

# Klasifikacija visokodimenzionalnih podataka temeljena na konvoluciji

---

Milotić, Josip

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:324477>

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

Download date / Datum preuzimanja: **2024-06-30**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Diplomski sveučilišni studij računarstva

Diplomski rad

**KLASIFIKACIJA  
VISOKODIMENZIONALNIH PODATAKA  
TEMELJENA NA KONVOLUCIJI**

Rijeka, studeni 2021.

Josip Milotić  
0069073972

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Diplomski sveučilišni studij računarstva

Diplomski rad

**KLASIFIKACIJA  
VISOKODIMENZIONALNIH PODATAKA  
TEMELJENA NA KONVOLUCIJI**

Mentor: doc. dr. sc. Goran Mauša

Rijeka, studeni 2021.

Josip Mlotić  
0069073972

Umjesto ove stranice umetnuti zadatak  
za završni ili diplomski rad

## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio diplomski rad prema zadatku povjerenstva za diplomske ispite diplomskog sveučilišnog studija računarstva, pod vodstvom mentora doc. dr. sc. Gorana Mauše.

Rijeka, studeni 2021.

-----  
Josip Milotić

# Zahvala

Zahvaljuem se mentoru doc. dr. sc. Goranu Mauši za pruženu pomoć i savjete pri izradi diplomskog rada, te svojoj obitelji i prijateljima koji su mi pružali podršku za vrijeme studija.

# Sadržaj

<b>Popis slika</b>	<b>viii</b>
<b>Popis tablica</b>	<b>x</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 Konvolucijske neuronske mreže</b>	<b>3</b>
2.1 Konvolucijske neuronske mreže . . . . .	3
2.1.1 Konvolucijski sloj . . . . .	7
2.1.2 Sažimajući sloj . . . . .	10
2.1.3 Potpuno povezani sloj . . . . .	12
2.2 Aktivacijske funkcije . . . . .	12
2.2.1 Linearna aktivacijska funkcija . . . . .	12
2.2.2 Sigmoidna aktivacijska funkcija . . . . .	13
2.2.3 Aktivacijska funkcija tangens hiperbolni . . . . .	14
2.2.4 Rectified Linear Unit . . . . .	16
2.3 Regularizacija . . . . .	17
2.3.1 Regularizacije L1 i L2 . . . . .	17
2.3.2 Rano zaustavljanje . . . . .	18
2.3.3 Dropout . . . . .	18

## *Sadržaj*

<b>3 Metodologija</b>	<b>20</b>
3.1 Skupovi podataka . . . . .	20
3.2 Predobrada podataka . . . . .	22
3.3 Algoritam transformacije podataka u slike . . . . .	23
3.3.1 Algoritam transformacije u matricu udaljenosti . . . . .	24
3.3.2 Algoritam transformacije u histogram . . . . .	25
3.4 Arhitektura neuronske mreže . . . . .	28
3.5 Metrike vrednovanja . . . . .	31
<b>4 Rezultati</b>	<b>34</b>
<b>5 Zaključak</b>	<b>40</b>
<b>Bibliografija</b>	<b>42</b>
<b>Pojmovnik</b>	<b>45</b>
<b>Sažetak</b>	<b>45</b>
<b>A Numerički prikaz rezultata</b>	<b>46</b>



# Popis slika

2.1	Općenita arhitektura neuronske mreže . . . . .	4
2.2	Usporedba konvolucije i unakrsne korelacije . . . . .	5
2.3	Generična arhitektura konvolucijske neuronske mreže . . . . .	6
2.4	Postupak konvolucije . . . . .	8
2.5	Mapa značajki naučena u prvom sloju . . . . .	8
2.6	Usporedba oblaganja postupcima valid i same . . . . .	9
2.7	Spajanje mapa značajki . . . . .	10
2.8	Usporedba max i average pooling algoritama . . . . .	11
2.9	Linearna aktivacijska funkcija . . . . .	13
2.10	Sigmoidna aktivacijska funkcija . . . . .	14
2.11	Aktivacijska funkcija tangens hiperbolni . . . . .	15
2.12	ReLU i Leaky ReLU aktivacijske funkcije . . . . .	16
2.13	Early stopping regularizacija . . . . .	19
2.14	Dropout regularizacija . . . . .	19
3.1	Slikovni prikaz dobiven transformacijom sonar skupa podataka . . .	24
3.2	Slikovni prikaz dobiven transformacijom uzorka iz skupa podataka Swarm behavior u matricu . . . . .	25
3.3	Slikovni prikaz dobiven transformacijom sonar skupa podataka . . .	26

*Popis slika*

3.4	Slikovni prikaz dobiven transformacijom uzorka iz Swarm behavior skupa podataka u histogram . . . . .	27
3.5	Arhitektura neuronske mreže . . . . .	29
4.1	Metrike za vrijeme treniranja umjetne neuronske mreže . . . . .	36
4.2	Metrike za vrijeme treniranja konvolucijske neuronske mreže . . . . .	37
4.3	Kutijasti dijagrami za metrike točnosti i F1 na skupu za testiranje . . . . .	38

# Popis tablica

3.1	Svojstva korištenih skupova podataka . . . . .	21
3.2	Hiperparametri konvolucijske neuronske mreže . . . . .	30
3.3	Hiperparametri neuronske mreže . . . . .	31
3.4	Matrica zabune . . . . .	32
4.1	Veličine skupova podataka u različitim formatima . . . . .	34
4.2	Prosječno vrijeme treniranja . . . . .	35
4.3	Točnost na validacijskom skupu . . . . .	36
4.4	Najbolja dobivena točnost na skupu za testiranje . . . . .	39
A.1	Točnost na testnom skupu . . . . .	46
A.2	F1 metrika na testnom skupu . . . . .	46

# Poglavlje 1

## Uvod

Konvolucijske neuronske mreže su jedan od najpopularnijih algoritama dubokog učenja, te se najčešće koriste u zadacima obrade slike, posebno klasifikacije i računalnog vida, te u tim područjima postižu vrlo dobre rezultate. Međutim, konvolucijske neuronske mreže primarno služe za obradu dvodimenzionalnih i u nekim slučajevima trodimenzionalnih podataka, ali nisu namijenjene za rad s visokodimenzionalnim podacima [1].

Kod klasifikacije podataka korištenjem umjetnih neuronskih mreža, količina podataka može predstavljati velike probleme. Obrada velikih količina podataka s mnogo značajki može biti spora i zahtijevati velike količine ljudskih i računalnih resursa. Zbog toga je potrebno veliku pažnju posvetiti pripremanju podataka i odabiru onih značajki koje najviše doprinose klasifikaciji. Taj proces može biti vrlo vremenski zahtjevan, ali također ponekad zahtjeva i dodatne ljudske resurse jer je za pojedine probleme potrebno uposliti i stručnjaka za tu domenu [2].

Zbog načina na koji konvolucijske neuronske mreže rade, ručni odabir značajki nije potreban zbog toga što mreža sama nauči važne značajke korištenjem konvolucijskih slojeva [2]. Zadatak je isprobati različite metode transformacije podataka u slikovni format koji je moguće klasificirati konvolucijskim neuronskim mrežama, i time smanjiti potrebu za pripremom i obradom podataka.

Cilj ovog rada je istražiti načine transformacije visokodimenzionalnih podataka u dvodimenzionalni oblik, te iskoristiti konvolucijske neuronske mreže za njihovu

## *Poglavlje 1. Uvod*

klasifikaciju. Na četiri korištena skupa podataka bit će testirana dva algoritma transformacije visokodimenzionalnih podataka u slikovni oblik - algoritam transformacije u histogram i algoritam transformacije u matricu udaljenosti. Njihove će performanse biti uspoređene međusobno, ali i s umjetnim neuronskim mrežama kao predstavnikom klasičnih algoritama za klasifikaciju podataka u numeričkom i visokodimenzionalnom obliku.

Rad je strukturiran na sljedeći način. U idućem poglavlju će biti dan pregled konvolucijskih neuronskih mreža i njihovih sastavnih dijelova, te će biti objašnjen koncept konvolucije. Zatim će detaljno biti opisana metodologija rada, korišteni skupovi podataka, algoritmi transformacije podataka u dvodimenzionalni oblik, korištena arhitektura neuronske mreže, te metrike vrednovanja rezultata. Na posljertku će biti predstavljeni i komentirani dobiveni rezultati, te predloženi načini njihova poboljšanja.

## Poglavlje 2

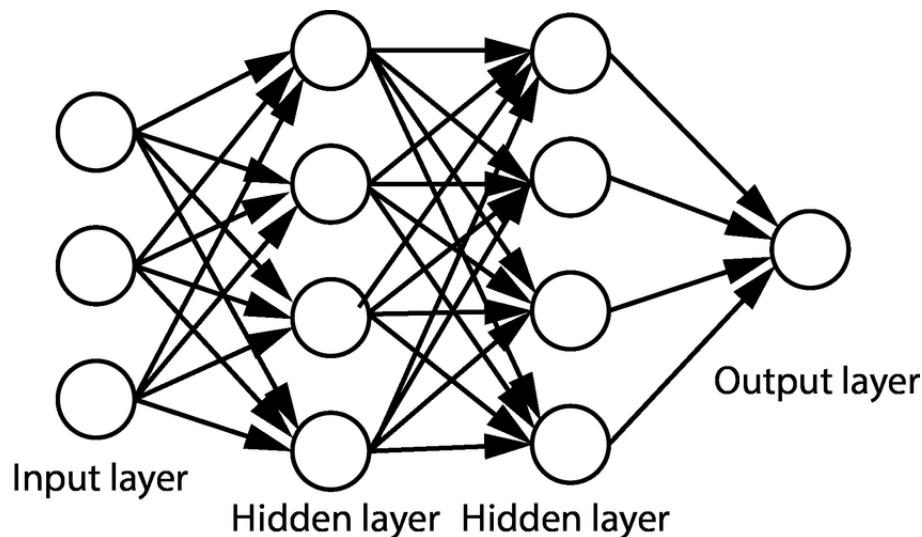
# Konvolucijske neuronske mreže

U ovom će poglavlju biti opisan algoritam konvolucijskih neuronskih mreža korišten u ovom radu, te proces konvolucije na kojem se temelje. Također će biti dan pregled često korištenih aktivacijskih funkcija i regularizacijskih metoda, te će biti objašnjena njihova uloga u neuronskoj mreži.

### 2.1 Konvolucijske neuronske mreže

Neuronske mreže su računalni sustavi inspirirani neuronima u biološkom mozgu. Tipično, umjetne neuronske mreže se sastoje od čvorova (umjetnih neurona), međusobno povezanih i organiziranih u slojeve. Neuroni su male jedinice koje na ulazu primaju ulazni impuls, obrađuju ga na temelju aktivacijske funkcije, te ga propagiraju dalje kroz neuronsku mrežu. Općenito, umjetne neuronske mreže rade s numeričkim podacima, ali postoje različite varijante za rad s drugim vrstama ulaznih podataka. Opća struktura neuronske mreže je prikazana na slici 2.1.

Konvolucijske neuronske mreže su specijalizirana vrsta neuronskih mreža koje služe za obradu podataka organiziranih u topologiju rešetke (eng. *grid topology*), kao što su slike. Kao što ime kaže, konvolucijske neuronske mreže se zasnivaju na matematičkoj operaciji konvolucije, te dodaju dodatne slojeve u neuronsku mrežu, čija je uloga izvlačenje značajki iz slika. Te se značajke kasnije koriste za treniranje mreže [3].



