

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

SOMENTOR:

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 mentorja<sup>1</sup>.

©2017 PRIMOŽ GODEC

---

<sup>1</sup>V dogovorju z mentorjem lahko kandidat magistrsko delo s pripadajočo izvirno 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 [5].



## 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**

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled Področja</b>	<b>5</b>
2.1	Globoke nevronske mreže . . . . .	5
2.2	Predstavite podatkov . . . . .	7
2.3	Obstoječe metode . . . . .	10
<b>3</b>	<b>Sklicevanje na besedilne konstrukte</b>	<b>13</b>
<b>4</b>	<b>Plovke: slike in tabele</b>	<b>15</b>
4.1	Formati slik . . . . .	15
<b>5</b>	<b>Razno</b>	<b>19</b>
5.1	Notacije . . . . .	19
5.2	Lepe tabele in psevdokoda . . . . .	19
<b>6</b>	<b>Kaj pa literatura?</b>	<b>21</b>
<b>7</b>	<b>Sklepne ugotovitve</b>	<b>23</b>
<b>A</b>	<b>Title of the appendix 1</b>	<b>25</b>



# Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
DBMS	database management system	sistem za upravljanje podatkovnih baz
SVM	support vector machine	metoda podpornih vektorjev
...	...	...



# Povzetek

**Naslov:** Barvanje črnobelih slik z globokimi modeli

V vzorcu je predstavljen postopek priprave magistrskega dela z uporabo okolja L<sup>A</sup>T<sub>E</sub>X. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavanega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

## Ključne besede

*Umetna inteligenca, Odkiravanje znanj iz podatkov, globoko učenje, nevronske mreže*



# Abstract

**Title:** Deep models for image coloring

This sample document presents an approach to typesetting your BSc thesis using L<sup>A</sup>T<sub>E</sub>X. 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*





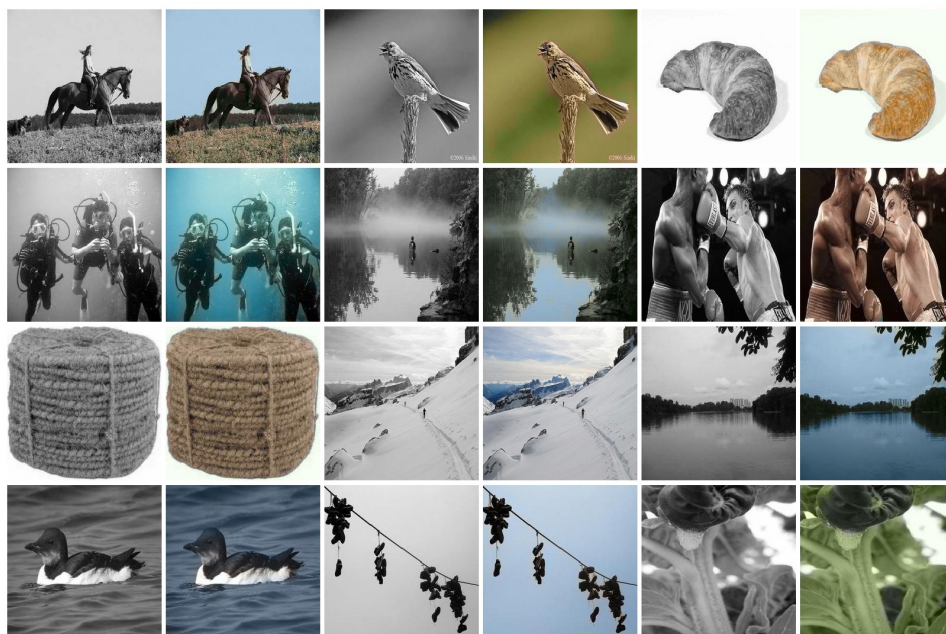
# Poglavje 1

## Uvod

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 1.1, 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 opraviu 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 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 nevronske mreže, 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 za-



**Slika 1.1:** 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 algoritem in obarvana slika - izhod algoritma.

stonj 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 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 najbolj delovala na slikah prilagodili še za video.



