezolucije*, koji služe za rekonstrukciju visokorezolutivnih djelova. (4) Rekonstruisani djelovi se slažu težinskim sumiranjem i dobija se konačni rezultat. Mnogi eksterni metodi uzorkovanja koriste isti ovaj obrazac. Ti metodi najviše pažnje posvjećuju učenju i optimizovanju rječnika, ili pak izgradnji efikasne funkcije mapiranja - kodiranja; ostali koraci se rijetko optimizuju i ne posmatraju se u sklopu neke zajedničke, optimizovane strukture.

Sa druge strane, primijećeno je da je ovaj proces obrade ekvivalentan dubokoj konvolucionoj mreži, i upravo zbog toga razloga došlo se na ideju da se konvolucione mreže primijene na problem SR. Ovaj model, nazvan SRCNN, direktno uči mapiranje s kraja na kraj između nisko LR i visokorezolutivnih HR slika, i u srži se razlikuje od ostalih metoda uzorkovanja - ovaj metod ne uči eksplicitno rječnike niti modelovanje prostora segmenata slike: sve to implicitno biva riješeno od strane skrivenih slojeva. Takođe, koraci izdvajanja segmenata i njihovo ponovno uređivanje su formulisani kao konvolucioni slojevi, pa i oni isto učestvuju u optimizaciji. Sve korake navedene SR obrade SRCNN dobija preko učenja, sa malo pre i postprocesiranja.

Nekoliko karakteristika izdvaja SRCNN metod od ostalih. U prvom redu to je jednostavnost njegove strukture i vrhunska preciznost pri obradi slike. Zatim, ovaj metod je dosta brz - brži je od mnogih metoda zasnovanih na uzorcima zato što ima

---

<sup>3</sup>Ili, sa što manjim brojem koeficijenata koji su što dalji od nule

<sup>4</sup>region, ili *patch* slike

nepovratnu i optimizovanu strukturu. SRCNN sa umjerenim brojem filtera i slojeva postiže veliku brzinu za praktičnu primjenu, čak i ako nemamo GPU. SRCNN može povećati svoj kvalitet restauracije ako se istrenira na većim i različitijim trening-uzorcima i/ili koristi veće filtere. Korišćenje većih baza može postojećim metodama uzorkovanja predstavljati problem, dok to nije slučaj sa SRCNN-om. Još jedna važna osobina mreže jeste to što može operisati na sva tri kanala boje uporedo i tako poboljšati svoje performanse, rezultirajući boljim kvalitetom slike.

U nastavku rada biće predstavljen sami model SRCNN-a, koji pokazuje da tehnike dubokog učenja i konvolucione mreže jesu korisne na polju super rezolucije i da one daju dobre rezultate - i dobar kvalitet i brzinu. Za kraj ostavljen je pregled eksperimenata i rezultati evaluacija i poređenja sa drugim SR tehnikama.

# Glava 2

## Osnovni koncepti

### 2.1 Algoritmi SR-a

Algoritmi za rješavanje problema super rezolucije, SR algoritmi, [10] se mogu podijeliti u 4 grupe: modeli predikcije, ivični modeli, statistički modeli i segmentni metodi uzorkovanja<sup>1</sup>, dobro dokumentovani u [12]. Od svih njih, metodi uzorkovanja postižu najbolje performanse. U toj grupi algoritama razlikujemo interne i eksterne modele uzorkovanja. Interni modeli koriste sličnosti koje postoje u samoj strukturi slike i generišu najbolje prikaze koji hvataju takvu strukturu i koji služe za povećanje kvaliteta. Eksterni modeli zavise od spoljašnjeg skupa uzoraka preko kojeg uče funkciju mapiranja ulaza na izlaz. Ti modeli razlikuju se između sebe u načinu koje reprezentacione šeme slika koriste, koje algoritme učenja rječnika primjenjuju ili da li topološki modeliraju prostor slikovnih elemenata. Kod prvog eksternog algoritma, rječnici su bili predstavljeni parovima LR i HR segmenata; na segmente ulaza se primjenjivao algoritam najbližeg susjeda (engl. nearest neighbor, skr. NN) da nađe njima najbliže u LR rječniku, a zatim bi ti djelovi bili projektovani na svoje HR parove. Alternative za ovakav pristup danas su mnogobrojne. Pristupi kao što

---

<sup>1</sup>koji vrše segmentaciju uzoraka, engl. patch or example based

su regresija jezgra (engl. kernel regression, [13]), randomizirana šuma (engl. random forest, [14]), regresija fiksiranog susjedstva (engl. anchored neighborhood regression, skr. ANR [15]) povećavaju preciznost i brzinu mapiranja. Metodi koji se temelje na rijetkom kodiranju spadaju u najbolje tehnike današnjice i predstavljaju glavnu konkurenciju SRCNN-u. Kod njih, korak ekstrakcije djelova slike i rekonstrukcije obavlja se odvojeno, i njihov primarni fokus je učenje rječnika segmenata slike.

Većina SR algoritama operiše nad samo jednim kanalom boje ili se bave crno-bijelom slikom. Za slike u boji, gorepomenuti metodi prvo transformišu sliku u drugu kolor-šemu (npr. YCbCr ili YUV) i onda obrađuju samo kanal luminiscencije (Y). Metod regresije jezgra, koji je u literaturi poznat kao KK metod, predložen je da istovremeno tretira sve kolor-kanale RGB šeme, koji se zatim stapaju da oforme konačni rezultat. I SRCNN nastoji da riješi problem istovremenog baratanja svim bojama - u glavi Ogledi predstavljeni su rezultati tih nastojanja.

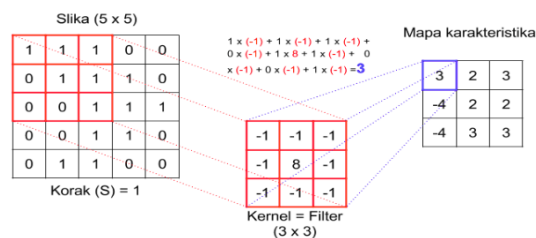
Za analizu algoritama SR-a koriste se evaluacione metrike. One se primjenjuju na njihove rezultate i gleda se odnos originalne i rekonstruisane HR slike. Najznačajne metrike [12] su: odnos signala i šuma - (engl. peak to noise ratio) *PSNR* metrika, metrika indeksa strukturalne sličnosti (engl. structural similarity index)- *SSIM*, kriterijum povjerenja informacije (engl. information fidelity criterion) *IFC*, mjera kvaliteta šuma *NQM* (engl. noise quality measure), težinski PSNR (engl. weighted PSNR, WPSNR) i multiskalni SSIM (engl. multiscale SSIM) - *MSSSIM* metrika. Izuzev IFC i NQM, nabrojani metodi ispitivanja pokazuju jaku korelaciju sa percepcijom - veća ocjena, veći vizuelni ugođaj. Sve ove mjere su iskorišćene u posmatranim ogledima.

## 2.2 Konvolucione mreže

Konvolucione neuronske mreže (engl. convolutional neural network, skr. CNN) predložene su još 1989. godine, i u posljednje vrijeme postale su veoma popularne. Razlog za to krije se u dostignućima koja su ostvarila na polju kompjuterske vizije, kao što je klasifikacija slika i raspoznavanje lica i objekata.

Inspiraciju za svoju struktura CNN mreža našla je u strukturi vidnog korteksa. Ova mreža je projektovana tako da prednost postiže u radu sa 2D strukturama, kao što su slike ili govorni signal, dok najnovije studije ukazuju na značajne rezultate i sa 3D strukturama. Pored ulaznog i izlaznog sloja, nju sačinjava jedan ili više *konvolucionih* slojeva i opciono jedan ili više potpuno povezanih slojeva, kao kod konvencionalnih višeslojnih neuronskih mreža (obično, poslednji sloj je potpuno povezan).

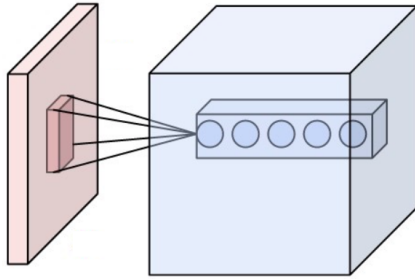
CNN je dobila ime po konvoluciji, široko korišćenom operatoru, koji se primjenjuje pri obradi slika i signala. Njen najznačajniji sloj je konvolucioni sloj (engl. convolutional layer, skr. KS) gdje se primjenjuju konvolucioni filteri. Konvolucioni filter, ili jezgro (kernel) konvo-



Slika 2.1: Primjer konvolucije: Na monohromatsku sliku se primjenjuje filter, pri čemu je korak  $K = 1$ .

lucije, je generalizovani linearni model za region ili segment slike (engl. image patch) na koji se primjenjuje (Slika 2.1). Filter se predstavlja dvodimenzionalnom matricom čije su dimenzije male u odnosu na sliku na koju se primjenjuje i sastoji se od realnih vrijednosti. Ideja kod CNN-a jeste da se postavi veći broj slojeva za otkrivanje bitnih osobina ulaznih podataka. U skladu sa tom idejom, konvolucioni filteri se primjenjuju na sliku kako bi se izvukle značajne karakteristike i kreirale njihove mape (engl.

feature maps). Mreža upravo uči te mape bitnih odlika<sup>2</sup>. Veličina mapa opada sa slojevima, odnosno dubinom mreže.



Slika 2.2: (Plavi) neuroni KS povezani sa (crvenim) RP: veze putuju kroz dubinu

Kod konvolucione mreže nijesu svi slojevi potpuno povezani, jer to bi dovelo do prevelike kompleksnosti izračunavanja. Neuroni su raspoređeni u 3D strukturu ( $\check{S} \times V \times D$ ) i povezuju se samo sa određenim regionom prošlog sloja, koji se naziva receptivnim poljem  $RP$  (engl. receptive field, slika 2.2), i to je primjer lokalne povezanosti neurona. Mreža ima osobinu *dijeljenja težina* u konvolucionim slojevima - što znači da se isti fil-

teri primjenjuju na svako receptivno polje u sloju. Filteri konvoluiraju po  $\check{S} \times V$ , i putujući kroz dubinu, stvaraju 2D aktivacione mape. Puni izlaz KS su združene mape svih filtera raspoređene po dubini.

