Pronalaženje teksta u prirodnim slikama

Pavić, Marko

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1656

PRONALAŽENJE TEKSTA U PRIRODNIM SLIKAMA

Marko Pavić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1656

PRONALAŽENJE TEKSTA U PRIRODNIM SLIKAMA

Marko Pavić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1656

Pristupnik:	**Marko Pavić (0036543775)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Pronalaženje teksta u prirodnim slikama**

Opis zadatka:

Pronalaženje teksta zanimljiv je problem računalnog vida s mnogim uzbudljivim primjenama. Zbog zakrivljenih napisa ovaj problem nije jednakovrijedan standardnom pronalaženju objekata. Ovaj rad istražit će primjenjivost koncepta na različitim zadatcima. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće duboke arhitekture utemeljene na konvolucijama i pažnji. Vrednovati generalizacijsku moć postupaka iz literature na javno dostupnim i privatnim slikama. Procijeniti složenost učenja modela. Prikazati i ocijeniti provedene eksperimente. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1656

PRONALAŽENJE TEKSTA U PRIRODNIM SLIKAMA

Marko Pavić

Zagreb, rujan 2024.

Zahvaljujem se mentoru prof. dr. sc. Siniši Šegviću na pomoći pri izradi rada i svojoj obitelji i prijateljima na podršci

Sadržaj

Uvod

Ljudi već godinama koriste razne tehnologije za prepoznavanje teksta u slikama, počevši s optičkim raspoznavanjem znakova (eng. Optical Character Recognition). Prvim OCR strojem može se smatrati uređaj za čitanje za slijepe osobe kojeg je napravio Emanuel Goldberg 1912.-te godine. Taj uređaj čitao bi slovo po slovo i pretvarao bi taj tekst u telegrafski kod [1]. Danas, preko sto godina kasnije, za takve probleme koristi se umjetna inteligencija.

Umjetna inteligencija je svakim danom sve popularnija tema, a i sve korištenija tehnologija. Od zdravstva, ekonomije i autonomnih vozila, pa do društvenih mreža i područja javne uporabe kao što je chatGPT, teško je ne doći u kontakt s tim područjem računarstva.

Pronalaženje teksta u slikama jedna je od mnogih točaka interesa u području umjetne inteligencije, točnije, računalnog vida. Računalni vid je termin koji predstavlja dovoljno razumijevanje slika ili videa od strane računala, da bi to računalo moglo iz slike ili videa „izvaditi“ neke informacije i na temelju tih informacija napraviti neku odluku. Te odluke mogu predstavljati npr. kategoriziranje slika ili, u kontekstu pronalaženja teksta, lokalizacija objekata na slikama.

Jedna od arhitektura koja se bavi ovakvim problemom je CharNet, potpuno konvolucijski jednofazni model koji je spojio probleme detekcije i prepoznavanja teksta u slikama te ta dva problema rješava zajedno u jednom koraku [2].

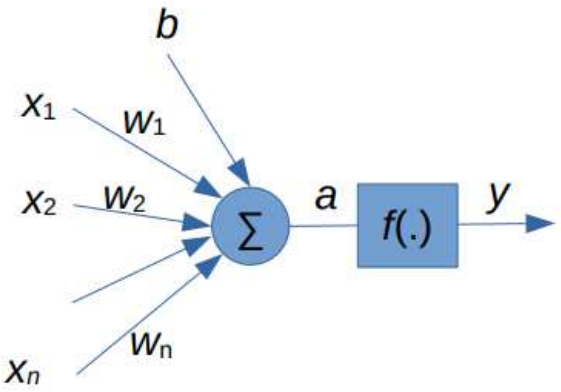
Ovaj rad predstavit će građevne elemente dubokih modela/neuronskih mreža kao što su konvolucija i pažnja. Detaljno ćemo objasniti model CharNet te će se razmotriti njegova generalizacijska moć na javno dostupnim i privatnim slikama. Svi provedeni eksperimenti bit će prikazani, objašnjeni i ocijenjeni.

1. Duboko učenje

Duboko učenje grana je umjetne inteligencije koja se zasniva na izravnoj optimizaciji svih parametara kompozitnog modela. Kompozitni modeli sastoje se od više slojeva koji modeliraju jednostavne transformacije. Najjednostavniji sloj odgovara afinoj vektorskoj transformaciji s nelinearnom aktivacijom funkcijom prema sljedećoj jednadžbi:

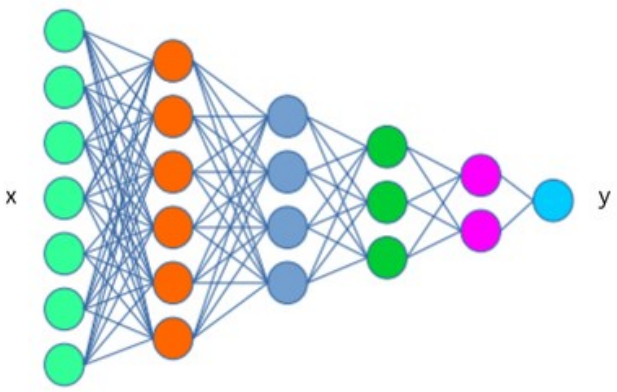
$$y = f(a) = f(\sum_{i=1}^{n} w[i] \times x[i] + b) \tag{1}$$

Ovakav sloj odgovara jednostavnom modelu umjetnog neurona prema sljedećoj slici:



Slika 1: Umjetni neuron [3].

Svaki neuron kao ulaz prima niz ulaznih varijabli (x), težine (w) i pomak (b) te kao izlaz daje informaciju y, koja je ili izlazna informacija iz mreže, ili informacija koja se šalje u sljedeći sloj mreže, tj kao ulaz u drugi neuron. Ilustrativan primjer jedne višeslojne potpuno povezane neuronske mreže dan je na slici Slika 2.



Slika 2: Primjer višeslojne potpuno povezane neuronske mreže [3].

Cilj učenja duboke neuronske mreže je optimizacija parametara w i b, kako bi oni davali što bolje rezultate. Za optimizaciju mreža najčešće se koriste tri skupa podataka:

- Skup za učenje – najčešće sadrži najveću količinu podataka i njega primarno koristi model za optimizaciju parametara.
- Skup za provjeru – služi kako bi se spriječila prenaučenost (kada greška krene rasti na ovom skupu, učenje se zaustavlja)
- Skup za testiranje – koristi se pri konačnoj provjeri mreže kako bi se ona ocijenila.