Slika 2.1 Općenita arhitektura neuronske mreže (preuzeto iz: [4])

Matematički, konvolucija se može opisati kao operacija koja na temelju dvije funkcije daje treću funkciju koja prikazuje utjecaj jedne ulazne funkcije na drugu. Postupak konvolucije može se opisati izrazom 2.1:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{+\infty} x(a)w(t - a) \quad (2.1)$$

gdje je:

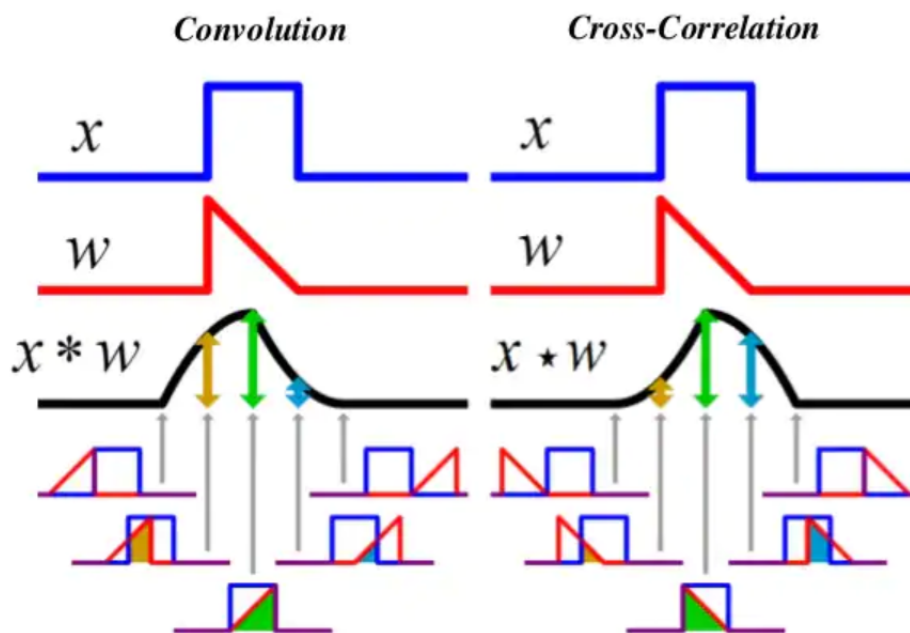
- $t$  - vremenski indeks
- $x(t)$  - prva ulazna funkcija
- $w(t)$  - druga ulazna funkcija

U slučaju konvolucijskih neuronskih mreža, ulazna vrijednost neće biti 1-D podatak, već će se koristiti 2-D podatci, kao što su slike. U takvom slučaju se postupak konvolucije primjenjuje na više osi istovremeno. Primjerice, ako je kao ulazni podatak dana dvodimenzionalna slika  $I$  i dvodimenzionalna jezgrena funkcija (eng. *kernel*)  $K$ , prethodni se izraz može zapisati kao izraz 2.2:

## Poglavlje 2. Konvolucijske neuronske mreže

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.2)$$

Postupak konvolucije se implementira u konvolucijskom sloju konvolucijskih neuronskih mreža, koji će detaljnije biti opisan kasnije. Važno je za napomenuti da se kod nekih programskih knjižnica umjesto postupka konvolucije koristi postupak unakrsne korelacije (eng. *cross-correlation*) koji je vrlo sličan konvoluciji [3, 5]. Vizualni prikaz ovih postupaka je vidljiv na slici 2.2.



Slika 2.2 Usporedba konvolucije i unakrsne korelacije (preuzeto iz: [6])

Konvolucija sadrži 3 svojstva koja uvelike pomažu pri poboljšanju algoritma strojnog učenja [3, 7]:

- rijetke interakcije,
- dijeljenje parametara,
- ekvivarijantna reprezentacija.

Svojstvo rijetkih interakcija (eng. *sparse interaction*) se postiže korištenjem jezgrene funkcije čija je veličina manja od veličine ulazne slike. Ovakvim se pristupom

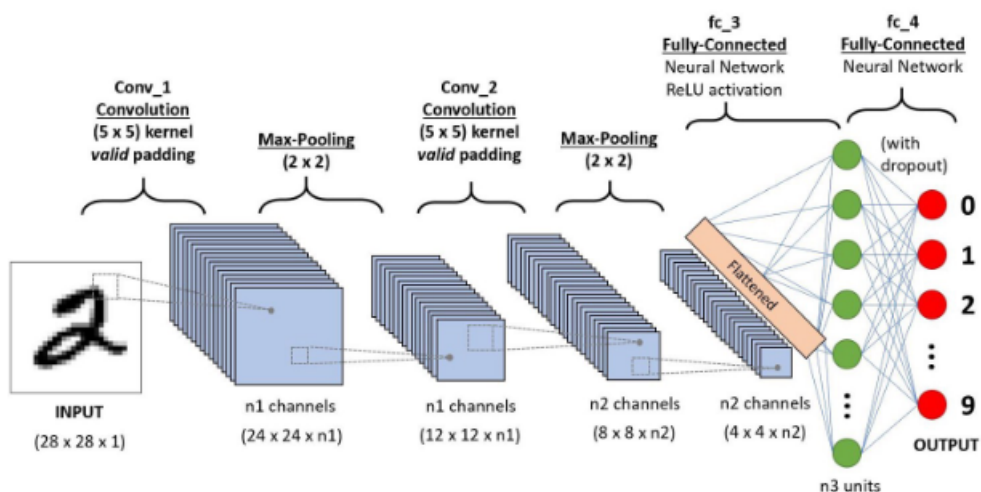


## Poglavlje 2. Konvolucijske neuronske mreže

algoritam može usredotočiti na manje dijelove slike čime se postiže lokalizirana identifikacija značajki. Samim time je broj značajki koje je potrebno pohraniti znatno manji, čime se smanjuje memorija potrebna za njihovu pohranu, te broj operacija potrebnih za računanje izlazne vrijednosti što povećava efikasnost algoritma.

Dijeljene parametara (eng. *parameter sharing*) se koristi za kontroliranje broja parametara i težinskih faktora u mreži. Kod umjetne neuronske mreže, svaki od težinskih faktora se koristi samo jednom, odnosno primjenjuje se samo na jednu vrijednost u mreži. S druge strane, kod konvolucijske neuronske mreže, svaki od parametara jezgrene funkcije će se iskoristiti jednom na svakom položaju na ulaznoj slici, što znači da se novi parametri neće morati računati ponovno za svaku od pozicija, već se računaju samo jednom. Ovim pristupom se također postiže smanjeno korištenje memorije.

Svojstvo ekvivarijantne reprezentacije (eng. *equivariant representation*) znači da se za svaku promjenu ulaznih vrijednosti izlazne vrijednosti mijenjaju na isti način. Ovo garantira da će za svaku promjenu na originalnoj ulaznoj slici, rezultat konvolucije imati identične promjene. Ovim svojstvom se omogućava dijeljenje značajki kroz sve slojeve mreže, neovisno od toga gdje su naučene.



Slika 2.3 Generična arhitektura konvolucijske neuronske mreže (preuzeto iz: [8])

## Poglavlje 2. Konvolucijske neuronske mreže

Kao što je ranije spomenuto, konvolucijske neuronske mreže koriste nekoliko dodatnih vrsta slojeva uz potpuno povezani sloj koji se koristi kod umjetnih neuronskih mreža. Glavna svrha tih dodatnih slojeva je identifikacija i izdvajanje važnih značajki iz slike, koje se potom koriste za daljnje treniranje mreže. Opća arhitektura konvolucijske neuronske mreže prikazana je na slici 2.3.

### 2.1.1 Konvolucijski sloj

Konvolucijski sloj je jedan od temeljnih slojeva konvolucijskih neuronskih mreža zbog toga što omogućava učenje značajki iz slika. Također je i sloj u kojem se obavlja najviše računski skupih operacija, pa se može smatrati i centralnim slojem konvolucijske neuronske mreže. Primarna svrha konvolucijskog sloja je odabir značajki iz danih ulaznih podataka, odnosno slika [9]. Funkcionalnost konvolucijskog sloja može se podijeliti na dva dijela:

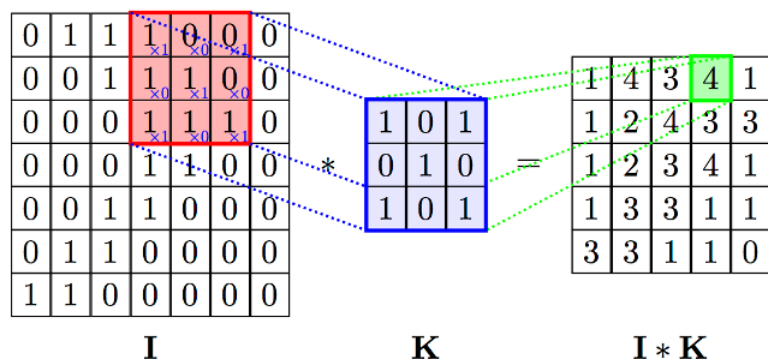
- Prepoznavanje značajki korištenjem konvolucije
- Primjena aktivacijske funkcije za dobivene značajke

U prvom dijelu konvolucijskog sloja, na sliku se primjenjuje niz malih filtera koji pomicanjem preko slike u malim koracima generiraju mapu značajki koje se mogu naučiti. Filteri se mogu shvatiti i kao skup od  $n$  neurona koji zajednički tvore jednu značajku koju, pomoću konvolucije, pretražuju u cijeloj slici. Postupak konvolucije ulazne slike s filtrom je prikazan na slici 2.4. Funkcionalnosti konvolucijskog sloja je moguće modificirati promjenom parametara modela:

- Broj filtara  $K$
- Prostorna dimenzija filtra  $F$
- Korak  $S$
- Količina obloge  $P$

Brojem i prostornom dimenzijom filtara se određuju dimenzije dobivenog izlaza iz konvolucijskog sloja. Sam filtar je definiran kao matrica dimenzija  $F \times F$ . Kao

Poglavlje 2. Konvolucijske neuronske mreže



Slika 2.4 Postupak konvolucije. Filtar  $K$  se pomiče preko ulazne slike  $I$ , te se njihovi elementi množe. Dobivene vrijednosti ćelija se potom zbrajaju, čime se dobiva nova vrijednost ćelije, te se smanjuju dimenzije ulazne slike.

što je ranije spomenuto, zbog svojstva rijetkih interakcija, dimenzije filtra se uvijek postavljaju na manju vrijednost od dimenzija ulazne slike, čime se stavlja naglasak na lokaliziranu identifikaciju značajki. Ako takav filter primijenimo na ulaznu sliku dimenzija  $N \times N$ , dobivamo mapu značajki dimenzija  $(N - F + 1) \times (N - F + 1)$ , koja će sadržavati težinske faktore neuronske mreže. Inicijalno, mapa značajki će se sastojati od jednostavnih oblika, koji u kasnijim slojevima mreže postaju kompleksniji oblici i konkretni objekti [10]. Primjer mape značajki dobivene u prvom sloju konvolucijske neuronske mreže prikazan je na slici 2.5.



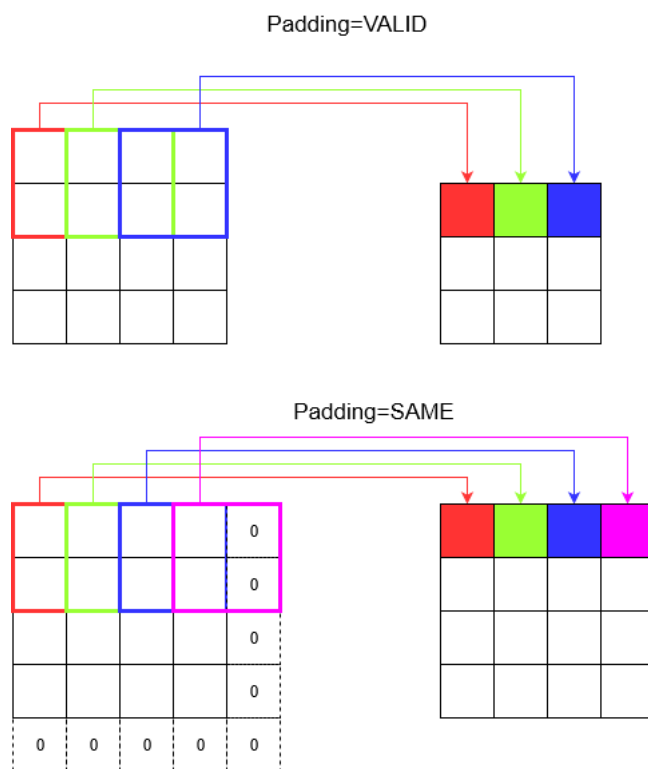
Slika 2.5 Mapa značajki naučena u prvom sloju mreže (preuzeto iz: [9])

Korak (eng. *stride*) određuje broj piksela za koji će se filter pomaknuti u svakoj

## Poglavlje 2. Konvolucijske neuronske mreže

iteraciji. Veći korak znači da će se smanjiti preklapanje kod operacije konvolucije [11]. To znači da se povećanjem ovog parametra dobiva manja mapa značajki, što se često koristi u sažimajućem sloju konvolucijske neuronske mreže, koji će biti objašnjen u sljedećem potpoglavlju.

Količina obloge (eng. *padding*) je parametar koji se koristi zajedno s parametrom koraka da bi se kontroliralo smanjenje izlazne mape značajki. Ovaj parametar služi za nadopunjavanje mape značajki nulama na rubovima, čime možemo spriječiti da se dobivena mapa značajki smanji u procesu konvolucije. Dva najčešće korištena načina oblaganja su *valid* i *same* oblaganje. Kod *valid* načina, mapa značajki se ne oblaže nulama, što znači da će se njene dimenzije smanjiti. S druge strane, pristup *same* oblaže mapu značajki nulama, i to tako da dobivena izlazna mapa zadrži jednake dimenzija kao ulazna [11]. Usporedba ovih pristupa je vidljiva na slici 2.6.