# Poglavje 2

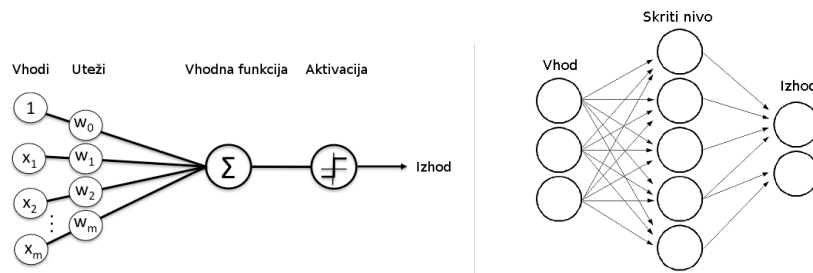
## Pregled Področja

### 2.1 Globoke nevronske mreže

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

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. 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 2.1. 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 nelinearnot, 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



**Slika 2.1:** Leva slika prikazuje zgradbo enega vozlišča, ki ima zasnovo podobno nevronom v možganih, desna pa zgradbo več nivojske nevronske mreže. Iz: Introduction to Deep Neural Networks [urlhttps://deeplearning4j.org/neuralnet-overview](https://deeplearning4j.org/neuralnet-overview) in Neural Networks, [urlhttp://docs.opencv.org/2.4/modules/ml/doc/neural\\_networks.html](http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html) (dostopano: 21. junij 2017)

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 po definiciji vsaj dva skrita nivoja, večinoma pa mnogo več. Vsak nivo globoke nevronske mreže prepozna določen lastnosti vhodnih podatkov. Nivoji, ki se nahajajo globlje lahko prepoznajo bolj kompleksne lastnosti podatkov, saj na vhodu, dobijo lastnosti oziroma aktivacije nivoja pred njim.

Da nevronska mreža daje zadovoljive rezultate je potrebno utežem določiti 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.

### 2.1.1 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*), uporabljamo 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  $\times$  višina  $\times$  globina*. 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.

## 2.2 Predstavite podatkov

Slike, ki jih uporabljamo za učenje so shranjene v *RGB* [?] barvnem prostoru. Kot je pokazano v [?] 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 [?].

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 .

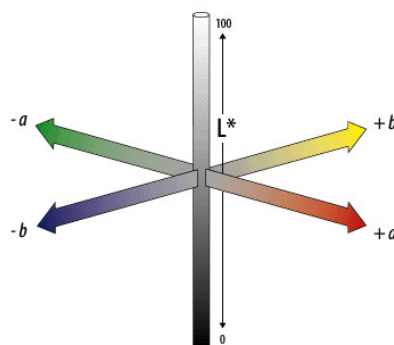
- **Nima kanala ločenega kanala za svetlost** Glede na to, da modeli za barvanje napovedujejo le barvne elemente v sliki, svetlost pa se vzame iz originalne slike, je najbolj priročno, če uporabljamo barvni prostor, ki ima ločen kanal za svetlost.

### 2.2.1 Izbira primerne barvnega prostora

Na podlagi teh predpostavk je izbira prostorov omejena na *Lab* [?], *YUV* in *HSL* [?]. Vsi ustrezajo drugi predpostavki iz 2.2. Edini, ki zares ustreza prvi predpostavki je *Lab*. Iz ugotovitev iz sorodnih del [?, ?, ?] se tudi najboljše 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 [?]. 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\** se 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 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 2.2:** 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,  $b^*$  se razteza od modre proti rumeni. Nasprotujoče barve na kanalih  $a^*$  in  $b^*$  se nikoli ne kombinirajo v odtenek. Iz: Adobe, Technical Guid, CIELAB, [http://dba.med.sc.edu/price/irf/Adobe\\_tg/models/cielab.html](http://dba.med.sc.edu/price/irf/Adobe_tg/models/cielab.html) (dostopano: 24. junij 2017)

**Tabela 2.1:** Največje in najmanjše 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

### 2.2.2 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 izbire referenčne bele barve. Običajno se izbere referenčno temperaturo belo točke  $D65$ , ki je tudi standardizirana<sup>1</sup>[?].
3. **Pretvorba iz XYZ v  $L^*a^*b^*$**  se izvede po enačbah opisanih v [?].

