

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Godec

**Barvanje črnobelih slik z globokimi
modeli**

MAGISTRSKO DELO
MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Blaž Zupan

Ljubljana, 2017

AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja¹.

©2017 PRIMOŽ GODEC

¹V dogоворju z mentorjem lahko kandidat magistrsko delo s pripadajočo izvorno kodo izda tudi pod drugo licenco, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL. V tem primeru na to mesto vstavite opis licence, na primer tekst [?].

ZAHVALA

Na tem mestu zapišite, komu se zahvaljujete za izdelavo magistrske naloge. V zahvali se poleg mentorja spodobi omeniti vse, ki so s svojo pomočjo prispevali k nastanku vašega izdelka.

Primož Godec, 2017

Vsem rožicam tega sveta.

"The only reason for time is so that everything doesn't happen at once."

— Albert Einstein

Contents

Povzetek

Abstract

1	Uvod	1
2	Pregled Področja	5
2.1	Predstavitev slikovnih podatkov in barvni prostori	5
2.2	Obstoječe metode za barvanje črno-belih slik	8
2.3	Globoke nevronske mreže	10
3	Barvanje črno-belih slik z globokimi nevronskimi mrežami	13
3.1	Arhitekture	13
3.2	Pristopi z regresijo	17
3.3	Pristopi s klasifikacijo	19
3.4	Postopek učenja	21
3.5	Podatki	23
3.6	Računanje napake	23
4	Rezultati in diskusija	27
4.1	Primerjava metod na manjši učni množici	27
4.2	Primerjava metod na večji učni množici	30
4.3	Barvanje večjih slik	30
5	Zaključek	31

CONTENTS

A Spearmanova korelacija rangov med metodami	33
B Podrobnosti arhitektur	35

Povzetek

Naslov: Barvanje črnobelih slik z globokimi modeli

V vzorcu je predstavljen postopek priprave magistrskega dela z uporabo okolja L^AT_EX. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavnega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

Ključne besede

Umetna inteligenco, Odkravanje znanj iz podatkov, globoko učenje, nevron-ske mreže

Abstract

Title: Deep models for image coloring

This sample document presents an approach to typesetting your BSc thesis using L^AT_EX. A proper abstract should contain around 100 words which makes this one way too short. A good abstract contains: (1) a short description of the tackled problem, (2) a short description of your approach to solving the problem, and (3) (the most successful) result or contribution in your thesis.

Keywords

Artificial intelligence, data mining, deep learning, neural networks

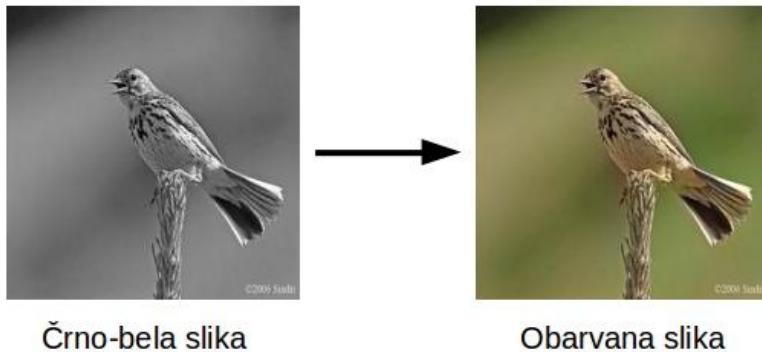
Poglavlje 1

Uvod

Čeprav se so prvo barvno fotografijo naredili že leta 1886 [?], se je barvna fotografija v vsakdanji uporabi uveljavila šele mnogo kasneje. Tako imajo naši stari starši še vedno veliko črno belih fotografij. Ker te prikazujejo realnost povsem drugače, bi jih radi obarvali. Pa je to sploh mogoče?

Ta problem rešujejo algoritmi za barvanje črno-belih slik. Ti dobijo kot vhod črno-belo fotografijo, ki ji na to dodajo barvo, kot je prikazano na sliki 1.1. Algoritmi za barvanje črno-belih slik se uporabljajo na več področjih: barvanje starih slik, barvanje črno-belih filmov in v pomoč pri umetnosti. Za človeka je barvanje črno-belih slik, ki so prikazane na sliki 1.2, enostavna naloga. Z vsakdanjim opazovanjem sveta se je človek naučil, da je nebo modro z belimi oblaki, drevesa so zelena in cesta je siva. Za objekte, ki nimajo enolično določene barve ljudje lahko ugibamo kakšne barve naj bi predmet bil. Pri tem opravilu je potrebno veliko razumevanja, saj iz sivinskih slik ni možno direktno razbrati barv, namreč pri nastanku sivinske slike se veliko informacij izgubi (dve od treh dimenzij).

Problem postane bolj kompleksen, ko ga želimo rešiti na avtomatski način z računalnikom. Pri tem nam je v pomoč dejstvo, da je možno iz tekstur objektov te prepoznati in jim na ta način določiti njihovo barvo. Pri objektih, ki nimajo enolično določeno barv (na primer avtomobili, stavbe in knjige) je izziv še mnogo težji. Pri tem nam delo olajša dejstvo, da ne želimo, da slika

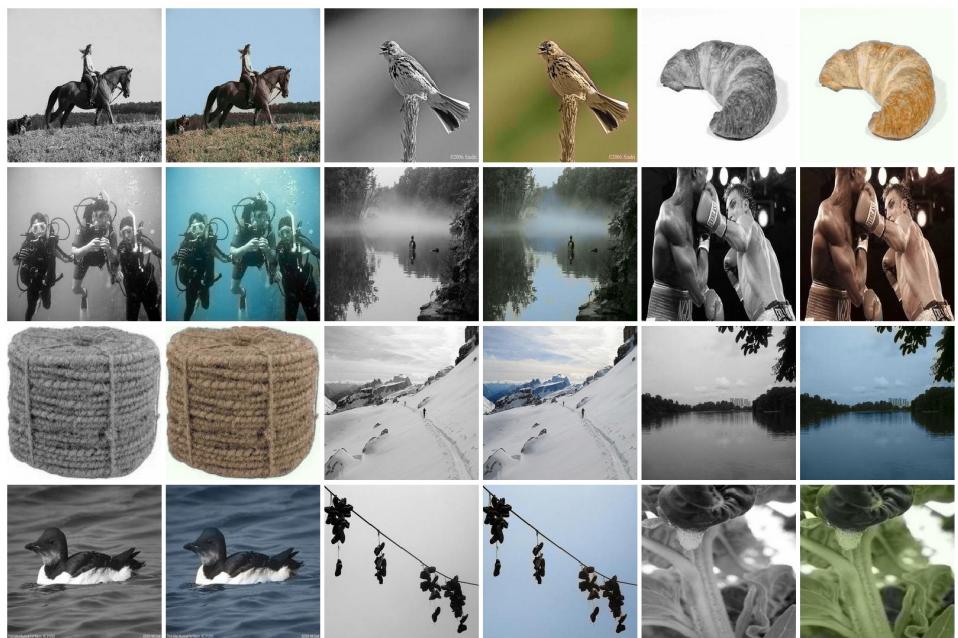


Slika 1.1: Algoritem za vhod vzame črno-belo (sivinsko) sliko, preko nivojev nevronske mreže določi barvne komponente in za konec združi sivinsko sliko z barvnimi komponentami.

zgleda enaka originalni ampak, da ta zgleda naravno. Nihče ne bo vedel, da je avtomobil, ki smo ga pobarvali rumeno bil v resnici rdeče barve.

Pri barvanju slik algoritmu, ki je zasnovan na podlagi nevronskega mrež, podamo sivinsko sliko (L kanal barvnega prostora *CIE Lab*), sistem pa vrne a in b kanal v istem barvnem prostoru. Za treniranje modela potrebujemo veliko količino črno-belih slik z referenčno barvno sliko, kar je praktično zastonj na voljo na spletu. Za treniranje lahko vzamemo katerokoli sliko, jo pretvorimo v barvni prostor *CIE Lab*, kjer L kanal predstavlja sivinsko sliko.

V tej magistrski nalogi rešujemo problem barvanja črno-belih slik in videov z večimi različnimi implementacijami. Za začetek smo implementirali več algoritmov za barvanje črno-belih slik. Rezultate smo med seboj primerjali s računanjem razlike med obarvano in originalno sliko. Rezultate smo primerjali tudi z dvema implementacijama iz sorodnih del. Ker pa obstaja veliko objektov, ki nimajo enolične barve (avtomobili, zgradbe, ...) in naš namen ni doseči enakega barvanja, vendar takega, ki da naravne rezultate, smo barvanje slik ocenjevali s pomočjo uporabnikov. V spletni anketi smo



Slika 1.2: Primeri barvanja črno-belih slik. Barvanje je bilo izvedeno z algoritmi, ki so bili razviti v okviru te magistrske naloge. Za vsako sliko je prikazana črno-bela slika, ki je bila vhod v algoritmom in obarvana slika - izhod algoritma.

uporabnike spraševali katera slika je bolj naravno obarvana (originalna ali slika obarvana z algoritmom).