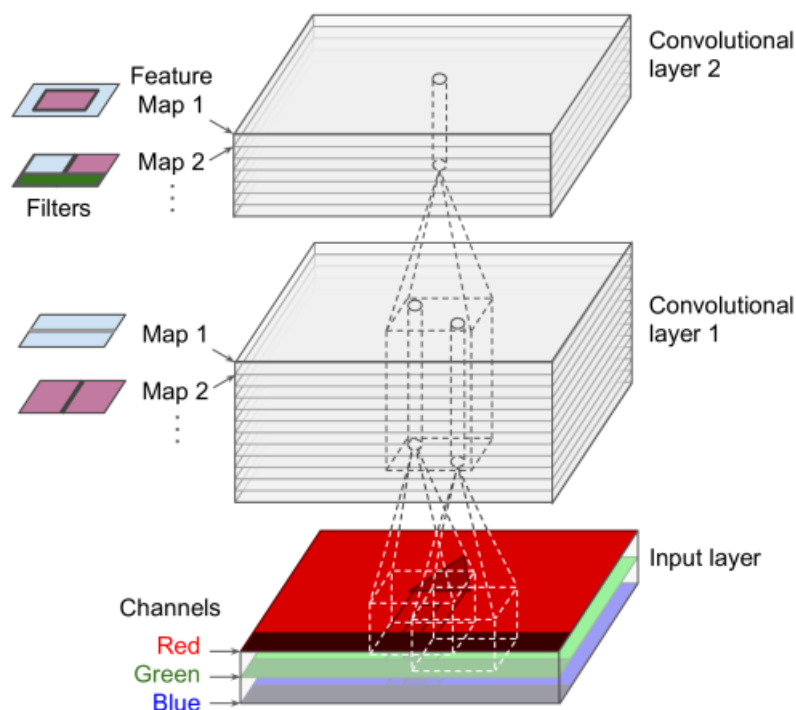


Slika 2.6 Usporedba oblaganja postupcima *valid* i *same* (preuzeto iz: [11])

Na dobivene značajke se zatim primjenjuje aktivacijska funkcija, koja obavlja

## Poglavlje 2. Konvolucijske neuronske mreže

transformaciju značajki da bi se one mogle proslijediti kao ulazna vrijednost i idući sloj. U slučaju konvolucijskih neuronskih mreža, kao aktivacijska funkcija se najčešće koristi *ReLU*. Glavno svojstvo ove funkcije je da za pozitivne ulaze vraća jednak izlaz, dok za negativne ulaze na izlazu vraća 0. Ovime se postiže otpornost na problem nestajućih gradijenata i postižu se bolje performanse pri treniranju. U slučaju da se koristi veći broj filtara, za svaki od njih ćemo dobiti jednu takvu mapu značajki. Tako dobivene mape značajki se potom slažu jedna na drugu te se propagiraju dalje kroz neuronsku mrežu kao ulaz u sljedeći sloj mreže, kao što je prikazano na slici 2.7.



Slika 2.7 Spajanje mapa značajki (preuzeto iz: [9])

### 2.1.2 Sažimajući sloj

Još jedan od često korištenih slojeva u konvolucijskoj neuronskoj mreži je sažimajući sloj (eng. *pooling layer*). Ovaj se sloj umeće između konvolucijskih slojeva mreže i glavna funkcionalnost mu je smanjenje dimenzija ulazne slike. Ovim procesom smanjenja broja uzoraka (eng. *downsampling*) se smanjuje količina detalja na slici, bez

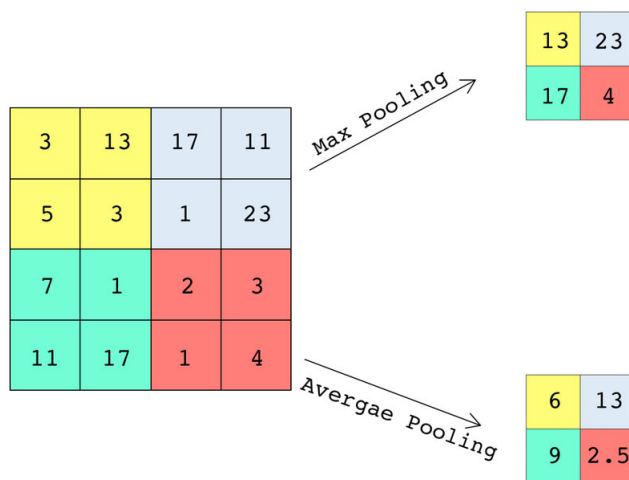
## Poglavlje 2. Konvolucijske neuronske mreže

uklanjanja važnih strukturalnih elemenata koji mogu pomoći pri klasifikaciji. Time se smanjuje broj parametara, te se kontrolira rizik pretjerane prilagodbe podacima.

Sam proces sažimanja funkcionira na sličnom principu kao i konvolucijski sloj. Razlika je u tome što je u ovom slučaju cilj smanjenje dimenzija slike, pa se tako ne koristi oblaganje, te se obično odabire veća vrijednost parametra koraka. Korišteni filtri su obično dimenzije  $2 \times 2$ . Sažimajući sloj je najčešće implementiran na jedan od dva načina [12]:

- Izborom najveće vrijednosti (eng. *max-pooling*)
- Izborom srednje vrijednosti (eng. *average-pooling*)

Kod *max-pooling* pristupa se za novu vrijednost postavlja najveća vrijednost unutar filtra, dok se kod *average pooling* pristupa za vrijednost postavlja srednja vrijednost svih elemenata unutar filtra. Usporedba ova dva pristupa je prikazana na slici 2.8. Kao što je ranije spomenuto, ovim procesom se smanjuju dimenzije slike koja se šalje dalje kroz neuronsku mrežu i time se naglašavaju potencijalne važne značajke [9].



Slika 2.8 Usporedba *max* i *average pooling* algoritama (preuzeto iz: [13])

### 2.1.3 Potpuno povezani sloj

Na kraju konvolucijske neuronske mreže se nalazi potpuno povezani sloj (eng. *fully connected layer*). Ovaj se sloj sastoji od neurona potpuno povezanih sa svakim od elemenata dobivenih kao izlaz iz prethodnog sloja. Na te značajke se potom primjenjuje aktivacijska funkcija, te se na temelju toga se vrši klasifikacija, isto kao i kod umjetne neuronske mreže. Kao izlaz iz potpuno povezanog sloja se dobiva vektor koji sadrži onoliko vrijednosti koliko je potencijalnih klasa.

## 2.2 Aktivacijske funkcije

Vrlo važan dio konvolucijske neuronske mreže su aktivacijske funkcije. Aktivacijska funkcija je matematička funkcija koja služi za transformaciju dobivenih ulaznih vrijednosti neurona u željenu izlaznu vrijednost. Postoji velik broj aktivacijskih funkcija koje se generalno mogu podijeliti u dvije kategorije [14]:

- Linearne aktivacijske funkcije
- Nelinearne aktivacijske funkcije

U nastavku će biti definirano nekoliko često korištenih ili relevantnih za ovaj rad.

### 2.2.1 Linearna aktivacijska funkcija

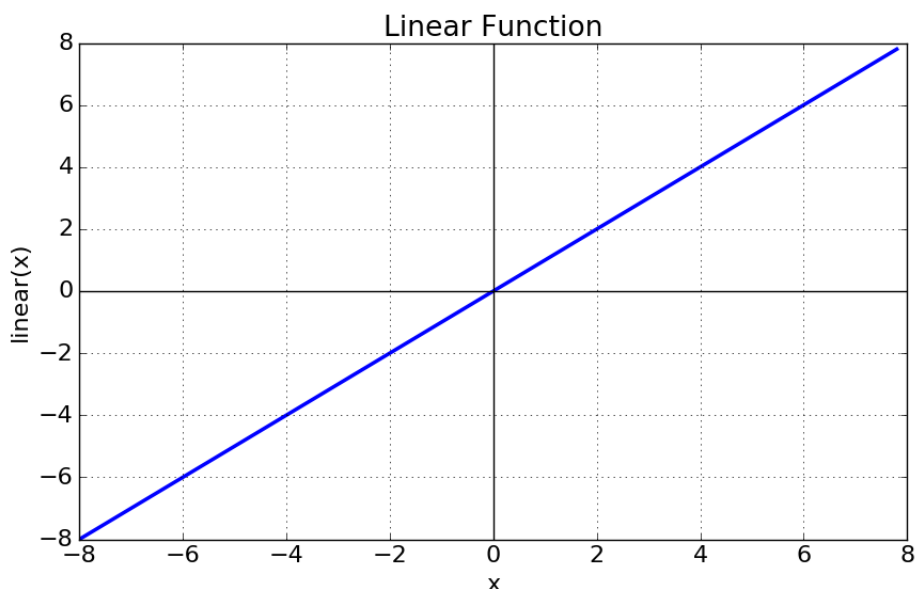
Najjednostavniji primjer aktivacijske funkcije. Ova funkcija prima ulaze i vraća izlaze proporcionalne ulazima. Zbog svoje linearnosti, ova funkcija ne ograničava izlazne vrijednosti. Matematički, može se opisati izrazom 2.3:

$$f(x) = c \cdot x \tag{2.3}$$

Zbog svoje linearnosti, ova funkcija ima nekoliko problema. Uz nju nije moguće primijeniti algoritam gradijentnog spusta jer se on oslanja na poznavanje derivacije funkcije da bi optimizirao težinske faktore, a kako je derivacija linearne funkcije konstanta, nije moguće odrediti kako težinski faktori utječu na ulaznu vrijednost.

## Poglavlje 2. Konvolucijske neuronske mreže

Također s obzirom na to da je funkcije linearna, zadnji sloj će biti linearna funkcija prvog sloja, što je ekvivalentno tome da u mreži postoji samo jedan sloj. Zbog ovih svojstava, linearna aktivacijska funkcije nema široku primjenu u dubokom učenju [14]. Linearna aktivacijska funkcija prikazana je na slici 2.9.



Slika 2.9 Linearna aktivacijska funkcija (preuzeto iz: [15])

### 2.2.2 Sigmoidna aktivacijska funkcija

Za razliku od linearnih aktivacijskih funkcija, nelinearne aktivacijske funkcije su derivabilne, što omogućava korištenje algoritama za optimizaciju težinskih faktora, kao što je algoritam propagacije unatrag. Nelinearnost također dopušta i dodavanje više uzastopnih slojeva u mrežu. Zbog ovih svojstava nelinearne aktivacijske funkcije imaju vrlo široku primjenu. Jedna od najpoznatijih nelinearnih aktivacijskih funkcija je sigmoidna aktivacijska funkcija, opisana izrazom 2.4:

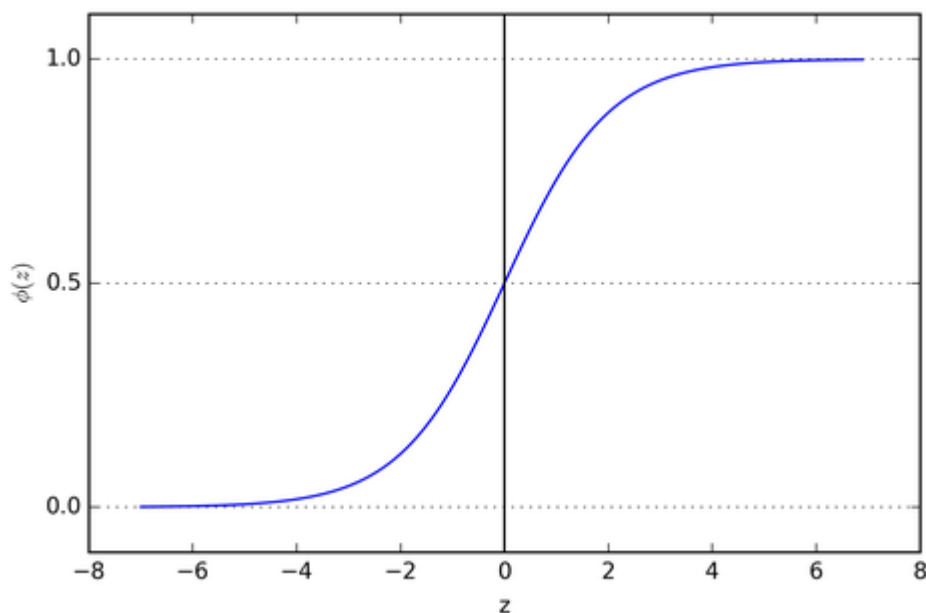
$$S(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

Jedno od najkorisnijih svojstava ove funkcije je to što poprima vrijednosti u rasponu  $[0, 1]$ , zbog čega automatski normalizira izlaznu vrijednost svakog neurona,



## Poglavlje 2. Konvolucijske neuronske mreže

što je čini izvrsnom u slučajevima kada kao izlaznu vrijednost tražimo vjerojatnost. Ova funkcija također posjeduje gladak gradijent (vrijednost prve derivacije). S druge strane, ova funkcija je računski vrlo zahtjevna, te je podložna problemu nestajućih gradijenata [14]. Bez obzira na te probleme, često je korištena jer je vrlo dobra kod problema binarne klasifikacije. Sigmoidna aktivacijska funkcija prikazana je na slici 2.10.