1.1. Pronalaženje teksta

Pronalaženje teksta je u osnovi podskup jednog drugog, opširnijeg problema, pronalaženja objekata. Prije je jedan od osnovnih koraka kod prepoznavanja oblika bio detekcija rubova. To je danas zastarjeli pristup i u praksi se koriste tehnike detekcije kompleksnijih značajki.

Ako se slika rastavi na niz piksela te se taj niz koristi kao ulazni niz, može se primijetiti da je međuovisnost piksela inherentno lokalna. Sama bitna informacija u slici je razlika između lokalnih piksela. Kako bi se prepoznao tekst na slici trebalo bi prepoznati neke karakteristične detalje teksta ili karakterističan oblik.

Osim samog ruba, granicu objekta također opisuju i karakteristike regija s obje strane te granice (tzv. lokalna susjedstva, odnosno razlike u lokalnim susjedstvima). Stoga je osim samog ruba, bitno i promotriti unutrašnjost i okolinu objekta za informacije [3].

Slojevi dubokih modela koji se mogu koristiti pri pronalaženju teksta su pažnja i konvolucija. Pažnja se više koristi pri radu sa sekvencijalnim podacima, dok se konvolucija najčešće koristi kada su ulazni podaci slike.

1.2. Pažnja

Sloj pažnje u strojnom učenju se temelji na miješanju elemenata skupa prema njihovoj sličnosti. Iz tog razloga, ta metoda se najčešće koristi nad skupovima. Pažnju možemo koristiti i nad nizovima, ali tada je položaj elementa potrebno zadati eksplicitno dodavanjem pozicijskog ugrađivanja (eng. positional embedding). Arhitekture koje se time bave nazivaju se transformatori [4].

Uzmimo za primjer model obrade prirodnog jezika koji koristi metodu pažnje. U takvom modelu dodjeljuje se različita razina važnosti pojedinim riječima u rečenici ovisno o njihovoj poziciji u rečenici i tzv. „kontekstnom prozoru“ [5].

Pažnja mijenja vektorske reprezentacije riječi u rečenici tako da nakon svakog sloja te reprezentacije dobivaju sve fokusiranije značenje. Promotrimo rečenicu: „Josip je otišao u dućan u susjednom gradu kako bi sreo prijatelja“. Moguć je slučaj da nakon prvog sloja, transformerski model na temelju sintagme "u susjednom gradu" može zaključiti da se imenica grad odnosi na naselje, a ne na meteorološku pojavu – tuču [4].

U kontekstu računalnog vida, koriste se transformatori vida (eng. Vision transformer, ViT) koji podijele ulaznu sliku u niz okana te nad njima radi operacije kao običan transformator nad tokenima. Naime, položaj okana u originalnim slikama zadaje se 1D pozicijskim ugrađivanjem, ali moguće je i 2D pozicijsko ugrađivanje. ViT se danas koristi u prepoznavanju slike, segmentaciji slike i autonomnoj vožnji [6].

1.3. Konvolucija

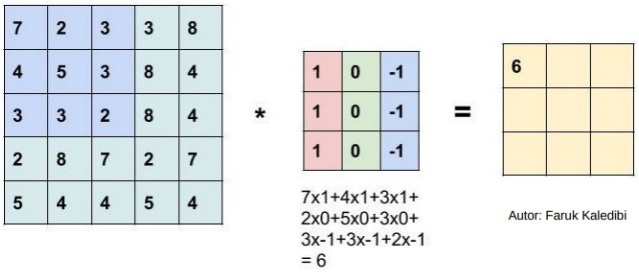
Uz pažnju, u mrežama za pronalaženje teksta prevladavajuće se koristiti konvolucija: matematička operacija nad dvije funkcije čiji je rezultat nova funkcija. U slučaju gdje se radi sa slikama koristi se 2D konvolucija gdje slike i dijelove slika možemo predstaviti kao dvodimenzionalna polja (izlazna funkcija je tada također dvodimenzionalna).

Ako se u slici I želi naći slika objekta G može se koristiti funkcija prema izrazu (2). Slike i dijelovi slika su predstavljeni funkcijom I(x,y), a ako je izgled objekta koji se traži poznat, onda konstruiranu masku predstavlja funkcija G(x,y). Dovoljno velika vrijednost u odzivu indicira lokalizaciju objekta.

$$(I * G)(x, y) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} I(x - m, y - n)G(m, n) \tag{2}$$

Maske (filtri) se mogu koristiti za pronalaženje rubova ili nekih kompliciranijih značajki na slikama (npr. kutevi). Filtri su najčešće manji od same slike pa prolaze kroz svaki dio te slike (kroz širinu/visinu) te računa matrični produkt kao što je prikazano na slici Slika 3, a kao izlaz se stvara mapa značajki [3][4].

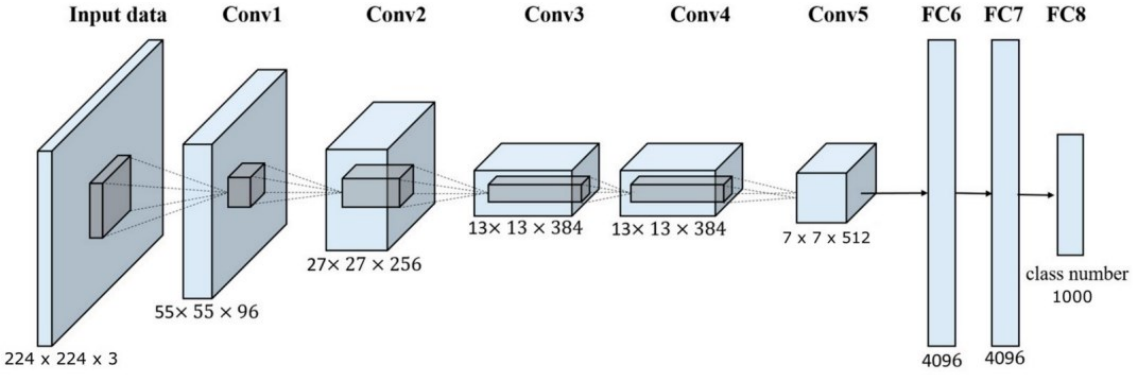
Bitno je napomenuti da moderni pristupi za lokalizaciju ne detektiraju rubove.



Slika 3. Primjer detekcije vertikalnih (lijevih) rubova [3].