Kasneje smo se odločili poskusiti tudi barvanja vida. Za barvanje smo arhitekturo nevronske mreže, ki je najbolje delovala na slikah prilagodili še za video.

Poglavlje 2

Pregled Področja

2.1 Predstavitev slikovnih podatkov in barvni prostori

Slike, ki jih uporabljamo za učenje so shranjene v *RGB* [1] barvnem prostoru. Kot je pokazano v [2] se izkaže, da prostor RGB ni direktno primeren za učenje algoritmov za barvanje iz dveh razlogov:

- **Sistem se ne ujema dobro s človeško percepcijo barv**, saj so razdalje med enako sorodnimi barvami različne glede na odtenek [3]. Na primer, če imamo dva para barv rdečo in svetlo rdečo, ter modro in svetlo modro, pri čemer sta barvi v vsakem paru za človekov vizualni sistem enako različni, sta razdalji v RGB barvnem prostoru različni .
- **Nima ločenega kanala za svetlost** [1]. Glede na to, da modeli za barvanje napovedujejo le barvne elemente v sliki, svetlost pa se vzame iz originalne slike, je najbolj priročno, če uporabljamo barvni prostor, ki ima ločen kanal za svetlost.

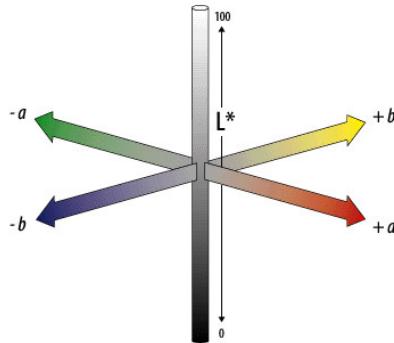
Tabela 2.1: Njavečje in njemanjše vrednosti posamezne komponente CIE $L^*a^*b^*$ barvnega prostora, pri pretvorbi vseh barv iz barvnega prostora RGB. Pretvorba je bila narejena z uporabo osvetlitve $D65$, ki določa temperaturo bele točke.

Kanal	Najmanjša vrednost	Največja vrednost
L^*	0	100
a^*	-86,185	98,254
b^*	-107,863	94,482

2.1.1 Izbira primernega barvnega prostora

Na podlagi teh predpostavk je izbira prostorov omejena na *Lab* [4], *YUV* [?] in *HSL* [1]. Vsi ustrezano drugi predpostavki iz 2.1. Edini, ki zares ustreza prvi predpostavki je *Lab*. Iz ugotovitev iz sorodnih del [2, 5, 6] se tudi najbolje izkaže prostor *CIE L*a*b*.

Obstaja več verzij barvnega prostora *Lab*, vendar se trenutno najbolj uporablja *CIE L*a*b**, ki naj bi bil najboljša aproksimacija človeškega vizualnega sistema [3]. Prostor ima to prednost, da je neodvisen od naprave. Prostor CIE $L^*a^*b^*$ predstavi vse barve, ki jih je možno zaznati z tremi barvnimi kanali. L^* predstavlja svetlost, a^* se razširja od zelene proti rdeči barvi in b^* od modre proti rumeni. L^* ser razteza od 0, ki predstavlja črno barvo, do 100, ki predstavlja belo barvo [?]. Prostor je grafično prikazan na sliki ?? a^* in b^* komponenti nimata uradne omejitve, vendar sta v implementacijah ponavadi omejene na vrednosti v intervalu $[-128, 127]$, kar je možno predstaviti z 8 bitnim celim številom [?]. Ker zaradi pretvorb iz barvnega prostora *RGB* vrednosti višje od 100 ali nižje -100 redko dosežemo smo opazili, da nekatere implementacije omejijo barvni prostor na interval $[-100, 100]$. Za pomoč pri implementaciji nevronske mreže smo sami preizkusili kakšen je dejanski interval barv pretvorjenih iz *RGB* barvnega prostora. Intervale si lahko pogledate v tabeli 2.1.



Slika 2.1: Slika prikazuje kanale barvnega prostora CIE $L^*a^*b^*$. L^* predstavlja svetlost, a^* se razteza od zelene barve v najbolj negativni točki proti rdeči barvi, b^* pa razteza od modre proti rumeni. Nasprotnoče barve na kanalih a^* in b^* se nikoli ne kombinirajo v odtonek. Iz: Adobe, Technical Guid, CIELAB, http://dba.med.sc.edu/price/irf/Adobe_tg/models/cielab.html (dostopano: 24. junij 2017)

2.1.2 Pretvarjanje med RGB in CIE $L^*a^*b^*$ barvnim prostorom

Za pretvorbo med prostoroma ni enostavne enačbe, saj je *RGB* barvni prostor odvisen od naprav, *CIE L^{*}a^{*}b^{*}* pa ne. Tako se pretvorba zgodi v treh korakih [?]:

1. **Pretvorba iz RGB v sRGB ali Adobe RGB**, saj sta ta barvna prostora neodvisna od naprave. Ta pretvorba je odvisna od naprave. Slike, ki jih bomo uporabili v našem delu so že v sRGB obliki, saj so bile pretvorjene, ko so bile zajete z fotoaparatom.
2. **Pretvorba CIE 1931 barvni prostor** ali drugače imenovan XYZ barvni prostor. Ta pretvorba se izvede s pomočjo linearne pretvorbe z matriko. Matrika je odvisna od izibire referenčne bele barve. Običajno se izbere referenčno temperaturo belo točke *D65*, ki je tudi standardizirana¹[7].

¹Zapis na uradni strani komisije International Commision on Illumination (krajše CIE),

3. Pretvorba iz XYZ v L*a*b* se izvede po enačbah opisanih v [8].

2.2 Obstojče metode za barvanje črno-belih slik

Metode za barvanje črno-belih slik delimo v dve večji skupini. Prva zahteva interakcijo uporabnika, pri drugi pa barvanje poteka popolnoma avtomatsko.

2.2.1 Metode, ki zahtevajo interakcijo uporabnika

To skupino metod delimo na tehnike, ki temeljijo na uporabnikovem barvanju manjših delov slik (ang. *scribble based*) [9, 10] in tiste, ki temeljijo na primerih (ang. *example based*) [11, 12, 13]. Pri prvih uporabnik določi barvo določenih točk v sliki, te pa algoritom avtomatsko razširi preko cele slike. Pri barvanju na primerih pa mora uporabnik izbrati sliko, ki je podobna tisti, ki jo želimo pobarvati, algoritom nato lastnosti izbrane slike razširi na drugo sliko ali množico slik. Tehnika barvanja s primeri se uporablja za barvanje videov, saj je v tem primeru potrebno ročno pobarvati na primer vsako stoto sliko, na ostale pa algoritom sam razširi lastnosti ročno barvane slike.

2.2.2 Popolnoma avtomatske metode

V magistrskem delu se osredotočamo na avtomatske metode barvanja. To so algoritmi, ki samostojno, brez uporabnikovega posredovanja, obarvajo celotno sliko. Prvi dve metodi, ki sta bili predlagani na tem področju, temeljita na značilkah (ang. *features*) pridobljenih iz slike. Tukaj gre predvsem za značilke, ki opisujejo intenziteto posamezne barve in opisnike, ki opisujejo robove v sliki. Prva metoda uporablja za barvanje nevronsko mrežo [14], ki pa vsebuje zgolj polnopovezane nivoje, druga pa za barvanje uporabi metodo naključnih gozdov (ang. *random forest*) [15].

Novejši pristopi barvanja črnobelih slik tipično temeljijo na konvolucijskih nevronskih mrežah, ki imajo to lastnost, da v vsakem nivoju same odkrijejo značilke, ki so pomembni. Prva tovrstna rešitev [16] gradi mrežo na podlagi že zgrajene šestnajst-nivojske mreže VGG-16, ki so jo razvili na univerzi v Oxfordu [17]. Rešitev uporablja evklidsko cenilno funkcijo in barvni prostor YUV. Slabost te rešitve je, da izhodne barvne slike niso dovolj nasičene.

