SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 5705

# Pronalaženje teksta dubokim konvolucijskim modelima

Kristijan Fugošić

Zagreb, srpanj 2018.

*Zahvaljujem se svojoj obitelji na bezuvjetnoj podršci tijekom cijelog školovanja i mentoru prof. dr. sc. Siniši Šegviću na pomoći pri izradi rada.*

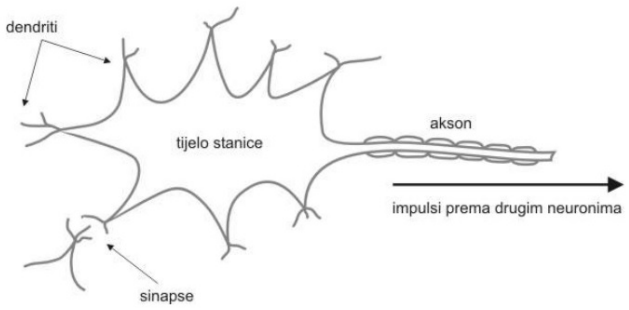
# Sadržaj

# 1. Uvod

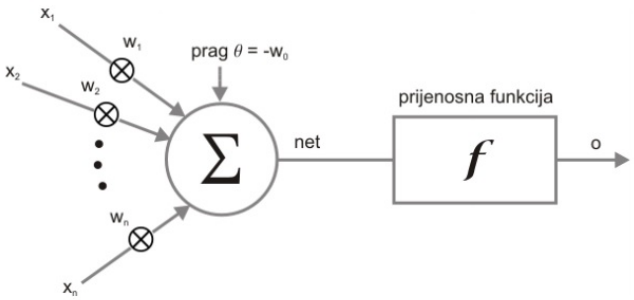
Računalni vid je područje umjetne inteligencije koje se bavi izlučivanjem i analiziranjem korisnih informacija iz slika. Formirao se u šezdesetim godinama prošloga stoljeća, a u posljednjih desetak godina doživljava pravi procvat koji se najbolje vidi kroz raznolike primjene. Kao primjer možemo navesti primjenu računalnog vida u medicini, u procesima proizvodnje, a samovozeći automobili su također sve popularnija tema. Neki mobiteli i računala pružaju nam mogućnost otključavanja uređaja prepoznavanjem lica, no prepoznavanje lica koristi se i u daleko ozbiljnijim stvarima. Kina trenutno ima najveći sustav nadzora koji pomoću kamera može prepoznati lice, procijeniti godine, spol i etnicitet, te već danas ostvaruje rezultate u pronalaženju kriminalaca. Računalni vid se koristi i za očitavanje tekstova i brojeva, a kao primjer možemo navesti prevođenje teksta vidljivog putem kamere u realnom vremenu ili evaluaciju rukom pisanih jednadžbi.
Ovaj rad usredotočuje se na prvi dio strojnog očitanja teksta, odnosno na pronalaženje teksta. Upotreba dubokog učenja u računalnom vidu omogućila je da cijeli proces ručnog pronalaženja značajki prepustimo dubokoj neuronskoj mreži. U radu će biti predstavljena osnovna ideja konvolucijskih i rezidualnih neuronskih mreža, te njihova primjena unutar modela za pronalaženje teksta EAST. Model EAST čini središnji dio ovoga rada, promotrit ćemo njegovu arhitekturu, performanse, te usporediti iste sa starijim i novijim modelima. Na kraju će biti prikazani eksperimentalni rezultati na kojima možemo vidjeti mane ovoga modela i prostor za daljnji napredak.

# 2. Umjetne neuronske mreže

Umjetne neuronske mreže predstavljaju konektivistički pristup umjetnoj inteligenciji, a nastale su kao pokušaj izrade softvera na temelju koncepata inspiriranih biološkim funkcijama mozga. Neuronske mreže uče samostalno na temelju iskustva, a mogu raditi i s nejasnim ili manjkavim podacima. Osnovna gradivna jedinica neuronskih mreža jest neuron koji se temelji na biološkom neuronu, te je između njih moguće povući analogiju: signali su numeričke vrijednosti, jakost sinapse opisuje težinski faktor $w$, zbrajalo ima ulogu tijela stanice, a akson je prijenosna (aktivacijska) funkcija $f$.




**Slika 2.1:** Biološki neuron. [4]




**Slika 2.2:** Umjetni neuron. [4]

Svaki neuron množi podatke s ulaza $x_i$ s pripadnim težinama $w_i$ (eng. *weights*) i umnošku pridodaje prag $b$ (eng. *bias*). Prag se često zapisuje kao $w_0$ uz vrijednost $x_0$=1. Izlaz iz zbrajala opisan je izrazom 2.1.

$$net = \sum_{i=0}^{n} \omega_i x_i \tag{2.1}$$

Na dobivenu vrijednost *net* primjenjuje se aktivacijska funkcija čija je osnovna zadaća učiniti mrežu nelinearnom. U izrazima 2.3 prikazane su neke aktivacijske funkcije. Logistička funkcija ima velik povijesni značaj upravo zato što je *"donijela"* nelinearnost u neuronske mreže. Trenutno je najpopularnija funkcija ReLU (Rectified linear unit) koja pretvara sve negativne vrijednosti u nulu. Rezultat prolaska kroz aktivacijsku funkcija ujedno je i izlaz iz neurona, odnosno ulaz u sljedeći neuron.

$$f(net) = net$$

**(a)** Funkcija identiteta (ADALINE)

$$f(net) = \begin{cases} 0 & \text{za net} < 0 \\ 1 & \text{inače} \end{cases}$$

**(b)** Skokovita funkcija (TLU)

$$f(net) = \frac{1}{1 + e^{-a*net}}$$

**(c)** Logistička funkcija (sigmoid)

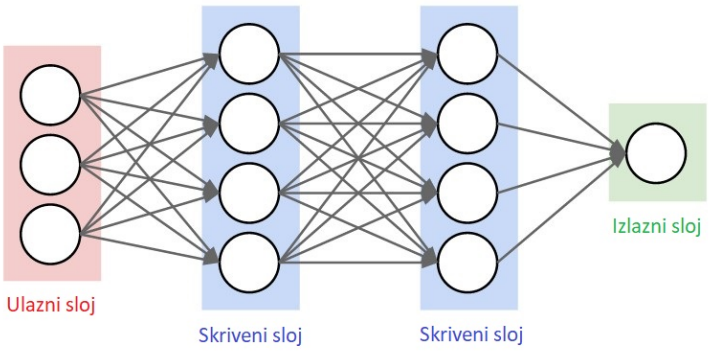
$$f(net) = \begin{cases} 0 & \text{za net} < 0 \\ net & \text{inače} \end{cases}$$

**(d)** ReLU

**Slika 2.3:** Primjeri aktivacijskih funkcija.

Znanje neuronskih mreža pohranjeno je u težinama $w_i$. Težine je potrebno učiti iterativnim predočavanjem ulaznih podataka, a kada govorimo o nadziranom učenju uz ulazne podatke se prilažu i očekivani izlazni podatci.

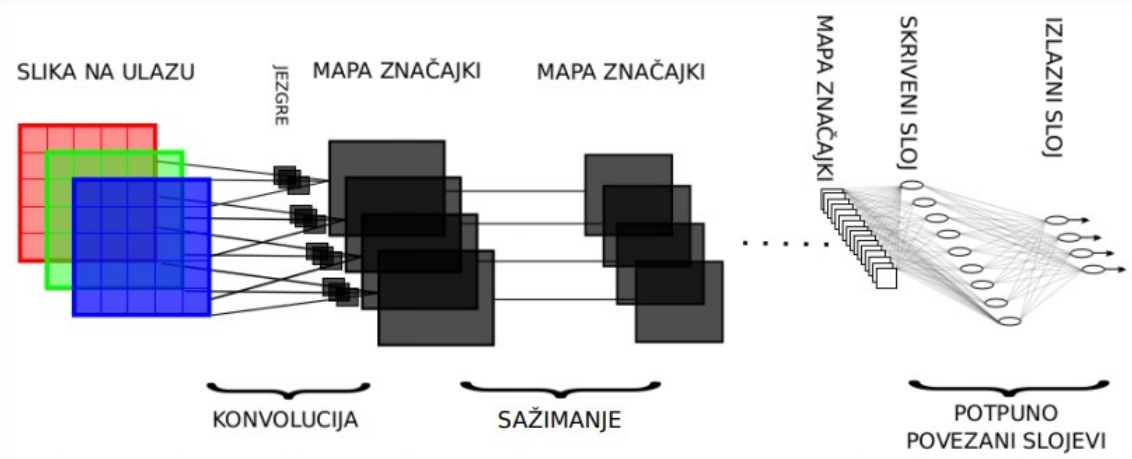
Neuronske mreže obično se modeliraju kao aciklični grafovi veće količine neurona. Neuroni su obično razdvojeni u slojeve, a kada je riječ o klasičnim neuronskim mrežama najčešći tip sloja je potpuno povezani sloj. Neuroni u potpuno povezanim slojevima povezani su sa svim izlazima prethodnog sloja, a neuroni unutar jednog potpuno povezanog sloja nisu međusobno povezani. Umjetna neuronska mreža se obično sastoji od ulaznog sloja, skrivenih slojeva i izlaznog sloja. Mreža koja se sadrži barem jedan skriveni sloj naziva se duboka neuronska mreža, a jedna takva mreža prikazana je na slici 2.4.