1.4. Konvolucijske neuronske mreže

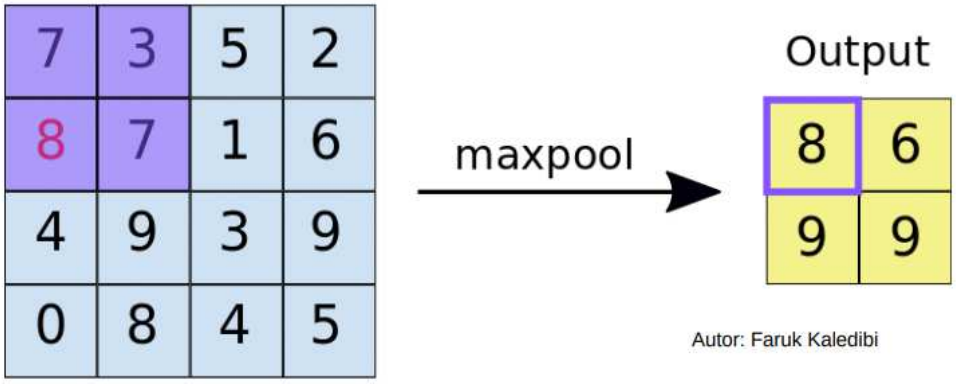
Više konvolucijskih filtra stvaraju konvolucijski sloj, koji u idući sloj šalje tenzor mapa značajki njegovih filtra. Neuronske mreže koje imaju barem jedan konvolucijski sloj nazivaju se konvolucijskim mrežama. Primjer moguće arhitekture konvolucijske mreže prikazan je na slici Slika 4.



Slika 4. Primjer klasifikacijske konvolucijske mreže [3].

Osim konvolucijskog sloja, konvolucijske mreže mogu sadržavati i:

- Sloj sažimanja – služi za smanjivanje dimenzija ulaznih podataka združivanjem prostorno bliskih značajki (npr. average pooling, max pooling prikazan na slici Slika 5, ROI pooling). Združivanjem lokalnih informacija smanjuje prostorne rezolucije, a korisno je i za postizanje invarijantnosti na geometrijske transformacije kao što su pomaci.



Slika 5. Primjer sažimanja maksimumom [3].

- Potpuno povezani sloj – svaki neuron iz prošlog sloja povezan je sa svakim neuronom iz sljedećeg. Najčešće se koristi na kraju mreže gdje se određuje kojoj klasi pripada određeni ulaz u mrežu (ako se radi o klasifikacijskom problemu).

Ako konvolucijska mreža nema potpuno povezani sloj, onda se radi o potpuno konvolucijskoj mreži. Takve mreže se najčešće koriste za semantičku segmentaciju slika ili segmentaciju instanci (kao što je problem pronalaženja teksta na slikama) [3][4][7][8].

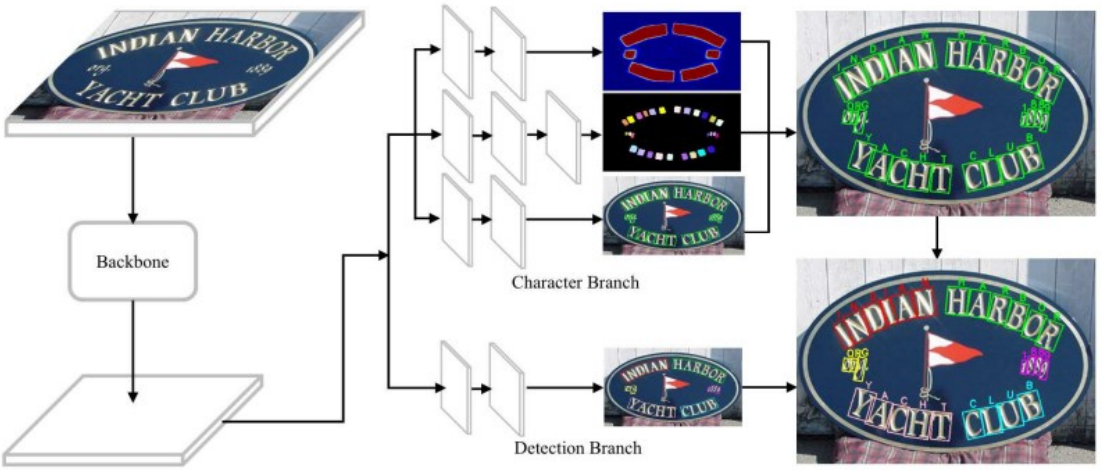
2. CharNet

Detekcija teksta u slikama i prepoznavanje teksta u slikama često se rješavaju odvojenim modelima, a ako je spojen, onda su ti procesi spojeni u dvofaznom okviru. Xing et al [2] su primijetili da takvi modeli često koriste ROI pooling koji zna degradirati generalizacijsku moć modela, ili koriste povratne neuronske mreže kod grana za prepoznavanje teksta [16][17]. Te mreže je teško optimizirati zajedno s granom za detekciju teksta jer zahtjevaju iznimno veću količinu podataka za treniranje [2].

Iz tog razloga, odlučili su razviti potpuno konvolucijski model zvan „convolutional character networks“ tj. CharNet [2] koji detekciju i prepoznavanje teksta radi u jednom koraku, istovremeno. Nakon razvoja, taj model je uspio nadmašiti tada najsuvremenije pristupe.

Mnogo modela za prepoznavanje teksta na slikama kao detekcijsku jedinicu koriste riječi. Oni se zapravo bave problemom označavanja sekvenci. U nekim jezicima riječi ne mogu biti jednako definirane kao u hrvatskom ili engleskom jeziku. Tako na primjer u kineskom jeziku, lakše je instance teksta definirati znakovima, umjesto riječima. Ovaj model stoga umjesto riječi, koristi znakove kao detekcijske jedinice. To znači da se problem označavanja sekvenci pretvara u problem detekcije objekata pa se može umjesto povratnih neuronskih mreža koristiti konvolucijska. Taj pristup također pomaže i kod detekcije zakrivljenog i višestruko orijentiranog teksta.

Model se sastoji od dvije paralelne grane: znakovna grana i grana detekcije teksta; gdje je znakovna grana zadužena za detekciju znakova u slici, a grana detekcije teksta služi za detekciju riječi. Arhitektura modela prikazana je na slici Slika 6. Mreže ResNet-50 i Hourglass (Hourglass-88 i Hourglass-57) korištene su kao okosnice (eng. backbone networks) [2].