Kreatori CNN mreže postavljaju veličinu filtera - modifikovanjem filtera, modifikuje se broj parametara mreže. Dva važna parametra koja su isto definisana od strane kreatora jesu - korak (pomjeraj, engl. stride) i dopuna (engl. padding). Korak  $K$  se odnosi na korak pomjeraja filtera preko slike. Kako bi se izvukle karakteristike nižeg nivoa (engl. low-level features), kao što su pikseli slike, cilj je da se u prvim slojevima CNN-a očuva što veći broj informacija. S tim ciljem, se uvodi drugi pojam – dopuna ( $P$ ). Parametar  $P$  se odnosi na oivičenje koje se sastoji od konstantnih vrijednosti piksela. Najčešća vrijednost piksela kojima se ovičava - popunjava, širi matrica - je nula, odnosno zero-padding. Ovo su tkz. hiperparametri, pored stope učenja, receptivnog polja  $RP$ , dimenzija mreže i ulaza. Parametri koji utiču na veličinu izlaza su dubina ulaza, korak i dopuna.

---

<sup>2</sup>Koriti se i termin *korisni obrazac*

Analiza konvergencije mreža pokazala je da je funkcija ReLU Rectified Linear Unit  $f(x) = x^+ = \max(x, 0)$  odličan izbor za aktivacionu funkciju.

Sve bolje i moćnije grafičke kartice i BigData pokret doprinio je tome da CNN svakodnevno poboljšavaju svoje performanse.

## 2.3 Tehnike dubokog učenja za restauraciju slike

Većina tehnika za restauraciju se najviše bave uklanjanjem šuma ili zamućenja. Predlagane su konvolucione mreže ranije koje bi se bavile takvim problemima - npr., jedan takav pristup uključuje konvolucionu mrežu za uklanjanje tragova kiše i nečistoća. Za sami SR problem, vještačke neuronske mreže su primjenjivane npr. u metodu kaskadnih mreža DNC - u okviru kanala SR obrade postavlja se više autoko-dirajućih mreža koje bi imale interni pristup uzorkovanja. Ali ovaj duboki model nije kreiran da obezbijedi mapiranje sa kraja na kraj, jer svaki sloj zahtijeva nezavisnu optimizaciju - procesa traženja sličnosti sa jedne strane, i procesa šifriranja sa druge. Tu SRCNN metod pokazuje svoju superiornost, jer ne samo da obezbjeđuje željeno mapiranje nego i veću brzinu.

## Glava 3

# Konvolucione mreže super rezolucije

### 3.1 Definicija

U ovoj sekciji formuliše se model SRCNN-a. Neka imamo sliku male rezolucije  $X$  i neka je  $s$  faktor skaliranja koliko želimo uvećati našu sliku. Bikubičnom interpolacijom (BI) skalirajmo sliku za faktor  $s$ . Ova interpolacija je takođe konvolucija i može se definisati kao konvolucionni sloj. Ipak, kako bi se iskoristile dobro optimizovane implementacije, ovaj korak se obavlja zasebno i to čini svu fazu preprocesiranja. Interpoliranu sliku označimo sa  $Y$  i nju sada nazivamo "slika male rezolucije". Cilj je dobiti iz  $Y$  sliku  $F(Y)$  koja treba da je u što većoj mjeri slična  $X^+$  - stvarnom "visokorezolutivnom" pandanu slike  $X$ .<sup>1</sup> Odnosno, naš cilj je naći preslikavanje  $F$ , do kojeg dolazimo u tri koraka, koji će biti detaljno objašnjeni:

1. *Dijeljenje slike na segmente i njihovo predstavljanje* - slika  $Y$  se dijeli na mnoge, preklapajuće djelove i svaki dio se predstavlja multidimenzionalnim vektorom. Ovi vektori u stvari predstavljaju skup mapa ili skup odlika, i njihov broj je jednak dimenzionalnosti vektora.
2. *Nelinearno mapiranje* - U ovom koraku svaki vektor se preslikava u drugi

---

<sup>1</sup>Slika  $Y$  se naziva slikom male rezolucije, iako ima istu veličinu kao i  $X^+$



multidimenzionalni vektor. Novodobijeni vektor predstavlja *segment visoke rezolucije*.

3. **Rekonstrukcija** - dobijeni segmenti visoke rezolucije se spajaju, primjenjuje se određena funkcija na okolinu piksela i formira se konačna slika.

Pokazuje se da sve ove operacije formiraju konvolucionu mrežu, koja je predstavljena na slici 3.1.

### 3.1.1 Izvlačenje i reprezentacija segmenata

Gusto segmentiziranje slike i reprezentacija tih segmenata skupom (ranije nađenih) baza je čest korak u algoritmima SR-a. Ovaj korak je ekvivalentan tome da se slika konvoluiru skupom filtera, gdje svaki filter predstavlja neku bazu. Prvi korak kanala obrade SR biće riješen prvim slojem mreže, koji se definiše kao operacija:

$$F_1(Y) = \max(0, W_1 \star Y + B_1), \quad (3.1.1)$$

gdje  $W_1$  predstavlja filtere a  $B_1$  *bias*-e, a  $\star$  predstavlja operaciju konvolucije. Ovdje,  $W_1$  predstavlja  $n_1$  proširenih filtera oblika  $c \times f_1 \times f_1$ , gdje je  $c$  broj kanala slike a  $f_1$  veličina osnovnog filtera. Drugim riječima,  $W_1$  predstavlja  $n_1$  konvoluciju i svaka konvolucija ima veličinu jezgra  $c \times f_1 \times f_1$ . Rezultat je  $n_1$  mapa, te shodno tome,  $B_1$  je  $n_1$ -dimenzioni vektor. Primjećujemo, ReLU funkcija se primjenjuje na krajnje odzive filtera. Ova funkcija može biti izostavljena u ovom koraku, te definisana u okviru drugog sloja nelinearnog mapiranja - prvi sloj bi u tom slučaju postao samo linearna konvolucija.

### 3.1.2 Nelinearno mapiranje

Rezultat prvog sloja je  $n_1$ -dimenzionalna odlika za svaki segment. U ovom koraku, tih  $n_1$ -dim. vektora će biti prevedeno u  $n_2$ -dimenzionalne. Opet, ovo je ekvivalento:

$$F_2(Y) = \max(0, W_2 \star F_1(Y) + B_2). \quad (3.1.2)$$

ako se  $W_2$  filteri primijene na rezultat prvog sloja.  $W_2$  predstavlja u stvari  $n_2$  filtera veličine  $n_1 \times 1 \times 1$ . Ipak, možemo generalizovati i na druge netrivialne prostorne veličine filtera  $f_2 \times f_2$ , kao  $3 \times 3$  ili  $5 \times 5$  - u tom slučaju, nelinearno mapiranje nije na segmentima slike, nego na  $f_2 \times f_2$  segmentima mapa.  $B_2$  je  $n_2$ -dimenzionalni vektor. Dobijeni segmenti operacije drugog sloja biće iskorišćeni za rekonstrukciju.

Dublje strukture, kada se doda još jedan konvolucioni sloj da poveća nelinearnost, od ovoga su istražene i daju iznenađujuće rezultate. Svakako, dublji modeli povećavaju kompleksnost ( $n_2 \times f_2 \times f_2 \times n_2$  parametara samo za još jedan sloj) i zahtijevaju duže vrijeme treniranja.

### 3.1.3 Rekonstrukcija

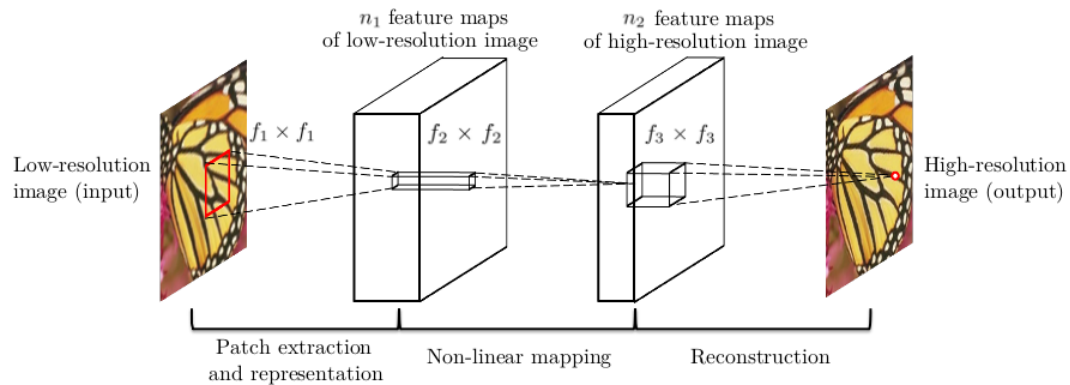
U većini algoritama SR-a, krajnje dobijeni, preklapajući segmenti visoke rezolucije se normalizuju ili usrednjavaju i tek onda se kroji cjelokupna slika. To usrednjavanje može se smatrati isto nekim filtriranjem skupa dobijenih odlika. Stoga, treći sloj biva definisan preko operacije:

$$F_3(Y) = W_3 \star F_2(Y) + B_3, \quad (3.1.3)$$

gdje je  $W_3$  -  $c$  filtera veličine  $n_2 \times f_3 \times f_3$ , a  $B_3$   $c$ -dimenzioni vektor.  $W_3$  se ponaša kao linearni filter. Ako reprezentacije visokorezolutnih segmenata pripadaju

domenu slike (t.j., možemo ih reorganizovati tako da predstavljaju segment slike),  $W_3$  ih usrednjava (u odnosu na okolinu); ako reprezentacije ne pripadaju domenu slike (npr. neki koeficijenti)  $W_3$  će prvo izvršiti projekciju tih koeficijenata na zahtijevani domen i onda izvršiti funkciju usrednjavanja nad njima.

Operacije grupišemo u okviru jedne strukture, odnosno definisani slojevi predstavljaju slojeve konvolucione neuronske mreže. Filtrirajuće težine i bias-ovi su optimizovani u sklopu nje. SRCNN se oslanja na analize i rezultate dobijene u eksperimentima sa tehnikama rijetkog kodiranja, te njima duguje svoju uspješnost.