Slika 2.10 Sigmoidna aktivacijska funkcija (preuzeto iz: [15])

U slučaju višeklasne klasifikacije, može se koristiti poopćena verzija sigmoidne funkcije, takozvana funkcija softmax, koja kao izlaz daje vektor realnih vrijednosti u rasponu između 0 i 1. Svaka od tih vrijednosti predstavlja vjerojatnost pripadnosti odgovarajućoj klasi. Može opisati izrazom 2.5:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.5)$$

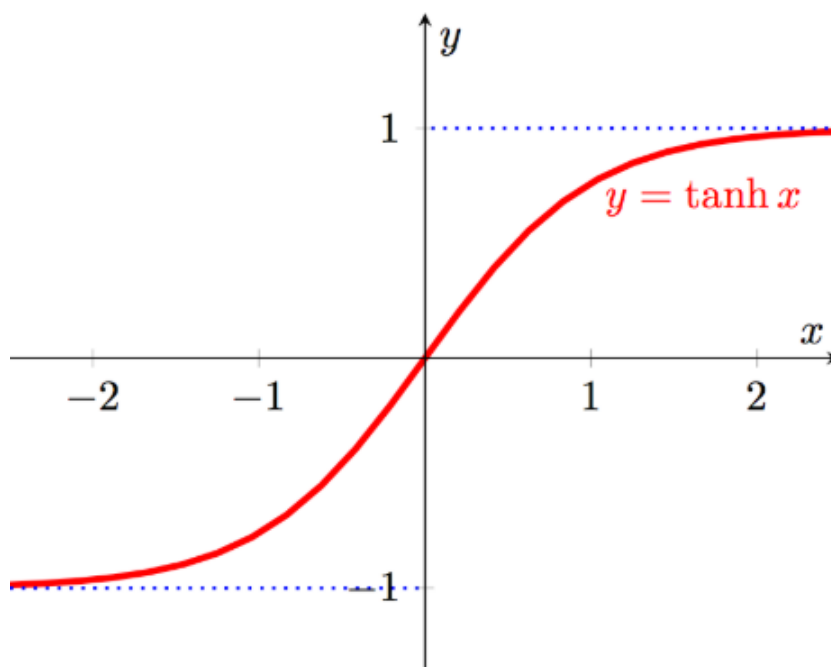
### 2.2.3 Aktivacijska funkcija tangens hiperbolni

Funkcija tangens hiperbolni može se opisati formulom 2.6:

## Poglavlje 2. Konvolucijske neuronske mreže

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (2.6)$$

Oblikom i svojstvima je ova funkcija vrlo slična sigmoidnoj aktivacijskoj funkciji. Ipak, sadrži neka svojstva koja ju u većini slučajeva čine boljom od sigmoidne funkcije [16]. Ova funkcija poprima vrijednosti u rasponu  $[-1, 1]$ , zbog čega možemo reći da je centrirana oko nule. To znači da će se negativne izlazne vrijednosti mapirati strogo negativno, dok će pozitivne vrijednosti biti mapirane strogo pozitivno, čime se smanjuju nagli skokovi u gradijentu i povećava brzina treniranja mreže. Aktivacijska funkcija tangens hiperbolni prikazana je na slici 2.11.



Slika 2.11 Aktivacijska funkcija tangens hiperbolni (preuzeto iz: [14])

U slučaju konvolucijskih neuronskih mreža, prethodno navedene aktivacijske funkcije se koriste pretežito na izlaznom sloju mreže sa svrhom klasifikacije naučenih podataka u dvije (sigmoidna, tangens hiperbolni) ili više klasa (*softmax*).

## 2.2.4 Rectified Linear Unit

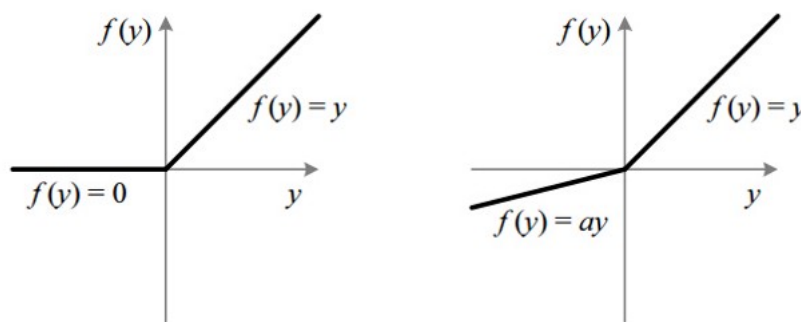
Aktivacijska funkcija ReLU (eng. *Rectified Linear Unit*) je najkorištenija aktivacijska funkcija kod konvolucijskih neuronskih mreža. Računski je znatno efikasnija od funkcija kao što su sigmoidna ili tangens hiperbolni, zbog čega se često koristi u mrežama s mnogo slojeva. Može se opisati formulom 2.7:

$$R(z) = \max(0, z) \quad (2.7)$$

Računska efikasnost ove funkcije proizlazi iz činjenice da poprima vrijednosti u rasponu  $[0, \infty]$ , što znači da će se svi negativni ulazi automatski mapirati u 0. Ipak, ovakav pristup ima nedostatak. U slučaju da se gradijent funkcije približi nuli ili postane negativan, mreža više neće moći izvoditi algoritam propagacije unatrag. Ovaj problem se naziva *Dying ReLU problem* i jedan od načina da se izbjegne je korištenje Leaky ReLU aktivacijske funkcije, koje se definira izrazom 2.8:

$$R(z) = \max(0.1z, z) \quad (2.8)$$

Ova varijanta funkcije više neće potpuno zanemarivati negativne vrijednosti, te će omogućiti propagaciju unatrag i u slučajevima kada to običan ReLU ne bi mogao. Ipak, predviđanja koja daje za negativne ulazne vrijednosti neće biti konzistentna. ReLU i Leaky ReLU aktivacijske funkcije prikazane su na slici 2.12.



Slika 2.12 ReLU i Leaky ReLU aktivacijske funkcije (preuzeto iz: [15])

Važno je napomenuti da Leaky ReLU nije jedina varijanta koja se bavi rješavanjem *Dying ReLU* problema. Postoji mnogo aktivacijskih funkcija baziranih na ReLU aktivacijskoj funkciji koje se zasnivaju na istoj ideji, kao što su [14]:

- Parametric ReLU (PReLU)
- Exponential Linear (ELU, SELU)
- Concatenated ReLU (CReLU)
- ReLU-6

## 2.3 Regularizacija

Regularizacija podrazumijeva skup tehnika koje se koriste da bi se penaliziralo algoritam strojnog učenja i smanjio rizik pretjerane prilagodbe podacima, te se ujedno povećala robusnost algoritma i njegova sposobnost generalizacije na novim podacima. Postoji nekoliko tehnika regularizacije, od kojih će neke biti predstavljene u nastavku.

### 2.3.1 Regularizacije L1 i L2

Regularizacije L1 i L2 često su korištene regularizacijske tehnike koje funkcioniraju tako da dodaju dodatni regularizacijski faktor na funkciju cijene što dovodi do smanjenja težinskih faktora. Time se smanjuje kompleksnost mreže, što umanjuje i šansu pretjerane prilagodbe podacima.

Razlika između ova dva tipa regularizacije je u regularizacijskom faktoru koji se dodaje na funkciju cijene. Regularizacija L1 može se opisati izrazom 2.9:

$$\text{Funkcija cijene} = \text{Gubitak} + \frac{\lambda}{2m} \cdot \sum \|w\| \quad (2.9)$$

gdje je:

- $\lambda$  - regularizacijski parametar

## Poglavlje 2. Konvolucijske neuronske mreže

- $w$  - vektor težinskih faktora

Ova regularizacija je poznata i kao Lasso regresija, i ona ima efekt minimiziranja koeficijenta manje važnih faktora, čime ih se isključuje iz proračuna. Ovakvim se postupkom postiže efekt odabira najvažnijih značajki.

S druge strane, regularizacija L2 je poznata kao Ridge regresija, i opisuje se izrazom 2.10:

$$\text{Funkcija cijene} = \text{Gubitak} + \frac{\lambda}{2m} \cdot \sum \|w\|^2 \quad (2.10)$$

Za razliku od regularizacije L1, regularizacija L2 nema efekt odabira značajki, već samo penalizira funkciju cijene da ne bi došlo do pretjerane prilagodbe podacima. Zbog toga će u većini situacija biti preferirana nad regularizacijom L1 [17].

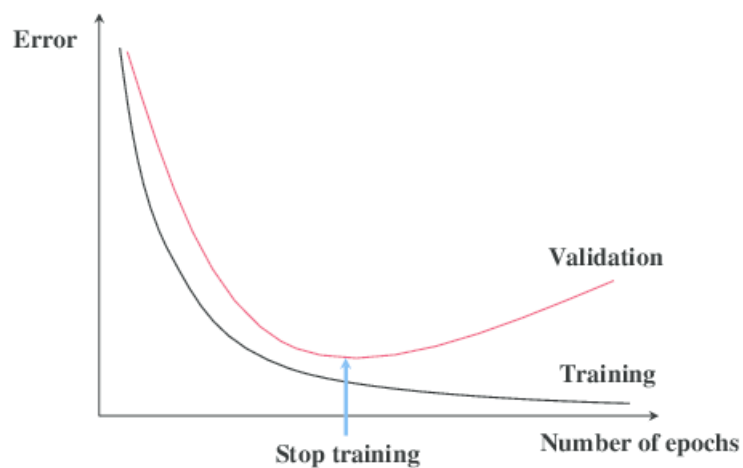
### 2.3.2 Rano zaustavljanje

Metoda ranog zaustavljanja (eng. *early stopping*) funkcionira tako da zaustavi treniranje algoritma ranije ukoliko primijeti pogoršanje u nekom od promatranih parametara. Uobičajeno se implementira tako da se pri treniranju koristi i validacijski skup, na kojem se nakon svake iteracije provjerava kvaliteta klasifikacije. Kada se primijeti pad performansi na validacijskom skupu, treniranje se zaustavlja da bi se izbjegla pretjerana prilagodba podacima. Primjer ranog zaustavljanja je prikazan na slici 2.13.

### 2.3.3 Dropout

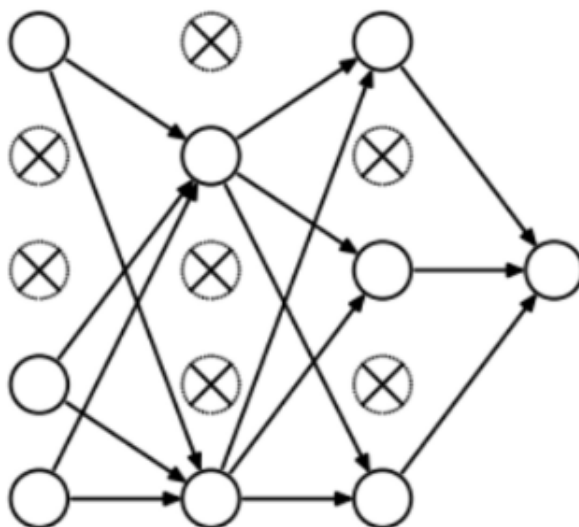
*Dropout* je regularizacijska tehnika koja, za razliku od prethodno navedenih regularizacija L1 i L2, ne utječe direktno na funkciju cijene. Umjesto toga, *dropout* funkcionira tako da se u svakoj iteraciji treninga nasumično odabiru čvorovi u neuronskoj mreži koji će za vrijeme te iteracije postati neaktivni, zajedno sa svim svojim ulazima i izlazima. Na ovaj se način postiže veća razina nasumičnosti u neuronskoj mreži, te se smanjuje šansa da će se neuronska mreža pri klasifikaciji previše oslanjati na određene čvorove, čime se smanjuje šansa za pretjeranu prilagodbu podacima. Šansa

Poglavlje 2. Konvolucijske neuronske mreže



Slika 2.13 Early stopping regularizacija (preuzeto iz: [18])

da čvor postane neaktivan se obično određuje korištenjem hiperparametra, te se povećavanjem vrijednosti tog hiperparametra može pojačati učinak regularizacije. Na slici 2.14 je prikazan primjer neuronske mreže u jednoj iteraciji koristeći *dropout* regularizaciju.



Slika 2.14 Dropout regularizacija (preuzeto iz: [18])

# Poglavlje 3

## Metodologija

Cilj ovog rada bio je istražiti mogućnost primjene konvolucijske neuronske mreže na jednodimenzionalne podatke, za čiju bi se klasifikaciju inače koristile klasične neuronske mreže. Uobičajeno, klasični pristup u nekim slučajevima zahtjeva veliku količinu domenskog znanja i predobrade podataka. U svrhu ovog projekta, istraženi su odabrani načini pretvorbe dvodimenzionalnih podataka u slikovni format, uz minimalnu prethodnu predobradu podataka.

U ovom će poglavlju biti opisani algoritmi pretvorbe podataka u slikovni oblik, arhitektura neuronske mreže korištene za klasifikaciju, te korišteni skupovi podataka i njihova svojstva.

### 3.1 Skupovi podataka

U svrhu ovog projekta korišteno je nekoliko skupova podataka. Pri odabiru skupova podataka obraćena je pozornost na parametre kao što su količina uzoraka, broj značajki i količina kategoričkih značajki, kako bi se algoritam mogao testirati u što više različitih slučajeva. Korišteni skupovi podataka su:

- Breast cancer dataset
- Thyroid dataset

### Poglavlje 3. Metodologija

- Sonar dataset
- Swarm behavior dataset

Svi skupovi podataka su preuzeti s UCI repozitorija [19]. U tablici 3.1 prikazani su osnovni podaci o svakom od korištenih skupova podataka.