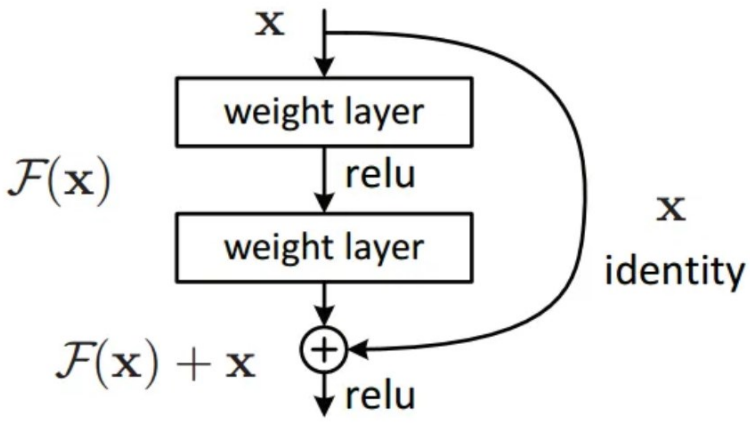


Slika 6. Arhitektura modela CharNet [2].

2.1. Okosnice (ResNet50 i Hourglass)

Rezidualne neuronske mreže (ResNet) su izmišljene kako bi riješile jedan specifičan problem. Što je neuronska mreža dublja njena preciznost pada, a gubitak raste. Rezidualne mreže smanjuju ovaj problem korištenjem rezidualnog bloka prikazanog na slici Slika 7. Njime se na izlazu iz određenog broja slojeva prije aktivacijske funkcije zbroji ono što je naučeno prije tih slojeva. Ta rezidualna povezanost pojednostavnjuje optimizaciju i omogućava učenje vrlo dubokih modela.

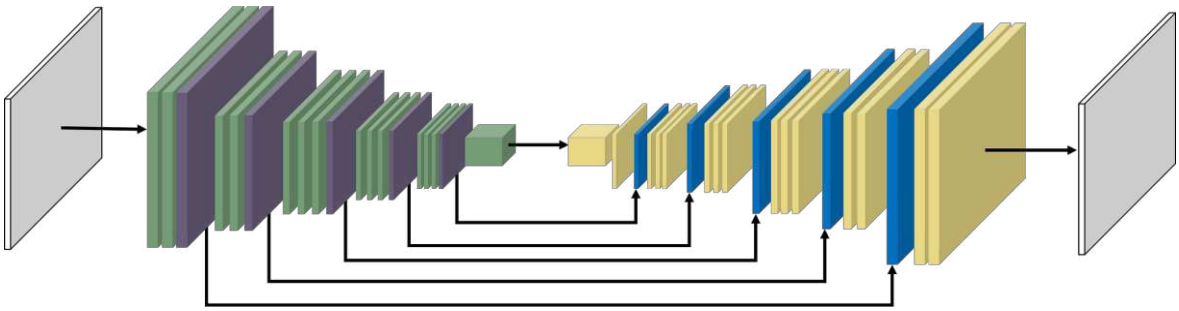
ResNet-50 je verzija rezidualne mreže s 50 slojeva i vrlo je dobra u klasifikacija slika i identificiranju objekata i scena u njima. Rezultat ove mreže je visoko rezolucijska mapa značaji koju CharNet koristi za identificiranje ekstremno malih instanci teksta [14].



Slika 7. Primjer rezidualnog bloka [15].

Hourglass arhitektura je potpuno konvolucijska mreža koja dobiva svoje ime jer njena arhitektura prikazana na slici Slika 8 izgledom podsjeća na pješčani sat. Nizom konvolucija i sažimanja smanjuje se dimenzionalnost ulazne slike, a onda se također konvolucijama, ali i naduzorkovanjem vraća slika na originalnu rezoluciju. Umjesto detekcije objekata, ovaj model služi za detekciju ključnih točaka na slikama [12].

Za CharNet, koristila su se dva Hourglass modula naredana jedan nakon drugog. Broj pored modula (npr. Hourglass-57, Hourglass-88) predstavlja dubinu.



Slika 8. Arhitektura Hourglass modela [13].

2.2. Znakovna grana

Ova grana služi za detekciju i prepoznavanje pojedinačnih znakova te postavljanje okvira i oznake pored svakog znaka. Za to koristi tri podgrane: segmentacija instanci teksta, detekcija znakova i prepoznavanje znakova. Podjela tih podgrana slikovno je prikazana na slici Slika 6.

Podgrana za segmentaciju instanci teksta sadrži tri konvolucijska sloja s filtrima veličina: 3 x 3, 3 x 3, 1 x 1. Izlaz je mapa značajki s dvije vrijednosti ovisno radi li se o tekstu u toj lokaciji slike ili ne.

Podgrana za detekciju znakova procjenjuje okvire znakova na slici koristeći pet vrijednosti: udaljenost od vrha, dna, lijeve strane i desne strane slike te orijentacijski parametar (stvara mapu značajki s pet kanala). Također sadrži tri konvolucijska sloja s filtrima iste veličine kao i kod podgrane za segmentaciju instanci teksta.

Podgrana za prepoznavanje znakova ima četiri slojeva (jedan 3 x 3 filtar više od druge dvije podgrane). Mapa značajki ima 68 kanala, jedan za svaki znak (26 engleskih znakova, 10 brojeva i 32 specijalna simbola).

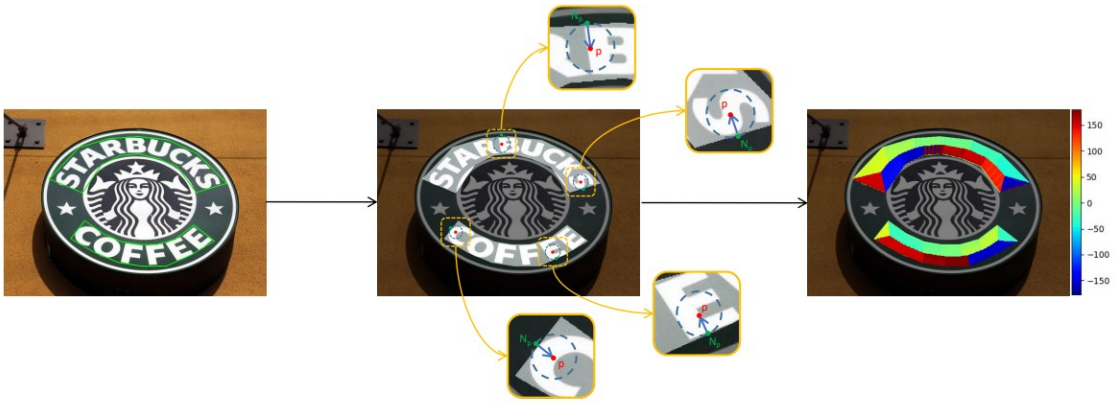
Generirani okviri zadržavaju se s vrijednosti povjerenja od 0.95 [2].