**Slika 2.4:** Duboka neuronska mreža sačinjena od ulaznog sloja, dva skrivena sloja i izlaznog sloja. [3]

# 3. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže su podvrsta neuronskih mreža i često su korištene u računalnom vidu. Zahtijevaju minimalnu ili nikakvu prethodnu obradu slika te imaju sposobnost samostalno naučiti značajke koje su se u tradicionalnim algoritmima ručno osmišljale. Konvolucijska neuronska mreža obično se sastoji od tri sloja: konvolucijskog, sloja sažimanja i potpuno povezanog sloja.

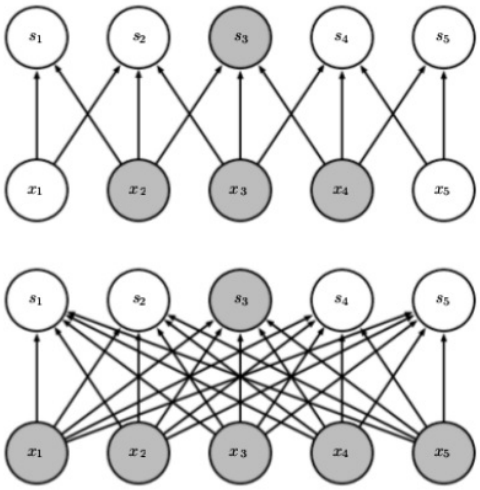



**Slika 3.1:** Primjer arhitekture konvolucijske mreže. [6]

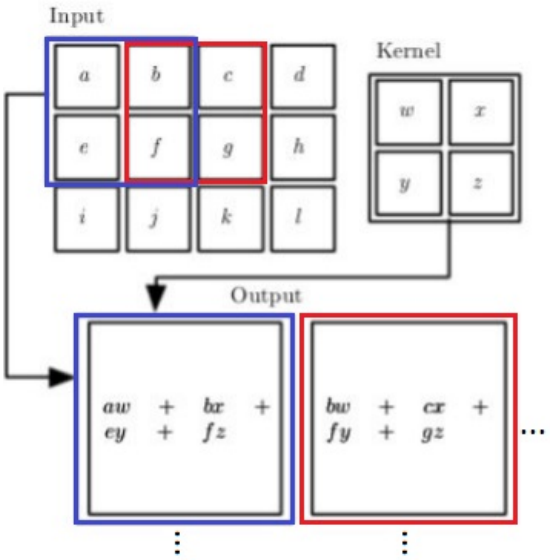
## 3.1. Konvolucijski sloj

Konvolucijski sloj je osnovni dio konvolucijskih neuronskih mreža i ono što ih razlikuje od ostalih neuronskih mreža. Konvolucijski sloj sličan je potpuno povezanom, no bitna je razlika u tome što je kod potpuno povezanog sloja svaki neuron povezan sa svim neuronima prethodnog sloja, a kod konvolucijskog samo s malim dijelom. Puno manji broj veza znači i puno manje parametara, odnosno težina, što povlači i puno bržu evaluaciju. U filterima (jezgrama) se nalaze parametri konvolucijskoj sloja. Filteri su uglavnom malih dimenzija po širini i visini, dok dubinom moraju odgovarati ulaznom sloju. Filteri se pomiču za određen korak

(eng. *stride*) po ulaznom sloju, te nad svakim djelom slike obavljaju matematičku operaciju. Preciznije, množe se vrijednosti ulaznog sloja s vrijednostima filtera, te se na kraju svi umnošci zbrajaju i tvore rezultat primjene filtera na određeni dio ulaznog sloja.



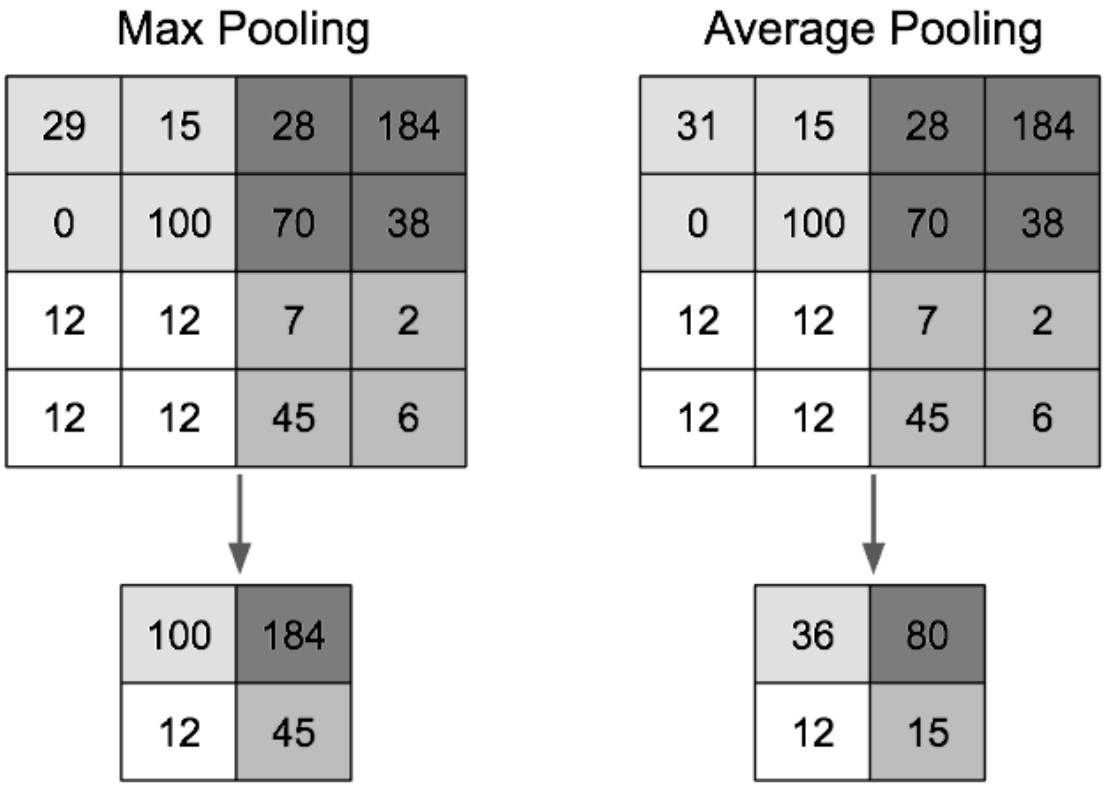

**(a)**
Usporedba konvolucijskog sloja (gore) i potpuno povezanog sloja(dolje). [6]




**(b)** Prikaz pomičnog prozora. [6]

## 3.2. Sloj sažimanja

Sloj sažimanja obično se nalazi između uzastopnih konvolucijskih slojeva. Funkcija sažimanja mapira skup prostorno bliskih značajki na ulazu u jednu značajku na izlazu. Sažimanjem se smanjuju prostorne dimenzije što kao posljedicu ima smanjenje broja parametara, odnosno manju količinu računanja. Slojevi sažimanja najčešće su dimenzije 2x2 s korakom 2 u širinu i visinu. Takvi slojevi eliminiraju 75% podataka. U prošlosti se često koristio *Average pooling* koji na izlazu iz pojedinog okvira vraća prosjek vrijednosti tog okvira, dok je danas najpopularniji *Max pooling* koji na izlazu vraća maksimalnu vrijednost pojedinog okvira, a postoje i drugi. Dobro je spomenuti i da postoje pristupi u kojima se izbjegava korištenje sloja sažimanja, te se umjesto njega, bez gubitka preciznosti, koristi konvolucijski sloj s većim korakom.