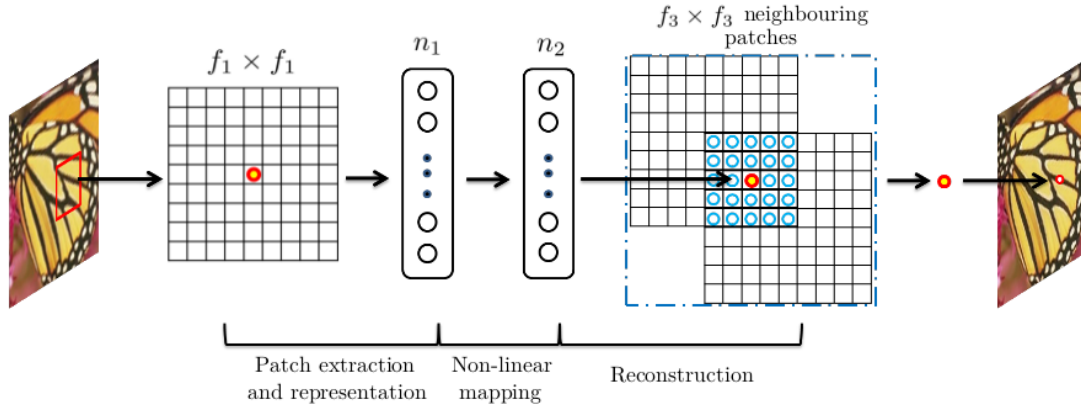


Slika 3.1: Vizuelna reprezentacija SRCNN mreže [1]

## 3.2 Analogija sa tehnikama SC-a

Kao što je već ranije rečeno, tehnike rijetkog kodiranja SC-a za SR problem se mogu posmatrati kao konvoluciona mreža. I zaista, intuitivno, dijele isti kanal. U ovoj sekciji se detaljnije pokazuje zašto je to tako. Neka slika 3.2 posluži kao vizuelna reprezentacija toga. Pretpostavimo da smo u prvom koraku SC tehnike izvukli odgovarajući  $f_1 \times f_1$  segment početne slike male rezolucije. U drugom koraku, algoritam rijetkog kodiranja, kao što je Feature-Sign [7], prevodi taj segment u rječnik niske

rezolucije. Algoritam rijetkog kodiranja se u literaturi naziva i *solver*. Ovim dobijamo  $N$  koeficijenata, t.j. dobijeni rječnik ima veličinu  $N$  - ovo znači da se proces SC algoritma može posmatrati kao primjenjivanje  $N$  linearnih filtera veličine  $f_1 \times f_1$  na ulaz. Oduzimanje srednje vrijednosti koju primjenjuje SC tehnika takođe je linearna operacija, pa isto može biti predstavljena filterima. Pretpostavimo da je  $N = n_1$ .



Slika 3.2: SC tehnike iz ugla CNN mreže [1]

Zatim SC algoritam iterativno prolazi svakim  $n_1$  koeficijentom i obrađuje ih. Izlaz tog algoritma će biti  $n_2$  koeficijenata. Kod SC tehnika, često je  $n_1 = n_2$ . Tih  $n_2$  koeficijenata upravo predstavlja segmente visoke rezolucije, što će reći da se SC algoritam ponaša kao specijalni slučaj operatora nelinearnog mapiranja koji ima trivijalnu osnovnu veličinu  $1 \times 1$ . Da to prevedemo na jezik SRCNN mreže - postavimo da je goredefinisano  $f_2 = 1$  - drugi sloj SRCNN-a uzima tada piksele slike<sup>2</sup> i u ovom slučaju u potpunosti je povezan. Razlika između nelinearnog mapiranja SRCNN-a i algoritma SC tehnike je u (ne)povratnosti. SC algoritam je iterativan, dok je nelinearno mapiranje SRCNN-a u potpunosti unaprijed propagirajuće, bez povratka. Stoga je nelinearni operator mapiranja SRCNN-a efikasan. Analogija solvera kod SRCNN-a su prva dva njena sloja, ne samo drugi. Tokom procesa učenja mreže, operator

<sup>2</sup>Kažemo da je piksel-orijentisani

mapiranja se sam optimizuje.

Nakon rijetkog kodiranja (desni dio slike 3.2), slijedi algoritam prevođenja  $n_2$  koeficijenata u kvalitetne rječnike kako bi se dobili HR segmenti. Dobijeni preklapajući segmenti u ovom koraku se usrednjavaju, i, kao što je pomenuto prije, ovo može biti predstavljeno linearnom konvolucijom (3.1.3) na  $n_2$  dobijenih mapa odlika. Ako kvalitetni segmenti korišćeni za rekonstrukciju imaju veličinu  $f_3 \times f_3$ , onda filteri konvolucije imaju isto prostornu veličinu  $f_3 \times f_3$ .

To pokazuje da su SC i SRCNN tehnike zaista srodne u smislu da obuhvataju strukturu konvolucione mreže (SC tehnike samo sa drugačije definisanom funkcijom nelinearnog mapiranja). Ali ono što je mana SC tehnika, što SRCNN nadoknađuje, jeste to što nijesu svi koraci optimizovani. Svi koraci u okviru SRCNN-a - i dobijanje rječnika i nelinearno mapiranje, sa funkcijama oduzimanja srednje vrijednosti i usrednjavanjem okoline - bivaju uhvaćeni filterima, koji se optimizuju. Zaista se dobija optimizovana struktura mapiranja ulaza na visokokvalitetni izlaz, bez segregacije komponenti.

Ova analogija služi i da se postave hiperparametri,  $n_1, n_2, f_1, f_2$  i  $f_3$ . Obično se uzima da je  $f_3 < f_1$  jer tako se više oslanjamo na centralni dio kvalitetnog segmenta slike. Ako je  $f_3 = 1$  posmatra se samo jedan piksel, te ne postoji usrednjavanje okoline. Takođe,  $n_2 < n_1$ , jer se očekuje *rjeđa*<sup>3</sup> dobijena mapa. Uobičajene postavke su:

$$f_1 = 9, f_2 = 1, f_3 = 5, n_1 = 64 \text{ i } n_2 = 32 \quad (3.2.1)$$

Za ovakvu postavku, za jedan piksel izlaza koristi se informacija od  $(9 + 5 - 1)^2 = 169$  piksela<sup>4</sup>, što je veći broj informacija nego kod postojećih tehnika SC-a (i to duplo

---

<sup>3</sup>predstavlja se manji preko većeg regiona, te manje koeficijenata/filtera

<sup>4</sup>Kažemo da je veličina RP-a  $13 \times 13$

kod nekih) što pruža još jedno objašnjenje zašto SRCNN daje bolje performanse.

### 3.3 Treniranje mreže

Kako bismo našli željenu funkciju  $F$ , koja će nam dati  $F(Y)$ , prvo moramo dovesti mrežu do toga da nauči parametre  $\Theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$ . Ovo se ostvaruje tako što minimalizuje funkcija gubitka između rekonstruisanih  $F(Y; \Theta)$  slika i odgovarajuće, stvarne slike  $X^+$ . Ako je  $\{X_i\}$  skup kvalitetnih slika i skup  $\{Y_i\}$  njima odgovarajuće manje kvalitetne slike, tada funkcija gubitka biće predstavljena sa:

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \|F(Y_i; \Theta) - X_i\|^2, \quad (3.3.1)$$

Ovdje je iskorišćena srednja kvadratna greška (MSE), a  $n$  predstavlja broj trening-uzoraka. Odabir MSE funkcije za loss-funkciju daje veliku PSNR ocjenu. I druge diferencijabilne funkcije se mogu iskoristiti umjesto nje. SRCNN je fleksibilna i u tom smislu što podržava prilagođavanje drugim percepcijom vođenim metrikama. Takva fleksibilnost se teško postiže u drugim metodama. Iako MSE funkcija favorizuje PSNR ocjenu, model SRCNN-a daje zadovoljavajuće rezultate i pri drugim evaluacijama.

Funkciju gubitka minimizujemo stohastičnim gradijentnim spustom, sa standardnom backpropagacijom. Bolje rečeno, težinske matrice se mijenjaju ovako:

$$\Delta_{i+1} = 0.9 \cdot \Delta_i - \eta \cdot \frac{\partial L}{\partial W_i^l}, \quad W_{i+1}^l = W_i^l + \Delta_{i+1}, \quad (3.3.2)$$

gdje  $l \in \{1, 2, 3\}$  je indeks sloja, a  $i$  redni broj iteracije.  $\eta$  je koeficijent učenja, a  $\frac{\partial L}{\partial W_i^l}$  je izvod. Težinski filteri  $W^l$  se inicijalizuju slučajno, sa raspodjelom  $\mathcal{N}(0, 0.001)$ , dok je bias  $B^l$  na početku 0. Koeficijent učenja  $\eta = 10^{-4}$ . Pokazuje se da je bolje odabrati manji koeficijent učenja za zadnji sloj (npr.,  $10^{-4}$  za prva dva sloja,  $10^{-5}$  za

zadnji), kako bi mreža što brže konvergirala.

Slike  $\{X_i\}$  se pripremaju tako što se nasumično "režu" na  $n_{sub} \times n_{sub} \times c$  subslika.<sup>5</sup> Za dobijanje  $\{Y_i\}$  koriste se iste ove subslike, tako što se one prvo zamute Gausovim filterom; umanje za neki faktor skaliranja  $s$ , a onda povećaju za isti faktor koristeći bikubičnu interpolaciju. Tako se dobijaju manje kvalitetne  $Y_i$ .

Ni jedan konvolucioni sloj nema dopunu  $P$ , kako bi se izbjegao efekat dobijanja granica slike<sup>6</sup>. Mreža za  $Y_i$  proizvodi izlaz veličine:

$$(n_{sub} - f_1 - f_2 - f_3 + 3)^2 \times c. \quad (3.3.3)$$

Funkcija gubitka procjenjuje se samo na centralnom pikselu ovog izlaza i  $X_i$ . Iako u toku treninga mreža operiše sa fiksnim veličinama slika, u fazi testiranja mreži može biti predata bilo koja veličina<sup>7</sup>.