2.3. Grana detekcije teksta

Grupiranje znakova u riječi ovisno o njihovoj lokaciji može biti kompliciranо u slučajevima gdje su mnogo instanci teksta locirani međusobno blizu. CharNet koristi postojeće detektore teksta koje su se mogli primijeniti s vrlo malo modifikacija, ovisno o tipu instance teksta.

Za višestruko orijentirani tekst koristi se modificirani model EAST (An Efficient and Accuracy Scene Text Detector). To je detektor koji radi segmentaciju instanci teksta te regresiju okvira koristeći IoU (intersection over union) gubitak. U arhitekturi se koriste dva 3 x 3 konvolucijska sloja te jedan 1 x 1 konvolucijski sloj [18][19].

Zakrivljeni tekst se detektira koristeći modificirani TextField model koji također koristi dva 3 x 3 konvolucijska sloja i jedan 1 x 1 konvolucijski sloj. TextField je dizajniran specifično za detekciju iregularnih oblika i orijentacija instanci teksta. Za to koristi tzv. orijentacijsko polje (eng. direction field). Ono se stvara tako da se svaki piksel p koji je dio instance teksta na slici, traži sebi najbliži piksel Np koji nije dio te instance teksta. Zatim se stvara dvodimenzionalni vektor koji pokazuje od Np prema p. Orijentacijsko polje čine takvi vektori, a primjer stvaranja jednog takvog polja dan je na slici Slika 9 [20].



Slika 9. Primjer stvaranja orijentacijskog poja [20].

Za kraj, okviri instanci riječi koje je generirala ova grana koriste se za grupiranje detektiranih znakova koje je generirala znakovna grana kao što je prikazano na slici Slika 6. Ako se okvir detektiranog znaka preklapa s okvirom detektirane instance teksta, onda se taj znak dodjeljuje toj instanci teksta [2].

2.4. Iterativna detekcija znakova

Za treniranje znakovne grane potrebne su istiniti okviri na razini znakova, što nije često prisutno u postojećim javnim skupovima podataka. Kako bi se spriječili dodatni troškovi pri nabavljanju takvih okvira na prirodnim slikama Xing et al [2] su uveli iterativnu detekciju znakova koja prolazi kroz sljedeće faze:

- Treniranje inicijalnog modela na skupovima sintetičkih slika kao što je Synth800k. Tamo su dostupne anotacije na razini riječi i na razini znakova.
- Taj model se zatim koristi na prirodnim slikama iz skupa za treniranje kako bi se na njima generirali granični okviri na razini znakova
- Iz generiranih graničnih okvira skupljaju se oni „točni“ za daljnje treniranje modela koristeći prirodne slike. Granični okviri su „točni“ ako je broj znakovnih graničnih okvira u instanci teksta jednak broju oznaka znakova u toj instanci teksta.

Taj postupak se ponavlja iterativno dok se ne prikupi zadovoljavajuća količina graničnih okvira koja se može koristiti za treniranje i testiranje CharNet modela [2]. Primjer 4 iteracije ove metode prikazan je na slici Slika 10.

Slika 10. Primjer iterativne detektcije znakova kroz 4 koraka (crveni okviri su smatrani "točnima", a plavi "netočnima") [2].

3. Programska podrška i implementacija

3.1. Python

Programski jezik Python izmišljen je 1991. godine, od strane nizozemskog programera Guida van Rossuma, a danas je jedan od najšire korištenih programskih jezika. Njegova glavna filozofija objašnjena je u kolekciji principa zvanoj „Zen of Python“ gdje se uglavnom naglašava čitljivost/urednost koda te fokus na jednostavnost.

Python podržava strukturno, funkcijsko te objektno orijentirano programiranje, a danas je vrlo popularan programski jezik u području strojnog učenja. Neki od najpopularnijih okvira za strojno učenje u Pythonu su: Matplotlib, Keras, TensorFlow i PyTorch [9][10].

3.2. PyTorch

Makar je relativno noviji okvir za strojno učenje (u usporedbi s ostalim), Pytorch se danas smatra jednim od najpopularnijih takvih okvira. Među glavnim privlačnostima okvira PyTorch je su lako korištenje i veliki broj biblioteka.

Klasa torch.Tensor definira tenzor multidimenzionalnih podataka nad kojima se mogu raditi razne operacije. To znači da ako u programu postoje dva tenzora s različitim dimenzionalnostima (npr. tenzori koji predstavljaju slike različitih rezolucija), oba su ista klasa i mogu se nad njima raditi iste operacije.

Osim podrške za predstavljanje tenzora, vrlo često koristimo i modul torch.nn koji sadrži osnovne slojeve i aktivacijske funkcije za izražavanje dubokih modela [10].

3.3. CUDA

Biblioteka PyTorch, kao i neke druge, često se koristi za razvoj programa koji se izvršavaju na paralelnim računalima. Biblioteka CUDA dozvoljava programu pristup grafičkim procesorima tvrtke NVidia. CUDA može raditi s programskim jezicima: C, C++ i Python te ima pristup razvojnim alatima i bibliotekama koji pomažu ubrzati izvođenje programa.

Tvrtka Nvidia je 2007. godine napravila platformu CUDA, a programi koji zahtijevaju tu platformu mogu se izvoditi samo na računalima koji imaju NVidijine grafičke kartice (tj. grafičke procesore). Jedno od popularnih rješenja ako računalo nema NVidijine grafičku karticu je servis Google Colab, koji dozvoljava stvaranje Python bilježnica gdje se može izvršavati kod u različitim okruženjima uključujući i pristup NVidijinim grafičkim procesorima [11].

3.4. Instalacija i pokretanje

Za evaluaciju sam koristio implementaciju modela s repozitorija „research-charnet“[1] u okruženju Google Colab koji dozvoljava pokretanje programa kojima je potrebna NVidijina grafička kartica.