Skup podataka	Broj uzoraka	Broj klasa	Broj značajki
Breast Cancer	569	2	30
Thyroid	215	3	5
Sonar	208	2	60
Swarm behavior	24,017	2	2,400

Tablica 3.1 Svojstva korištenih skupova podataka

Kako bi se ispitala kvaliteta algoritma konverzija za različite kombinacije broja uzoraka i značajki, odabrani su skupovi različitih parametara. Obraćena je i pažnja na to da se testira i više tipova klasifikacije, pa tako jedan dio skupova obuhvaća problem binarne, a drugi dio problem višeklasne klasifikacije. U nastavku je dan detaljniji pregled za svaki od skupova podataka.

Prije samog treninga, skup podataka je podijeljen na tri podskupa. Skup za treniranje obuhvaća 80% originalnog skupa podataka, dok se preostalih 20% podijeli na skup za validaciju i skup za testiranje u jednakom omjeru [1]. Učenje modela predviđanja se vrši na skupu za treniranje, dok se validacijski skup koristi za provjeru točnosti dobivenog modela nakon svake epohe treniranja, primarno radi implementacije ranog zaustavljanja. Testni skup se ostavlja po strani dok trening se završi, te se potom na njemu još jednom provjeravaju performanse izgrađenog algoritma kako bi se utvrdilo da daje dobre rezultate na nikad viđenim podacima.

Kao prvi skup podataka odabran je skup o dijagnostičkim rezultatima raka dojke. Skup se sastoji od 569 uzoraka, podijeljenih u dvije grupe: maligni i benigni. Malignih uzoraka je 37% i označeni su klasom M, dok je benignih 63% i označeni su klasom B. S obzirom na format klasa, potrebno je provesti njihovu transformaciju, tako da se benigni uzorci označe klasom 0, a maligni klasom 1. Svaki uzorak se sastoji od



### Poglavlje 3. Metodologija

30 značajki, koje su sve brojčane vrijednosti, te ih je potrebno normalizirati da bi se dovele u raspon  $[0, 1]$ .

Drugi skup sadrži podatke o funkciji tiroidne žlijezde, i sastoji se od 215 uzoraka, podijeljenih u 3 klase. Za razliku od prethodnog skupa, u ovom skupu su klase već označene brojkama, gdje 0 predstavlja normalan rad žlijezde i sadrži 70% uzoraka, 1 predstavlja pojačan rad žlijezde i sadrži 16% uzoraka, te 2 predstavlja slabiji rad žlijezde i sadrži 14% uzoraka. Skup ima 5 značajki, koje su brojčane vrijednosti i zahtijevaju normalizaciju. Iz skupa su uklonjena 3 uzorka s vrijednostima koje nedostaju.

Treći skup podataka se sastoji od podataka dobivenim očitanjima sonara u potrazi za skrivenim minama. Skup sadrži 208 uzoraka, podijeljenih u dvije klase. Od tih uzoraka, 53% predstavlja očitavanje mine, dok preostalih 47% predstavlja očitavanje kamena. Skup je označen tako da su uzorci mina označeni s  $M$ , dok je kamenje označeno s  $R$ . Kao i u prethodnom slučaju, na ovom skupu će biti obavljen proces transformacije klasa, tako da će se klasa  $R$  označiti s 0, a klasa  $M$  s 1. Za razliku od ostalih skupova, ovaj skup ne zahtijeva normalizaciju podataka jer svih 60 značajki već u rasponu  $[0, 1]$ .

Posljednji od korištenih skupova podataka, sastoji se od 24,017 uzoraka i 2,400 značajki, te predstavlja najizazovniji skup podataka. Skup se sastoji od podataka o koordinatama članova roja, na temelju kojih se može odrediti ponašanje roja. Podijeljen je u dvije klase, koje predstavljaju *flocking* (označena s 1) i *non-flocking* (označena sa 0). Iz skupa je uklonjen 1 uzorak zbog vrijednosti koje su nedostajale.

## 3.2 Predobrada podataka

Kao što je ranije spomenuto, jedan od ciljeva ovog rada bio je istražiti načine da se umanjuje količina predobrade podataka, no ono se ipak ne može potpuno zanemariti. Tako su na sve skupove podataka primijenjene dvije osnovne tehnike predobrade:

- Uklanjanje uzoraka s nedostajućim vrijednostima
- Normalizacija podataka

### Poglavlje 3. Metodologija

U slučaju kada u skupu podataka postoje uzorci kojima nedostaju vrijednosti nekih značajki, postoje 2 uobičajena pristupa. Jedan od pristupa je da se vrijednost koja nedostaje popuni srednjom vrijednosti te značajke u skupu. Takav pristup može dovesti do problema u slučaju da su vrijednosti u širokom rasponu i imaju veliko odstupanje od srednje vrijednosti, te također nije koristan u slučajevima kada set podataka sadrži mnogo kategoričkih vrijednosti. S obzirom na to da u odabranim skupovima podataka postoji velik broj kategoričkih vrijednosti, odabran je drugi pristup, a to je uklanjanje uzoraka kojima nedostaju vrijednosti. Uklonjena su samo 3 uzorka iz skupa o tiroidnoj žlijezdi, te 1 iz skupa o ponašanju roja.

Skaliranje podataka je uobičajeno kod primjene većine algoritama strojnog učenja jer svođenje podataka u jednak raspon može ubrzati proces treniranja i pomoći da se izbjegnu problemi eksplodirajućih ili nestajućih gradijenata [20]. U slučaju ovog projekta, skaliranje je važno jer podaci moraju biti svedeni na jedinstvenu skalu da bi nakon konverzije u slikovni oblik njihov prikaz na slici bio razlučiv. Normalizacija je vrsta skaliranja podataka koja se provodi tako da se svaka od značajki stavlja u raspon  $[0, 1]$ . Matematički, ovaj postupak se može opisati izrazom 3.1:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (3.1)$$

gdje je:

- $X'$  - skalirana vrijednost
- $X$  - originalna vrijednost
- $X_{\max}$  - najveća vrijednost u skupu
- $X_{\min}$  - najmanja vrijednost u skupu

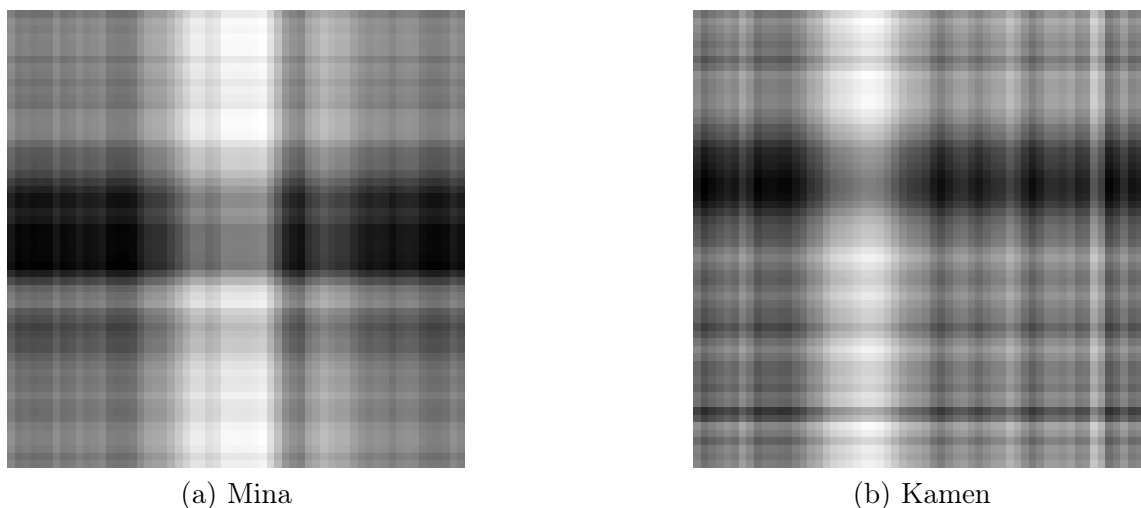
### 3.3 Algoritam transformacije podataka u slike

Da bi se konvolucijska neuronska mreža mogla primijeniti na brojčane podatke, potrebno ih je transformirati u slikovni oblik. U svrhu ovog rada, korištena su dva

algoritma predložena u [1], koja će biti detaljno opisana u nastavku. Važno je napomenuti da iako dobivene slike nisu praktične za ljudsko tumačenje, one su dovoljne da konvolucijska neuronska mreža nauči značajke potrebne za njihovu klasifikaciju.

### 3.3.1 Algoritam transformacije u matricu udaljenosti

Ideja ovog algoritma je prikazati razliku među značajkama u matricnom obliku, gdje se svaka vrijednost prikazuje jednim pikselom. Za skup podataka sa  $n$  uzoraka i  $m$  značajki, algoritam će generirati  $n$  slika, veličine  $m \times m$  piksela. Da bi se to moglo ostvariti, podatke je prvo potrebno skalirati u raspon  $[0, 1]$  da bi razlika bila jasno vidljiva. Na slici 3.1 su prikazane slike dobivene pretvorbom skupa podataka o očitanjima sonara u slike, po jedan primjerak za svaku klasu.

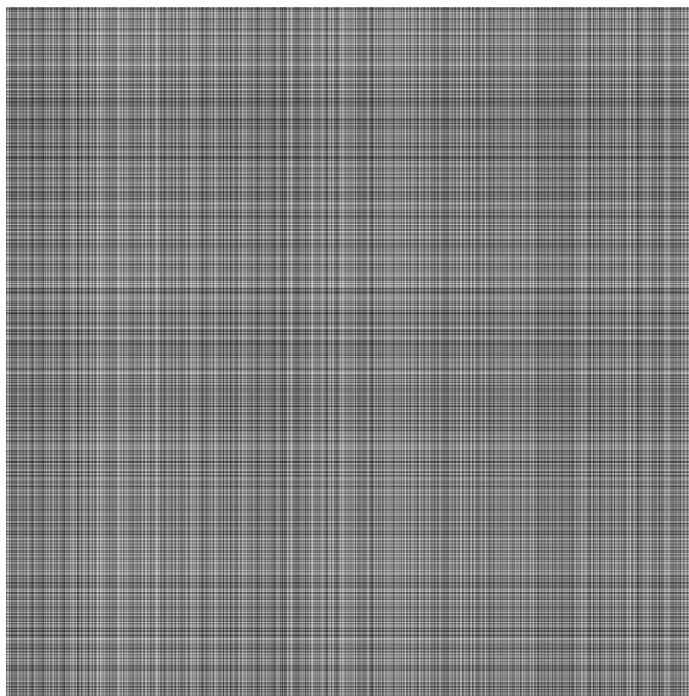


Slika 3.1 Slikovni prikaz dobiven transformacijom sonar skupa podataka

S obzirom na to da je u izvršavanju algoritma uključen veliki broj računskih operacija za svaku od slika slici, duljina trajanja procesa transformacije će uvelike zavisiti o veličini skupa podataka. Za male skupove razlika se pokazala minimalna, međutim kod velikih skupova s mnogo značajki, vrijeme izvršavanja se može uvelike produžiti. Također, generirane slike zauzimaju veliku količinu memorijskog prostora, što čini

### Poglavlje 3. Metodologija

čuvanje skupa podataka u ovoj formi umjesto u originalnoj brojčanoj znatno zahtjevnijim. Na slici 3.2 je prikazan rezultat transformacije uzorka iz skupa podataka Swarm behavior.



Slika 3.2 Slikovni prikaz dobiven transformacijom uzorka iz Swarm behavior skupa podataka u matricu

Za svaki od uzoraka u skupu podataka, inicijalizira se prazna matrica dimenzija  $n \times n$ , gdje  $n$  predstavlja broj značajki. Zatim se od svake značajki oduzima vrijednost svake druge značajke, te se matrica popunjava dobivenim vrijednostima. Postupak je opisan pseudokodom danom u algoritmu 1 [21].

#### 3.3.2 Algoritam transformacije u histogram

Ovaj se algoritam također zasniva na normalizaciji značajki, ali za razliku od prethodnog ne računa se razlika među značajkama, već se one prikazuju u obliku histograma.