Ilustracija primjera: na slici 3.3a možete pogledati naučene filtere prvog sloja [1] kada je mreža trenirana na velikoj bazi ImageNet<sup>8</sup>-a, sa faktorom skaliranja 3, dok na slici 3.3b možete vidjeti izvučene mape. Filteri imaju različite funkcionalnosti: izdvojeni ( $g$ ) i ( $h$ ) ponašaju se kao Gausov ili Laplasov filter, a ( $a$ )-(e) pronalaze ivice slike; ( $f$ ) hvata teksturu, itd.

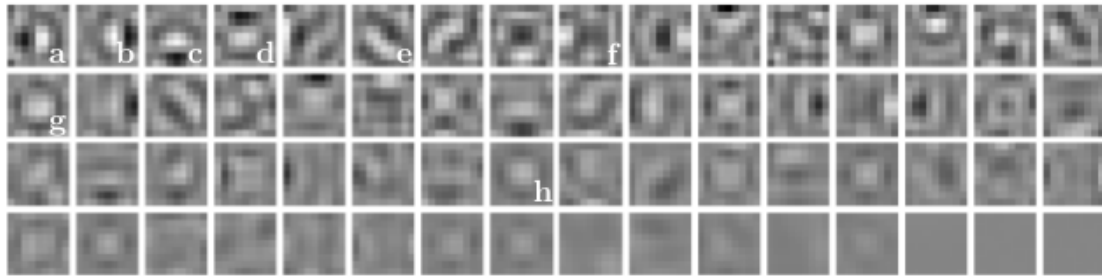
---

<sup>5</sup>Termin „subslika” se ovdje koristi u drugačijem značenju od regiona, segmenata: Slikovni segmenti mogu biti preklapajući djelovi i zahtijevaju neku obradu na kraju, dok subslike ne.

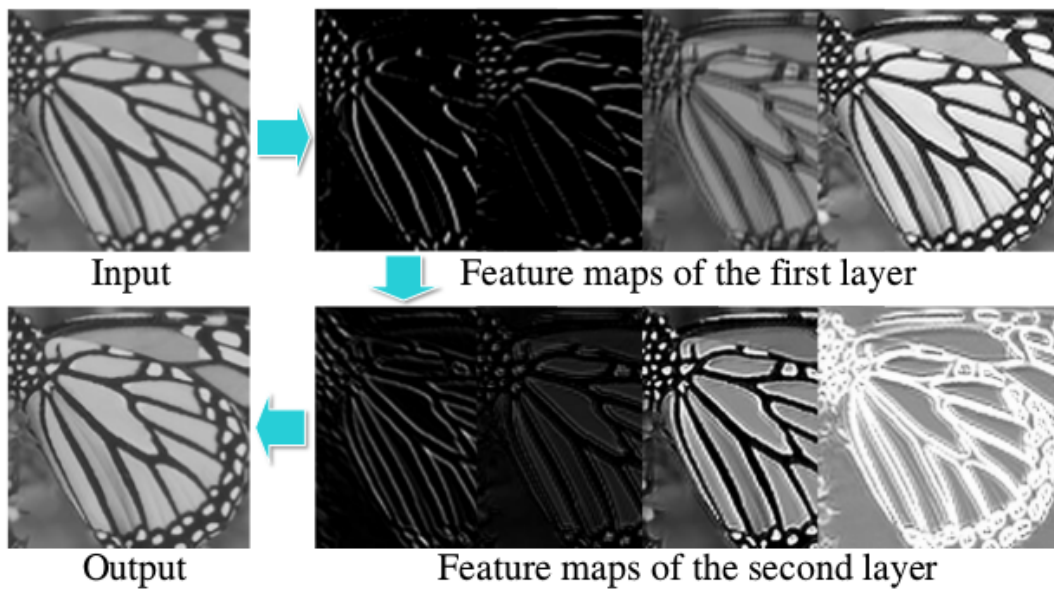
<sup>6</sup>Rezultati kada postoji dopuna se mogu vidjeti na [21]

<sup>7</sup>„bilo koja” veličina ipak zavisi od jačine računara: dostupne memorije i moći grafičke kartice

<sup>8</sup><http://www.image-net.org/>



(a) Naučeni filteri prvog sloja: svaki ima posebnu namjenu



(b) Izvučene mape: šta "vide" filteri ova dva sloja

Slika 3.3: Rezultati treniranja i testiranja: primjer



# Glava 4

## Ogledi

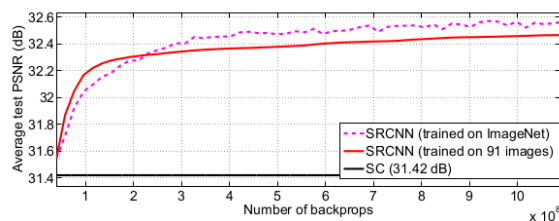
U ovoj sekciji predstavljaju se rezultati eksperimenata koji su sprovedeni sa SRCNN mrežom: kako drugačije trening-baze poboljšavaju performanse i koje rezultate daju drugačije postavke modela. Istraživani su efekti različitih metaparametara  $n_1, n_2, f_1, f_2$  i  $f_3$ , odnosno ispitivani su uticaji promjene veličine i broja filtera. Ogledi koji su sprovedeni sa različitim dubinskim strukturama SRCNN-a otkrivaju iznenađujuće rezultate. Takođe, ostavljen je pregled evaluacija sa različitim metrikama i poređenja sa state-of-the-art tehnikama današnjice, kao i eksperiment sa svim kolor-kanalima. Svi rezultati uzeti su iz titularnog rada autora SRCNN-mreže [1], po njihovom optimizovanom kodu, i biće prikazani u tabelama i graficima sa brojem backpropagacija i PSNR ocjenom.

### 4.1 Skup trening-uzoraka

Tehnike dubokog učenja napredovale su zahvaljujući mnogostrukosti podataka, kao što je ImageNet. Ogled koji se prezentuje u ovom poglavlju zasnivao se na poređenju rezultata koje daje mreža kada uči na malom skupu podataka, nasuprot

učenju na velikoj bazi. Poblje, prva baza sastojala se od 91 slike, skup  $A^1$ , druga od 395909 slika. Sa pomjerajem 14 i  $n_{sub} = 33$ , prvi set slika davao je 24800 subslika, drugi (ImageNet skup) preko 5 miliona i sa korakom rezanja 33. Za ovaj eksperiment koristile su se tipične postavke sistema (3.2.1) i faktor skaliranja  $s = 3$ . Za referentnu tačku koristi se SC-metod [11] koji daje prosječnu vrijednost  $PSNR = 31.42\text{dB}$ . Radi preciznosti ocjene, za test koristila su se dva skupa koji su u literaturi poznati kao *Set5* i *Set14* [21].

Rezultati ovog oglada prikazani su na slici 4.1. Vrijeme za treniranje obje baze je skoro isto, budući da je jednak broj prolaza širenja greške unazad, odnosno backpropagacija. Sa  $8 \cdot 10^8$  backpropagacija,  $PSNR$  vrijednost dobijena na većoj bazi je veća od  $PSNR$  vrijednosti dobijenoj na manjoj ( $32.52\text{dB} > 32.39\text{dB}$ ),



Slika 4.1: Treniranje na većoj bazi popravljja performanse

što pozitivno ukazuje da SRCNN može poboljšati kvalitet obrade na većim BigData podacima. Ipak, rezultat koji je dobijen korišćenjem velike baze nije toliko značajno bolji. Razlog za to - prvi set je uhvatio dovoljnu raznolikost slika.<sup>2</sup>

## 4.2 Postavke modela

U ovim ogledima proučavana je veza između parametara i performansi mreže. Osnovni parametri su mijenjani i posmatrana je mreža u odnosu na rezultate uobičajene konfiguracije. Trening-set je ImageNet, faktor skaliranja  $s = 3$ . Testni set je Set5, i

<sup>1</sup>Folder Basic na sajtu programske realizacije [21]

<sup>2</sup>Opasnost od preobučavanja ne postoji - naša mreža je mala - sveukupno 8032 parametra, tako da se nije mogla pretrenirati na 91 sl. i svih 24800 uzoraka.

rezultati su čitani u trenutku  $8 \cdot 10^8$  backpropagacija, i ukoliko nije drugačije naglašeno, isto važi i za naredne eksperimente.

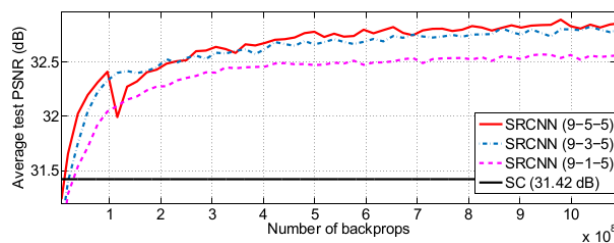
### 4.2.1 Broj filtera

U opštem slučaju, povećanje broja filtera vodi poboljšanju kvaliteta, ali na uštrb vremena. Tabela 8.1 pokazuje kako mijenjanje parametara  $n_1$  i  $n_2$  vodi drugačijim rezultatima. Testirana je mreža sa  $n_1 = 128$  i  $n_2 = 64$  filtera, kao i mreža sa  $n_1 = 32$  i  $n_2 = 16$ . Iako SRCNN sa većim brojem filtera daje kvalitetniji izlaz, mreža sa manjim brojem njih je poželjnija ukoliko se zahtijeva velika brzina rekonstrukcije - ona i dalje daje bolje rezultate od referentnog SC-metoda.

### 4.2.2 Veličina filtera

Osnovna konfiguracija jeste  $f_1 = 9, f_2 = 1$  i  $f_3 = 5$  i neka se takva mreža zove  $9-1-5$  mreža. Kada se posmatrala mreža  $11-1-7$  (veličina filtera drugog sloja nepromijenjena), ona je davala rezultat bolje *PSNR* vrijednosti ( $32.57\text{dB} > 32.52\text{dB}$ ). Ovakav ishod ukazuje na to da i (razumno<sup>3</sup>) veći filteri vode boljim performansama, jer hvataju više strukturalnih informacija.