V zadnjem času predlagane rešitve popravijo problem nenasičenosti z uporabo softmax funkcije v zadnjem nivoju nevronске mreže, kar pomeni, da so problem spremenili iz regresijskega v klasifikacijskega. Zang in sod. [18] uporabijo konvolucijsko nevronsko mrežo z več nivoji in aktivacijskimi funkcijami ReLU. Posebnost te mreže je cenilna funkcija. Uporablja križno entropijo, ki pa je v tem primeru izvedena na primerjavi barv posameznih delov slike glede na barvni prostor, ki je kvantiziran. Napake so pomnožene z utežjo, ki določa pogostost barve. Bolj redke barve so obtežene tako, da prispevajo večji delež k napaki, ki jo izračuna cenilna funkcija. S tem so avtorji izboljšali rezultate, tako da se bolj pogosto pojavljajo tudi močnejši odtenki (tisti z višjimi vrednostmi v prostoru a^*b^* , ki so bili prej redkeje zastopani zaradi bolj pogostega pojavljanja nežnejših barv v slikah (barve bližje vrednostim $(0, 0)$ v a^*b^* prostoru. Pogostost je bila izračunana z analizo vseh slik v podatkovni zbirkki Imagenet [19]. Uporabljajo barvni prostor L^*a^*b . Larsson in sod. [20] za osnovo uporabijo mrežo VGG-16, iz katere vzamejo napovedi vsakega nivoja, ki jih združijo v enotno matriko. Sledi še en polno-povezan nivo na nivoju točk v sliki. Rezultat klasifikacije je histogram za vsako točko v sliki (histogram z verjetnostmi). Uporabljajo barvni prostor HSV. Cenilna funkcija, ki jo uporabljajo je KL-divergenca, ki primerja izhodni histogram z v histogram pretvorjeno originalno sliko. Iizuka in sod. [2] uporabijo nevronsko mrežo sestavljenou iz dveh delov. Prvi del poskrbi za napovedovanje vsebine slike, ki se potem združi z drugim delom in izboljša natančnost barvanja. Uporabili so križno entropijo (ang. *cross entropy*) v kombinaciji s cenilno funkcijo *povprečna kvadratna napaka* (ang. *Mean squared error*) in barvni prostor L^*a^*b . Za razliko od prejšnjih dveh metod zadnja ne napo-

veduje histograma na podlagi kvantiziranega prostora ampak direktno a^* in b^* vrednost, kar pomeni, da ne uporablja klasifikacije ampak regresijo.

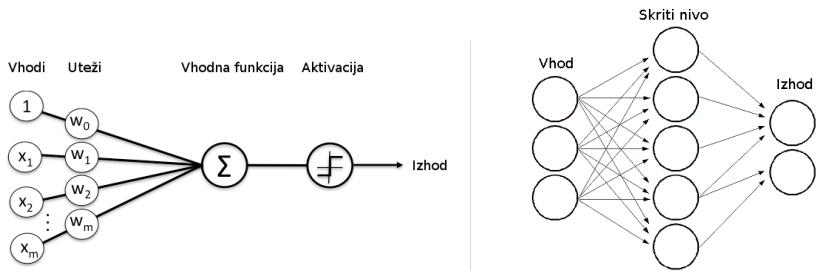
2.3 Globoke nevronske mreže

Globoke nevronske mreže so algoritmi, ki so zgrajeni na podlagi opazovanja strukture možganov. Veliko se uporablja za klasifikacijo, regresijo, gručenje in napovedovalno analizo. Predvsem se uporablja na področju slik, kjer je zelo pomembno prepoznavanje objektov in obrazov, razvrščanje slik v skupine glede na podobnost, prepoznavanje gest in barvanje slik [21].

Nevronska mreža je v osnovi funkcija $f(x)$, ki preslika vhod x v izhod y . Med postopkom učenja je ta funkcija optimizirana tako, da najde najboljšo povezavo med vhodnimi in izhodnimi podatki [21]. Nevronske mreže so ime za strukturo, ki je sestavljena iz več nivojev. Nivoje si lahko predstavljamo kot vrsto vozlišč, ki se odzovejo v primeru da je vzburjenje na njih zadovoljivo - odvisno od aktivacijske funkcije. Struktura vozlišča in nivojev je predstavljena na sliki 2.2. Vozlišče pomnoži vsak vhod s trenutnimi vrednostmi uteži, vrednosti sešteje in moč aktivacije izračuna s pomočjo tako imenovane aktivacijske funkcije, ki tvori izhod vozlišča. Aktivacijski funkciji rečemo tudi neliearnot, saj poskrbi za to, da nevronska mreža ni le linearna funkcija. [?] Uteži se skozi postopek učenja spreminjajo in s tem določijo aktivacijo vozlišča.

Nivojem v nevronskih mrežah, ki se nahajajo med vhodnim in izhodnim nivojem rečemo *skriti nivoji* (angl. *hidden layers*) [?]. Tradicionalni algoritmi na področju strojnega učenja so sestavljeni iz vhodnega, izhodnega nivoja in enega skritega nivoja, *globoka nevronska mreža* (ang. *deep neural network*) pa ima vsaj dva skrita nivoja [?], večinoma pa mnogo več. Vsak nivo globoke nevronske mreže prepozna določen lastnosti vhodnih podatkov. Nivoji, ki se nahajajo globje lahko prepozna bolj kompleksne lastnosti podatkov, saj na vhodu, dobijo lastnosti ozioroma aktivacije nivoja pred njim.

Da nevronska mreža daje zadovoljive rezultate je potrebno utežem določiti



Slika 2.2: Leva slika prikazuje zgradbo enega vozlišča, ki ima zasnovno podobno nevronom v možganih, desna pa zgradbo več nivojske nevronske mreže. Iz: Introduction to Deep Neural Networks url<https://deeplearning4j.org/neuralnet-overview> in Neural Networks, urlhttp://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html (dostopano: 21. junij 2017)

prave vrednosti. To naredimo s postopkom učenja. Vsaka nevronska mreža ima cenilno funkcijo (ang. loss function), ki pove kako dobre rezultate na testnih podatkih nevronska mreža daje trenutno. V postopku učenja zmanjšujemo vrednost cenilne funkcije z enim od algoritmov optimizacije.

2.3.1 Konvolucijske nevronske mreže

Ker bi bilo na primeru slik pri uporabi klasičnih nevronskih mrež hitro preveč parametrov, kar bi poleg podaljšanja časa učenja povzročilo tudi prekomerno prilagajanje (ang. overfitting), uporabljam za take primere konvolucijske nevronske mreže. Te so zelo podobne običajnim nevronskim mrežam. Sestavljene so iz nevronov, ki imajo svoje uteži in bias, ki so učljivi. Operacije znotraj nevrona so podobne tistim pri običajnih nevronskih mrežah, le da so prilagojene pričakovanim vhodnim podatkom - slikam. Vhod v vsak nivo nevronske mreže je torej matrika z obliko *širina x višina x globina* [22]. Te so v osnovi sestavljene iz treh vrst nivojev:

- **Konvolucijski nivo** je glavni gradnik konvolucijske nevronske mreže. Parametri tega nivoja so sestavljeni iz majhnih konvolucijskih jeder, ki

pokrivajo majhno polje v širino in višino obenem pa pokrivajo celotni nivo v globino. Med prehodom po nevronski mreži izvedemo konvolucijo po celotni višini in širi vhodne matrike, po globini pa se te izhode teh konvolucij sešteje enako kot pri običajni nevronski mreži. Izhod konvolucije z enim setom jeder je dvodimenzionalna matrika. [?]

- **Pooling nivo** je namenjen pod-vzorčenju (ang. downsampling) na določenem nivoju. S tem zmanjšamo število parametrov, kar vpliva zmanjšanje računske zahtevnosti in prekomernega prilagajanja. Deluje na principu, da je točna lokacija značilke manj pomembna kot približna lokacija glede na ostale značilke. [?]
- **Polno povezni nivo** je nivo enak skritim nivojem pri klasični nevronski mreži. Uporabi se za zadnjih nekaj nivojev pri konvolucijski nevronski mreži.

Poglavlje 3

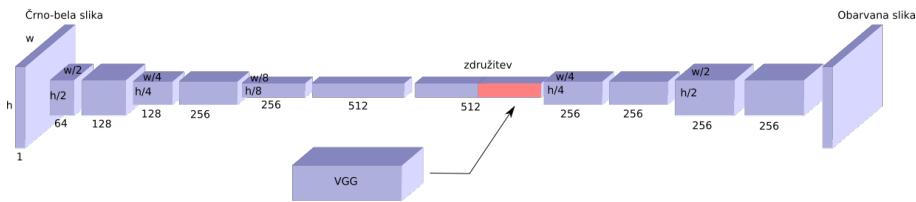
Barvanje črno-belih slik z globokimi nevronskimi mrežami

3.1 Arhitekture

V grobem smo v okviru magistrske naloge implementirali 4 arhitekture nevronskih mrež, kasneje smo te arhitekture kombinirali z različnimi cenilnimi funkcijami, pristopi (regresijski ali klasifikacijski) in načini napovedovanja (napovedovanje po delih ali na celi sliki).