**Slika 3.3:** Sloj sažimanja dimenzija 2x2 s korakom 2. [5]

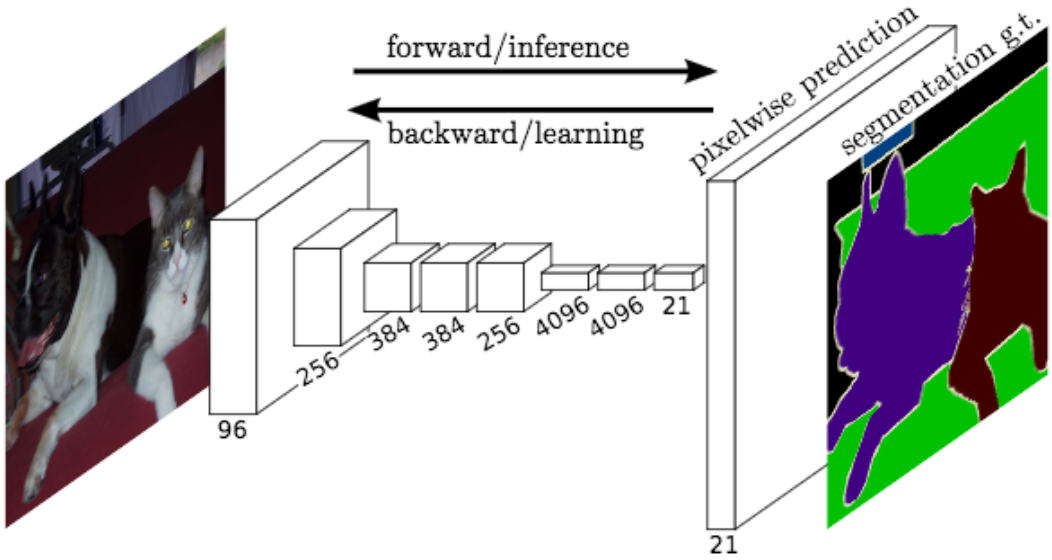
## 3.3. Potpuno povezani sloj

Osnovna svojstva potpuno povezanih slojeva spomenuta su u drugom poglavlju, a ovdje ćemo razmatrati njihovu ulogu unutar konvolucijskih neuronskih mreža. Potpuno povezani sloj obično se nalazi na kraju konvolucijske mreže, a njegova je uloga iz ranije naučenih značajki formirati vektor određene dimenzije. Na primjer, ako želimo da naša konvolucijska neuronska mreža određuje je li na slici pas, mačka ili ptica, potpuno povezani sloj vratit će nam vektor dimenzije 3, koji ćemo normalizirati korištenjem softmax funkcije kako bi naš rezultat bio u obliku tri realna broja između 0 i 1 koja se zbrajaju u 1 i koja označavaju vjerojatnosti je li na slici pas, mačka ili ptica. Potpuno povezani slojevi imaju fiksan broj ulaza i izlaza, što u nekim slučajevima može biti nepoželjno, no moguće ih je pretvoriti u konvolucijske[1].

[1] https://stackoverflow.com/a/39367644

## 3.4. Potpuno konvolucijske neuronske mreže

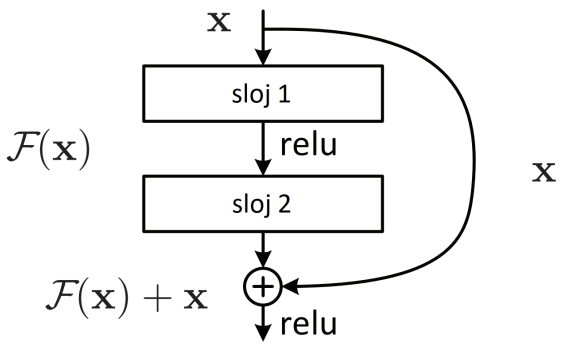
Potpuno konvolucijske neuronske mreže su one konvolucijske neuronske mreže koje ne sadrže potpuno povezani sloj, što znači da na ulazu može biti slika proizvoljnih dimenzija. Koriste se kada su nam bitne informacije na bazi piksela, najčešće u segmentaciji slika.




**Slika 3.4:** Potpuno konvolucijska neuronska mreža za segmentaciju slika. [10]

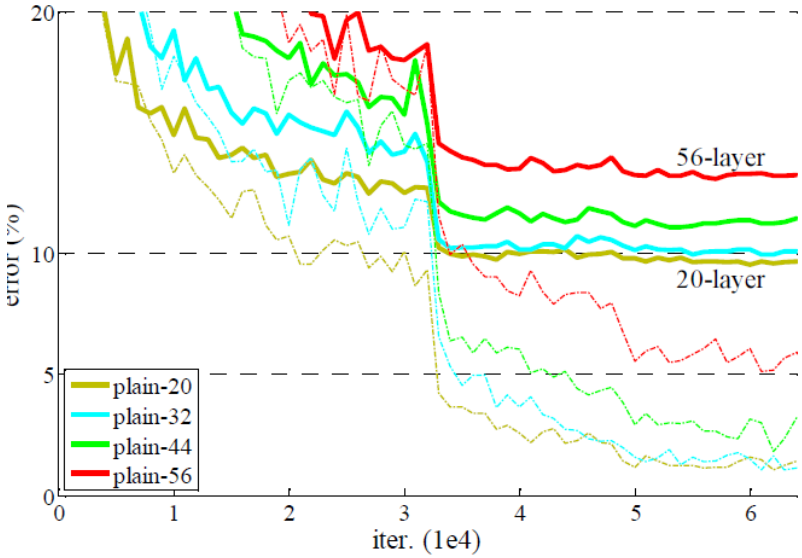
# 4. Rezidualne neuronske mreže

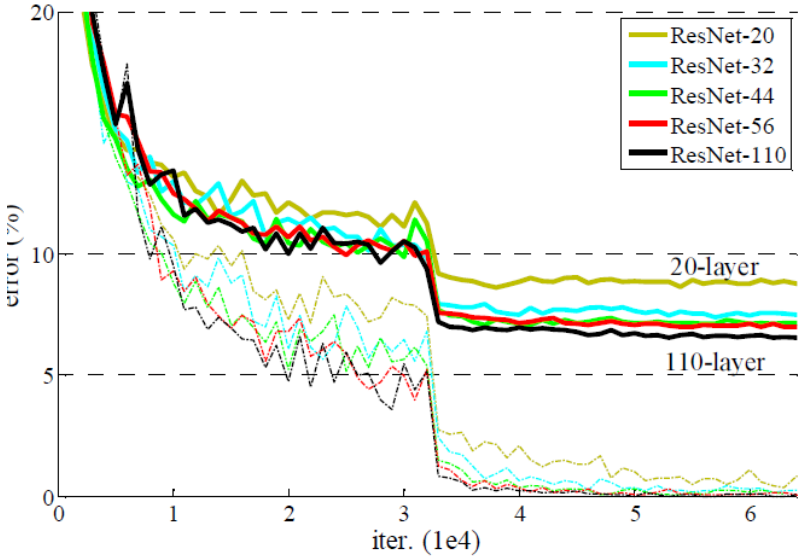
Duboke konvolucijske neuronske mreže postepeno uče značajke od jednostavnijih ka složenijima. Na primjer, ako imamo mrežu za klasifikaciju životinja, prvi sloj će naučiti pronalaziti jednostavne stvari poput rubova, sljedeći sloj prepoznavat će malo složenije stvari poput jednostavnih oblika ili tekstura, a idući sloj prepoznavat će stvari još više razine poput lica. Iako veći broj slojeva doprinosi performansama, pokazalo se da dodavanje prevelikog broja slojeva ima negativan utjecaj. Rezidualne neuronske mreže rješavaju taj problem. Motivacija je sljedeća: Zamislimo mrežu A koja ima određenu grešku pri treniranju. Konstruirajmo mrežu B tako da na mrežu A dodamo još slojeve X i Y, ali na način da ne mijenjaju izlaz iz mreže A. Očekivali bi da mreža B nema veću grešku pri treniranju od mreže A, no to nije slučaj. To pokazuje da slojevi X i Y utječu na mrežu, iako je njihova uloga prepisivanje podataka sa svog ulaza na izlaz. Element specifičan za rezidualne mreže je rezidualni blok prikazan na slici 4.1. Njime se postiže da se rezultatu na izlazu proizvoljnog broja slojeva prije aktivacijske funkcije pridoda ono što je u mreži naučeno prije tih slojeva. Ako slojevi ne doprinose mreži, ovime se barem osiguravamo da joj neće niti pretjerano naštetiti. Usporedba performansi klasičnih i rezidualnih mreža na skupu CIFAR-10[1] prikazana je na slici 4.2.