U Google Colab bilježnici potrebno je promijeniti „runtime type“ na T4 GPU te klonirati udaljeni repozitorij naredbom „!git clone https://github.com/msight-tech/research-charnet“. To će stvoriti kopiju udaljenog repozitorija u lokalnoj bilježnici. Klonirani repozitorij bit će vidljiv s lijeve strane među datotekama gdje treba dodati sljedeće mape i datoteke:

- Mapa „weights“ u koju se treba dodati trenirani model skinut s interneta[2]
- Mapa „input_images“ gdje se trebaju umetnuti slike nad kojima se želi provesti pronalaženje teksta
- Mapa „result_dir“ gdje će se spremiti izlazne slike i tekstualne datoteke rezultata

Potrebno je i instalirati neke biblioteke bez kojih bi kod javljao greške. Te biblioteke mogu se instalirati naredbom pip na sljedeći način:

```
!pip install torch torchvision
!pip install editdistance
!pip install pyclipper
!pip install shapely
!pip install yacs
```

Za kraj je potrebno pokrenuti i konfiguracijsku/kompajlirajuću komandu:

[1] https://github.com/msight-tech/research-charnet

[2] https://github.com/HilaManor/Scene-Understanding-Based-on-Text-Extraction/issues/24

```
!python setup.py build develop
```

A sama komanda koja pokreće program je:

```
!python tools/test_net.py configs/icdar2015_hourglass88.yaml input_images
results_dir
```

Primjer izlaza na Google Colab bilježnici prikazan je na slici Slika 11.

```
✓ 48s [9]  !python tools/test_net.py configs/icdar2015_hourglass88.yaml input_images results_dir

CHAR_DICT_FILE: datasets/ICDAR2015/test/char_dict.txt
CHAR_MIN_SCORE: 0.25
CHAR_NMS_IOU_THRESH: 0.3
CHAR_STRIDE: 4
INPUT_SIZE: 2280
MAGNITUDE_THRESH: 0.2
NUM_CHAR_CLASSES: 68
RESULTS_SEPARATOR: ,
SIZE_DIVISIBILITY: 128
WEIGHT: weights/icdar2015_hourglass88.pth
WORD_DETECTOR_DILATION: 1
WORD_LEXICON_PATH: datasets/ICDAR2015/test/GenericVocabulary.txt
WORD_MIN_SCORE: 0.95
WORD_NMS_IOU_THRESH: 0.15
WORD_STRIDE: 4
Processing test_image0.jpg...
Processing test_image1.jpg...
Processing test_image10.jpg...
Processing test_image11.jpg...
Processing test_image12.jpg...
Processing test_image2.jpg...
Processing test_image3.jpg...
Processing test_image4.jpg...
Processing test_image5.jpg...
Processing test_image6.jpg...
Processing test_image7.jpg...
Processing test_image8.jpg...
Processing test_image9.jpg...
```

Slika 11. Primjer pokretanja programa.

U mapu results_dir bit će spremljene slike s pravokutnicima oko znakova i riječi koje je model uspio pronaći te će pored tih pravokutnika pisati znakovi i riječi koje je model prepoznao. Osim slika nalazit će se i tekstualne datoteke (jedna za svaku sliku) u kojima će pisati koordinate kuteva pravokutnika oko svih pronađenih riječi (i riječ o kojoj se radi).

4. Evaluacija rezultata

4.1. Eksperimentalni rezultati na javno dostupnim podatkovnim skupovima

Za eksperimente na javno dostupnim skupovima, model CharNet treniran je 5 epoha na sintetičkim podacima iz skupa Synth800k, a sljedeća tri iterativna kruga treniran je na prirodnim slikama 100, 400 i 800 epoha. Početna stopa učenja na sintetičkim podacima je 0.0002, a na prirodnim je 0.002.

Metoda iterativne detekcije znakova opisana u poglavlju 2.3 uspješno skuplja 92.65% točnih riječi iz prirodnih slika nakon 4 iteracije. Xing et al [2] su zaključili da je to dovoljno anotacija za treniranje modela, te se ta količina i koristila za dobivanje rezultata koji su prikazani u ovom poglavlju.

Evaluacija je provedena na tri standardna skupa podataka: ICDAR 2015, ICDAR 2017 MLT te TotalText. Za evaluaciju se koristi usporedba sljedećih vrijednosti:

- Preciznost – označava omjer između točno prepoznatih elemenata i svih prepoznatih elemenata

$$preciznost = \frac{točna\ pozitivna\ detekcija}{točna\ pozitivna\ detekcija + lažna\ pozitivna\ detekcija}$$

- Odziv – označava omjer između točno prepoznatih elemenata i svih točnih elemenata

$$odziv = \frac{točna\ pozitivna\ detekcija}{točna\ pozitivna\ detekcija + lažna\ negativna\ detekcija}$$

- F1-mjera – predstavlja težinsku harmonijsku sredinu preciznosti i odziva; služi za simetričnu reprezentaciju te dvije vrijednosti u jednoj vrijednosti

$$F1 - mjera = \frac{2}{\frac{1}{preciznost} + \frac{1}{odziv}}$$

$$F1 - mjera = 2 \times \frac{preciznost \times odziv}{preciznost + odziv}$$

Bitno je razlikovati rezultate jednorezolucijskog i višerezolucijskog mjerenja. Jednorezolucijsko mjerenje radi se nad svim slikama koristeći jednu fiksiranu rezoluciju, a višerezolucijsko mjerenje koristi više verzija iste slike s različitim veličinama. Ako pored modela piše MS to označava da se radi o višerezolucijskom (eng. multi-scale) modelu, a inače se radi o jednorezolucijskom. Višerezolucijske primjene su uglavnom bolje u detekciji vrlo malih instanci teksta.

4.1.1. ICDAR 2015

ICDAR 2015 sadrži 1000 slika za treniranje i 500 slika za evaluaciju, a sve slike skupljene su korištenjem Google naočala. Najkompliciranije probleme u ovom skupu podataka predstavljaju višestruko orijentirani tekst i tekst malih razmjera.

Metoda	Odziv	Preciznost	F1-mjera
WordSup	77.03	79.33	78.16
EAST	78.33	83.27	80.72
R2CNN	79.68	85.62	82.54
Mask TextSpotter	81.00	91.60	86.00
FOTS R-50	85.17	91.00	87.99
FOTS R-50 MS	87.92	91.85	89.84
CharNet R-50	88.30	91.15	89.70
CharNet H-57	88.88	90.45	89.66
CharNet H-88	89.99	91.98	90.97
CharNet R-50 MS	90.90	89.44	90.16
CharNet H-57 MS	**91.43**	88.74	90.06
CharNet H-88 MS	90.47	**92.65**	**91.55**

Tablica 1. rezultati detekcije teksta na skupu ICDAR 2015 [2].