3.1.1 Plitva arhitektura z globalno mrežo

Ta arhitektura je sestavljena iz dveh delov, ki se kasneje združita v enotno mrežo. Glavni del predstavlja zaporedje konvolucijskih nivojev, ki na vhodu vzamejo sivinsko sliko, izhod pa je obarvana slika. Po osmih kovolucijskih nivojih se mreža združi z tako imenovano globalno mrežo, ki napoveduje objekt, ki ga slika predstavlja. Za globalno mrežo smo vzeli že naučeno mrežo VGG-16 [17], ki smo ji odvzeli zadnji polno-povezani nivo in ji dodali nov polno-povezani nivo z izhodnim tenzorjem dolžine 256. Ker je ta mreža namenjena sprejemu barvnih RGB slik smo vhod prilagodili tako, da sprejme sivinsko sliko na vseh treh kanalih. Arhitektura nevronske mreže je predstavljena na sliki 3.1 in v tabeli B.1 v prilogi.



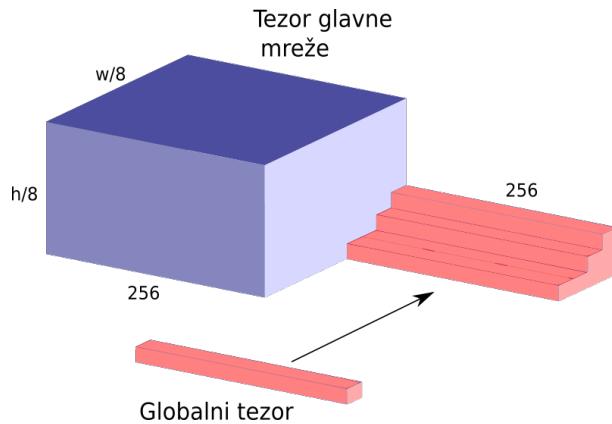
Slika 3.1: Slika prikazuje velikosti tenzorjev skozi plitvo arhitekturo z globalno nevronsko mrežo. w in h predstavljata širino in višino vhodne slike. Podrobnosti nivoja združitev so predstavljene na sliki 3.2, nivo izhodne slike ni natančneje označen, saj se razlikuje v različnih implementacijah, ki bodo podrobnejše opisane v poglavju ??.

Na tem mestu bi bilo smiselno opisati še kako poteka združevanje dveh mrež. Vhod v element za združevanje sta tenzorja velikosti $w/8 \times h/8 \times 256$ iz glavne mreže in enodimensionalni tenzor velikosti 256 iz globalne mreže. Pri tem w in h predstavljata širino in višino vhodne slike v mrežo. Pri združevanju vsakemu elementu širine in višine prvega tenzorja pridružimo tenzor globalne mreže, kot prikazuje slika 3.2. Tako na izhodu dobimo tenzor velikosti $\frac{w}{8} \times \frac{h}{8} \times 512$.

3.1.2 Globja arhitektura z globalno mrežo

Ta arhitektura ima v osnovi enako zasnovo, kot v poglavju 3.1.1 opisana arhitektura. Razlika se pojavi pri globini glavne mreže. Ta ima namreč mnogo več nivojev kot lahko vidite na sliki 3.3 in tabeli B.2 v prilogi. Za razliko od arhitekture opisane v poglavju 3.1.1 ta za zmanjševanje prostorskih (ang. *spatial*) dimenzij uporablja maksimalno združevanje (ang. *max pooling*) in za povečevanje le teh v zadnjih nivojih uporablja transponirano konvolucijo (ang. *transpose convolution*) [?]. Združevanja glavne in globalne mreže se izvede na način opisan v poglavju 3.1.1.

Ta arhitektura prinaša še eno spremembo. To so tako imenovane rezidualne povezave, ki so bile prvič uporabljeni v nevronski mreži ResNet [?], ki je leta 2015 zmagala na tekmovanju ImageNet [19]. Te povezave so na sliki 3.3



Slika 3.2: Prikaz delovanje nivoja združevanja glavne nevronske mreže z globalno nevronske mrežo. K izhodu glavne nevronske mreže dodamo tenzor iz globalne mreže, tako da ga priključimo k vsem prostorskim lokacijam kot dodatnih 256 kanalov.

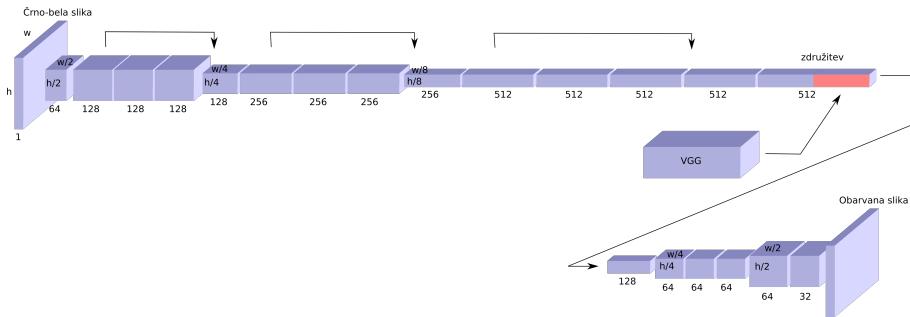
označene s puščicami nad nevronske mreže in predstavljajo povezavo, ki na mestu kamor kaže puščica, združi trenutni tenzor z tenzorjem izračunanim pred dvema nivojema. Operacija združevanja je seštevanje.

3.1.3 Globja arhitektura brez globalne mreže

Ta arhitektura je enaka tisti opisani v poglavju in prikazani na sliki 3.3 v glavnem delu in se razlikuje po tem, da nima globalne mreže. Torej nima mreže, ki se pridruži v nivoju združitve, zaradi tega smo tudi ta nivo izpustili. Mreža za vhod vzame črno-belo sliko z enim kanalom in izračuna barvno sliko na izhodu. Ta arhitektura je bila načrtovana z namenom, da se preveri če ima globalna mreža prisotna v arhitekturah predstavljenih v poglavjih 3.1.2 in 3.1.1 kakšen vpliv na rezultate.

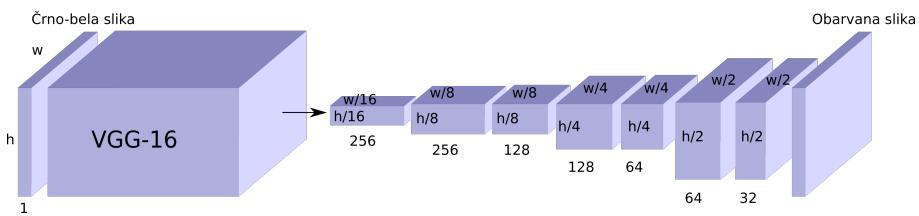
3.1.4 Dopolnjena VGG-16 arhitektura

Arhitektura je zgrajena tako, da za vrh mreže uporabimo mrežo VGG-16 [17], kateri smo odstranili vse polno povezane nivoje. Arhitektura je narejena



Slika 3.3: Slika prikazuje velikosti tenzorjev skozi globjo arhitekturo z globalno nevronsko mrežo. w in h predstavljata širino in višino vhodne slike. Podrobnosti nivoja združitev so predstavljene na sliki 3.2, nivo izhodne slike ni natančneje označen, saj se razlikuje v različnih implementacijah, ki bodo podrobnejše opisane v poglavju ??.

tako, da na vhodu sprejme črno-belo sliko, ki se potem prilagodi na vhod mreže VGG-16. Tenzor, ki ga vrne zadnji konvolucijski nivo omenjene mreže podam na vhod mreže, ki je prikazana na sliki 3.4 in podrobno opisana v tabeli B.3 v prilogi.

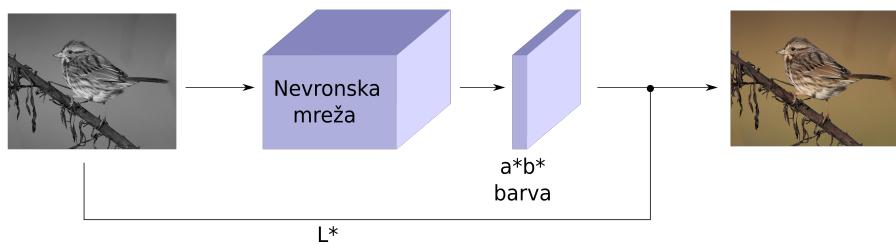


Slika 3.4: Slika prikazuje velikosti tenzorjev skozi dopolnjeno VGG-16 arhitekturo. w in h predstavlja širino in višino vhodne slike. Nivo izhodne slike ni natančneje označen, saj se razlikuje v različnih implementacijah, ki bodo podrobneje opisane v poglavju ??.

3.2 Pristopi z regresijo

V tem poglavju bomo prestavili regresijske metode, ki smo jih uporabili. Te se imenujejo tako, saj pri napovedovanju barve direktno napovejo vrednosti a^* in b^* barvne komponente v prostoru $CIE L^*a^*b^*$. Tu mreža predstavlja regresijsko funkcijo $y = f(x)$ za vsako točko na slike, ki na vhod dobi točko sivinske slike x , izhod pa je kar vrednost y , ki predstavlja določeno barvo v našem primeru sta to dve vrednosti a^* in b^* .