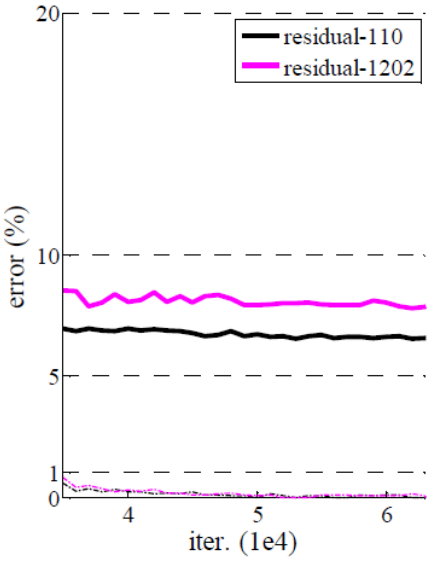



**Slika 4.1:** Rezidualni blok. [7]

[1]https://www.cs.toronto.edu/ kriz/cifar.html



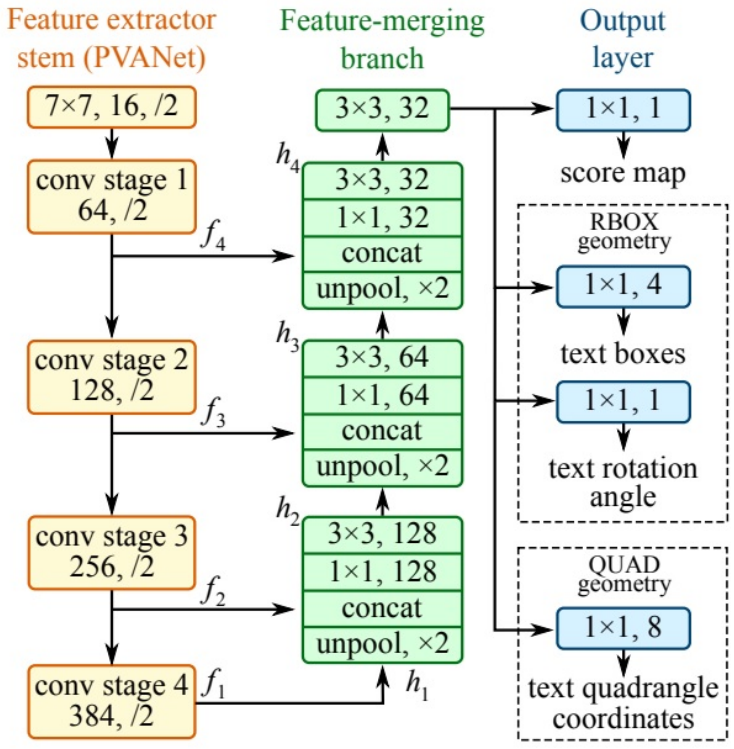






**Slika 4.2:** Usporedba performansi, tanke linije označavaju greške na skupu za treniranje, a debele linije označavaju greške na skupu za testiranje. [7]

# 5. EAST

EAST (An Efficient and Accuracy Scene Text Detector) je model za pronalaženje (eng. *Localization*) teksta objavljen u travnju 2017. godine. Model EAST bazira se na jednostavnoj arhitekturi koja po preciznošću i brzini nadmašuje tadašnje najbolje modele za pronalaženje teksta. Tadašnji modeli su pronalaženje teksta izvršavali u više koraka, kao primjer možemo navesti Zhang et al. [13] i neke od koraka koje koristi: pronalaženje linije teksta, prepoznavanje slova i razdvajanje na riječi. EAST model direktno pronalazi riječi ili linije teksta proizvoljne orijentacije, a kao rezultat dobiven je model koji je istovremeno jednostavniji, precizniji i brži od svojih prethodnika. Model se može jednostavno modificirati tako da prepoznaje cijele linije teksta ili zasebne riječi, u obliku četverokuta ili rotiranih pravokutnika.




**Slika 5.1:** Arhitektura modela EAST. [14]

## 5.1. Arhitektura modela EAST

Ključna komponenta detektora teksta EAST je potpuno konvolucijska neuronska mreža koja pronalazi riječi ili linije teksta na razini piksela. Shematski prikaz modela (Slika 5.1) razdvojen je u tri dijela. Prvi dio sačinjen je od rezidualne neuronske mreže s 50 slojeva ResNet50. Kako bi uspješno pronašli riječi i jako malih i jako velikih dimenzija, potrebno je koristiti mape značajki koje su dobivene na različitim stadijima rezidualne mreže. Kako bi to ostvarili, prihvaćena je ideja iz [9] o postupnom zbrajanju značajki. Možemo primijetiti da je na slici 5.1 rezidualna mreža razdvojena na četiri stadija, te da je izlaz iz svakog stadija označen s $f_i$. $f_1..f_4$ označava značajke dobivene na dimenzijama 1/4, 1/8, 1/16 i 1/32 početne slike.

Središnji dio modela EAST čini grana za spajanje, a redoslijed operacija na grani za spajanje je sljedeći:

1. Radimo *unpooling* na mapi značajki $h_{i-1}$ kako bi mape značajki $h_{i-1}$ i $f_i$ bile istih dimenzija
2. Pribrajamo kanale mapi značajki $h_{i-1}$ i $f_i$
3. Rezultat drugog koraka provlačimo kroz konvolucijski sloj 1x1 kako bi smanjili broj kanala
4. Rezultat trećeg koraka provlačimo kroz konvolucijski sloj 3x3

Redoslijed operacija također je opisan izrazima 5.1 i 5.2 preuzetim iz [14].

$$g_i = \begin{cases} unpool(h_i) & \text{za i} \leq 3 \\ \text{conv}_{3x3}(h_i) & \text{za i} = 4 \end{cases} \tag{5.1}$$

$$h_i = \begin{cases} f_i & \text{za i} = 1 \\ \text{conv}_{3x3}(\text{conv}_{1x1}([g_{i-1}; f_i])) & \text{inače} \end{cases} \tag{5.2}$$

Nakon sjedinjavanja mapa značajki s različitih stadija rezidualne neuronske mreže, prolazimo kroz još jedan konvolucijski sloj dimenzija 3x3 iz kojega dobivamo završne mapu značajki.

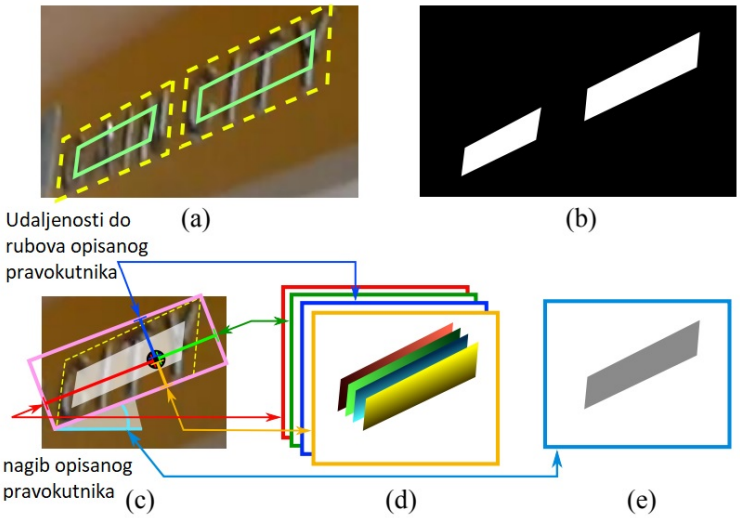
Završni dio modela EAST sadrži više jezgri 1x1 koji iz 32 kanala završne mape značajki , ovisno o potrebi, izvode:

1) Mapu sačinjenu od jednog kanala - mapa vrijednosti pojavljivanja teksta za svaki piksel

2) Mapu sačinjenu od četiri kanala - udaljenosti od piksela do rubova pravokutnika (gore, desno, dolje, lijevo)

3) Mapu sačinjenu od jednog kanala - nagib pravokutnika

4) Mapu sačinjenu od osam kanala - Četiri para x, y koordinata gdje svaki par označava udaljenost piksela do određenog vrha četverokuta




**Slika 5.2:** Proces generiranja opisanih okvira [14]

a) Opisani četverokut - pošto mreža ima tendenciju kao rezultat vratiti preširoke rezultate (žuta isprekidana linija), četverokut se sužava (zelena linija). b) Mapa vrijednosti pojavljivanja teksta c) Tekst opisan rotiranim pravokutnikom d) 4 kanala koja predstavljaju udaljenost svakog piksela do rubova pravokutnika e) Nagib pravokutnika

## 5.2. Suzbijanje nevažnih detekcija

Svakom opisanom četverokutu pridružena je vrijednost koja predstavlja vjerojatnost da se unutar tog četverokuta nalazi tekst, koju ćemo označavati s $p_c$.
Prvi korak pri suzbijanju nevažnih detekcija je eliminiranje svih četverokuta čije pridružene vrijednosti $p_c$ ne zadovoljavaju određeni prag[1].
U sljedećem koraku potrebno je izbaciti višestruke opisane četverokute za istu riječi, drugim riječima potrebno je za svaku lokaliziranu riječ ostaviti samo četverokut s najvećom vjerojatnošću detekcije teksta $p_c$, dok se ostali eliminiraju. Opisani zadatak rješava se upotrebom NMS (Non-max Suppression) algoritma:

1. Biramo opisani četverokut s najvećom $p_c$ vrijednošću
2. Iteriramo po svim ostalim četverokutima i tražimo one koji se preklapaju s izabranim. Ako je omjer presjeka i unije površina dvaju četverokuta (IoU[2]) veći od zadane konstante[3], eliminiramo četverokut s manjom vrijednošću $p_c$.

   *Ponavljamo korake dok nismo prošli sve četverokute, jednom odabrani četverokut u 1. koraku više ne biramo ponovno.*

U izradi modela EAST uvelike se brinulo o performansama, stoga se umjesto klasičnog NMS algoritma koje se izvodi u $O(n^2)$ koristio LANMS (Locality-Aware NMS). LANMS umjesto biranja četverokuta s najvećom vrijednošću $p_c$, vadi prosjek između četverokuta koji se međusobno preklapaju redak po redak. Iako LANMS u najgorem slučaju ima istu složenost kao i NMS, u praksi se pokazuje da je puno brži. Algoritam je implementiran u programskom jeziku C++ i pobliže je opisan u [14].