U tablici Tablica 1 prikazani su rezultati raznih modela pri detekciji teksta na podatkovnom skupu ICDAR 2015. CharNet modeli su najuspješniji u ovom primjeru, a pogotovo model koji koristi Hourglass-88 model koji je najsnažniji od sve tri okosnice.

Nakon detekcije teksta, također se provjerava uspješnost modela pri prepoznavanju teksta na područjima slika gdje je „detektiran“ tekst. Za prepoznavanje teksta uspoređuje se samo F1-mjera, ovisno o tome koristi li se jak, slab, generičan leksikon ili ako se ne koristi leksikon. Leksikoni se koriste za poboljšanje preciznosti prepoznavanja teksta kako bi se dao kontekstno bolji rezultat. Tablica Tablica 2 pokazuje da se rezultati prepoznavanja teksta više manje poklapaju s rezultatima detekcije teksta.

Metoda	Jak leksikon	Slab leksikon	Generičan leksikon	Bez leksikona
Deep text spotter	54.00	51.00	47.00	-
TextProp.+DictNet	53.30	49.61	47.18	-
Mask TextSpotter	79.30	73.00	62.40	-
FOTS R-50	81.09	75.90	60.80	-
FOTS R-50 MS	83.55	79.11	65.33	-
CharNet R-50	80.14	74.45	62.18	60.72
CharNet H-57	81.43	77.62	66.92	62.79
CharNet H-88	83.10	79.15	69.14	65.73
CharNet R-50 MS	82.46	78.86	67.64	62.71
CharNet H-57 MS	84.07	80.10	69.21	65.26
CharNet H-88 MS	**85.05**	**81.25**	**71.08**	**67.24**

Tablica 2. rezultati prepoznavanja teksta na skupu ICDAR 2015 [2].

4.1.2. ICDAR 2017 MLT

Skup ICDAR 2017 MLT koristi 7200 slika za treniranje, 1800 slika za validaciju i 9000 slika za testiranje. Glavni razlog testiranja modela na ovom skupu je zastupljenost tekstom 9 različitih jezika, što pokazuje uspješnost modela na slikama s većom raznolikosti podataka.

Makar je CharNet među boljim modelima što se tiče detekcije teksta na ovom skupu podataka, tablica Tablica 3 ukazuje na to da je višerezolucijska primjena modela FOTS preciznija od najsnažnijeg CharNet modela, makar ima manji odziv.

Metoda	Odziv	Preciznost	F1-mjera
SARI_FDU_RRPN	55.50	71.17	62.37
SCUT_DLVClab	54.54	80.28	64.96
FOTS	57.51	80.95	67.25
FOTS MS	62.30	**81.86**	70.75
CharNet R-50	70.10	77.07	73.42
CharNet H-88	**70.97**	81.27	**75.77**

Tablica 3. rezultati detekcije teksta na skupu ICDAR 2017 MLT [2].

4.1.3. TotalText

Najmanji od ova tri skupa podataka, TotalText u skupu za treniranje sadrži 1255 slika, a u skupu za treniranje 300 slika. U slikama ovog skupa nalaze se razni tekstovi s višestruko orijentiranim i zakrivljenim tekstom.

U tablici Tablica 4 vidi se usporedba različitih modela pri detekciji i prepoznavanju teksta na skupu TotalText. Kao i kod ostalih skupova, CharNet daje najbolje rezultate. Zanimljiv detalj je da je jednorezolucijski Charnet H-88 model skoro 2% precizniji od višerezolucijske primjene modela, ipak F1 je bolji. Za prepoznavanje teksta se nije koristio leksikon.

Metoda	Detekcija			Prepoznavanje
	Odziv	Preciznost	F1-mjera	F1-mjera
Textboxes	45.5	62.1	52.5	36.3
Mask TextSpotter	55.0	69.0	61.3	52.9
TextNet	59.5	68.2	63.5	54.0
TextField	79.9	81.2	80.6	-
CharNet H-57	81.0	88.6	84.6	63.6
CharNet H-88	81.7	**89.9**	85.6	66.6
CharNet H-57 MS	**85.0**	87.3	86.1	66.2
CharNet H-88 MS	**85.0**	88.0	**86.5**	**69.2**

Tablica 4. rezultati detekcije i prepoznavanja teksta na skupu TotalText [2].

4.2. Eksperimentalni rezultati na vlastitom podatkovnom skupu

Slike iz vlastitog podatkovnog skupa prikupljene su diljem Zagreba te smo na njima primijenili model CharNet kao što je opisano u poglavlju 3.4. Model je treniran na skupu podataka ICDAR 2015 s mrežom Hourglass-88 kao okosnicom.

Oko znakova su generirani zeleni granični okviri, a oko riječi su generirani crveni granični okviri (istim bojama označili smo i prepoznati znakovi/riječi).

Slika 12. Primjer dobre detekcije i prepoznavanja teksta 1.

Slika 13. Primjer dobre detekcije i prepoznavanja teksta 2.

Slika 14. Primjer slabije detekcije i prepoznavanja teksta 1.

Slika 15. Primjer slabije detekcije i prepoznavanja teksta 2.

Slika 16. Primjer loše detekcije i prepoznavanja riječi.

Slike Slika 12 i Slika 13 su među boljim primjerima detekcije i prepoznavanja teksta (riječi i znakova). U slikama Slika 14 i Slika 15 detektirani su uglavnom svi znakovi, ali ne i sve riječi.

Slika Slika 16 je primjer loše detekcije riječi. Riječ „Miloša“ je prepoznata kao riječ „MIMOSA“, a iz riječi „BUS-a“ izrezan je znak a pa je riječ prepoznata kao „BUSS“. Također nisu prepoznate sve riječi. Ovaj primjer pokazuje kako grafiti i slična ometanja mogu loše utjecati na ovaj model koji takve stvari može prepoznati kao znakove.

Zaključak

Napredak u dubokom učenju pronašao je svoje primjene kod pronalaženja teksta u prirodnim slikama. Taj problem ima razne korisne primjene, kao npr. autonomna vozila ili aparati za slabovidne/slijepe ljude.

Uz pomoć operacija konvolucije i pažnje mogu se stvarati različite arhitekture neuronskih mreža ovisno o tome za što je namijenjena ta mreža. U kontekstu pronalaženja teksta, najčešće se radi o konvolucijskim neuronskim mrežama.