Na sliki 3.5 je prikazan postopek delovanja, ki je skupen vsem regresijskim metodam opisanim v nadaljevanju. Da dobimo barvno sliko moramo izhod iz mreže, ki predstavlja barvne komponente združiti s sivinsko.



Slika 3.5: Shematski prikaz delovanja regresijske metode, ki na vhod prejme črno-belo sliko, s pomočjo nevronske mreže izračuna barvni komponenti a^* in b^* barvnega prostora $CIE L^*a^*b^*$ ter to združi s sivinsko sliko L^* , da dobi obarvano sliko.

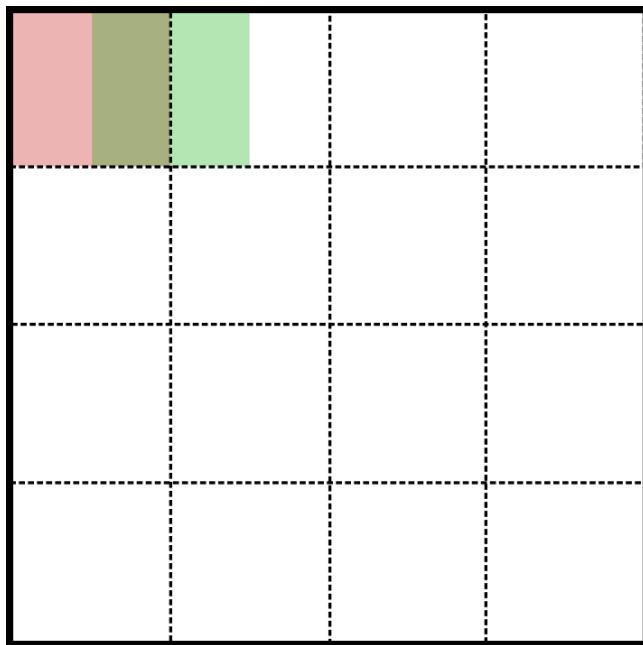
3.2.1 Pristopi na delih slik

V to skupino lahko uvrstimo tri metode, ki imajo skupno to, da smo barvanje izvajali na majhnih delih slik. Ta princip smo razvili z opazovanjem delovanja človeškega zaznavnega sistema, ki se bi lotil barvanja po delih, na sliki bi zaznal objekte ločeno barval. Na primer najprej vodo, nato gozd in kasneje še nebo.

Sliko smo preden smo jo podali nevronski mreži razdelili na koščke velikosti 32×32 . Pri tem sta se sosednja koščka prekrivala za 16 slikovnih točk kot

je prikazano na sliki 3.6. V arhitekturah, kjer je bila prisotna ločena globalna mreža, je ta še vedno na vhodu prejela celotno sliko, s katero smo pridobili globalni koncept slike. Dele slik smo kasneje sestavili z metodo prekrivanja, tako da so imele vrednosti točk pri robu manjši vpliv kot tiste pri sredini. Vpliv barvne točke se je izračunal po enačbi 3.1, kjer x predstavlja vrednost slikovne točke, d predstavlja oddaljenost od središča v številu slikovnih točk. Enačba se ločeno uporablja v vertikalni in horizontalni smeri.

$$y = \frac{d}{16} * x \quad (3.1)$$



Slika 3.6: Shema prikazuje razdeljevanje slike velikosti 128×128 slikovnih točk, ki jo v vsaki smeri razdelimo na 7 enakih delov velikosti 32. Rdeč in zelen kvadrat prikazujeta prva dva dela s konceptom prekrivanja. Enak sistem uporabimo v vertikalni smeri.

Z takim načinom dela pričakujemo pohitritev učenja, saj menimo, da je že del slike dovolj da se mreža nauči celotne tekture objekta. Na primer, da se naučimo barvati vodo ne potrebujemo celotnega območja vode na sliki,

ki včasih lahko prekriva tudi pol ali več slike, ampak le en del. Z učenjem mreže na manjših vhodnih tenzorjih, kar dosežemo z učenjem po delih, se čas, ki ga mreža porabi za učenje ene serije (ang. *batcha*) podatkov zmanjša.

V tabeli 3.1 so predstavljene podrobnosti vsake od metod barvanja po delih.

Tabela 3.1: Regresijski pristopi po delih, njihove arhitekture, ki so podrobneje opisane v poglavju 3.1 in cenilne funkcije uporabljene za učenje. *MSE* je povprečna kvadratna napaka (ang. *mean squared error*).

Ime metode	Arhitektura	Cenilna funkcija
Reg. po delih	Globja arh. z glob. mr.	MSE
- brez softmax	Globja arh. z glob. mr.	MSE
- brez globalne mreže	Globja arh. brez glob. mr.	MSE

3.2.2 Pristopi na celih slikah

Za primerjavo točnosti metod na delih slik s tistimi na celih slikah. Smo dve metodi opisane v poglavju 3.2.1 pretvorili v metode za barvanje na celih slikah. Tej smo dodali še eno metodo, saj zaradi večkratnega pomanjšanja prostorskih dimenzij znotraj arhitekture ne more biti realizirana na manjših delih slik.

V tabeli 3.2 so podrobno predstavljene lastnosti teh metod.

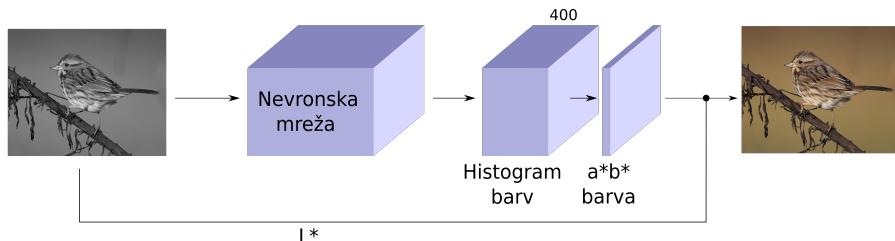
3.3 Pristopi s klasifikacijo

Razvili smo tudi štiri metode, ki namesto regresije uporablja klasifikacijo. To so pristopi, kjer direktno ne napovemo številčne vrednosti barve ampak to določimo s pomočjo klasifikacije v enega od razredov, ki predstavljajo nekaj sosednjih odtenkov v sliki. Klasifikacija se izvede z uporabo softmax

Tabela 3.2: Regresijski pristopi na celih slikah, njihove arhitekture, ki so podrobneje opisane v poglavju 3.1 in cenilne funkcije uporabljene za učenje. *MSE* je povprečna kvadratna napaka (ang. *mean squared error*).

Ime metode	Arhitektura	Cenilna funkcija
Reg. cela slika	Globja arh. z glob. mr.	MSE
- brez globalne mreže	Globja arh. brez glob. mr.	MSE
Reg. cela slika VGG	Dop. VGG-16 arh.	MSE

funkcije v zadnjem nivoju mreže. Izhodi teh mrež so verjetnosti za vsakega od razredov. Kot je prikazano na sliki 3.7 se te vrednosti pretvorijo v vrednosti a^* in b^* .



Slika 3.7

Razrede smo dobili tako, da barvni prostor *CIE L*a*b** razdelimo v 400 razredov po komponentah a^* in b^* . Vsako od komponent smo razdelili v 20 razredov med vrednostima –100 do 100, kar pomeni da vsak razred zajema interval širine 10.

Za delovanje mreže moramo pretvoriti $a^* b^*$ zapis v histogram in obratno. Pretvorba v iz zapisa $a^* b^*$ v histogram je potrebna saj moramo pri učenju referenčno sliko (ang. *ground truth*) pretvoriti v histogram, da lahko izračunamo napako. To se izvedemo s enačbo 3.2, kjer a in b predstavljata

$a*$ ali $b*$ vrednost slikovne točke, y pa indeks razreda v histogramu.

$$y = 20 \left\lfloor \frac{a + 100}{10} \right\rfloor + \left\lfloor \frac{b + 100}{10} \right\rfloor \quad (3.2)$$

Pretvorba iz histograma v $a*$ in $b*$ se uporabi pri pretvorbi ocenjenih barvnih vrednosti s pomočjo mreže in se izvede po enačbah 3.3. Pri tem a in b predstavlja $a*$ in $b*$ barvne vrednosti slikovne točke, y predstavlja indeks razreda v histogramu, ki je bil napovedan z največjo verjetnostjo.

$$\begin{aligned} a &= 10 \left\lfloor \frac{y}{20} \right\rfloor - 100 + 5 \\ b &= 10(y \mod 20) - 100 + 5 \end{aligned}$$

Pristope s klasifikacijo smo preizkusili, ker so v delih [6, 5] ugotovili, da so rezultati teh metod, slike, ki so bolj nasičene in realnih barv. V tabeli 3.3.