---

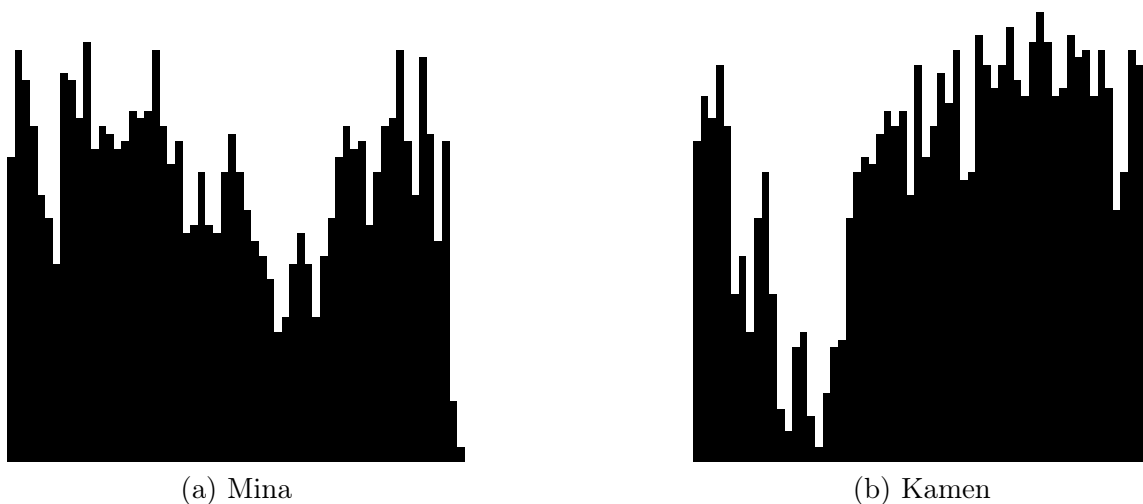
**Algorithm 1** Algoritam transformacije u matricu udaljenosti

---

```
for  $i = 1 : n\_samples$  do  
     $image \leftarrow M_{n \times n}$   
    for  $j = 1 : image\_size$  do  
        for  $k = 1 : image\_size$  do  
             $image[j][k] \leftarrow dataset[i][j] - dataset[i][k]$   
        end for  
    end for  
     $save(image)$   
end for
```

---

Svaka od značajki je prikazana redosljedom kojim se pojavljuju u skupu podataka, koji nije strogo definiran. Ovaj će algoritam također dati jednake veličine slika, dakle za skup podataka sa  $m$  značajki, algoritam će generirati slike veličine  $m \times m$  piksela. Na slici 3.3 su prikazane slike dobivene konverzijom skupa podataka o očitanjima sonara u slike, po jedan primjerak za svaku klasu.

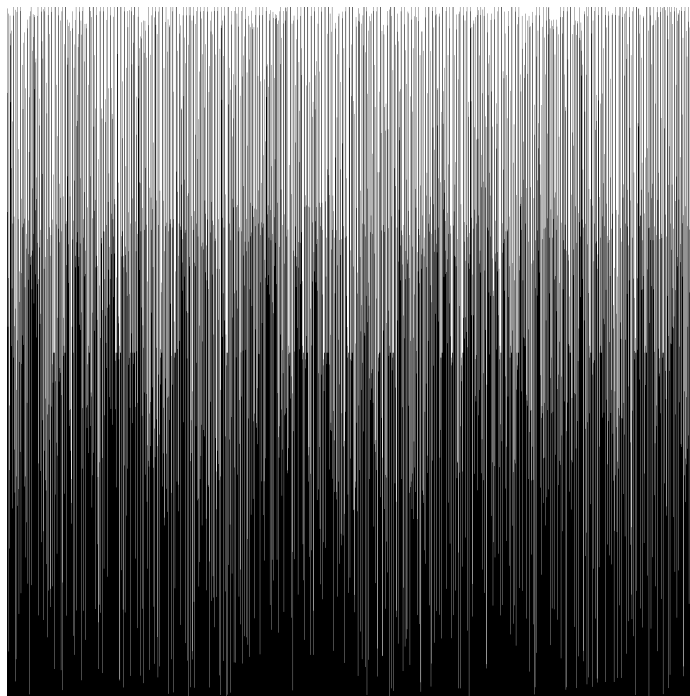


Slika 3.3 Slikovni prikaz dobiven transformacijom sonar skupa podataka

Za razliku od prethodno opisanog algoritma, slike generirane ovim algoritmom

### Poglavlje 3. Metodologija

zauzimaju manje memorije, te je trajanje izvođenja algoritma znatno kraće, što može napraviti primjetnu razliku kod skupova podataka velikih dimenzija. Na slici 3.4 je prikazana slika dobivena transformacijom jednog uzorka iz Swarm behavior skupa podataka.



Slika 3.4 Slikovni prikaz dobiven transformacijom uzorka iz Swarm behavior skupa podataka u histogram

Za svaki od uzoraka u skupu podataka, inicijalizira se prazna matrica dimenzija  $n \times n$ , gdje  $n$  predstavlja broj značajki. Zatim se od svake značajki oduzima vrijednost svake druge značajke, te se matrica popunjava dobivenim vrijednostima.

Inicijalno, generira se matrica dimenzija  $n \times n$ , te se vrijednost svih elemenata matrice postavljaju na 1. Potom se za svaku od značajki računa veličina stupca histograma, na temelju vrijednosti normalizirane značajke i određenog parametra za visinu stupca. Ovaj parametar je postavljen na veličinu manju od veličine slike, kako bi se ostvario pregledniji prikaz. U algoritmu 2 je pseudokodom opisan postupak pretvorbe [21].

---

**Algorithm 2** Algoritam transformacije u histogram

---

```
image_size ← n_samples
bar_height ← image_size − 2
for i = 1 : n_samples do
    image ←  $M_{n \times n}$ 
    for j = 1 : image_size do
        bar ← floor(bar_height * dataset[1 : n_samples][j])
        for k = 1 : range(bar[i]) do
            image[k][j] ← 0
        end for
    end for
    save(image)
end for
```

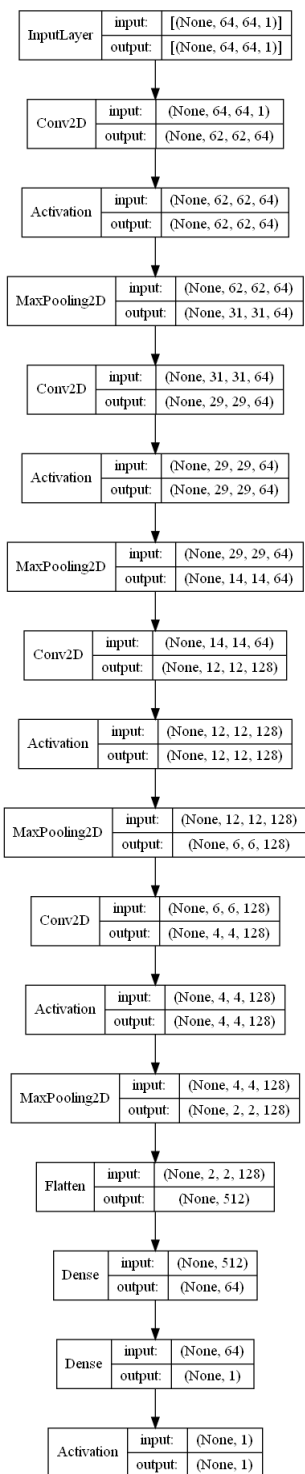
---

### 3.4 Arhitektura neuronske mreže

U ovom će poglavlju biti opisana arhitektura konvolucijske neuronske mreže, te odabrani hiperparametri. Uz konvolucijsku neuronsku mrežu, također je korištena i umjetna neuronske mreža za klasifikaciju podataka u izvornom obliku, te će biti dan i kratak pregled njenih hiperparametara.

U inicijalnom testiranju je korištena jedna od poznatih arhitektura konvolucijske neuronske mreže, VGG16 [22]. Međutim, s obzirom na to da je većina korištenih skupova podataka vrlo jednostavna, kroz testiranje se pokazalo da pretjerana složenost modela u nekim situacijama dovodi do pretjerane prilagodbe podacima. Zbog toga je odabrana pojednostavljena varijanta inicijalnog modela, za koju se pokazalo da i dalje daje dobre rezultate, ali ima i mogućnost generalizacije na novim podacima. Primjer arhitekture odabranog modela koji na ulaznom sloju prima slike veličine 64x64 je prikazana na slici 3.5.

### Poglavlje 3. Metodologija



Slika 3.5 Arhitektura neuronske mreže



### Poglavlje 3. Metodologija

Kao što je vidljivo sa slike, model se sastoji od četiri konvolucijska sloja, jednog potpuno povezanog sloja, te izlaznog sloja. Prva dva konvolucijska sloja sadrže 64 filtera veličine 3x3, dok dublja dva sloja sadrže 128 filtera, također veličine 3x3. Uz svaki konvolucijski sloj, također dolazi i sažimajući sloj. Svaki sažimajući sloj koristi sažimanje izborom najveće vrijednosti, s veličinom filtera 2x2. Svaki od konvolucijskih slojeva kao aktivacijsku funkciju koristi ReLU.

Na izlazu iz mreže se nalaze potpuno povezani i izlazni sloj. Kao aktivacijska funkcija na izlaznom sloju se koriste sigmoidna ili softmax aktivacijska funkcija, ovisno o broju klasa. U slučaju da se radi o problemu binarne klasifikacije, koristi se sigmoid, dok se za višeklasnu klasifikaciju koristi softmax funkcija.

Da bi se spriječila pretjerana prilagodba modela podacima, implementirane su i regularizacijske metoda. Za vrijeme treninga se koristi spomenuta L2 regularizacija, kako bi se povećala brzina treninga i robusnost modela. Također je implementirano i rano zaustavljanje. Metoda ranog zaustavljanja je implementirana tako da prati točnost na validacijskom skupu za vrijeme treniranja, te u slučaju da nakon određenog ne dođe do poboljšanja validacijske točnosti, ranije zaustavlja trening.

U tablici 3.2 je dan pregled hiperparametara konvolucijske neuronske mreže korištenih pri treniranju modela za svaki od korištenih skupova podataka. Kod određivanja hiperparametara, kao početna točka su uzeti hiperparametri predloženi u [1], te su kroz daljnje testiranje prilagođeni.

<b>Parametar</b>	<b>Vrijednost</b>
Max epochs	50
Initial learning rate	1e-4
Regularization	L2 (1e-3)
Filter size	3x3
Pool size	2x2
Padding	valid

Tablica 3.2 Hiperparametri konvolucijske neuronske mreže

U svrhu usporedbe rezultata sa standardnim algoritmima klasifikacije, korištena

### Poglavlje 3. Metodologija

je i umjetna neuronska mreža za klasifikaciju podataka u njihovom izvornom obliku brojčanom obliku. U tablici 3.3 je dan pregled korištenih hiperparametara neuronske mreže. Kao i kod konvolucijske neuronske mreže, ugrađen je i mehanizam ranog zaustavljanja kako bi se spriječila pretjerana prilagodba podacima.

Parametar	Vrijednost
Max epochs	50
Initial learning rate	1e-3
Regularization	L2 (1e-7)
Hidden layers	5

Tablica 3.3 Hiperparametri neuronske mreže

Kod konvolucijske neuronske mreže se kao ulazne vrijednosti koriste 1-kanalne crno-bijele slike, dimenzija 64x64 piksela. S obzirom na to da svaki piksel poprima vrijednosti u rasponu  $[0, 255]$ , vrijednost svakog piksela se dijeli da najvećom vrijednošću da bi se svela u raspon  $[0, 1]$ . Ovime se smanjuje rizik nedovoljne prilagodbe podacima, te se ubrzava treniranje.

## 3.5 Metrike vrednovanja

Postoji veliki broj metrika vrednovanja koje se koriste za ocjenu kvalitete izgrađenog modela predviđanja, a one najčešće korištene za klasifikacijske probleme su:

- Matrica zabune (eng. *confusion matrix*)
- Točnost (eng. *accuracy*)
- Preciznost (eng. *precision*)
- Specifičnost (eng. *specificity*)
- Osjetljivost (eng. *recall*)
- F1 rezultat

### Poglavlje 3. Metodologija

Matrica zabune jedan je od najpoznatijih i najintuitivnijih načina vrednovanja klasificiranih podataka. Funkcionira tako da se svaki rezultat binarne klasifikacije može svrstati u jednu od 4 kategorije:

- Točno pozitivno (TP) - podatak je ispravno klasificiran kao pozitivan
- Lažno pozitivno (FP) - podatak je neispravno klasificiran kao pozitivan
- Točno negativno (TN) - podatak je ispravno klasificiran kao negativan
- Lažno negativno (FN) - podatak je neispravno klasificiran kao negativan

U slučaju klasifikacije u dvije klase, matrica zabune se može prikazati na način opisan tablicom 3.4.

Stvarno	Predviđeno	
	Pozitivan	Negativan
Pozitivan	TP	FN
Negativan	FP	TN

Tablica 3.4 Matrica zabune

Točnost je metrika koja opisuje udio ispravno predviđenih uzoraka u ukupnom broju predviđanja. Može se izraziti formulom 3.2.

$$\text{Točnost} = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.2)$$

Metriku točnosti je najbolje koristiti u slučajevima kada je skup podataka ujednačen (udio klasa u skupu podataka je podjednak). S druge strane, preporučuje se izbjegavati korištenje ove metrike u slučajevima kada postoji velika neuravnoteženost klasa [23].

Slično kao i točnost, metrika preciznosti opisuje udio ispravno predviđenih uzoraka, ali za razliku od točnosti, umjesto svih predviđenih uzoraka, mjeri se udio u uzorcima koji su predviđeni kao pozitivni. Matematički, može se opisati formulom 3.3.

