Primjena modela dubokog učenja za prepoznavanje teksta u minskim zapisnicima

Krmpotić, Klara

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 472

PRIMJENA MODELA DUBOKOG UČENJA ZA PREPOZNAVANJE TEKSTA U MINSKIM ZAPISNICIMA

Klara Krmpotić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 472

PRIMJENA MODELA DUBOKOG UČENJA ZA PREPOZNAVANJE TEKSTA U MINSKIM ZAPISNICIMA

Klara Krmpotić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 472

Pristupnica:	**Klara Krmpotić (0036519753)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	doc. dr. sc. Marko Horvat

Zadatak: **Primjena modela dubokog učenja za prepoznavanje teksta u minskim zapisnicima**

Opis zadatka:

Pomoću naprednih arhitektura dubokih neuronskih mreža danas se uspješno rješavaju brojni problemi iz području računalnog vida uključujući detekciju teksta i optičko prepoznavanje znakova u slikama. Istodobno, u domeni humanitarnog razminiranja važan izvor informacija su minski zapisnici koji predstavljaju rukom pisane dokumente s informacijama o razmještaju i vrsti minsko-eksplozivnih sredstava koja se nalaze u nekom području. U okviru diplomskog rada potrebno je upoznati se s modelima dubokog učenja koji se temelje na tradicionalnim i unaprijednim konvolucijskim neuronskim mrežama (eng. convolutional neural networks, CNN) te zatim proučiti radni okvir YOLO (engl. You Only Look Once). Upoznati se s problemom optičkog prepoznavanja znakova u slikama. Razviti programsku podršku i implementirati odabrani YOLO model za određivanje regija u slikama minskih zapisnika te prepoznavanje teksta unutar detektiranih regija. Provesti eksperimentalno vrednovanje modela i statističku obradu rezultata. Radu priložiti izvorni i izvršni kod razvijenog sustava uz potrebna dodatna objašnjenja i dokumentaciju. Također, radu priložiti označene skupove podataka te citirati korištenu literaturu.

Rok za predaju rada: 28. lipnja 2024.

Zahvaljujem se mentoru, doc. dr. sc. Marku Horvatu na strpljenu i pomoći kod pisanja rada.
Isto tako, svojoj obitelji i prijateljima na podršci tokom studiranja.

Sadržaj

Uvod

U humanitarnom razminiranju bitno je imati precizne i pouzdane senzore te ispravnu opremu koja će olakšati posao na terenu i pružiti sigurnost osoblja i rada, no isto tako potrebno je imati pouzdane tehnike za obradu i analizu podataka. [7] Iz tih razloga je osobito važno da minski zapisnici budu ispunjeni uredno i u potpunosti te da su što bolje očuvani, kako bi se njihovi podatci kasnije mogli koristiti i pomoći u budućim istraživanjima.

Minski zapisnici su dokumenti koji sadrža informacije o položaju i vrsti minsko-eksplozivnih sredstava koja se nalaze u nekom području. Dosad, većima minskih zapisnika je bilo pisano rukom te još 1990-tih godina te su bili pohranjeni na raznim mjestima. Razvojem računalnog vida, nastala je mogućnost za automatsko prepoznavanje rukom pisanog teksta i optičko prepoznavanje znakova u slikama te zapise digitalizirati i učiniti ih dostupnima u digitalnom formatu, što je ubrzalo proces obrade i analize tih podataka. Digitalizacija minskih zapisnika olakšava njihovo pretraživanje, dijeljenje I arhiviranje, što doprinosi sigurnijem i učinkovitijem upravljanju minskih polja. Na taj način, računalni vid značajno doprinosi modernizaciji i unapređenju minskih evidencija.

Računalni vid je tehnologija koja se još razvija. Prepoznavanje objekata i teksta koji su razmješteni na drugačijim mjestima ili različite boje ili fonta, čovjek vrlo jednostavno može prepoznati kao iste, dok je za računalo to puno kompleksnije. Da bi računala bila uspješnija u tome, potrebno ih je trenirati. Tu se primjenjuju duboke neuronske mreže i modeli koji ih koriste. [8]

U konkretnom slučaju ovoga rada riječ je o modelu YOLO (*You Only Look Once*), koji se koristi za prepoznavanje objekata na slikama. YOLO model se trenira na skupu slika na kojima oko svakoga objekta, područja interesa, se definira takozvane granične okvire (*engl. bounding boxes*), za koje definira parametre da bi na neviđenim slikama mogao odrediti poziciju istih objekata ili područja. Pošto je riječ o skeniranim dokumentima, potrebno je nakon što se odrede ta područja interesa na kojima se nalazi tekst, da se tekst I pročita. U tu svrhu se u model ubacuje posebno područje računalnog vida koje služi za raspoznavanje znakova na slici, koje se naziva optičko prepoznavanje znakova (engl. *Optical Character Recognition*, OCR).

Tema kojom se ovaj rad bavi je da se uz pomoć YOLO modela, koji će na poznatom skupu podataka minskih zapisnika naučiti gdje se nalaze područja interesa, tekst koji je potrebno pročitati i definirati okvire za njih. Zatim korištenjem određenog OCR-a i pročitati taj tekst te rezultate pohraniti u json formatu.

