

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6391

**Detekcija i klasifikacija tekstovnih  
elemenata na slici koristeći duboke  
neuronske mreže**

Lukas Šestić

Zagreb, travanj 2019.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

*Zahvaljujem se prof. dr. sc. Domagoju Jakoboviću na pruženoj pomoći i sredstvima danim u svrhu uspješne izrade završnog rada.*

*Također se zahvaljujem svojim roditeljima na prilici za studiranje i stalnoj podršci.*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
1.1. Uvod . . . . .	1
1.2. Računalni vid . . . . .	1
1.2.1. Pregled . . . . .	1
1.2.2. Konvolucijske neuronske mreže . . . . .	2
<b>2. Generiranje Slika</b>	<b>4</b>
2.1. Apstrakcija postupka . . . . .	4
<b>Literatura</b>	<b>5</b>

# 1. Uvod

## 1.1. Uvod

Računalna moć uređaja koje gotovo neprestano nosimo sa sobom kao što su pametni telefoni i prijenosna računala je unazad deset godina eksponencijalno narasla. Naravno, sa modernim alatima dolaze i moderni problemi.

Jedan najraširenijih alata, koji se sve više i više koristi za rješavanje problema koji su do nedavno bili nemogući, ili izrazito algoritamski komplicirani za riješiti je *umjetna inteligencija*. Umjetna inteligencija podrazumjeva skup načina i metoda koje računalu opisuju početno i konačno stanje do kojeg mora doći sam.

Ovaj rad, bavit će se najkorištenijom metodom umjetne inteligencije, *dubokim učenjem*, i njegovim podskupom *računalnim vidom*. Kroz rad i programsku implementaciju, prihvatiti ću se problema detekcije napisanog teksta na slici i daljnom obradom istog.

Detaljno ću kroz poglavlja obraditi postupke koje sam primjenio za generiranje raznolikih slika koje imitiraju rukopis i proces potreban da računalo nauči prepoznavati isti na slici.

Na kraju, izlučeni tekst sa slike, biti će moguće obraditi na željeni način. Način koji ću ja predstaviti biti će primjena jednostavne matematike, slično onome što pruža *Photomath, Inc.*. Na primjer, za sliku na kojoj je napisan tekst " $2 + 2$ ", izlaz će biti slika sa kvadratima oko prepoznatih simbola, i rješenje obrađenog teksta, u ovom slučaju " $4$ ".

## 1.2. Računalni vid

### 1.2.1. Pregled

Na najvišoj razini, *računalni vid* su metode koje računalima daju mogućnost razumjevanja slike na visokoj razini, najčešće s ciljem automatiziranja ljudskih

poslova. Osnovni zadatak je raspoznavanje veze između obrazaca na slici i rješenja na problem koji želi riješiti. Svi procesi koji koriste strojno učenje, u konačnici se svode na detekciju i klasifikaciju elemenata na slici. Metode računalnog vida temelje se na geometriji, statistici, fizici i teoriji učenja.

Danas, se velika količina problema rješava uz pomoć računalnog vida, često da ljudi za to nisu ni svjesni:

- Prepoznavanje znakova (Slika 1.1)
- Prepoznavanje lica
- Kompresija i restauracija slike
- Prepoznavanje elemenata na slici
- Analiza medicinskih snimki u svrhu detaljnije analize
- Itd.



**Slika 1.1:** Maskiranje elemenata na slici prometa

### 1.2.2. Konvolucijske neuronske mreže

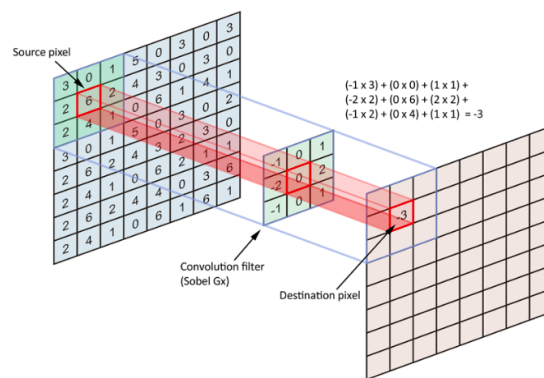
Konvolucijske neuronske mreže danas se koriste kao najefektivniji način postizanja računalnog vida. Glavna prednost nad potpuno povezanim neuronskim mrežama je manji broj *težina* za treniranje što znatno ubrzava treniranje. Ipak, ono što je možda najvažnije za napomenuti je to što pozicija traženog elementa na slici *konvolucijskoj neuronskoj mreži* ne igra ulogu.

Svaki sloj *duboke konvolucijske neuronske mreže* funkcionira kao filter koji se kreće po slici, pamteći što ga je najviše aktiviralo. Najčešće se koristi filter veličine  $3 \times 3$ . (Slika 1.2)

Dalje, uz konvolucijski sloj, nerijetko se postavlja *max pooling sloj*. Na apstraktnoj razini, princip rada max pooling sloja je sljedeći: Ako uzmemo veličinu pooling filtra kao onu koja se najčešće koristi, to jest  $2 \times 2$ , on izlaz iz prethodnog sloja raspodjeli na kvadrate iste veličine. Zatim, filter se postavi između 4 kvadrata i sebi za vrijednost stavi najveću iz svakog u pripadajuće polje.

Prirodno je pitati se zašto se to koristi i zašto to radi.

Pooling filter jednostavno smanjuje "rezoluciju" prethodnog sloja, ne mjenjajući važne čimbenike potrebne za daljnji rad mreže. Na primjer, vertikalna linija, krug, ili elipsa, ostaje ono što je, jedino manje razlučivo. Bitno je napomenuti da smanjivanjem rezolucije dobivamo puno manje parametara za treniranje. Stavimo to u brojeve. Slike unutar *mnist* seta podataka su veličine  $28 \times 28$ . To znači da bi se treniralo  $28 \times 28$  parametara. Primjenom *Max pooling sloja* veličine  $2 \times 2$ , treniralo bi se  $\frac{1}{4} \times (28 \times 28)$  parametara.



**Slika 1.2:** Klizeći konvolucijski filter

Spomenute prednosti, referenciraju se na glavnu značajku *konvolucijskih mreža*. Cilj je ići dublje, ne šire. Za sliku veličine  $100 \times 100$ , potpuno povezanoj neuronskoj mreži u prvom sloju treba 10 000 čvorova, svaki sa svojim parametrom za treniranje, dok konvolucijskoj to ne treba.

Svaki sljedeći sloj ima drugu ulogu. Prvi najčešće ima ulogu raspoznavanja najosnovnijih elemenata slike kao što su različiti rubovi, dok sve dublji koriste podatke od prošlih i osnovne elemente grupiraju u apstraktne strukture koji predstavljaju značajnije elemente slike.

## 2. Generiranje Slika

### 2.1. Apstrakcija postupka

Prikupljanje podataka za treniranje duboke neuronske mreže prvi je praktični korak treninga. (O'Shea i Nash, 2015, p. 199)



# LITERATURA

Računalni vid. [https://en.wikipedia.org/wiki/Computer\\_vision](https://en.wikipedia.org/wiki/Computer_vision).

Keiron O'Shea i Ryan Nash. An introduction to convolutional neural networks.  
*arXiv preprint arXiv:1511.08458*, 2015.