Tabela 3.3: Klasifikacijski pristopi po delih, njihove arhitekture, ki so podrobneje opisane v poglavju 3.1 in cenilne funkcije uporabljene za učenje. KL – divergenca je Kullback-leibler divergenca [?], CCE predstavlja kategorično križno entropijo [?].

Ime metode	Arhitektura	Cenilna funkcija
Klas. brez uteži - plitva arh.	Plitva arh. z glob. mr.	KL-divergenca
Klas. brez uteži - globja arh.	Globja arh. z glob. mr.	CCE
Klas. z utežmi - plitva arh.	Plitva arh. z glob. mr.	CCE
Klas. z utežmi - globja arh.	Globja arh. z glob. mr.	CCE

3.4 Postopek učenja

V tem poglavju bomo predstavili podrobnosti učenja na manjši množici, na večji učni množici in si za konec še pogledali kakšne značilke prepozna posamezni nivo nevronske mreže.

3.4.1 Učenje na manjši učni množici

Za učenje na manjši množici smo izbrali vse metode opisane v poglavjih 3.2 in 3.3. Za primerjavo natančnosti smo dodatno še implementirali metode razvite s strani R. Dahl-a [16], Iizuka in sod. [2] in Zhang in sod. [18].

Za posodabljanje parametrov smo uporabili Adam optimizator [?]. Pri tem sta se za dobre izkazali naslednji parametri, ki so prikazani v tabeli 3.4. Pri vseh metodah smo uporabili velikost serije (ang. *batch size*) 32, razen pri treniranju metode Zhang in sod., kjer smo morali zaradi večjih dimenzij tenzorjev in posledično pomakanju spomina uporabiti velikost serije 8.

Tabela 3.4: Parametri, s katerim smo nastavili Adam optimizator.

Parameter	Vrednost parametra
stopnja učenja (ang. <i>learning rate</i>)	10^{-4}
beta 1	0.9
beta 2	0.99
epsilon	10^{-8}

Pri treniranju smo beležili vrednosti cenilne funkcije tako na množici podatkov za treniranje in množici podatkov za validacijo po končanem vsakem prehod čez vse podatke (ang. *epoch*). S tem smo opazovali kdaj je določena metoda optimalno naučena. To se v našem primeru zgodi v trenutku, ko vrednost cenilne funkcije na validacijski množici preneha padati ali celo začne naraščati. V tem trenutku je naša mreža optimalno naučena, zato smo te uteži uporabili za testiranje. Grafi padanja cenilnih funkcij za vse pristope so prikazani na sliki 3.8.

3.4.2 Učenje na večji učni množici

3.4.3 Pomen nivojev mreže

Tukaj bi dali tudi slike po nivojih - kaj so znacilke posameznega nivoja

3.5 Podatki

Podatke, ki smo jih uporabili v za treniranje in validacijo pristopov smo pri-dobili iz podatkovne zbirke ImageNet [19], ki vsebuje približno 14 milijonov slik. Iz zbirke smo naključno izbrali množico podatkov jih za namen učenja pretvorili v $CIE L^*a^*b^*$ barvni prostor. Pri preizkusu na manjši množici smo za učenje naključno izbrali 100000 slik za vlidacijo pa 10000 slik iz nabora.

3.6 Računanje napake

Za primerjavo metod smo napake računali na validacijski množici. Pri tem smo uporabili dve metrike: kvadratni koren povprečne napake (ang. *Root mean squared error*) in razmerje med signalom in šumom (ang. *Peak signal-to-noise ratio*).

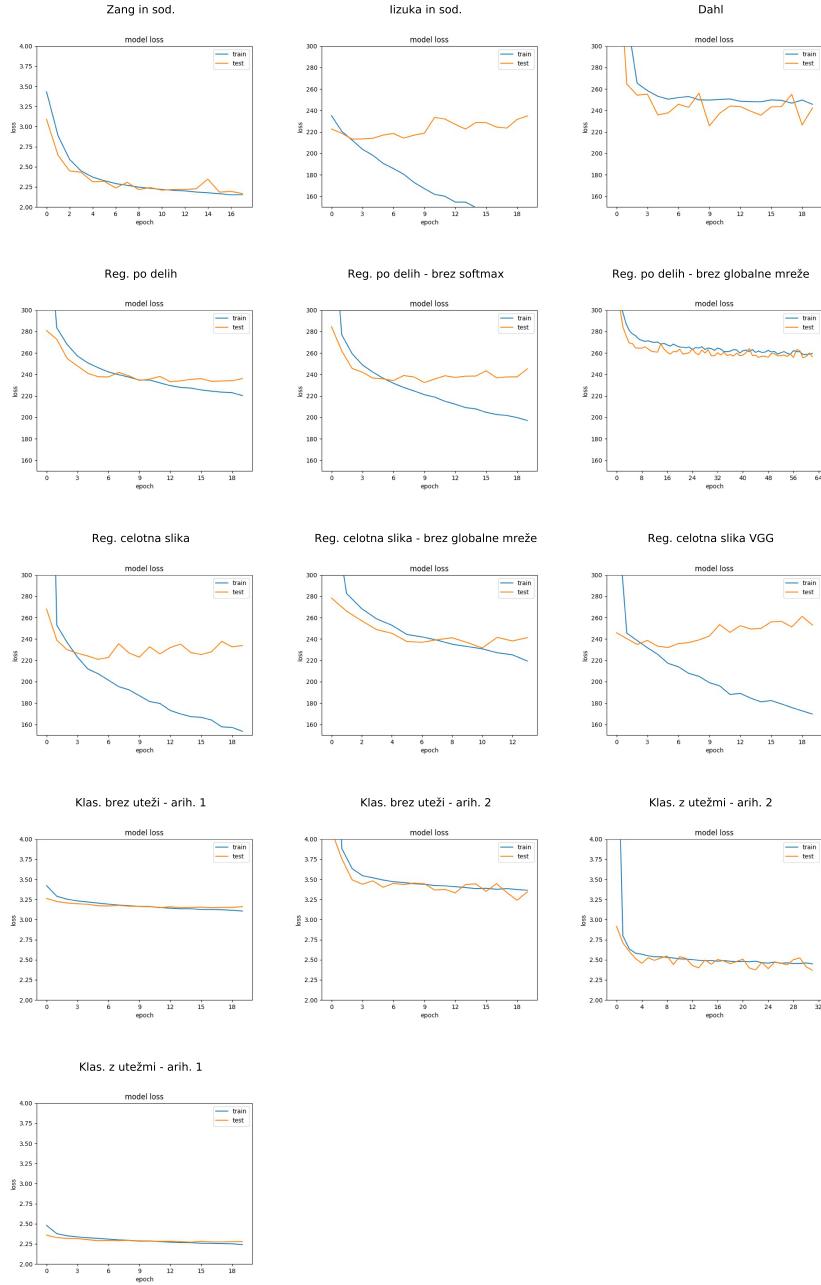
Kvadratni koren povprečne napake (RMSE) za vsako sliko smo izračunali s pomočjo enačbe 3.3, kjer w in h prestavlja širino in višino slike ter c predstavlja število kanalov slike. d je originalna slika (ang. *ground truth*) in \hat{d} obarvana slika. Napaka je bila izračunana za vsako sliko posebej in kasneje povprečena preko vseh slik. Napaka RMSE je bila izračunana na slikah v CIE $L^*a^*b^*$ barvnem prostoru le za a^* in b^* barvni kanal, saj za L^* barvni kanal izračun napake ni smiseln, ker so vrednosti vzete iz originalne slike.

$$RMSE = \sqrt{\sum_{i=1}^w \sum_{j=1}^h \sum_{k=1}^c (d_{i,j,k} - \hat{d}_{i,j,k})^2} \quad (3.3)$$

Razmerje med signalom in šumom (PSNR) je metrika, ki kaže razmerje med največjo možno močjo signala in močjo šuma, ki signal pokvari. Primerna je za primerjave rekonstruiranih podatkov, kot so v našem primeru validacijske slike, ki jih poskušamo rekonstruirati s pomočjo nevronske mreže. Vrednosti razmerja med signalom in šumom se merijo v enoti decibel (dB) [?]. Zadovoljive vrednosti rekonstrukcije slike 8 bitnih podatkov, kar naši podatki so, saj smo primerjali slike v barvnem prostoru *RGB*, je med 30 in 50 dB [?].

Napako PSNR smo izračunali z enačbo ??, v kateri ima MAX_I največjo možno vrednost slikovne točke, kar je v barvnem prostoru RGB 255, $RMSE$ pa je napaka izračuana z enačbo 3.3.