## 2.3 Obstoječe metode

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

### 2.3.1 Metode, ki zahtevajo interakcijo uporabnika

To skupino metod delimo na tehnike, ki temeljijo na uporabnikovem barvanju manjših delov slik (ang. *scribble based*) [?, ?] in tiste, ki temeljijo na primerih (ang. *example based*) [?, ?, ?]. Pri prvih uporabnik določi barvo določenih točk v sliki, te pa algoritem avtomatsko razširi preko cele slike. Pri barvanju

---

<sup>1</sup>Zapis na uradni strani komisije International Commission on Illumination (krajše CIE), ki je postavila standard: [http://cie.co.at/index.php?i\\_ca\\_id=484](http://cie.co.at/index.php?i_ca_id=484)

na primerih pa mora uporabnik izbrati sliko, ki je podobna tisti, ki jo želimo pobarvati, algoritem 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 algoritem sam razširi lastnosti ročno barvane slike.

### 2.3.2 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 nevronske mreže [?], ki pa vsebuje zgolj polnopravne nivoje, druga pa za barvanje uporabi metodo naključnih gozdov (ang. *random forest*) [?].

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 [?] gradi mrežo na podlagi že zgrajene šestnajst-nivojske mreže VGG-16, ki so jo razvili na univerzi v Oxfordu [?]. 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 nevronske mreže, kar pomeni, da so problem spremenili iz regresijskega v klasifikacijskega. Zang in sod. [?] uporabijo konvolucijsko nevronske mreže 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 zbirki Imagenet [?]. Uporabljajo barvni prostor  $L*a*b$ . Larsson in sod. [?] 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. [?] uporabijo nevronska mrežo sestavljeno 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 napoveduje histograma na podlagi kvantiziranega prostora ampak direktno  $a*$  in  $b*$  vrednost, kar pomeni, da ne uporablja klasifikacije ampak regresijo.

## Poglavje 3

# Sklicevanje na besedilne konstrukte

Matematična ali popolna indukcija je eno prvih orodij, ki jih spoznamo za dokazovanje trditev pri matematičnih predmetih.

**Izrek 3.1** *Za vsako naravno število  $n$  velja*

$$n < 2^n. \tag{3.1}$$

*Dokaz.* Dokazovanje z indukcijo zahteva, da neenakost (3.1) najprej preverimo za najmanjše naravno število — 0. Res, ker je  $0 < 1 = 2^0$ , je neenaka (3.1) za  $n = 0$  izpolnjena.

Sledi indukcijski korak. S predpostavko, da je neenakost (3.1) veljavna pri nekem naravnem številu  $n$ , je potrebno pokazati, da je ista neenakost v veljavi tudi pri njegovem nasledniku — naravnem številu  $n + 1$ . Računajmo.

$$n + 1 < 2^n + 1 \tag{3.2}$$

$$\leq 2^n + 2^n \tag{3.3}$$

$$= 2^{n+1}$$

Neenakost (3.2) je posledica indukcijske predpostavke, neenakost (3.3) pa enostavno dejstvo, da je za vsako naravno število  $n$  izraz  $2^n$  vsaj tako velik kot 1. S tem je dokaz Izreka 3.1 zaključen.  $\square$

Opazimo, da je  $\text{\LaTeX}$  številko izreka podredil številki poglavja.