**Slika 5.3:** Prije NMS-a - 101 detekcija.

[1]U slučaju modela EAST, prag iznosi 0.8

[2]Intersection over Union https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[3]U modelu EAST ta konstanta iznosi 0.2

# 6. Programske podrška

## 6.1. Programski jezik Python

Python je interpretirani programski jezik visoke razine razvijen od strane nizozemskog programera Guida van Rossuma. Prvi puta je objavljen 1991. godine, popularnost stječe desetak godina kasnije, te je od 2003. do danas među deset najpopularnijih programskih jezika prema TIOBE indeksu. Opseg primjena veoma je širok, a uz biblioteke poput NumPy, SciPy i Matplotlib programski jezik Python efikasno se koristi u računarskoj znanosti, a naročito je popularan u području umjetne inteligencije.

## 6.2. Biblioteka Tensorflow

Tensorflow je trenutno najpopularniji programski okvir za razvoj modernih AI sustava. TensorFlow je razvijen od strane Googleovog tima Google Brain te je javno dostupan od studenog 2015. godine. TensorFlow izvodi izračune pomoću računskih grafova. Ime TensorFlow proizlazi iz operacija koje neuronske mreže obavljaju nad višedimenzionalnim podatkovnim poljima. Takva polja se nazivaju "tenzori". Tensorflow omogućava komputaciju na jednom ili više procesora ili grafičkih kartica, a dostupan je na 64-bitnim platformama Linux, macOs i Windows, ali i na mobilnim operacijskim sustavima iOs i Android što ga čini iznimno rasprostranjenim. Također, može se izvoditi i na ugradbenim sustavima poput TX2.

## 6.3. Instalacija programske podrške

Implementacija detektora teksta EAST je javno dostupna u GitHub repozitoriju[1]. Za potrebe evaluacije potrebno je imati instaliran Python 3.[2] Za pokretanje programa potrebna je biblioteka Tensorflow[3] verzije 1.0 ili novija. Program zahtijeva i neke dodatne module, te će dojaviti pogrešku prilikom pokretanja programa ako nisu instalirani. U tekstu pogreške bit će navedeno ime potrebnog modula, te se preporučuje instalacija putem pip upravljača paketima[4]. Obratiti pozornost na to da su i programski jezik Python i upravljač paketima pip dodani pod sistemske varijable (eng. *System variables*). Ako se implementacija EAST modela instalirava na računalu s operativnim sustavom Windows, upute za prevođenje ranije opisane LANMS biblioteke također su dostupne u GitHub repozitoriju[5].

## 6.4. Evaluacija na vlastitom podatkovnom skupu

Prije evaluacije potrebno je pribaviti podatkovni skup. Slike nad kojima želimo pokrenuti evaluaciju moramo smjestiti u isti direktorij. Dimenzije slika su proizvoljne, a podržani formati su jpg, png, jpeg i JPG. Datoteke svih ostalih formata će biti ignorirane, a listu podržanih formata moguće je promijeniti u datotekama *eval.py* i *icdar.py*.

U sklopu završnog rada korišten je model[6] treniran na setu za treniranje ICDAR 2013 i ICDAR 2015. Moguće je i treniranje vlastitog modela, a upute se nalaze u ranije spomenutom GitHub repozitoriju.

Osim ulaznog direktorija i lokacije predtreniranog modela, potrebno je navesti i izlazni direktorij. U tom direktoriju generirat će se slike na kojima će svaka prepoznata riječ imati plavi opisani pravokutnik, a uz sliku generirat će se i tekstualna datoteka s koordinatama vrhova opisanog pravokutnika.

[1] https://github.com/argman/EAST

[2] Python 3 nije kompatibilan sa starijim verzijama Pythona, stoga treba voditi računa o verziji Pythona s kojom pokrećemo program

[3] https://www.tensorflow.org/install/

[4] Upute za instalaciju na operacijskim sustavima Windows - http://bit.ly/pipwindows

[5] https://github.com/argman/EAST/issues/120

[6] https://drive.google.com/file/d/0B3APw5BZJ67ETHNPaU9xUkVoV0U/view

Generirane slike i tekstualne datoteke bit će istog imena kao i pripadajuće slike ulaznog direktorija, a ako nam takav format imenovanja ne odgovara možemo ga promijeniti unutar datoteke *eval.py*. U istoj datoteci moguće je i promijeniti oblik označavanja riječi na slikama, poput debljine i boje linija opisanih pravokutnika. Generirane slikovne i tekstualne datoteke bit će prikazane u sljedećem poglavlju.

Evaluacija se može pokrenuti na jednoj ili više grafičkih kartica i započinje se sljedećom naredbom unutar naredbenog retka:

```
python eval.py
```

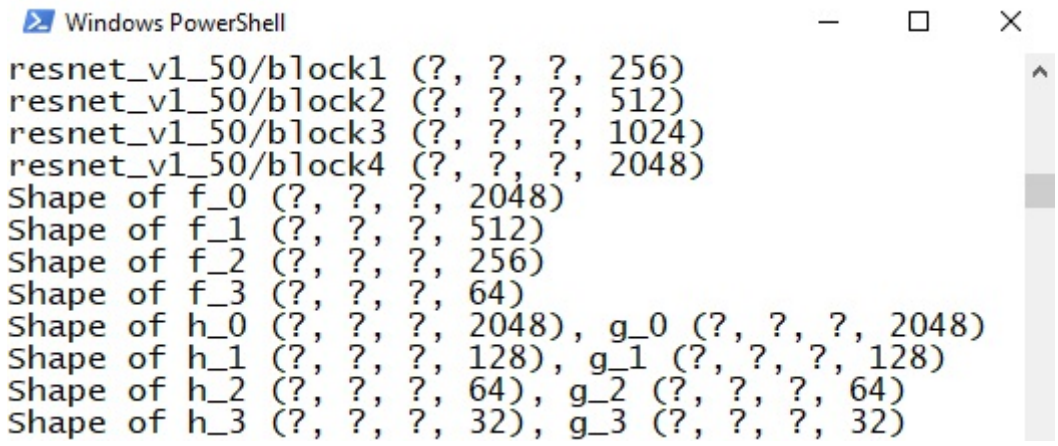
Uz parametre:

```
--test_data_path - putanja do direktorija sa slikama koje želimo evaluirati
--gpu_list - odabir na kojoj grafičkoj kartici želimo izvršiti evaluaciju
(npr. --gpu_list=0,1,2,3)
--checkpoint_path - lokacija predtreniranog modela
--output_dir  - direktorij u koji želimo pohraniti rezultate
```

Primjer naredbe unutar naredbenog retka:

```
python eval.py --test_data_path=tmp/images/ --gpu_list=0
--checkpoint_path=tmp/east_icdar2015_resnet_v1_50_rbox/
--output_dir=tmp/results
```

Nakon pokretanja evaluacije prvo će se ispisati sažet prikaz arhitekture iz kojega možemo iščitati broj značajki pojedine komponente (Slika 6.1). Pojmovi *resnet*, $f_i$, $g_i$ *i* $h_i$ odnose se na sliku 5.1 iz ranijeg poglavlja u kojem smo opisivali arhitekturu modela EAST.

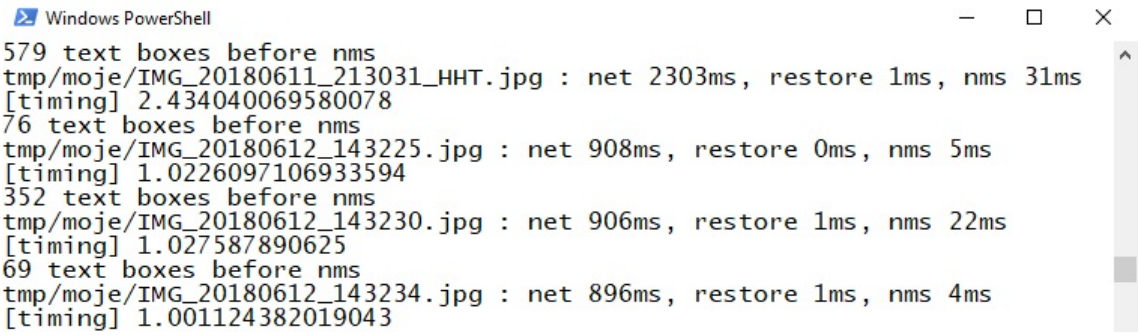



**Slika 6.1:** Ispis neposredno nakon pokretanja programa.

U nastavku će za svaku sliku biti ispisana tri retka:

1. broj generiranih opisnih okvira prije prolaska kroz LANMS algoritam.
2. Vrijeme prolaska kroz neuronsku mrežu, vrijeme potrebno za formiranje pravokutnika iz podataka dobivenih na izlazu mreže te vrijeme potrebno za provlačenje generiranih pravokutnika kroz LANMS algoritam
3. Ukupno vrijeme potrošeno na obradu slike

Kao što je prikazano na slici 6.2.