### Poglavlje 3. Metodologija

$$\text{Preciznost} = \frac{TP}{TP + FP} \quad (3.3)$$

Ova metrika je korisna kada je cijena neispravnog klasificiranja uzorka kao pozitivnog visoka, odnosno kada je važno da broj lažno pozitivnih uzoraka bude što manji [23].

Metrika osjetljivosti opisuje udio ispravnih pozitivnih predviđanja, u ukupnom broju pozitivnih uzoraka. Može se izraziti formulom 3.4.

$$\text{Osjetljivost} = \frac{TP}{TP + FN} \quad (3.4)$$

Ova se metrika koristi u slučajevima kada je cilj što više smanjiti broj lažno negativnih predviđanja [23].

Specifičnost je metrika koja je suprotna od metrike od metrike osjetljivosti, tako da mjeri udio ispravno predviđenih negativnih uzoraka u postojećim negativnim uzorcima [23]. Izražava se formulom 3.5.

$$\text{Specifičnost} = \frac{TN}{TN + FP} \quad (3.5)$$

F1 metrika kombinira metrike preciznosti i osjetljivosti tako da računa njihovu harmonijsku sredinu. To se postiže formulom 3.6.

$$F1 = 2 \times \frac{\text{Preciznost} * \text{Osjetljivost}}{\text{Preciznost} + \text{Osjetljivost}} \quad (3.6)$$

Zbog toga, ova metrika je korisna u slučajevima kada se pokušava pronaći ravnoteža između preciznosti i osjetljivosti, čak i ako postoji neujednačena raspodjela klasa [24].

# Poglavlje 4

## Rezultati

U ovom će poglavlju biti prikazani rezultati dobiveni primjenom opisane metodologije na korištenim skupovima podataka. S obzirom na to da je za ovaj pristup nužna transformacija podataka u sliku, važno je uzeti u obzir njezinu cijenu. Kao što je ranije spomenuto, za velike skupove podataka se ova operacija može pokazati vrlo računski skupom, jer uz to što inicijalni proces transformacije može biti vremenski zahtjevan, skupovi podataka pohranjeni u ovom formatu također zauzimaju znatno više prostora nego kada su pohranjeni u numeričkom formatu. U tablici 4.1 je dan prikaz veličina skupova podataka u svakom od formata.

<b>Skup podataka</b>	<b>CSV</b>	<b>Histogram</b>	<b>Matrica</b>
Breast Cancer	123KB	153KB	1.01MB
Thyroid	5KB	37.8KB	46.9KB
Sonar	86KB	90.7KB	868KB
Swarm Behavior	230MB	812MB	153.92GB

Tablica 4.1 Veličine skupova podataka u različitim formatima

Kao što se može vidjeti iz tablice, skupovi podataka zapisani u numeričkom formatu očekivano zauzimaju manje memorijskog prostora od preostala dva pristupa, što je očekivan rezultat. Između skupova podataka pretvorenih u slike, skup u obliku histograma zauzima manje prostora od skupa u obliku matrice. Na manjim skupo-

## Poglavlje 4. Rezultati

vima podataka ta razlika nije značajna, no vrlo je primjetna kod velikih skupova podataka. Iz tog razloga, ne savjetuje se trajna pohrana podataka u ovom formatu, već samo privremena pretvorba u svrhu treniranja modela.

Uz razliku u veličinama datoteka, također postoji i razlika u duljini vremena treniranja algoritma. Pokazalo se kako se umjetna neuronska mreža trenira kraće od konvolucijske. U prosjeku, vrijeme treniranja za jednu epohu je 7 do 12 puta kraće kod umjetne neuronske mreže. S druge strane, konvolucijske neuronska mreža završi treniranje u 2 puta manjem broju epoha od umjetne, no se obzirom na znatno kraće trajanje po epohi, ukupno trajanje treniranja je niže kod umjetne neuronske mreže. U tablici 4.2 su prikazani rezultati za oba algoritma, s trajanjem treniranja po epohi, te prosječnim brojem epoha. Pokazalo se kako je za oba pristupa transformacije podataka prosječno trajanje treniranja konvolucijske neuronske mreže jednako, pa su prikazani zajedno.

Skup podataka	ANN		CNN	
	Vrijeme epohe	Broj epoha	Vrijeme epohe	Broj epoha
Breast Cancer	0.5 s	18	3.2 s	10
Thyroid	0.15 s	15	1.16 s	7
Sonar	0.16 s	21	1.2 s	10
Swarm Behavior	27 s	12	325 s	8

Tablica 4.2 Prosječno vrijeme treniranja

U tablici 4.3 su prikazani rezultati dobiveni provjerom točnosti modela na validacijskom skupu u vrijeme treniranja. Akronim *ANN* označava algoritam umjetnih neuronskih mreža, dok oznake *Matrica* i *Histogram* podrazumijevaju klasifikaciju konvolucijskim neuronskim mrežama uz korištenje navedenog načina transformacije podataka.

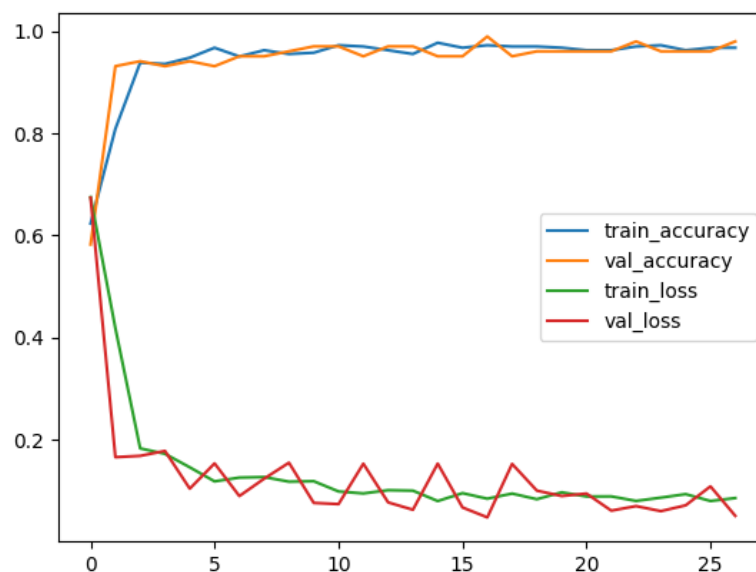
Provjerom modela u fazi treniranja se pokazalo da za skupove podataka s malim brojem uzoraka algoritam neuronskih mreža teže uči, što je vidljivo rezultata prikazanih u prethodnoj tablici. Proces treniranja je nestabilniji nego kod skupova podataka s većim brojem uzoraka i daje širi raspon rezultata, što se može vidjeti po

## Poglavlje 4. Rezultati

Skup podataka	ANN	Histogram	Matrica
Breast Cancer	97.7%(±0.8)	95.3%(±0.8)	94.7%(±1.5)
Thyroid	74.2%(±2.6)	83.4%(±2.4)	82.3%(±2.3)
Sonar	80.0%(±2.1)	90.5%(±2.1)	89.1%(±2.7)
Swarm Behavior	93.8%(±0.1)	99.9%(±0.1)	99.1%(±0.9)

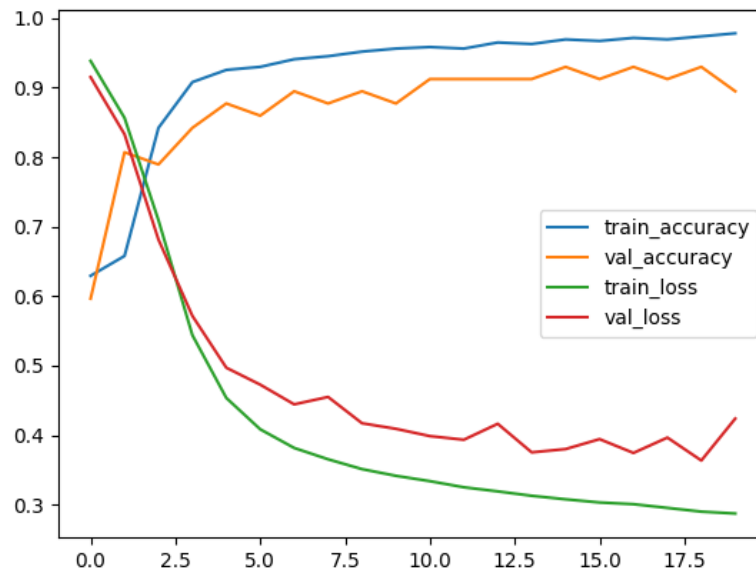
Tablica 4.3 Točnost na validacijskom skupu

vrijednostima standardne devijacije koje su veće na skupovima s manje uzoraka. Na slikama 4.1 i 4.2 su prikazane krivulje koje prate promjenu točnosti i funkcije gubitka na trening i validacijskom skupu kroz epohe za vrijeme treniranja. Prikazane metrike su dobivene kroz jedno treniranje algoritma na skupu podataka o raku dojke.



Slika 4.1 Metrike za vrijeme treniranja umjetne neuronske mreže

## Poglavlje 4. Rezultati



Slika 4.2 Metrike za vrijeme treniranja konvolucijske neuronske mreže

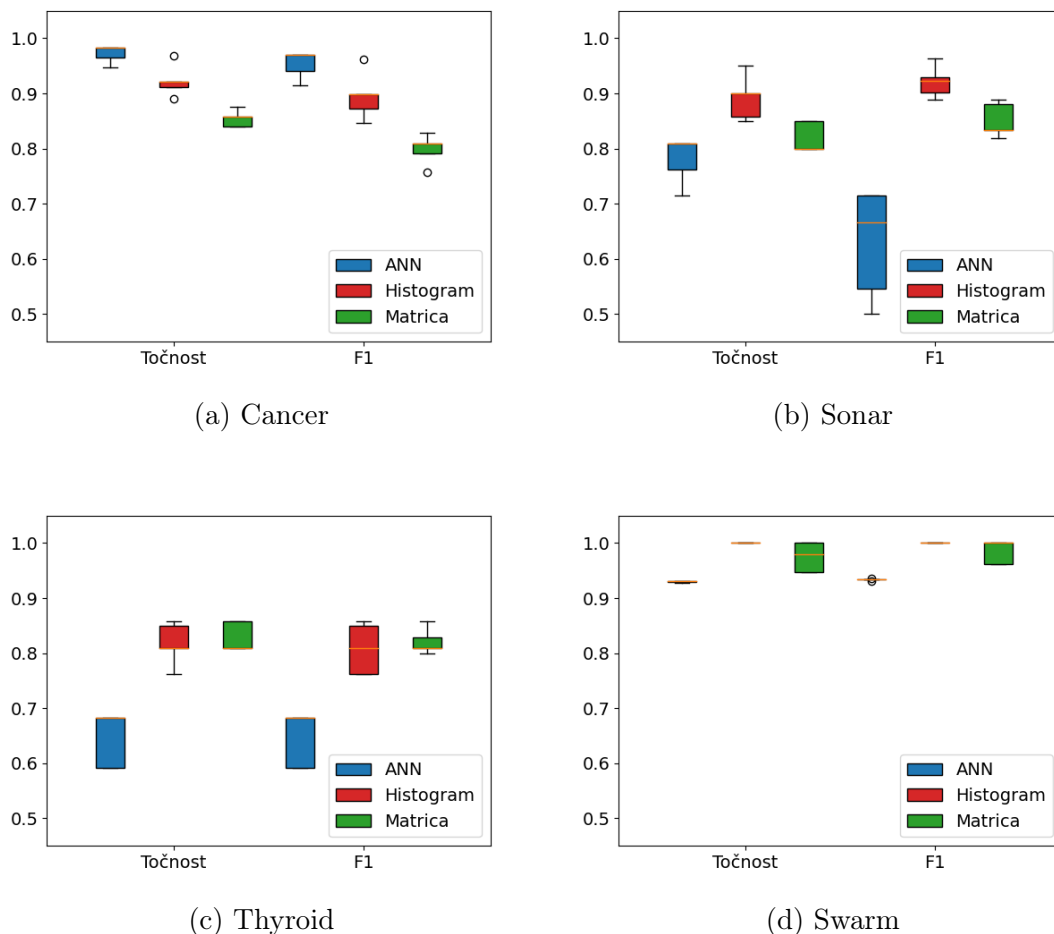
Nakon treninga algoritma, on je dodatno testiran na testnom skupu. Pri tome je uz točnost u obzir uzeta i F1 metrika, da bi se bi se dobile informacije i o performansama klasifikatora u slučaju eventualne neuravnoteženosti klasa. Dobiveni rezultati su prikazani kutijastim dijagramom na slici 4.3. Na dijagramu je plavom bojom označena metrika točnosti, a zelenom F1 metrika za svaki od korištenih načina klasifikacije.

Iz dobivenih rezultata može se vidjeti da korišteni algoritmi mogu izvršiti transformaciju podataka u slikovni oblik iskoristiv za treniranje konvolucijske neuronske mreže. Pokazalo se da algoritmi u većini slučajeva postižu obećavajuće rezultate jer na nekim od skupova u prosjeku postižu i oko 8% bolje rezultate od umjetne neuronske mreže s kojom su uspoređeni.