U drugom dijelu ovog eksperimenta su bile fiksirane veličine  $f_1$  i  $f_3$ , a mijenjana  $f_2$ . Posmatrane su mreže  $9-3-5$  i  $9-5-5$ . Slika 4.2 pokazuje da korišćenje 2-slojnih filtera većih veličina odvodi znatno



Slika 4.2: Veći filteri, bolji rezultati

boljim rezultatima. Jasnije rečeno, dobijene PSNR vrijednosti su  $32.66\text{dB}$  i  $32.75\text{dB}$ ,

<sup>3</sup>imajući na umu povećanje kompleksnosti mreže

za  $f_2 = 3$  i  $f_2 = 5$ . Test sa većim filterima ukazuje na to da je korišćenje okolnih informacija u fazi mapiranja veoma korisno.

Ipak, zbog povećanja kompleksnosti mreže, opada brzina dobijanja output-a: 8032, 24416 i 57184 redom predstavljaju tačan broj parametara za  $9-1-5$ ,  $9-3-5$  i  $9-5-5$  mrežu. Vidimo da je kompleksnost treće mreže skoro dva puta veća od druge, dok je dobijanje na kvalitetu neznatno.

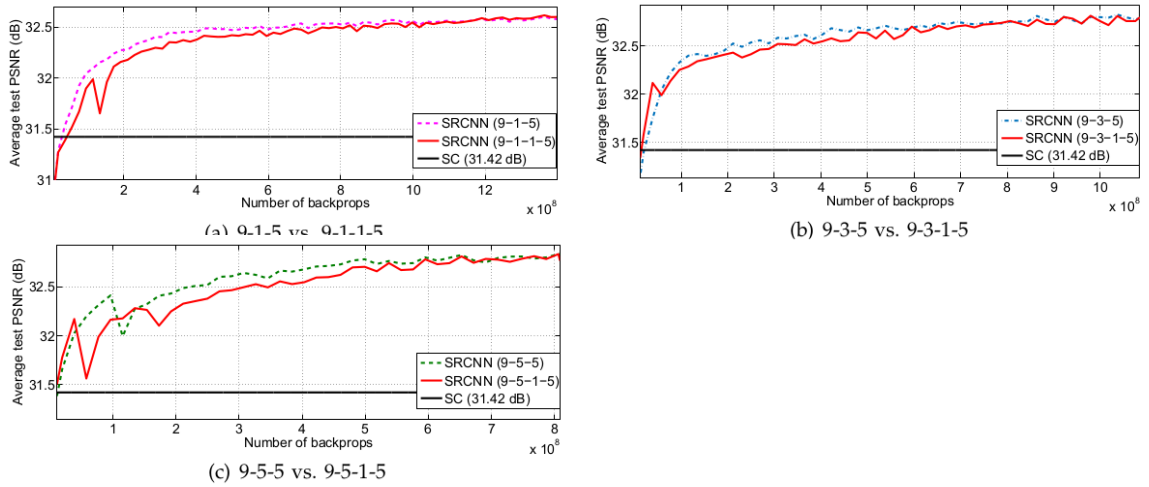
### 4.2.3 Dubina mreže

Studije nad CNN mrežama ukazuju da povećanje broja slojeva dovodi do boljitka u performansama. Ranije posmatrane mreže  $9-1-5$ ,  $9-3-5$  i  $9-5-5$  u ovom ogledu bivaju proširene sa još jednim dodatnim slojem koji ima koeficijent učenja 0.0001 i  $n_{22} = 16$  filtera trivijalne veličine  $1 = f_{22}$ ; njegove težine bivaju isto nasumično inicijalizovane kao i kod ostalih slojeva. Vrijeme koje je potrebno dubljim mrežama da dosegnu konvergenciju je veće (slika 4.3).

Ono što iznenađuje, uprkos ranije objavljenim studijama, jeste da dublje strukture SRCNN ne pokazuju poboljšanje performansi. Ako bi se mreži  $9-1-5$  dodao konvolucioni sloj sada sa  $n_{22} = 32$  filtera i  $f_{22} = 1$ , dolazi do degradiranja i opadanja kvaliteta krajnjeg izlaza, što pokazuje slika 4.4a. Dublje strukture daju još gore rezultate - mreža  $9-1-1-1-5$  sa  $n_{22} = 32$  i  $n_{23} = 16$  filtera (koja ima manji koeficijent učenja kako bi se osigurala konvergencija), i posle nedjelju dana učenja pokazuje pogoršanje. Postuliralo se da li je to možda zbog veličine filtera, ali i to pitanje je riješeno - slika 4.4b pokazuje da ni povećavanje veličine filtera u dodatim slojevima ne dovodi do nekog poboljšanja. Mreže  $9-3-3-3$  i  $9-3-3-5$  ne ostvaruju bolje rezultate od  $9-3-1-5$ .

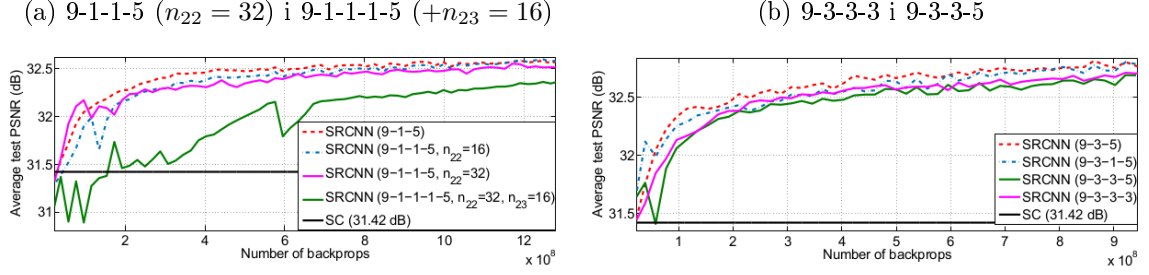
Razlog zašto SRCNN pokazuje tako loše rezultate pri većim dubinama krije se u teškoći učenja tako raslojene strukture. Eksperimenti pokazuju njenu osjetljivost

na mijenjanje vrijednosti parametara i brzine učenja. SRCNN nema sloj sažimanja (engl. pooling layer) koji smanjuje veličinu ulazne slike, a samim tim i broja parametara. Duboke strukture SRCNN, koje broje 4 ili 5 slojeva, imaju poteškoće da stignu u stanje konvergencije, zbog teškoće postavljanja odgovarajućeg koeficijenta  $\eta$ . A ukoliko i dosegnu konvergenciju, naučeni filteri pokazuju manju raznolikost čak i nakon dužeg treniranja, dok i dalje postoji opasnost od upadanja u lokalni minimum. Nepažljivo podešavanje slojevitosti strukture nije jedinstveno za SR problem - i samo polje klasifikacije slika zabilježilo je degradiranje performansi. Duboke arhitekture i njihova analiza još treba da odgonetnu zašto ne vode uvijek boljim rezultatima, te ukažu na bezbjedan način raslojavanja.



Slika 4.3: 3-slojna nasuprot 4-slojne arhitekture

Kako je pokazano da je 3-slojna mreža SRCNN-a najbolja, ona će biti iskorišćena za poređenje sa drugim tehnikama.



Slika 4.4: Dublje strukture sa većim filterima

### 4.3 Poređenja sa drugim tehnikama

Ovdje će biti predstavljene kvalitativne i kvantitativne analize SRCNN-a u odnosu na najsavremenije SR metode. Model SRCNN-a koji je uzet jeste  $9-5-5$  sa  $n_1 = 64$  i  $n_2 = 16$ , treniran na ImageNet-u. Faktori skaliranja koji su razmatrani jesu 2, 3 i 4, i za svaki od njih bila je istrenirana zasebna mreža. Ulaz,  $Y$  slika, svakog od posmatranih metoda dobijao se isto bikubičnom interpolacijom BI. Ocjene kojima su vršene analize nabrojane su u glavi Osnovni koncepti. Više baza uzoraka se koristilo za testiranje metoda, ranije pomenuti Set5, Set14 i novi set od 200 slika BSD200<sup>4</sup>.

Poređenja su se vršila sa sljedećim metodama:

- **SC-tehnika** Yanga i autora [11]
- **NE+LLE** - topološko uranjanje susjeda lokalno linearnim metodom (engl. neighbour embedding + locally linear embedding method) [17]
- **ANR/A+** - regresija fiksiranog susjedstva i poboljšana regresija ([15],[16])
- ranije pomenuta **KK tehnika** - jedan od najbolje ocijenjenih eksternih metoda uzorkovanja[13]

<sup>4</sup>Berkeley Segmentation Dataset: Images

### 4.3.1 Analiza i ocjene

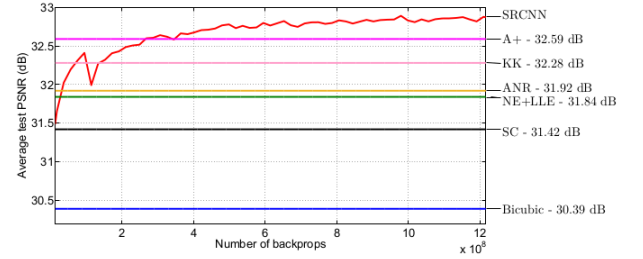
SRCNN daje najbolje rezultate u skoro svim posmatranim metrikama (tabela 8.3)<sup>5</sup>. Za faktor  $s = 3$ , poboljšanja *PSNR*-a, gledano u odnosu na drugi najbolji A+ metod, iznose 0.15, 0.17 i 0.13dB po skupovima. Jedine ocjene koje SRCNN-u ne daju primat jesu IFC i NQM metrika, ali kao što je ranije pomenuto, one ne ocrtavaju stvarni vizuelni kvalitet rezultatne slike.

Sa slike 4.5 se vidi da SRCNN prevazilazi bikubičnu interpolaciju na samom startu, a nakon umjereno dugog vremena treniranja pobjeđuje i najsavremenije metode. Kao i obično, SRCNN može još više da se poboljša većim intervalom treniranja (jer mreža još uvijek nije došla u stanje konvergencije). Slike na sajtu [19] jasnije demonstriraju nadmoćnost SRCNN metoda (pogledati sliku 8.1).