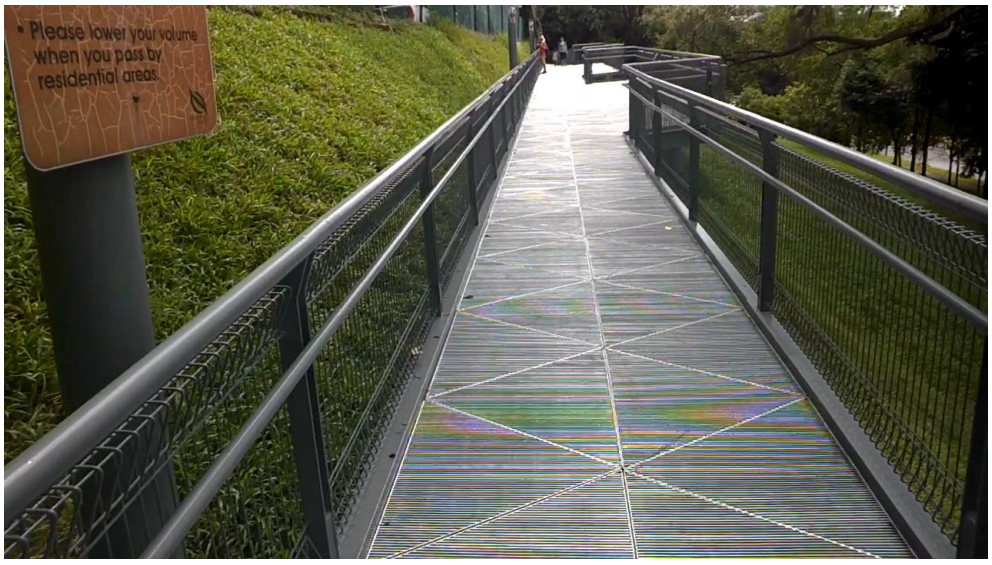



**Slika 6.2:** U nastavku ispisa prikazanog na slici 6.1 ispisat će se detalji za svaku pojedinu sliku iz ulaznog direktorija.

## 6.5. Evaluiranje na javno dostupnom skupu

Robust Reading Competition[7] je natjecanje u robusnom čitanju koje se u razdoblju od 2003. do 2015. godine održavalo pet puta, a od 2017. godine nastavlja se u promijenjenom obliku pod imenom ICDAR 2017 Robust Reading Competition. Robusno čitanje (eng. *Robust reading*) se odnosi na tumačenje pisane komunikacije u nepovoljnim okruženjima, a najčešće je riječ o pronalaženju i prepoznavanju teksta u sceni.

Skup slika koji se koristio na natjecanju ICDAR Incidental Scene Text 2015[8] sastoji se od 1000 slika za treniranje, 500 slika za testiranje te 170 slika za rangiranje najboljih natjecatelja. Na fotografijama su prikazane slučajne scene s tekstom, odnosno scene koje sadrže tekst čija se pozicija i kvaliteta nije nastojala popraviti prije fotografiranja. Spomenuti skup za testiranje koristio se za evaluaciju modela EAST u sklopu ovog rada.




**Slika 6.3:** Primjer fotografije iz skupa za testiranje.

Uz svaku fotografiju unutar skupa za testiranje dana je i tekstualna datoteka s očekivanim rezultatima (eng. *ground-truth*). Svakom tekstu na fotografiji odgovara jedan redak unutar tekstualne datoteke u kojem se nalazi 8 koordinata koje su međusobno razdvojene zarezom u obliku "x1,y1,x2,y2,x3,y3,x4,y4,tekst". Tih 8 koordinata predstavlja vrhove četverokuta unutar kojega se nalazi tekst, a zadnji element predstavlja riječ koju je potrebno prepoznati. Ovaj rad se fokusira

[7] http://rrc.cvc.uab.es/

[8] http://rrc.cvc.uab.es/?ch=4&com=tasks

na pronalaženje teksta, stoga zadnji član zanemarujemo. Riječi s jednim ili dva znaka, kao i riječi koje se smatraju nečitljivima, vode se kao *Do Not Care* te su u tekstualnoj datoteci označene sa znakovima ### umjesto očekivane riječi. Riječi označene kao *Do Not Care* ne uzimaju se u obzir prvi evaluaciji. Primjer jedne takve tekstualne datoteke dan je u nastavku.

Datoteka

```
27,17,103,22,106,47,30,45,Please
107,20,159,26,159,48,109,47,lower
161,26,198,27,199,51,163,51,your
201,28,251,31,251,48,201,46,volume
35,52,97,51,100,76,39,79,when
101,55,140,53,143,80,103,81,you
141,55,181,53,183,77,144,79,pass
182,51,205,52,205,76,185,77,###
41,83,148,77,151,103,45,113,residential
152,82,198,80,199,99,153,101,areas
```

Za svaku fotografiju iz skupa za testiranje potrebno je generirati rezultat u obliku tekstualne datoteke u ranije navedenom obliku, s razlikom da zadnji element ne ispisujemo. Primjer datoteke koja sadrži rezultate dan je u nastavku.

Datoteka

```
38,87,145,76,148,102,41,113
26,17,105,19,105,46,26,45
33,55,97,52,97,76,34,79
201,29,253,28,254,47,201,48
140,54,182,51,184,73,142,76
150,84,198,79,199,98,152,103
97,55,139,53,140,75,98,77
108,23,159,23,159,47,108,47
162,28,198,26,199,47,162,48
c@FancyVerbLinee9,47,162,48
```

Usporedbom ranije navedene dvije tekstualne datoteke možemo primijetiti da redoslijed kojim su tekstovi prepoznati nije bitan, te da koordinate četverokuta nisu identične. Naime, rezultat će biti priznat kao točan ukoliko metodom "Intersection-over-Union", odnosno omjerom presjeka površina dvaju četverokuta i njihove unije, dobijemo kao rezultat broj jednak ili veći od 0.5. Također možemo primijetiti da model EAST u ovom slučaju nije prepoznao tekst označen kao *Do Not Care*, kao što je i očekivano.

Kako bismo evaluirali model EAST na skupu za testiranje Robust Reading Competitiona, potrebno je preuzeti skup slika s web stranica natjecanja[9]. Preuzeti skup slika potrebno je raspakirati, te ćemo direktorij koji sadrži slike koristiti kao ulazni direktorij u postupku evaluacije koji je opisan u prethodnom potpoglavlju. Robust Reading Competition za svaku sliku očekuje rezultat u obliku tekstualne datoteke imena res_imeslike.txt. Potrebno je u datoteci *eval.py* izmijeniti format imena tekstualnih datoteka, a moguće je i zakomentirati dio koda u kojem se generiraju slike s opisanim okvirima.

```
172         # save to file
173         if boxes is not None:
174             res_file = os.path.join(
175                 FLAGS.output_dir,
176                 'res_{}.txt'.format(
177                     os.path.basename(im_fn).split('.')[0]))
```

Zatim je potrebno pokrenuti evaluaciju i dobivene rezultantne datoteke arhivirati u *.zip* datoteku i istu učitati na stranicama natjecanja[10]. Evaluaciju je moguće napraviti i *offline*, a potrebne skripte nalaze se na istoj stranici.
Nakon uspješnog podnošenja rezultata na stranicu natjecanja, moguće je vidjeti vlastitu preciznost, odziv i F1-mjeru na rang listi svih podnositelja. Također, moguće je vidjeti pojedinačne rezultate za svaku sliku u obliku slike, IoU matrice ili preciznosti, odziva i F1-mjere.

[9] http://rrc.cvc.uab.es/?ch=4&com=downloads
[10] http://rrc.cvc.uab.es/?ch=4&com=mymethods&task=1

# 7. Eksperimentalni rezultati

U tablici 7.1 prikazani su rezultati[1] dobiveni 2015. godine na natjecanju čiji je skup korišten i za evaluiranje u sklopu ovog rada. Nakon horizontalne linije prikazan je rezultat koji je ostvaren u sklopu ovog rada, a nakon njega trenutno vodeća metoda.