U slučaju modela CharNet koristi se potpuno konvolucijska neuronska mreža koja spaja detekciju i prepoznavanje teksta na prirodnim slikama u jednofazni proces. CharNet također koristi znakove kao osnovnu jedinicu pri traženju teksta na slikama. Za to koristi revolucionarne metode tog vremena poput iterativne detekcije znakova, a pri detekciji riječi koristi i neke druge modele kao što su EAST i Textfield.

U ovom radu opisana je cijela arhitektura modela CharNet te se proučila usporedba njegove generalizacijske moći s drugim tada najsuvremenijim modelima. Rezultati su pokazali da CharNet uglavnom nadmašuje sve druge modele na tri različita skupa podataka. Također su prikazani i neki primjeri prepoznavanja teksta na slikama iz vlastitog skupa podataka koristeći javnu implementaciju modela.

Ipak, područje pronalaženja teksta još uvijek nije savršeno te će vjerojatno nastaviti napredovati u narednim godinama.

Literatura

[1] Thomas, E., *A Century Ago, the Optophone Allowed Blind People to Hear the Printed Word,* IEEE Spectrum (2021., srpanj) Poveznica: https://spectrum.ieee.org/alternative-for-gps; pristupljeno 26. kolovoza 2024.

[2] Xing, L., Tian, Z., Huang, W., Scott, M. R., *Convolutional Character Networks*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), Shenzen, Adelaide, (2019.)

[3] Subašić, M., *Detekcija objekata u slikama pomoću dubokih neuronskih mreža, prezentacija s predmeta Obrada informacija na Fakultetu elektrotehnike i računarstva,* (2023., Prosinac). Poveznica: https://www.fer.unizg.hr/_download/repository/Detekcija.pdf; pristupljeno 26. kolovoza 2024.

[4] Suri, Z. K., *Convolution vs. Attention,* Github Curiosity, (2023., ožujak). Poveznica: https://zshn25.github.io/CNNs-vs-Transformers/; pristupljeno 26. kolovoza 2024.

[5] Kohli, V., *Context window*, TechTarget, (2023., listopad). Poveznica: https://www.techtarget.com/whatis/definition/context-window; pristupljeno 26. kolovoza 2024.

[6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., *An image is worth 16x16 words: transformers for image recognition at scale*. International Conference on Learning Representations (ICLR) 2021, (2020.)

[7] Krapac J., Šegvić, S., *Konvolucijski modeli, prezentacija s predmeta Obrada informacija na Fakultetu elektrotehnike i računarstva.* Poveznica: https://www.zemris.fer.hr/~ssegvic/du/du2convnet.pdf; pristupljeno 26. kolovoza 2024.

[8] Tsang, S., *Review: FCN - Fully Convolutional Network (Semantic Segmentation),* Medium, (2018., listopad). Poveznica: https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1; pristupljeno 26. kolovoza 2024.

[9] W3schools, *Python Introduction*. Poveznica: https://www.w3schools.com/python/python_intro.asp; pristupljeno 26. kolovoza 2024.

[10] Nederkoorn, C., *Top 10 Python Packages for Machine Learning*, ActiveState, (2020., veljača). Poveznica: https://www.activestate.com/blog/top-10-python-machine-learning-packages/; pristupljeno 26. kolovoza 2024.

[11] Supermicro, *What is CUDA?* Poveznica: https://www.supermicro.com/en/glossary/cuda; pristupljeno 26. kolovoza 2024.

[12] Law, H., Deng, J., *CornerNet: Detecting Objects as Paired Keypoints*. European Conference on Computer Vision (ECCV), Sveučilište Princeton, (2019.), str. 6-7.

[13] Li, S., *Simple Introduction about Hourglass-like Model*, Medium, (2017., listopad). Poveznica: https://medium.com/@sunnerli/simple-introduction-about-hourglass-like-model-11ee7c30138; pristupljeno 26. kolovoza 2024.

[14] Potrimba, P., *What is ResNet-50?*, roboflow, (2024., ožujak). Poveznica: https://blog.roboflow.com/what-is-resnet-50/; pristupljeno 26. kolovoza 2024.

[15] Sahoo, S., *Residual blocks - Building blocks of ResNet*, (2018., studeni). Poveznica: https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec; pristupljeno 26. kolovoza 2024.

[16] He T., Tian Z., Huang W., Shen C., Qiao Y., Sun C., *An end-to-end textspotter with explicit alignment and attention*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Adelaide, Shenzen, (2018., ožujak)

[17] Li H., Wang P., Shen C., *Towards End-to-end Text Spotting with Convolutional Recurrent Neural Networks*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), Adelaide, (2019., srpanj)

[18] Fugošić K., *Pronalaženje teksta dubokim konvolucijskim modelima*. Zagreb, (2018., srpanj)

[19] Zhou X., Yao C., Wen H., Wang Y., Zhou S., He W., Liang J., *East: an efficient and accurate scene text detector*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Peking, (2017., srpanj)

[20] Xu Y., Wang Y., Zhou W., Wang Y., Yang Z., Bai X., *TextField: Learning A Deep Direction Field for Irregular Scene Text Detection*. IEEE Transactions on Image Processing (TIP), (2019., srpanj)

Sažetak

PRONALAŽENJE TEKSTA U PRIRODNIM SLIKAMA

Napredak u dubokom učenju pokazuje svoje tragove i u pronalaženju teksta u prirodnim slikama. Pažnja je često korištena metoda kod implementacije neuronskih mreža, ali kod segementacije slika popularnija je metoda konvolucije. Model CharNet je potpuno konvolucijski model koji sadrži dvije grane: znakovnu granu i granu detekcije teksta. Taj model koristi i metode poput iterativne detekcije znakova, ali i neke druge modele kao što su EAST i Textfield. Pokazalo se da CharNet ima bolju generalizacijsku moć od tada najsuvremenijih modela, kod detekcije i kod prepoznavanja teksta.

Ključne riječi: umjetna inteligencija, duboko učenje, neuronske mreže, konvolucija, računalni vid, pronalaženje teksta, CharNet

Summary

TEXT LOCALIZATION IN NATURAL IMAGES

Advances in deep learning also show their traces in finding text in natural images. Attention is a frequently used method in the implementation of neural networks, but in image segmentation, the convolution method is more popular. The CharNet model is a fully convolutional model that contains two branches: a character branch and a text detection branch. This model uses methods such as iterative character detection, but also some other models such as EAST and Textfield. It has been shown that CharNet has a better generalization power than the most modern models at the time, both in text detection and in text recognition.

Keywords: artificial intelligence, deep learning, neural networks, convolution, computer vision, text retrieval, CharNet