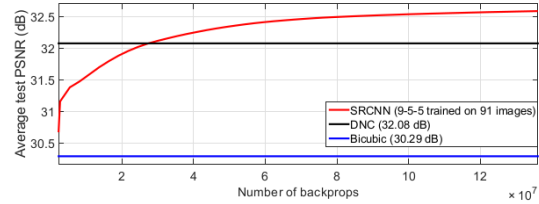
SRCNN je upoređen i sa metodom kaskadnih mreža DNC. U ovom slučaju, treniranje SRCNN-a se vršilo kada se u fazi preprocesiranja primjenjivao drugačiji Gausov filter za zamućenje, sa standardnom devijacijom 0.55, jer isti

koristi DNC. Odabrala se 9-5-5 SRCNN mreža za ovaj ogled i trenirala se skupom A od 91 slike; faktor skaliranja  $s = 3$ ; testni skup je Set5. Slika 4.6 pokazuje da SRCNN vlada i nad ovim metodom, kojeg prestiže sa samo  $2.7 \times 10^7$  backpropagacija. DNC je još jedan primjer strukture kod kojeg dubina ne znači superiornije rezultate.

<sup>5</sup>U radu će biti prikazani samo rezultati na BSD200 skupu; detaljnije [1]



Slika 4.5: Krive konvergencije za testni Set5



Slika 4.6: Grafik konvergencije SRCNN-a i DNC-a na Set5 skupu

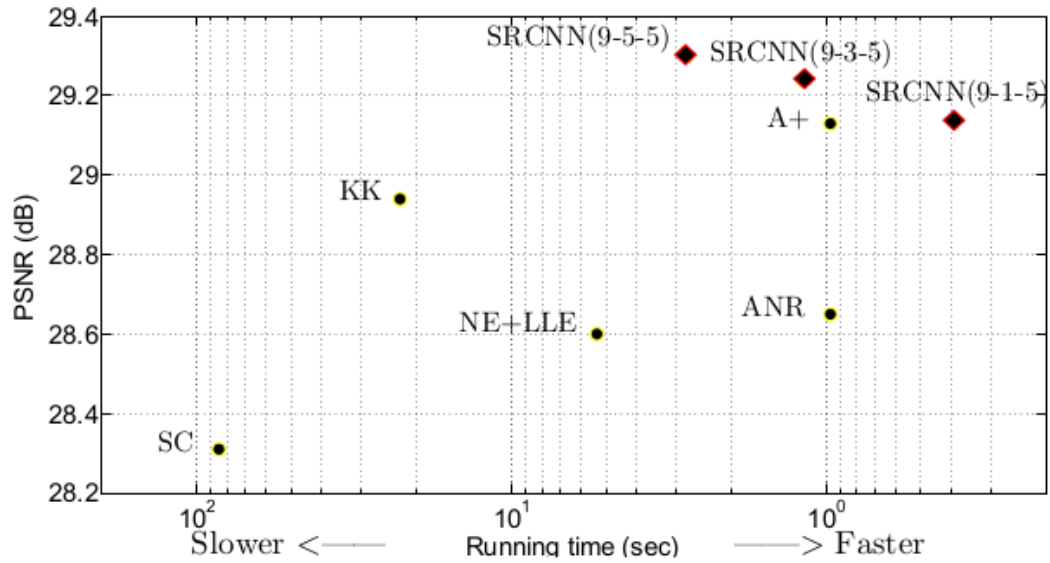
### 4.3.2 Vrijeme izvršavanja

Na slici 4.7 dat je grafik vremena izvršavanja najsavremenijih metoda i SRCNN-a. Računar na kojem je rađeno ima Intelov procesor 3.10 Ghz i 16GB RAM-a; koristila se C++ implementacija SRCNN-a, dok se za druge metode koristila MATLAB+MEX-implementacija samih kreatora. Vrijeme izvršavanja SRCNN-a linearno zavisi od rezolucije početne testne slike, jer sve slike prolaze kroz isti broj konvolucija. Ovaj ogled još jednom je pokazao da se mora vagati između brzine izvršavanja i kvaliteta izlaza SR konvolucione mreže. To pokazuju testovi sa 9-1-5, 9-3-5 i 9-5-5 mrežama na grafikonu 4.7. Najbrži metod svakako je 9-1-5 mreža, dok su drugi sporiji. Iako mreže 9-3-5 i 9-5-5 daju znatno bolje rezultate (najbolji 9-5-5 mreža), one nijesu brže od A+ metoda. Važno je napomenuti da ove razlike u performansama ne potiču zbog različitih implementacija (MATLAB nasuprot C++) nego od samih struktura drugih metoda koje nijesu u cjelosti optimalne (kao kod tehnika rijetkog kodiranja, što je ranije naglašeno). Vrijeme izvršavanja SRCNN-a može biti još više smanjeno, koristeći neke već poznate metode pojednostavljivanja i aproksimacije mreža, ali to možda odvede manjem kvalitetu izlaza.

## 4.4 Kolor-kanali

U svim ranije navedenim eksperimentima slika se transformisala u YCbCr šemu i algoritam se primjenjivao na kanal luminiscencije. Kanali Cb i Cr su skalirani bikubičnom interpolacijom. U ovom eksperimentu posmatrao se učinak primjenjivanja algoritma SRCNN-a na sva tri kanala. Odnosno, postavkom  $c = 3$  mreža SRCNN je modelirana na taj način da prihvati sva 3 kanala boje. Metaparametri sada izgledaju ovako:  $c = 3, f_1 = 9, f_2 = 1, f_3 = 5, n_1 = 64$  i  $n_2 = 32$ . Treniranje se sprovodilo na skupu A, a testiranje na skupu Set5, faktor skaliranja  $s = 3$ .





Slika 4.7: SRCNN je brz i veoma kvalitetan metod (testni skup: Set14)

Za poređenje koristila se bikubična interpolacija i KK, jedan od najmoćnijih kolor SR metoda. Za dobijanje što kompletnijih rezultata analize, drugačiji mehanizmi učenja su primjenjivani, i to:

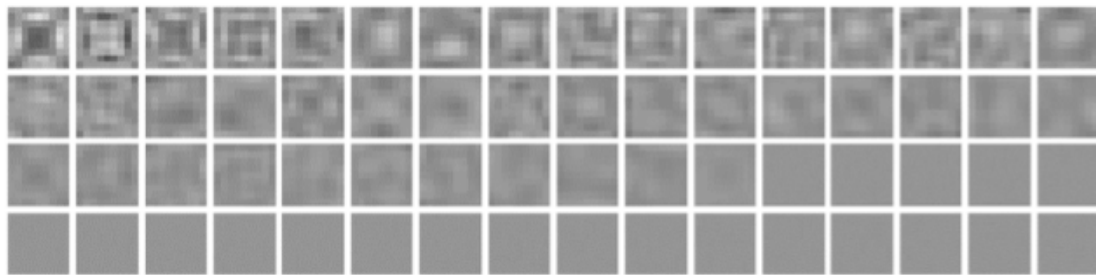
1. Osnovni,  $c = 1$ , ***Y-metod*** - do sada opisan
2. ***YCbCr metod*** - "združeno" treniranje svakog kanala YCbCr
3. ***Predtreniranje Y kanala*** - ovako se obezbjeđuje performansa kanala luminiscencije, tako što se MSE-funkcija u fazi predtreniranja primjenjuje samo na njemu, a onda u glavnoj fazi treninga uče se svi kanali
4. ***Predtreniranje CbCr kanala*** - analogno
5. ***RGB treniranje*** - slika je prevedena u RGB prostor i ti kanali se uče

Tabela 8.2 detaljnije pokazuje rezultate ovog eksperimenta. Vidno je sljedeće:

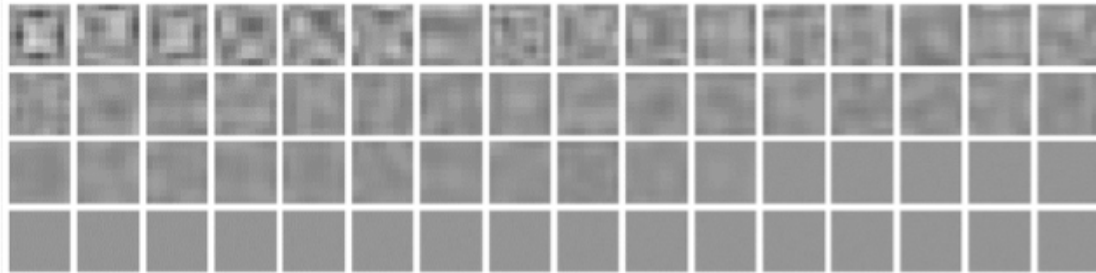
- Treniranje dato 2. strategijom daje lošije rezultate nego bikubična interpolacija, jer mreža upada u lokalni minimum, usljed velikih razlika koje postoje između Y, Cb i Cr kanala.
- 3. i 4. strategija su bolje od 2., ali ipak ne bolje od osnovne 1. strategije, što ukazuje da Cb i Cr kanali mogu degradirati performanse Y kanala.
- Iznenadjujuće, Cb i Cr kanali imaju veću PSNR ocjenu kod 3., nego kod 4. strategije. Razlog zašto je to tako krije se u činjenici da su Cb i Cr kanali "mutniji" od Y kanala, pa nijesu toliko pogođeni preprocesiranjem, a nakon prve faze treniranja na njima, samo mali broj filtera biva promijenjen, što će značiti da u glavnoj fazi treninga mreža zapada u loš lokalni minimum. Sa druge strane, pretreniranje Y kanala aktivira više filtera. Ipak, broj filtera koji je promijenjen nije dovoljno velik da nadmaši osnovnu strategiju. (slika 4.8)
- Konačno, 5. strategija - treniranje na RGB kanalima - pokazuje poboljšanja i daje najbolje rezultate. Za razliku od YCrCb šeme, R, G i B kanali su tijesno povezani i SRCNN pri učenju hvata tu povezanost i koristi je pri rekonstrukciji. RGB strategija na Y kanalu postiže slične rezultate kao Y-samo strategija a bolje rezultate na Cb i Cr kanalima od bikubičnog jezgra, što za rezultat ima kvalitetniju sliku.
- KK metoda je naklonjenija Y-kanalu. Naime, kod KK tehnike algoritam se primjenjuje na sva tri RGB kanala. Kada se rezultat predstavi u YCbCr prostoru, PSNR vrijednost koju ima kanal luminiscencije je slična vrijednost koja se postiže na tom kanalu pri strategiji Y-samo, ali vrijednosti Cb i Cr su gori nego kod BI.