**Tablica 7.1:** Usporedba rezultata.

| Datum | Metoda | Preciznost | Odziv | F1-mjera |
|---|---|---|---|---|
| 2015 | Stradvision-2 | **77.46** | 36.74 | **49.84** |
| 2015 | Stradvision-1 | 53.39 | 46.27 | 49.57 |
| 2015 | NJU | 70.44 | 36.25 | 47.87 |
| 2015 | AJOU | 47.26 | **46.94** | 47.1 |
| 2015 | HUST-MCLAB | 44.0 | 37.79 | 40.66 |
| 2015 | Deep2Text-MO | 49.59 | 32.11 | 38.98 |
| 2015 | CNN MSER | 34.71 | 34.42 | 34.57 |
| 2015 | TextCatcher-2 | 24.91 | 34.81 | 29.04 |
| 31.07.2017[2] | EAST | 84.66 | 77.32 | 80.83 |
| 31.01.2018 | Alibaba-PAI | 93.84 | 87.34 | 90.47 |

Rezultati su iskazani trima vrijednostima. Preciznost predstavlja omjer točnih pozitivnih detekcija i zbroja točnih pozitivnih i lažnih pozitivnih detekcija. Točna pozitivna detekcija predstavlja dio slike koji je označen kao riječ i na kojem se uistinu nalazi riječ, dok lažna pozitivna detekcija predstavlja dio slike na kojem se ne nalazi niti jedna riječ, a algoritam ga je označio kao riječ.

[1] http://rrc.cvc.uab.es/files/short_rrc_2015.pdf, Table III

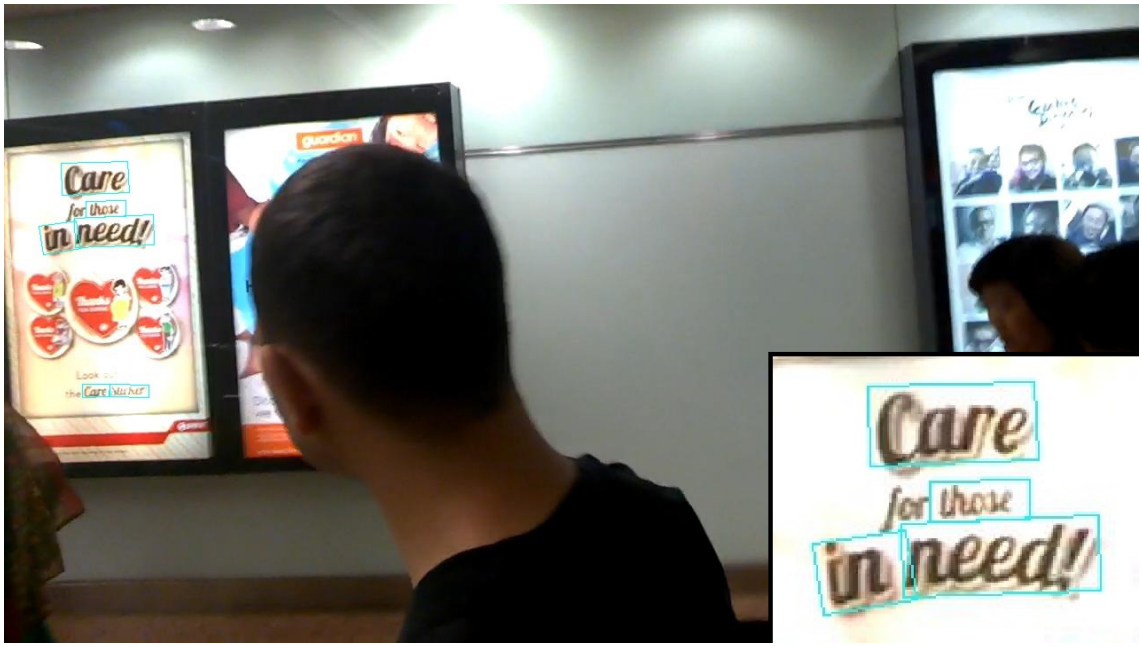
[2] EAST detektor teksta izvorno je iz 2017. godine, stoga je taj datum naveden u tablici (umjesto datuma evaluacije u sklopu završnog rada) kako bi bolje prikazali napredak od 2015. do danas

Na primjer, do pada preciznosti doći će kada algoritam prepozna znakove ili neke druge čudne oblike kao riječi. Odziv predstavlja omjer točnih pozitivnih detekcija i zbroja točnih pozitivnih i lažnih negativnih detekcija. Lažna negativna detekcija predstavlja dio slike na kojem se nalazi neka riječ, a algoritam ju nije prepoznao i označio. Drugim riječima, odziv pada kada algoritam ne prepozna određene riječi. Često se događa da algoritam riječ "prepolovi" i prepozna ju kao dvije, ili pak dvije riječi prepozna kao jednu, što dovodi do pada i preciznosti i odziva. F1-mjera predstavlja harmonijsku sredinu preciznosti i odziva.
Analizom rezultata možemo primijetiti značajan napredak u proteklih nekoliko godina, no trenutno najbolja preciznost od 93.84% još uvijek je daleko od savršene, te možemo očekivati napredak i u narednim godinama.
U nastavku su prikazane neke od slika s označenim riječima koje su dobivene pokretanjem implementacije EAST detektora teksta.


**Slika 7.1:** Očekivani rezultat.

Na slici 7.1 prikazan su svi opisani okviri koji su navedeni unutar datoteke s očekivanim rezultatima. Riječi koji se trebaju prepoznati označene su zelenim četverokutima, a *Do Not Care* riječi označene su sivim četverokutima. Na slici 7.2 prikazan je rezultat dobiven EAST detektorom teksta gdje sve prepoznate riječi označene su plavim opisanim okvirima.


**Slika 7.2:** Dobiveni rezultat.

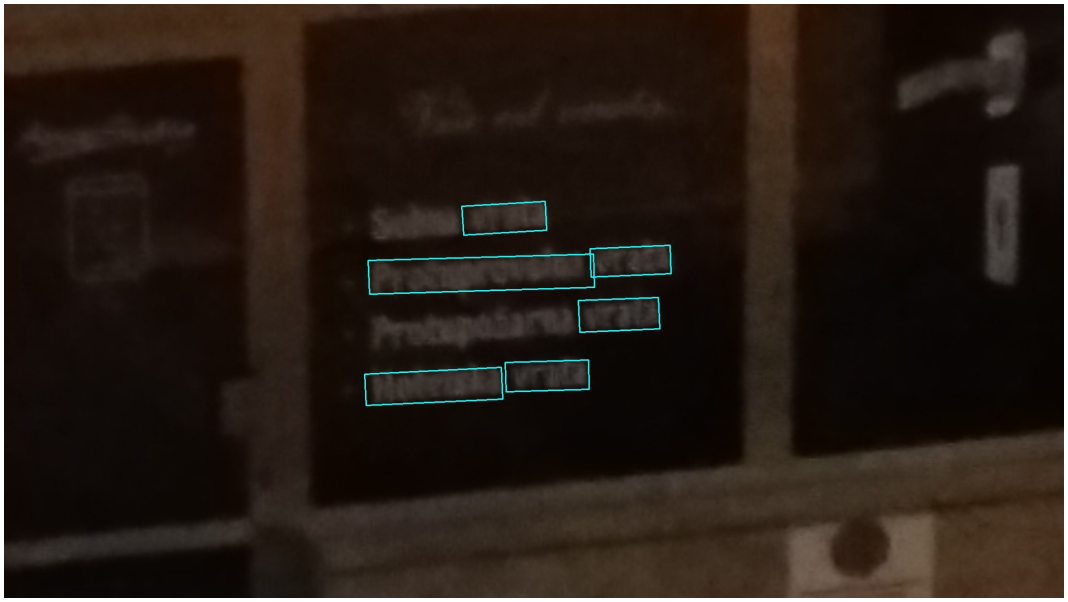



**Slika 7.3:** Primjer pogrešnog pronalaženja riječi - riječ "for" nije detektirana.

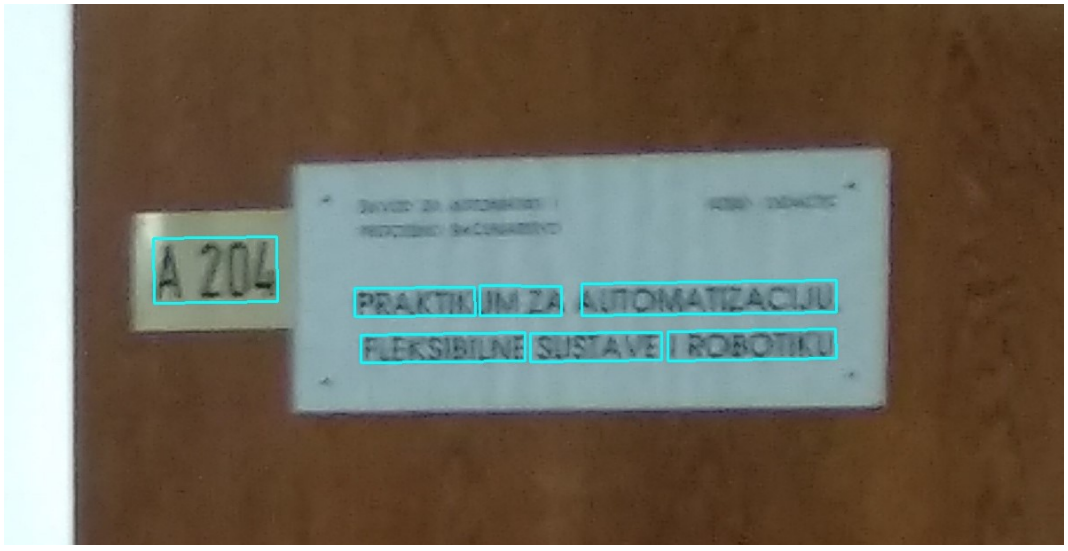
Na slikama 7.3 i 7.4 prikazane su tipične pogreške koje nastaju pri pronalaženju teksta. Na prvoj slici možemo vidjeti kako riječ "for" nije detektirana, dok je na drugoj slici logo marke autobusa detektiran kao dio riječi.