Kod skupova podataka s malim brojem uzoraka, umjetna neuronska mreža postiže slabije rezultate od konvolucijske. To se može djelomično pripisati lošim rezultatima umjetne neuronske mreže na tim skupovima. Kod srednjih i velikih skupova



## Poglavlje 4. Rezultati



Slika 4.3 Kutijasti dijagrami za metrike točnosti i F1 na skupu za testiranje

podataka, oba pristupa daju znatno bolje rezultate. Kod skupa podataka o raku dojke, umjetna neuronska mreža postiže bolje rezultate u obje praćene metrike. S druge strane, na najvećem skupu podataka konvolucijska neuronska mreža nadmašuje umjetnu neuronsku mrežu.

Na manjim skupovima podataka oba algoritma transformacije podataka u slike postižu podjednako dobre rezultate. Veća razlika je primjetna kod skupova podataka s većim brojem uzoraka, gdje algoritam transformacije u histogram postiže višu klasiifikacijsku točnost od algoritma transformacije u matricu udaljenosti. No, treba uzeti

## Poglavlje 4. Rezultati

u obzir da su dobiveni rezultati ostvareni bez dodatnih postupaka optimizacije hiperparametara, te je moguće da bi njihova optimizacija pomogla u poboljšanju rezultata klasifikacije, kako kod konvolucijskih, tako i kod umjetnih neuronskih mreža.

Skup podataka	ANN	Histogram	Matrica
Breast Cancer	<b>98.3%</b>	96.9%	87.5%
Thyroid	78.0%	<b>85.7%</b>	<b>85.7%</b>
Sonar	81.0%	<b>92.9%</b>	90.0%
Swarm Behavior	93.1%	<b>100.0%</b>	<b>100.0%</b>

Tablica 4.4 Najbolja dobivena točnost na skupu za testiranje

Iako je maksimalan broj epoha treniranja postavljen na 50, u većini slučajeva je algoritam zbog implementiranog ranog zaustavljanja s treningom završio i prije nego je dosegnut maksimalan broj epoha. Kod manjih skupova podataka se pokazalo da čak i uz implementirane mjere regularizacije može doći do pojave pretjerane prilagodbe podacima, što čini ovaj pristup nepogodan za skupove podataka s malim brojem instanci. Jedan od mogućih načina da se ovaj problem zaobiđe je povećanja broja uzoraka primjenom transformacija kao što su rotacija ili skaliranje na postojeće uzorke. U tablici 4.4 su prikazani najbolji dobiveni rezultati korištenjem svakog od algoritama.

Kada se, uz generalno bolje performanse, uzmu u obzir i faktori brzine treniranja i veličine transformiranog skupa podataka, algoritam transformacije u histogram se pokazao kao bolji izbor u većini slučajeva. U usporedbi s umjetnom neuronskom mrežom, konvolucijska neuronska mreža se sporije trenira, te podaci u slikovnom formatu potrebni za njeno treniranje zauzimaju više memorijskog prostora, što čini umjetnu neuronsku mrežu praktičnijom za korištenje.

# Poglavlje 5

## Zaključak

U ovom je radu opisan način transformacije visokodimenzionalnih podataka u slikovni oblik, te njihova klasifikacija korištenjem konvolucijske neuronske mreže. Testirana su dva načina transformacije podataka - transformacija u histogram i transformacija u matricu udaljenosti. Također je dan i pregled algoritma konvolucijskih neuronskih mreža i njegovih najvažnijih dijelova, te je opisana arhitektura konvolucijske neuronske mreže korištena za klasifikaciju podataka.

Kroz testiranje se pokazalo da pristup transformacije podataka u slikovni oblik u svrhu klasifikacije daje obećavajuće rezultate. Ovisno o skupu podataka, dobiveni rezultati su podjednako dobri ili bolji od rezultata dobivenih klasifikacijom koristeći klasičan pristup. Od dva testirana algoritma, pokazalo se da algoritam transformacije u histogram generalno daje do 7% bolje rezultate od algoritma transformacije u matricu udaljenosti. Međutim, pokazalo se da performanse oba algoritma variraju ovisno o veličini skupa podataka. Pokazalo se kako veći skupovi, oni s više značajki ili uzoraka, daju stabilnije rezultate s većom točnošću klasifikacije. U slučaju seta podataka o ponašanju roja ima 7% veću točnost klasifikacije. Kod setova podataka o očitanjima sonara i tiroidne žlijezde, konvolucijske neuronske mreže daju bolje rezultate od umjetnih neuronskih mreža, prvenstveno iz razloga što umjetne neuronske mreže nisu uspjele ostvariti dobre rezultate treniranja. Kod skupa podataka o raku dojke, umjetne neuronske mreže daju bolje rezultate od konvolucijskih neuronskih mreža, iako su oba algoritma postigla visoku točnost klasifikacije. Između dva korištena algoritma transformacije podataka u slikovni oblik, u većini slučajeva oba

## *Poglavlje 5. Zaključak*

algoritma postižu podjednako dobre rezultate, s razlikom manjom od 3%. Jedina iznimka je skup podataka o raku dojke, gdje algoritam transformacije u histogram postiže 7% veću točnost klasifikacije.

Također se pokazalo da sam proces transformacije podataka u slikovni oblik može biti računski i vremenski vrlo zahtjevan, pogotovo kod skupova podataka velikih dimenzija. Pohrana podataka u slikovnom obliku se također je znatno zahtjevnija od pohrane u njihovom izvornom obliku. U usporedbi algoritama transformacije, pokazalo se da slike u formatu histograma koriste znatno manje računalnih resursa. Kod najvećeg korištenog skupa podataka, slike histograma zauzimaju čak 190 puta manje memorijskog prostora od slika matrice. U kombinaciji s boljim rezultatima klasifikacije, može se zaključiti da je algoritam transformacije u histogram bolji izbor u većini analiziranih slučajeva.

Daljnje istraživanje ove teme bi podrazumijevalo testiranje novih metoda transformacije podataka u slikovni oblik, te poboljšanje rezultata klasifikacije kod trenutno korištenih algoritama. Kod velikih skupova podataka, vrijedi se usredotočiti na pronalaženje metode transformacije koja daje kompaktnije rezultate, te je manje računski kompleksna. Kod manjih skupova podataka, fokus bi se trebao staviti na način da se poveća dimenzionalnost podataka, bilo uvođenjem dodatnih značajki na temelju postojećih, ili dodavanjem dodatnih uzoraka korištenjem neke od metoda manipulacije slikom da se dobiju varijacije trenutno postojećih uzoraka.

# Bibliografija

- [1] A. Sharma and D. Kumar, “Non-image data classification with convolutional neural networks,” 7 2020.
- [2] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, “Conceptual understanding of convolutional neural network- a deep learning approach,” *Procedia Computer Science*, vol. 132, pp. 679–688, 2018, international Conference on Computational Intelligence and Data Science. , s Interneta, <https://www.sciencedirect.com/science/article/pii/S1877050918308019>
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [4] A. Nait Aicha, G. Englebienne, K. van Schooten, M. Pijnappels, and B. Krose, “Deep learning to predict falls in older adults based on daily-life trunk accelerometry,” *Sensors*, vol. 18, p. 1654, 05 2018.
- [5] Cross-Correlation, 16.9.2021. , s Interneta, <https://mathworld.wolfram.com/Cross-Correlation.html>
- [6] K. Bai, A Comprehensive Introduction to Different Types of Convolutions, 16.9.2021. , s Interneta, <https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>
- [7] R. Khandelwal, Convolutional Neural Network(CNN) Simplified, 18.9.2021. , s Interneta, <https://medium.datadriveninvestor.com/convolutional-neural-network-cnn-simplified-ecafd4ee52c5>
- [8] S. Saha, A comprehensive guide to convolutional neural networks the easy way, 15.9.2021. , s Interneta, <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [9] CS231n Convolutional Neural Networks for Visual Recognition, 25.9.2021. , s Interneta, <https://cs231n.github.io/convolutional-networks/>

## Bibliografija

- [10] K. Džomba, “Konvolucijske neuronske mreže,” Master’s thesis, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, 11 2018.
- [11] Stride and padding, 26.9.2021. , s Interneta, <https://deepai.org/machine-learning-glossary-and-terms/stride>
- [12] J. Brownlee, A Gentle Introduction to Pooling Layers for Convolutional Neural Networks, 25.9.2021. , s Interneta, <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
- [13] N. Aljaafari, “Ichthyoplankton classification tool using generative adversarial networks and transfer learning,” Ph.D. dissertation, King Abdullah University of Science and Technology, 02 2018.
- [14] Activation Functions in Deep Learning, 3.10.2021. , s Interneta, <https://xzz201920.medium.com/activation-functions-linear-non-linear-in-deep-learning-relu-sigmoid-softmax-swish-leaky-relu-a6333be712ea>
- [15] S. Sharma, Activation Functions in Neural Networks, 3.10.2021. , s Interneta, <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- [16] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, *Efficient BackProp*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. , s Interneta, [https://doi.org/10.1007/978-3-642-35289-8\\_3](https://doi.org/10.1007/978-3-642-35289-8_3)
- [17] A. Nagpal, L1 and L2 regularization methods, 5.10.2021. , s Interneta, <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>
- [18] S. Jain, Regularization techniques in deep learning, 5.10.2021. , s Interneta, <https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques>
- [19] D. Dua and C. Graff, “UCI machine learning repository,” 2017. , s Interneta, <http://archive.ics.uci.edu/ml>
- [20] Feature scaling, 25.9.2021. , s Interneta, <https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization>
- [21] A. Sharma and D. Kumar, “Classification with 2-d convolutional neural networks for breast cancer diagnosis,” *Synthesis lectures on artificial intelligence and machine learning*, 7 2020.
- [22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

## *Bibliografija*

- [23] Performance Metrics for Classification problems in Machine Learning, 15.10.2021. , s Interneta, <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>
- [24] K. P. Shung, Accuracy, Precision, Recall or F1, 15.10.2021. , s Interneta, <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>

# Sažetak

U ovom je radu opisan način transformacije visokodimenzionalnih podataka u slikovni oblik, te njihova klasifikacija korištenjem konvolucijske neuronske mreže. Testirana su dva načina transformacije podataka - transformacija u histogram i transformacija u matricu udaljenosti. Dobiveni rezultati su zatim uspoređeni sa algoritmom umjetne neuronske mreže koji je korišten za klasifikaciju podataka u njihovom izvornom obliku. Pokazalo se da algoritam transformacije u histogram na dva ispitana skupa podataka daje podjednako dobre rezultate kao algoritam transformacije u matricu, dok na preostala dva skupa daje do 7% bolje rezultate. U kombinaciji sa činjenicom da zahtjeva manje računskih resursa, algoritam transformacije u histogram se pokazao kao bolji izbor u većini slučajeva.

***Ključne riječi*** — **strojno učenje, konvolucijske neuronske mreže, konvolucija, duboko učenje, transformacija podataka**

## Abstract

This paper describes a method of transforming high-dimensional data into an image form, and their classification using a convolutional neural network. Two ways of data transformation were tested - transformation into a histogram and transformation into a distance matrix. The obtained results were then compared with the artificial neural network algorithm used to classify the data in their original form. The results that the algorithm of transformation into a histogram gives equally good results as the algorithm of transformation into a matrix on two of the examined data sets, while on the remaining two sets it gives up to 7% better results. Combined with the fact that it requires less computational resources, the histogram transformation algorithm has proven to be a better choice in most cases.

***Keywords*** — **machine learning, convolutional neural networks, convolution, deep learning, data transformation**



# Dodatak A

## Numerički prikaz rezultata

U ovom prilogu je prikazan numerički zapis rezultata dobivenih na testnom skupu podataka, koji je u obliku kutijastog dijagrama već prikazan na slici 4.3. U tablici A.1 je prikazana metrika točnosti, dok je u tablici A.2 prikazana F1 metrika. Kako se radi o prosječnoj vrijednosti, uz nju je dopisana i vrijednost standardne devijacije za svaku od metrika.

<b>Skup podataka</b>	<b>ANN</b>	<b>Histogram</b>	<b>Matrica</b>
Breast Cancer	97.2%(±1.4)	92.3%(±2.5)	85.3%(±1.3)
Thyroid	74.2%(±2.6)	81.8%(±3.4)	82.8%(±2.3)
Sonar	78.1%(±3.8)	88.1%(±2.3)	90.5%(±2.4)
Swarm Behavior	92.9%(±0.1)	100.0%(±0.0)	97.4%(±2.4)

Tablica A.1 Točnost na testnom skupu

<b>Skup podataka</b>	<b>ANN</b>	<b>Histogram</b>	<b>Matrica</b>
Breast Cancer	95.1%(±2.1)	89.5%(±3.8)	79.9%(±2.4)
Thyroid	64.5%(±4.4)	81.8%(±3.4)	82.1%(±2.0)
Sonar	66%(±6.9)	92%(±2.4)	85.1%(±2.8)
Swarm Behavior	93.4%(±0.1)	100.0%(±0.0)	98.5%(±1.9)

Tablica A.2 F1 metrika na testnom skupu