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{RMSE} \right) \quad (3.4)$$



Slika 3.8: Prikaz padanja napake modela pri učenju. Za vsak prehod preko vseh podatkov (ang. *epoch*) je prikazana vrednost celine funkcije na množici za treniranje in testni množici.

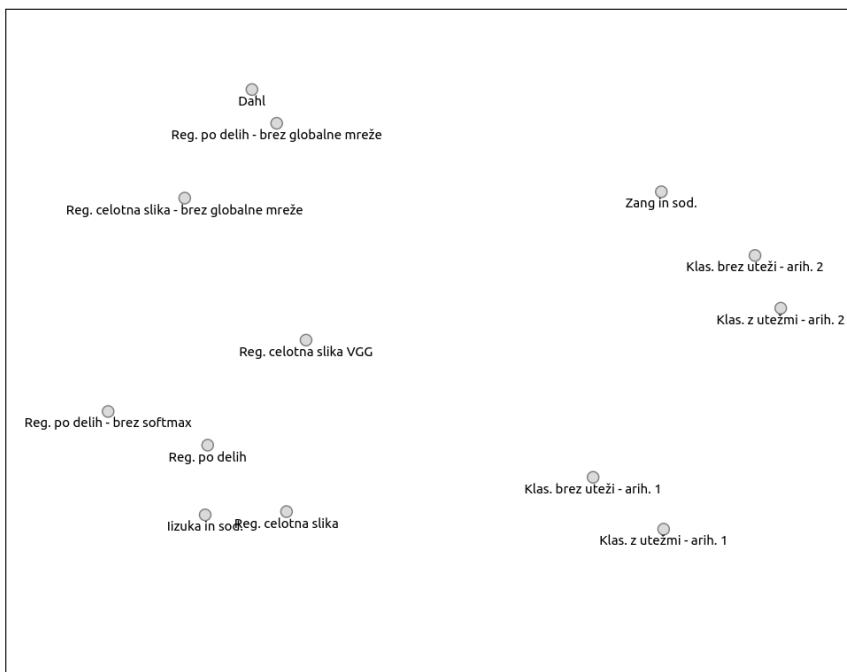
Poglavlje 4

Rezultati in diskusija

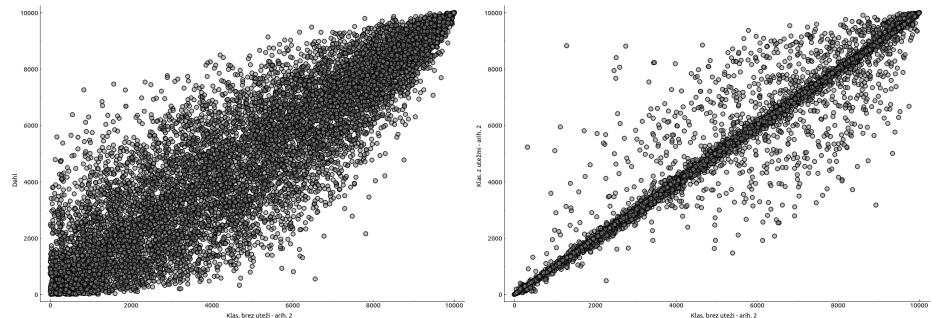
4.1 Primerjava metod na manjši učni množici

Tabela 4.1: Napake izračunane na testni podatkovni zbirki.

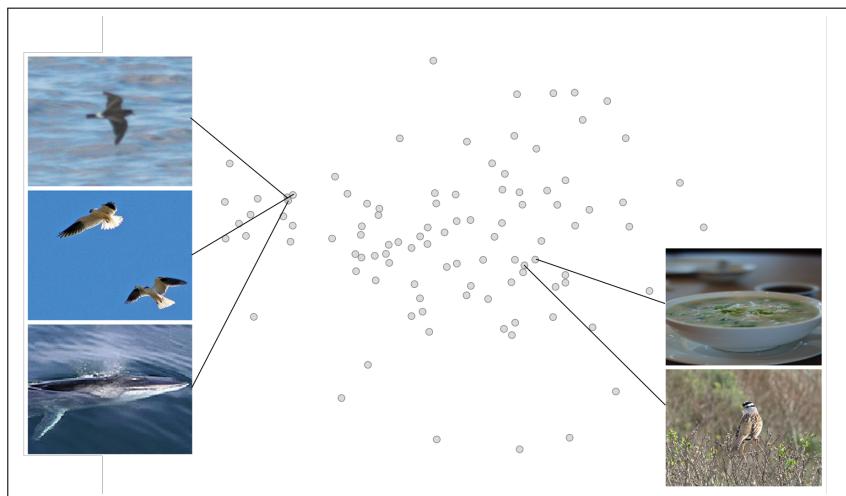
Metoda	RMSE	PSNR
Zhang in sod.	15.004	22.252
Iizuka in sod.	12.941	23.439
Dahl	13.936	22.551
Reg. po delih	13.216	23.199
- brez softmax	13.206	23.183
- brez globalne mreže	13.767	22.840
Reg. celotna slika	13.007	23.434
- brez globalne mreže	13.334	23.068
Reg. celotna slika VGG	13.387	23.131
Klas. brez uteži - plitva arh.	14.336	22.738
Klas. brez uteži - globoka arh.	15.086	22.380
Klas. z utežmi - plitva arh.	14.573	22.610
Klas. z utežmi - globoka arh.	15.137	22.395



Slika 4.1: Primerjava metod v prostoru MDS kaže sorodnosti med metodami na način napovedovanja barve (klasifikacija proti regresiji) in glede na arhitekturo mreže. Mreže, ki uporabljajo mrežo VGG za napovedovanje objekta v sliki so bližje skupaj in modeli brez VGG mreže so bližje skupaj.



Slika 4.2: Graf prikazuje rangiranje slik glede na napako $RMSE$ pri dveh različnih metodah. X os predstavlja rang pri prvi metodi, Y pa rang pri drugi metodi. Prva slika prikazuje primerjave rangov Dahlove metode in klasifikacijske metode z arhitekturo 2. Druga metoda prikazuje range pri dveh klasifikacijskih metodah z istimi arhitekturami.



Slika 4.3: Razporeditev slik v prostoru MDS, ki zajema 100 sli, ki kjer natančnosti najbolj odstopajo pri različnih metodah. V prostoru so točke podobnih slik bliže skupaj. Dve taki podobni skupini slik sta prikazani ob robu.

4.2 Primerjava metod na večji učni množici

Tukaj bi primerjali metode, ki bi bile naučene na več slikah, tako bi pridobili tudi boljša barvanja za pokazat izboljševanje barvanja glede na epoche. Primerjavo bi izvedli na manj manj metodah.

4.3 Barvanje večjih slik

Poglavlje 5

Zaključek

Dodatek A

Spearmanova korelacija rangov med metodami

Tabela A.1: Spearmanova korelacija med metodami.

	Zang	Iizuka	Dahl	Reg. po de	Reg. po de	Reg. po de	Reg. celot	Reg. celot
Zang in so	1.0000	0.8607	0.8688	0.8637	0.8407	0.8898	0.8779	0.8800
Iizuka in	0.8607	1.0000	0.8936	0.9454	0.9420	0.9015	0.9472	0.9200
Dahl	0.8688	0.8936	1.0000	0.9008	0.9058	0.9816	0.8889	0.9510
Reg. po de	0.8637	0.9454	0.9008	1.0000	0.9440	0.9096	0.9392	0.9140
Reg. po de	0.8407	0.9420	0.9058	0.9440	1.0000	0.9120	0.9313	0.9190
Reg. po de	0.8898	0.9015	0.9816	0.9096	0.9120	1.0000	0.9005	0.9640
Reg. celot	0.8779	0.9472	0.8889	0.9392	0.9313	0.9005	1.0000	0.9140
Reg. celot	0.8804	0.9206	0.9518	0.9149	0.9194	0.9642	0.9140	1.0000
Reg. celot	0.8969	0.9398	0.9196	0.9309	0.9211	0.9263	0.9334	0.9370
Klas. brez	0.9092	0.9012	0.8636	0.9004	0.8795	0.8784	0.9030	0.8750
Klas. brez	0.9409	0.8519	0.8861	0.8622	0.8434	0.8979	0.8657	0.8650
Klas. z ut	0.9242	0.8535	0.8722	0.8578	0.8390	0.8817	0.8632	0.8550
Klas. z ut	0.9023	0.8889	0.8431	0.8861	0.8640	0.8582	0.8904	0.8540

Dodatek B

Podrobnosti arhitektur

Tabela B.1: Tabela prikazuje nivoje plitve arhitekture z globalno mrežo in njihove parametre. K predstavlja število kanalov izhoda nevronske mreže, J pove velikost jedra, Ko je korak (ang. *stride*) uporabljen na nivoju in Akt predstavlja aktivacijo po vsakem nivoju. Zadnji konvolucijski nivo nima definiranega števila izhodnih kanalov, saj je to odvisno od pristopa, ki smo ga uporabili. Pri nadzorčenju smo vedno nadzorčili s faktorjem 2, kar pomeni, da ima tensor, ki predstavlja izhod po širini in višini dvakratno velikost.