**Slika 7.4:** Primjer pogrešnog pronalaženja riječi - logo marke autobusa detektiran kao dio riječi.

U nastavku ćemo komentirati rezultate dobivene na vlastitom skupu za testiranje. Na slici 7.5 vidimo kako se model EAST ponaša kada na ulaz dobije fotografiju uslikanu u noćnim uvjetima s mutnim tekstom. Na slikama 7.7 i 7.8 vidimo da se model zadovoljavajuće ponaša i kada je tekst na slici pod određenim kutem, no rotacija za 23 stupnja i ponovna evaluacija nad "uspravnim" tekstom ipak daje bolje rezultate. Ako uzmemo u obzir da bi većina mutnih ili sitnih riječi na natjecanju bila označena kao *Do Not Care*, dobiveni rezultati su veoma dobri.

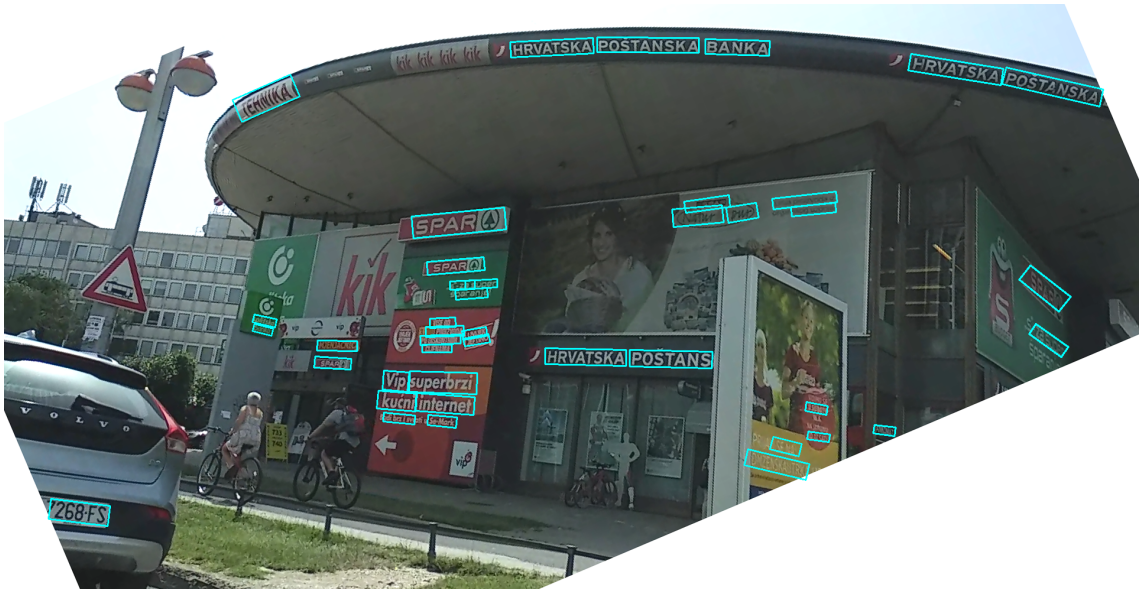


**Slika 7.5:** Pronalaženje riječi - noć, mutan tekst.




**Slika 7.6:** Pogrešno pronalaženje riječi - "PRAKTIK UMZA".


**Slika 7.7:** Pronalaženje riječi - solidni rezultati.




**Slika 7.8:** Pronalaženje riječi - ako sliku 7.7 rotiramo prije evaluacije, dobivamo bolje rezultate.

# 8. Zaključak

Razvoj dubokog učenja doveo je do velikog napretka na području računalnog vida. Zbog sposobnosti samostalnog učenja i mogućnosti rada s nejasnim i manjkavim podacima, neuronske mreže često se koriste u računalnom vidu, pa tako i za pronalaženje teksta na slikama. U ovom radu opisane su neuronske mreže, navedena su osnovna svojstva konvolucijskih neuronskih mreža i njihove prednosti. Uspoređene su performanse tradicionalnih i rezidualnih neuronskih mreža.
Glavni dio rada činila je analiza modela za pronalaženje teksta EAST. Model EAST efikasno koristi rezidualne mreže i prihvaća ideju o postupnom zbrajanju značajki, te pokazuje da jednostavna arhitektura može proizvesti veoma dobre rezultate. Opisan je NMS algoritam i njegova izmijenjena verzija LANMS koja se koristila u sklopu modela EAST.
Ukratko je opisan dio natjecanja Robust Reading Competition vezan za pronalaženje teksta, te je evaluiran model EAST na skupu za testiranje ICDAR Incidental Scene Text 2015. Dobiveni rezultati uspoređeni su s rezultatima iz 2015. godine te s trenutno najboljim modelom. Pokazalo se da model EAST ima problema s detekcijom kratkih riječi, tekstova malih dimenzija i tekstova pod većim kutem, te da ponekad elemente u neposrednoj blizini teksta označava kao dio teksta. Ipak, metode za pronalaženje teksta su sve bolje, te se u ovom području očekuje daljnji napredak.

# Literatura


[1] *ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks.* URL http://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/.

[2] *Stranice natjecanja "Robust Reading Competition".* URL http://rrc.cvc.uab.es.

[3] *Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition*, 2018. URL http://cs231n.github.io/neural-networks-1/.

[4] Bojana Dalbelo Bašić, Marko Čupić, i Jan Šnajder. *Umjetne neuronske mreže, prezentacija s predmeta Umjetna inteligencija na Fakultetu elektrotehnike i računarstva*, 2017. URL http://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf.

[5] Mohit Deshpande. *Introduction to Convolutional Neural Networks for Vision Tasks*, 2017. URL https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/.

[6] Siniša Šegvić. *Konvolucijski modeli, prezentacija s predmeta Duboko učenje na Fakultetu elektrotehnike i računarstva.* URL http://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. *Deep Residual Learning for Image Recognition*, 2015. URL https://arxiv.org/abs/1512.03385.

[8] Joyce Liu i Wang Xiqing. In your face: China's all-seeing state. URL http://www.bbc.com/news/av/world-asia-china-42248056/in-your-face-china-s-all-seeing-state.

[9] Olaf Ronneberger, Philipp Fischer, i Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015. URL https://arxiv.org/abs/1505.04597.

[10] David Silver. *Literature Review: Fully Convolutional Networks*, 2017. URL https://medium.com/self-driving-cars/literature-review-fully-convolutional-networks-d0a11fe0a7aa.

[11] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. URL http://szeliski.org/Book/.

[12] Matthew D. Zeiler i Rob Fergus. *Visualizing and Understanding Convolutional Networks*, 2013. URL https://arxiv.org/abs/1311.2901.

[13] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, i X. Bai. *Multi-oriented text detection with fully convolutional networks*, 2015. URL http://openaccess.thecvf.com/content_cvpr_2016/papers/Zhang_Multi-Oriented_Text_Detection_CVPR_2016_paper.pdf.

[14] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, i Jiajun Liang. *EAST: An Efficient and Accurate Scene Text Detector*, 2017. URL https://arxiv.org/abs/1704.03155.

**Pronalaženje teksta dubokim konvolucijskim modelima**

## Sažetak

Korištenje dubokog učenja u metodama pronalaženja teksta, ali i u mnogim drugim problemima računalnog vida, dovelo je do značajnog napretka. Rezidualne mreže imaju puno bolje performanse od tradicionalnih neuronskih mreža pri većim dubinama te su se pokazale kao veoma korisne u svrhu pronalaženja teksta. Promatranjem rezultata natjecanja poput Robust Reading Competitiona vide se raznoliki pristupi pri rješavanju ovoga problema i napredak koji se ostvaruje kroz godine. Model za pronalaženje teksta EAST ima naglasak na jednostavnosti arhitekture i brzini izvođenja, te postiže veoma dobre rezultate.

**Ključne riječi:** računalni vid, duboko učenje, neuronske mreže, pronalaženje teksta, east

**Text Localization With Deep Convolutional Models**


## Abstract

The use of deep learning in text localization methods, but also in many other computer vision problems, has brought significant progress to the field. Residual networks perform much better than traditional neural networks at higher depths, and have proved to be very useful for text localization. By observing the results of competitions like Robust Reading Competition, we can see many different approaches to addressing this problem, as well as the progress that has been made over the years. A text-localization model EAST has an emphasis on performance and simple architecture, and achieves very good results.



**Keywords:** computer vision, deep learning, neural networks, text localization, east