## Poglavje 4

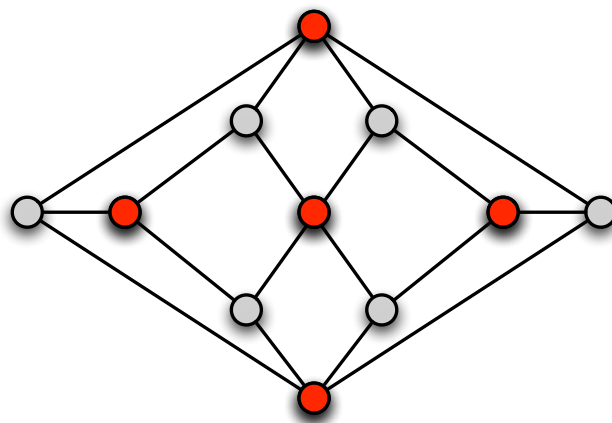
# Plovke: slike in tabele

Slike in daljše tabele praviloma vključujemo v dokument kot plovke. Pozicija plovke v končnem izdelku ni pogojena s tekom besedila, temveč z izgledom strani.  $\text{\LaTeX}$  bo skušal plovko postaviti samostojno, praviloma na vrh strani, na kateri se na takšno plovko prvič sklicujemo. Pri tem pa bo na vsako stran končnega izdelka želel postaviti tudi sorazmerno velik del besedila. V skrajnem primeru, če imamo res preveč plovk, se bo odločil za stran popolnoma zapolnjeno s plovkami.

### 4.1 Formati slik

Bitne slike, vektorske slike, kakršnekoli slike, z  $\text{\LaTeX}$ om lahko vključimo vse. Slika 4.1 je v `.pdf` formatu. Pa res lahko vključimo slike katerihkoli formatov? Žal ne. Programski paket  $\text{\LaTeX}$  lahko uporabljamo v več dialektih. Ukaz `latex` ne mara vključenih slik v formatu Portable Document Format `.pdf`, ukaz `pdflatex` pa ne prebavi slik v Encapsulated Postscript Formatu `.eps`. Strnjeno v Tabeli 4.1.

Nasvet? Odločite se za uporabo ukaza `pdflatex`. Vaš izdelek bo brez vmesnih stopenj na voljo v `.pdf` formatu in ga lahko odnesete v vsako tiskarno. Če morate na vsak način vključiti sliko, ki jo imate v `.eps` formatu, jo vnaprej pretvorite v alternativni format, denimo `.pdf`.

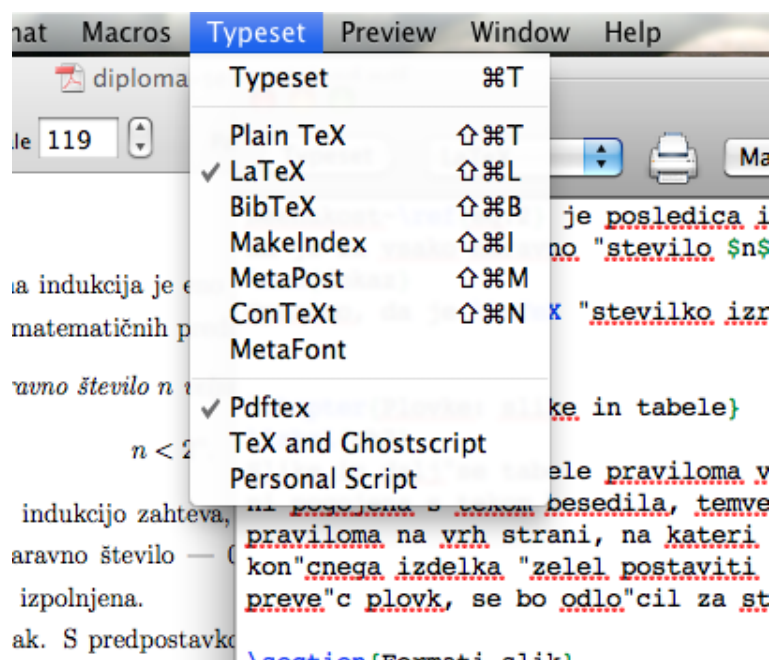


**Slika 4.1:** Herschelov graf, vektorska grafika.

**Tabela 4.1**

ukaz/format	.pdf	.eps	ostali formati
<code>pdflatex</code>	da	ne	da
<code>latex</code>	ne	da	da





Slika 4.2: Kateri dialekt uporabljati?