Zaključak je da SRCNN metod treniran na sva tri RGB kanala daje bolje rezultate nego tipična strategija ove mreže i KK tehnika. Ipak to poboljšanje u odnosu na jednokanalni sistem nije veliko (svega 0.07 dB), što još jednom svjedoči da Cb i Cr kanali ne pomažu u optimizovanju performansi.

**Kritike:** Drugi eksperimenti - na poznatom skupu Urban100, čak i na BSD100 skupu - pokazuju da su A+ i KK metoda bolje od SRCNN mreže. [18]



(a) First-layer filters – Cb channel



(b) First-layer filters – Cr channel

Slika 4.8: Filteri prvog sloja Cb i Cr kanala kod strategije Y-prije: uporediti sa 3.3a, filterima Y kanala kod osnovne strategije

## Glava 5

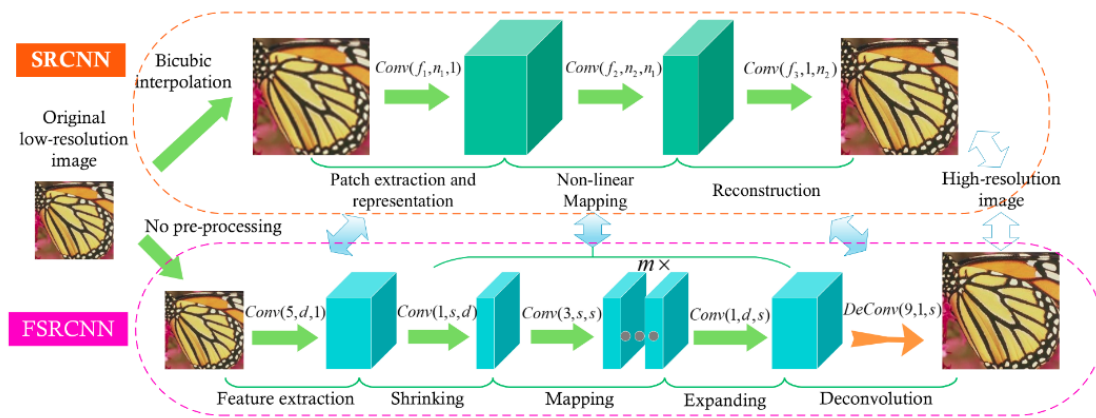
# Poboljšanje modela

U ovoj sekciji razmatra se dalji razvoj, odnosno poboljšanje trenutnog modela SRCNN-a. Jedan takav predlog uslijedio je od samih kreatora SRCNN tehnike. Dong et al. predlažu ubrzani model SRCNN-a kojeg nazivaju FSRCNN. [6]

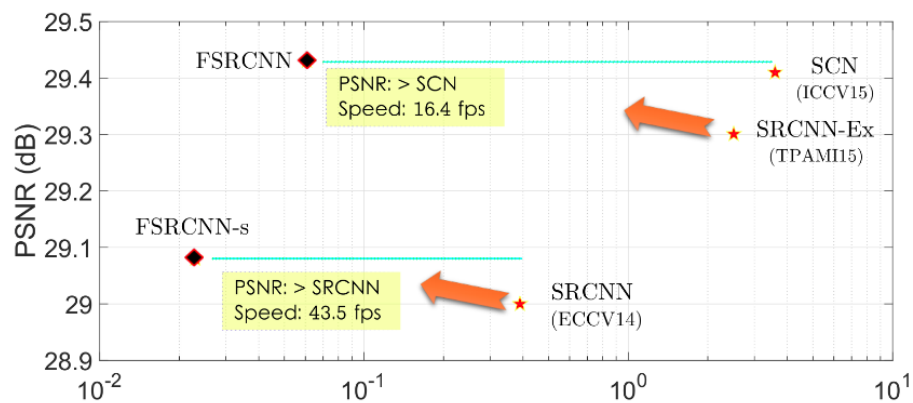
Redizajnirana struktura FSRCNN-a se od originalnog modela razlikuje u 4 aspekta: *(1)* ulaznom sloju FSRCNN-a biva predata originalna slika bez preprocesiranja, odnosno bez bikubične interpolacije, *(2)* na kraju je dodat dekonvolucionni sloj koji izvodi uvećavanje slike, *(3)* sloj nelinearnog mapiranja SRCNN-a grana se na tri koraka kod FSRCNN-a: na sloj smanjivanja (koji smanjuje sliku prije mapiranja), zatim korak mapiranja, i sloj širenja i *(4)* sami sloj mapiranja ima više slojeva i koriste se manji filteri. Iako je struktura dublja, FSRCNN pruža bolje performanse za manji broj izračunavanja - FSRCNN je, u odnosu na prethodni model, ubrzan preko 40 puta sa većom i boljom HR slikom. Autori u novom radu predlažu i nova podešavanja hiperparametara kako bi se osiguralo izvršavanje u realnom vremenu i na opštem procesoru, i dalje vodeći računa o kvalitetu. Nove strategije su predložene za brže treniranje i testiranje za različite faktore  $s$ . Naredne slike ocrtavaju rečeno, dok za detaljniji pregled rezultata poređenja pogledajte [20]. Na slici 5.2, SCN predstavlja mrežu rijetkog kodiranja, SRCNN-Ex duboku SRCNN i FSRCNN-s jednostavniji

model FSRCNN-a.

Predložena je još i tehnika [4] koja koristi višeskalne filtere i drugačiju aktivacionu funkciju, kao i metoda višekanalne SRCNN mreže [8] - koju autori nazivaju MC-SRCNN, gdje se ulaz predstavlja sa više kanala. Razmatrana je i metoda postepenog povećavanja dubine, kod kaskadno treniranih CT-SRCNN mreža [9]. I za ove metode zabilježeno je da pružaju bolje rezultate, ali opsežnije teorijske analize trebaju biti sprovedene.



Slika 5.1: Vizualizacija prelaza sa SRCNN na FSRCNN



Slika 5.2: Grafik poboljšanja, test Set14

## Glava 6

# Programska realizacija

Za programsku realizaciju SRCNN mreže odabran je u `tensorflow`<sup>1</sup>, biblioteka koja omogućava brza matematička izračunavanja. Jedan od glavnih razloga zašto baš taj alat jeste činjenica da pruža veliku podršku mašinskom učenju. Što se same mreže tiče, implementirana je osnovna struktura 9-5-1 i Y-samo strategija, sa bazičnim parametrima, kao i RGB struktura, po glavnom uzoru na MATLAB kod samih autora<sup>2</sup>. Trenirano je na skupu A, za faktor skaliranja  $s = 3$ . Ispitivano je na originalnim skupovima Setu 5 i Setu 14, kao i na nekim novoformiranim (skupovi se nalaze u folderu `Testiranje/`). Mreža je prošla kroz 15000+ za Y-samo strategiju i svega 3000+ epoha za RGB strategiju, u trenutku pisanja rada. Stanja mreže su periodično čuvana na svakih 500 koraka. Mreža pokazuje sjajne rezultate. Cjelokupni projekat, zajedno sa rezultatima testiranja, se mogu vidjeti na mom github nalogu: [21]. (U radu se ipak poziva na same primjere autora mreže jer su ti primjeri mnogo reprezentativniji.)

Prikaz modula dat je na sljedećoj strani.

---

<sup>1</sup><https://www.tensorflow.org/>

<sup>2</sup>Druge reference se nalaze na sajtu mog projekta.

- ***main.py*** - pokretački modul, koji parsira argumente. Korisnik postavlja hiperparametre prilikom pokretanja, npr:

```
$ python3 main.py -training=True -eta=1e-5 -rgb=True
```

pokreće fazu treniranja RGB strategije za koeficijent učenja  $10^{-5}$ . Ako se ne navede ni jedan argument, uzimaju se default vrijednosti. Za više informacija:

```
$ python main.py -help
```

- ***treniranje.py*** - ovaj modul se poziva ako je postavljeno `-training=True` i on obučava mrežu, po instrukcijama koje su postavili autori. Osnovna strategija je Y-samo. Prije treninga, vrši se procesiranje podataka - slika se dijeli na subslike, koje imaju veličinu : `i_size` za LR subslike, `l_size` za ground-truth subslike. Ako se promijeni `i_size`, mora se takođe promijeniti i `l_size` - zbog konvolucije:  $l\_size = i\_size - 12$  (sjetimo se forme (3.3.3)). Periodično se čuvaju stanja, kako bi mreža u sljedećoj trening fazi mogla da nastavi od trenutka gdje je stala. Definiše se jedna *tensor* sesija i u okviru nje se sve odvija. Za svaku epohu, mreži se predaju subslike i poziva se funkcija optimizovanja gubitka, čiju vrijednost možemo da pratimo. Funkcija gubitka računa se za cjelokupni segment, ne samo za centralnu tačku.
- ***testiranje.py*** - ukoliko nije drugačije naglašeno, mreža se testira. Njen folder za testiranje definiše se preko `test_dir`, što može biti i fajl; i svi ti fajlovi treba da se nalaze u folderu `Testiranje/`. Tensor sesija se inicijalizira, čita se sačuvano stanje mreže i vrši se evaluacija ulaza. Izlaz mreže, sa malo postprocesiranja, zajedno se čuva sa slikom koja je dobijena samo bikubičnom interpolacijom, kao referentnom tačkom. Čuvaju se i ocjene za ulazne slike po metrikama PSNR i SSIM. Zaista vidimo da SRCNN pobjeđuje BI metod, po vizuelnom ugođaju

i većini ocjena. Sa dužim vremenom treniranja, SRCNN pokazao bi još bolje performanse. Takođe, ovaj modul omogućava da, ako je korisnik specificirao pri pozivu parametar `-e=True`, da skripta direktno uveća samu testnu sliku za faktor skaliranja  $s$ . Svi izlazi mreže su sačuvani u folderu `rezultati/`.

- ***dodaci.py*** - pomoćni modul, koji sadrži mnoge pomoćne funkcije kao što su nalaženje checkpoint-a ili pretvaranje 3-kanalnog sistema u jednokanalni (izvlačenje kanala  $Y$  kako bi se nad njemu vršila obrada), kao i sama gradnja konvolucione mreže i ispisivanje analize (ocjena).

Kod je teško iskomentarisano da se može pratiti, i projekat sadrži mnoge napomene i primjere.