Nivo	K	J	K	Akt
Vhod	1	-	-	-
2D konvolucija	64	3	2	Relu
2D konvolucija	128	3	1	Relu
2D konvolucija	128	3	2	Relu
2D konvolucija	256	3	1	Relu
2D konvolucija	256	3	2	Relu
2D konvolucija	512	3	1	Relu
2D konvolucija	512	3	1	Relu
2D konvolucija	256	3	1	Relu
Združitev z globalno mrežo	512	-	-	-
2D konvolucija	256	3	1	Relu
Nadvzorčenje	256	-	-	-

2D konvolucija	256	3	1	Relu	
2D konvolucija	256	3	1	Relu	
Nadvzorčenje	256	-	-	-	
2D konvolucija		3	1	Relu	

Tabela B.2: Tabela prikazuje nivoje globje arhitekture z globalno mrežo in njihove parametre. K predstavlja število kanalov izhoda nevronskne mreže, J pove velikost jedra, Ko je korak (ang. *stride*) uporabljen na nivoju, Akt predstavlja aktivacijo po vsakem nivoju in Reg določa stopnjo regularizacije. V vseh nivojih uporabljamo $L2$ regulirzacijo [?]. Zadnji konvolucijski nivo nima definiranega števila izhodnih kanalov, saj je to odvisno od pristopa, ki smo ga uporabili.

Zap. št.	Nivo	K	J	Ko	Akt	Reg
0	Vhod	1	-	-	-	-
1	2D konvolucija	64	3	1	Relu	0.01
2	Maks. združevanje	64	2	2	-	-
3	2D konvolucija	128	3	1	Relu	0.01
4	2D konvolucija	128	3	1	Relu	0.01
5	2D konvolucija	128	3	1	Relu	0.01
6	Vsota z 3	128	-	-	-	-
7	2D konvolucija	128	3	1	Relu	0.01
8	Maks. združevanje	128	2	2	-	-
9	2D konvolucija	256	3	1	Relu	0.01
10	2D konvolucija	256	3	1	Relu	0.01
11	2D konvolucija	256	3	1	Relu	0.01
12	Vsota z 9	256	-	-	-	-
13	2D konvolucija	256	3	1	Relu	0.01
14	Maks. združevanje	265	2	2	-	-
15	2D konvolucija	512	3	1	Relu	0.01
16	2D konvolucija	512	3	1	Relu	0.01

17	2D konvolucija	512	3	1	Relu	0.01
18	Vsota z 15	512	-	-	-	-
19	2D konvolucija	512	3	1	Relu	0.01
20	2D konvolucija	256	3	1	Relu	0.01
21	Združitev z globalno mrežo	512	-	-	-	-
22	2D konvolucija	128	3	1	Relu	0.01
23	2D transponirana kovnolucija	64	3	2	Relu	0.01
24	2D kovnolucija	64	3	1	Relu	0.01
25	2D kovnolucija	64	3	1	Relu	0.01
26	2D transponirana kovnolucija	64	3	2	Relu	0.01
27	2D kovnolucija	32	3	1	Relu	0.01
28	2D kovnolucija		3	1	Relu	0.01

Tabela B.3: Tabela prikazuje nivoje dopolnjene VGG mreže in njihove parametre. K predstavlja število kanalov izhoda nevronске mreže, J pove velikost jedra, Ko je korak (ang. *stride*) uporabljen na nivoju, Akt predstavlja aktivacijo po vsakem nivoju in Reg določa stopnjo regularizacije. V vseh nivojih uporabljamo $L2$ regulrizacijo [?]. Zadnji konvolucijski nivo nima definiranega števila izhodnih kanalov, saj je to odvisno od pristopa, ki smo ga uporabili. Vsako nadzorčenje prostorske dimenzije poveča za 2 krat.

Zap. št.	Nivo	K	J	Ko	Akt	Reg
0	Vhod	1	-	-	-	-
1	VGG-16 mreža	512	-	-	-	-
2	Nadvzorčenje	512	-	-	-	-
3	2D konvolucija	256	3	1	Relu	0.01
4	2D konvolucija	256	3	1	Relu	0.01
5	Nadvzorčenje	256	-	-	-	-
6	2D konvolucija	128	3	1	Relu	0.01
7	2D konvolucija	128	3	1	Relu	0.01
8	Nadvzorčenje	128	-	-	-	-

9	2D konvolucja	64	3	1	Relu	0.01
10	2D konvolucja	164	3	1	Relu	0.01
11	Nadvzorčenje	64	-	-	-	-
12	2D konvolucja	32	3	1	Relu	0.01
13	2D konvolucja		3	1	Relu	0.01

Literatura

- [1] N. Pm, R. M. Chezian, VARIOUS COLOUR SPACES AND COLOUR SPACE CONVERSION ALGORITHMS, Journal of Global Research in Computer Science 4 (1).
URL <http://jgrcs.info/index.php/jgrcs/article/viewFile/587/430>
- [2] S. Iizuka, Edgar Simo-Serra, H. Ishikawa, Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification, in: SIGGRAPH '16, Vol. 35, ACM, 2016, p. 110. doi:10.1145/2897824.2925974.
- [3] L. Prangnell, Visible Light-Based Human Visual System Conceptual Model.
URL <https://arxiv.org/pdf/1609.04830.pdf>
- [4] S. Bansal, D. Aggarwal, Color Image Segmentation Using CIELab Color Space Using Ant Colony Optimization.
URL <http://www.ijcset.net/docs/Volumes/volume1issue7/ijcset2011010715.pdf>
- [5] R. Zhang, P. Isola, A. Efros, Colorful image colorization, European Conference on Computer Vision.
URL http://link.springer.com/chapter/10.1007/978-3-319-46487-9{_}40

- [6] G. Larsson, M. Maire, G. Shakhnarovich, Learning Representations for Automatic Colorization, arXiv preprint [arXiv:1603.06668](https://arxiv.org/abs/1603.06668).
URL <http://arxiv.org/abs/1603.06668>
- [7] N. Ohta, A. R. Robertson, Colorimetry: Fundamentals and Applications, Vol. null, 2005. doi:10.1002/0470094745.
URL <https://books.google.com/books?hl=en&lr=&id=U8jeh1uhSHgC&oi=fnd&pg=PR2&dq=Colorimetry%20Fundamentals+and+Applications&ots=SUbub0kPfU&sig=FtBn43qgQvrKCOpXi7lAeVB0FGk>
- [8] J. Schwiegerling, Society of Photo-optical Instrumentation Engineers., Field guide to visual and ophthalmic optics, SPIE, 2004.
URL <https://spie.org/Publications/Book/592975>
- [9] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, in: ACM Transactions on Graphics (TOG), Vol. 23, ACM, 2004, pp. 689–694.
- [10] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, J.-L. Wu, An adaptive edge detection based colorization algorithm and its applications, in: Proceedings of the 13th annual ACM international conference on Multimedia, ACM, 2005, pp. 351–354.
- [11] M. Koleini, S. A. Mobadjemi, P. Moallem, Automatic Black and White Film Colorization Using Texture Features and Artificial Neural Networks, Journal of the Chinese Institute of Engineers 33 (7) (2010) 1049–1057.
- [12] P. Shirley, Color transfer between images, IEEE Corn 21 (2001) 34–41.
- [13] Y.-W. Tai, J. Jia, C.-K. Tang, Local color transfer via probabilistic segmentation by expectation-maximization, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, IEEE, 2005, pp. 747–754.

- [14] Z. Cheng, Deep Colorization, in: CVPR, Vol. 1, 2015, pp. 415–423.
- [15] A. Deshpande, J. Rock, D. Forsyth, Learning Large-Scale Automatic Image Colorization (2015). doi:10.1109/ICCV.2015.72.
- [16] R. Dahl, Automatic Colorization.
URL <http://tinyclouds.org/colorize/>
- [17] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition arXiv:1409.1556.
URL <http://arxiv.org/abs/1409.1556>
- [18] R. Zhang, P. Isola, A. A. Efros, Colorful Image Colorization, ECCV, 2016.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. doi: 10.1007/s11263-015-0816-y.
- [20] G. Larsson, M. Maire, G. Shakhnarovich, Learning Representations for Automatic Colorization, arXiv preprint arXiv:1603.06668.
- [21] A. Gibson, C. Nicholson, J. Patterson, Introduction to Deep Neural Networks.
URL <https://deeplearning4j.org/neuralnet-overview>
- [22] A. Karpathy, CS231n Convolutional Neural Networks for Visual Recognition, University Lecture (2016).