Včasih se da v okolju za uporabo programskega paketa  $\text{\LaTeX}$  nastaviti na kakšen način bomo prebavljali vhodne dokumente. Spustni meni na Sliki 4.2 odkriva uporabo  $\text{\LaTeX}$ a v njegovi pdf inkarnaciji — `pdflatex`. Vključena Slika 4.2 je seveda bitna.



# Poglavje 5

## Razno

### 5.1 Notacije

Za notacijo spremenljivk ter skalarjev uporabimo običajno notacijo, t.j., spremenljivka  $x$  in skalar  $a$ . Pri notaciji matrik ter vektorjev pa se poslužujemo krepega fonta. Torej, matrika  $\mathbf{A}$  ter vektor  $\mathbf{v}$ ,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1q} \\ a_{21} & a_{22} & \dots & a_{2q} \\ \vdots & & & \\ a_{p1} & a_{p2} & \dots & a_{pq} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix}.$$

### 5.2 Lepe tabele in psevdokoda

Psevdokoda 1 prikazuje primer delovanja genetskega algoritma, medtem ko Tabela 5.1 prikazuje primer lepe tabele brez vertikalnih črt.

**Tabela 5.1:** Primer enostavne tabele.

Ime	Vrednost	Opis
$a$	0.03	skalar
$x$	-1	spremenljivka

---

**Algorithm 1** Psevdokoda genetskega algoritma

---

```
1:  $t \leftarrow 0$ 
2:  $InitPopulation[P(t)] \leftarrow$  inicializiraj populacijo
3:  $EvalPopulation[P(t)] \leftarrow$  evaluiraj populacijo
4: repeat
5:    $P'(t) \leftarrow Variation[P(t)] \leftarrow$  generiraj novo populacijo
6:    $EvalPopulation[P'(t)] \leftarrow$  evaluiraj novo populacijo
7:    $P(t+1) \leftarrow ApplyGeneticOperators[P'(t) \in Q]$ 
8:    $t \leftarrow t + 1$ 
9: until prekinitev
10: if rezultat dovolj dober then
11:   shrani rezultat
12: end if
```

---

## Poglavje 6

# Kaj pa literatura?

Kot smo omenili že v uvodu, je pravi način za citiranje literature uporaba `BIBTeX` [4]. Programski paket `LaTeX` je prvotno predstavljen v priročniku [3] in je v resnici nadgradnja sistema `TeX` avtorja Donalda Knutha, znanega po denimo, če izpustim njegovo umetnost programiranja, Knuth-Bendixovem algoritmu [2].

Vsem raziskovalcem s področja računalništva pa svetujem v branje mnenje L. Fortnowa [1].



## Poglavje 7

# Sklepne ugotovitve

Izbira  $\text{\LaTeX}$  ali ne  $\text{\LaTeX}$  je seveda prepuščena vam samim. Res je, da so prvi koraki v  $\text{\LaTeX}$ u težavni. Ta dokument naj vam služi kot začetna opora pri hoji.





# **Dodatek A**

## **Title of the appendix 1**

Example of the appendix.



# Literatura

- [1] L. Fortnow, “Viewpoint: Time for computer science to grow up”, *Communications of the ACM*, št. 52, zv. 8, str. 33–35, 2009.
- [2] D. E. Knuth, P. Bendix. “Simple word problems in universal algebras”, v zborniku: *Computational Problems in Abstract Algebra* (ur. J. Leech), 1970, str. 263–297.
- [3] L. Lamport. *LaTEX: A Document Preparation System*. Addison-Wesley, 1986.
- [4] O. Patashnik (1998) BiBT<sub>E</sub>Xing. Dostopno na: <http://ftp.univie.ac.at/packages/tex/biblio/bibtex/contrib/doc/btxdoc.pdf>
- [5] licence-cc.pdf. Dostopno na: <https://ucilnica.fri.uni-lj.si/course/view.php?id=274>