Za dubljim strukturama i većim filterima se nije posezalo zbog ograničenosti hardvera. Naime, rađeno je na računaru koji nema grafičku karticu, ima Intelov Core i3 procesor i svega 4GB RAM-a<sup>3</sup>. Jedan prolaz kroz cijelu epohu na mom računaru, i pored optimizovanja tensorflow-paketa, iznosio je u prosjeku 96.55sek za `batch=64`, `i_size=33` i `l_size=21`, dok smanjivanjem veličina subslika i batcha, ubrzava se proces (i preko 5 puta, za `i_size=21` i `l_size=9`). Glavna pomoć pri treniranju takođe je bilo korišćenje već postojećih checkpoint-a. Iako se za različiti faktor skaliranja treba trenirati nova mreža<sup>4</sup>, ostavljena je mogućnost pokretanja za drugačije  $s$ , radi upoređivanja rezultata i dobijanja jasnijeg uvida o kvalitetu performansi.

Importovanje tensorflow-a može da potraje. Kod se može još optimizovati, a prirodni njegov razvoj, pored duže faze treniranja, išao bi u smjeru implementacije FSRCNN mreže, koja je, po tvdrnji autora, znatno brža, ili nekih drugih na SRCNN-u baziranih metoda.

---

<sup>3</sup>U stvari, na dva računara ali performanse su slične.

<sup>4</sup>Kao kod uklanjanja šuma, kada se za različiti nivo šuma, treba trenirati nova mreža



# Glava 7

## Zaključak

SRCNN, model konvolucionih neuronskih mreža koji se bavi problemom super rezolucije, predstavlja jedan od ključnih trenutaka i napredaka na polju SR koji je dao izuzetno dobre rezultate. State-of-the-art SRCNN uči mapiranje s kraj na kraj, uzimajući sliku manje rezolucije a dajući HR sliku. Cijela njena struktura je laka, fleksibilna i optimizovana, sa samo par koraka pred i poslije procesiranja. Svakodnevno se istražuju novi filteri i različiti pristupi učenja - više slojeva, duboke rekurzivne strukture, drugačije arhitekture mreže - kako bi se mreža unaprijedila. SRCNN pruža inspiraciju i temelj novim i sve boljim tehnikama.

SRCNN nije striktno vezana za polje SR; ona se može primijeniti i na druge probleme "niske" kompjuterske vizije (engl. low computer vision) kao što su uklanjanje šuma ili zamućenja sa slike, kao i za prepoznavanje lica. Zbog svoje praktičnosti, kvaliteta i brzine, SRCNN nalazi mnoge važne primjene u stvarnom svijetu (npr. [5], i mnoge druge [18]).

# Glava 8

## Prilozi

| $n_1 = 128$<br>$n_2 = 64$ |              | $n_1 = 64$<br>$n_2 = 32$ |              | $n_1 = 32$<br>$n_1 = 16$ |              |
|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|
| PSNR                      | Vrijeme(sec) | PSNR                     | Vrijeme(sec) | PSNR                     | Vrijeme(sec) |
| 32.60                     | 0.60         | 32.52                    | 0.18         | 32.26                    | 0.05         |

Tabela 8.1: Rezultati treniranja mreže sa različitim brojem filtera

| Strategije         | PSNR vrijednost kanala |       |       |           |
|--------------------|------------------------|-------|-------|-----------|
|                    | Y                      | Cb    | Cr    | RGB-slika |
| Bikubična s.       | 30.39                  | 45.44 | 45.42 | 34.57     |
| Y-samo s.          | 32.39                  | 45.44 | 45.42 | 36.37     |
| YCbCr              | 29.25                  | 43.30 | 43.39 | 33.47     |
| Y-pretreniranje    | 32.19                  | 46.49 | 46.45 | 36.32     |
| CbCr-pretreniranje | 32.14                  | 46.38 | 45.84 | 36.25     |
| RGB                | 32.33                  | 46.18 | 46.20 | 36.44     |
| KK                 | 32.37                  | 44.35 | 44.22 | 36.32     |

Tabela 8.2: Prosječna PSNR vrijednost kolor-kanala pri različitim strategijama (Set5 test.)

| Metrika | Faktor $s$ | BI     | SC     | NE+LLE | KK     | ANR    | A+           | SRCNN         |
|---------|------------|--------|--------|--------|--------|--------|--------------|---------------|
| PSNR    | 2          | 28.38  | -      | 29.67  | 30.02  | 29.72  | 30.14        | <b>30.29</b>  |
|         | 3          | 25.94  | 26.54  | 26.67  | 26.89  | 26.72  | 27.05        | <b>27.18</b>  |
|         | 4          | 24.65  | -      | 25.21  | 25.38  | 25.52  | 25.51        | <b>25.60</b>  |
| SSIM    | 2          | 0.8524 | -      | 0.8886 | 0.8935 | 0.8900 | 0.8966       | <b>0.8977</b> |
|         | 3          | 0.7469 | 0.7729 | 0.7823 | 0.7881 | 0.7843 | 0.7945       | <b>0.7971</b> |
|         | 4          | 0.6727 | -      | 0.7037 | 0.7093 | 0.7060 | 0.7171       | <b>0.7184</b> |
| IFC     | 2          | 5.3    | -      | 7.1    | 6.33   | 7.28   | <b>7.51</b>  | 7.21          |
|         | 3          | 3.05   | 2.77   | 3.82   | 3.52   | 3.91   | <b>4.07</b>  | 3.91          |
|         | 4          | 1.95   | -      | 2.45   | 2.24   | 2.51   | <b>2.62</b>  | 2.45          |
| NQM     | 2          | 36.84  | -      | 41.52  | 38.54  | 41.72  | <b>42.37</b> | 39.66         |
|         | 3          | 28.45  | 28.22  | 34.65  | 33.45  | 34.81  | <b>35.58</b> | 34.72         |
|         | 4          | 21.72  | -      | 25.15  | 24.87  | 25.27  | <b>26.01</b> | 25.65         |
| WPSNR   | 2          | 46.15  | -      | 52.56  | 52.21  | 52.69  | 53.56        | <b>53.58</b>  |
|         | 3          | 38.60  | 40.48  | 41.39  | 41.62  | 41.53  | 42.19        | <b>42.29</b>  |
|         | 4          | 34.86  | -      | 36.52  | 36.80  | 36.64  | 37.18        | <b>37.24</b>  |
| MSSSIM  | 2          | 0.978  | -      | 0.9869 | 0.9876 | 0.9872 | 0.9883       | <b>0.9883</b> |
|         | 3          | 0.9426 | 0.9533 | 0.9575 | 0.9588 | 0.9581 | 0.9609       | <b>0.9614</b> |
|         | 4          | 0.9005 | -      | 0.9203 | 0.9215 | 0.9214 | 0.9256       | <b>0.9261</b> |

Tabela 8.3: Prosječne ocjene metrika - testiranje metoda na BSD200 skupu.  
Podebljane najveće ocjene



Slika 8.1: Primjer: *Lena*. SRCNN daje mnogo oštrije i čistije ivice, bez vidnih artefakata po slici.

# Bibliografija

- [1] Dong, C., Loy, C.C., He, K., Tang, X.: Image Super-Resolution Using Deep Convolutional Networks, <https://arxiv.org/abs/1501.00092>, 2015.
- [2] Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution; European Conference on Computer Vision, pp. 184–199 (2014)
- [3] Marko M. Dabović, Igor I. Tartalja: Duboke konvolucijske neuronske mreže – koncepti i aktuelna istraživanja, Zbornik 61. Konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku, ETRAN, 2017.
- [4] Xiaofeng Du, Xiaobo Qu, Yifan He, Di Guo: Single Image Super-Resolution Based on Multi-Scale Competitive Convolutional Neural Network, Sensors, 2018., 18(3), 789, <http://dx.doi.org/10.3390/s18030789>
- [5] Umehara, K., Ota, J., Ishimaru, N., Ohno, S., Okamoto K., Suzuki, T., Shirai, N., Ishida, T.,: Super-resolution convolutional neural network for the improvement of the image quality of magnified images in chest radiographs, SPIE Medical Imaging, 2017, <https://doi.org/10.1117/12.2249969>
- [6] Chao Dong, Chen Change Loy, Xiaoou Tang: Accelerating the Super-Resolution Convolutional Neural Network, <https://arxiv.org/abs/1608.00367>

- [7] Honglak Lee, Alexis Battle, Rajat Raina, Andrew Y. Ng: Efficient sparse coding algorithms, NIPS 2006, <https://papers.nips.cc/>
- [8] Yu Kato, Shinya Ohtani, Nobutaka Kuroki, Tetsuya Hirose, Masahiro Numa: Image super-resolution with multi-channel convolutional neural networks, 14th IEEE International New Circuits and Systems Conference (NEWCAS), 2016.
- [9] Ren, H., El-Khamy, M., Lee, J.: CT-SRCNN: Cascade Trained and Trimmed Deep Convolutional Neural Networks for Image Super Resolution, 2017., <https://arxiv.org/abs/1711.04048>
- [10] Željko Lukač: Algoritmi za interpolaciju uz očuvanje strukture slike, Novi Sad, 2016.
- [11] Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation, IEEE Transactions on Image Processing 19(11), 2861–2873 (2010)
- [12] Yang, C.Y., Ma, C., Yang, M.H.: Single-image super-resolution: A benchmark, European Conference on Computer Vision, pp. 372–386 (2014)
- [13] Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior, IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1127–1133 (2010)
- [14] Schuler, S., Leistner, C., Bischof, H.: Fast and accurate image upscaling with super-resolution forests, IEEE Conference on Computer Vision and Pattern Recognition. pp. 3791–3799 (2015)
- [15] Timofte, R., De Smet, V., Van Gool, L.: Anchored neighborhood regression for fast example-based super-resolution, IEEE International Conference on Computer Vision. pp. 1920–1927 (2013)

- [16] Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution, IEEE Asian Conference on Computer Vision (2014)
- [17] Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding, IEEE Conference on Computer Vision and Pattern Recognition (2004)
- [18] Khizar Hayat, Super-Resolution via Deep Learning, 2017., <https://arxiv.org/abs/1706.09077>
- [19] Dodatak, <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>
- [20] Dodatak, <http://mmlab.ie.cuhk.edu.hk/projects/FSRCNN.html>
- [21] <https://github.com/pdjerkovic/Diplomski>