1. Minski zapisnici u humanitarnom razminiranju

Svake godine u procesu razminiranje je uklonjeno i deaktivirano otprilike 100 tisuća mina, dok je približno 2 milijuna novih mina postavljeno. Međunarodni odbor Crvenog križa procjenjuje da stopa žrtava od mina trenutno premašuje 26 000 osoba svake godine. Procjenjuje se da 800 osoba bude ubijeno i 1200 ozlijeđeno svakog mjeseca od mina diljem svijeta. Primarne žrtve su nenaoružani civili, a među njima su posebno pogođena djeca. [3]

Humanitarno razminiranje je postupak koji služi da bi se uklonila minsko eksplozivna sredstva iz poslijeratnih područja. Podijeljeno je na skupljanje podataka, njihovu obradu te pohranu i daljnje korištenje. Kod razminiranja podaci se prikupljaju iz raznih izvora, primjerice pomoću magnetskih detektora, radara koji prodiru u zemlju i hiperspektralne slike iz zraka snimljene bespilotnim letjelicama (UAV), zrakoplovima i satelitima. [4]

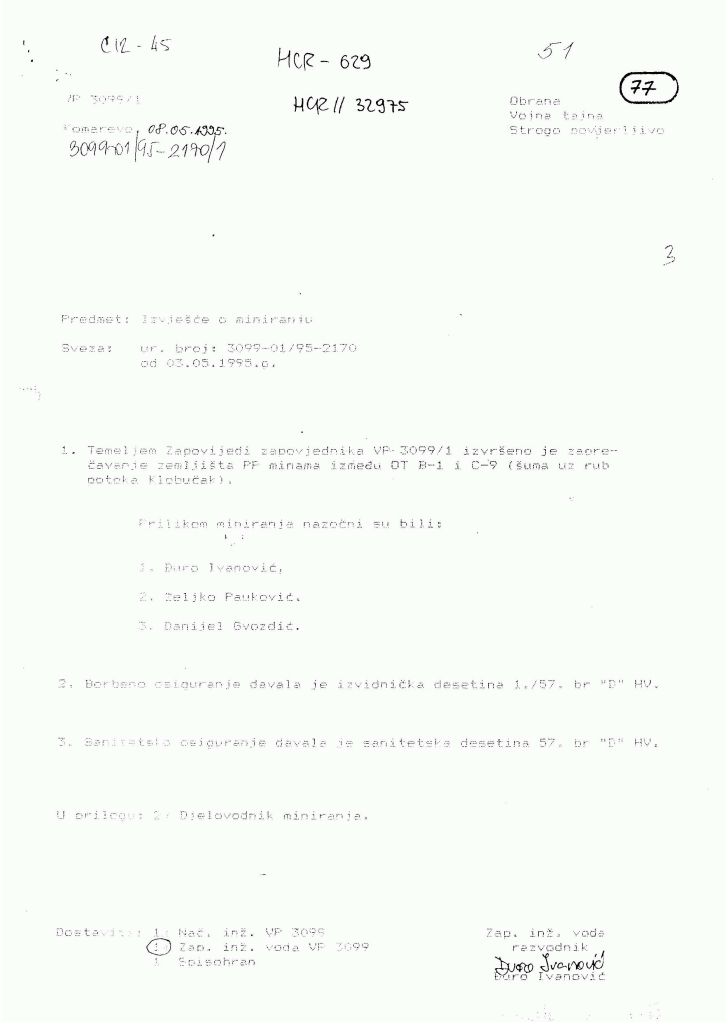
Jednom kada su ti podaci prikupljeni, njihovo dijeljenje i upravljanje postaje pomalo izazovno. Izrazito je važno da su ti podaci pouzdani i sigurno pohranjeni. Isto tako, ti podaci se dijele između različitih skupina dionika, na primjer između vlada, službenih i međunarodnih organizacija.

Postoje više tipova zapisnika koji se ispunjavaju. Postoji službeni formulari koji imaju točno označena polja gdje koja informacija ide i njih je izrazito lako čitati, to je format koji je odabran u ovome radu. Njegov primjer je prikazan na Slika 3 i Slika 4. No, uz ovaj format postoje još i drugi, koji su manje strukturirani, jedan primjer je pokazan na Slika 1. No njih nije bilo u tako velikom broju, a pošto nisu u istom formatu kao i ostali, samo bi modelu bilo teže odrediti ista područja interesa na kojima je tekst, što bi rezultiralo lošijom generalizacijom.

Što je bila česta situacija kod minskih zapisnika je da su na papiru bile mrlje od tinte te su tako mogla nastati „oštećenja“ na tekstu, što bi moglo utjecat na OCR i koliko točne će moći pročitati tekst. Primjer jedne manje mrlje na papiru se vidi na Slika 4 te će recimo to moći biti uklonjeno jer nema teksta ispod i nije preveliko područje. Proces kako će se to izbrisati je opisan u sljedećem poglavlju.

Također, imaju zapisnici gdje su crna područja toliko velika da ih nije moguće tako jednostavno ukloniti. Primjerice, Slika 2, ima gore područje zbog kojega se ne vidi tekst,

registarski broj, u gornjem lijevom kutu. No ostatak teksta zapisnika se vidi pa će zapisnik biti uključen u skup podataka.



CR - 45 HCR - 629 51

HCR // 32975

77

VP 3099/1

Obrana
Vojna tajna
Strogo povjerljivo

Komarevo, 08.05.1995.
3099-01/95-2190/1

3

Predmet: Izvješće o miniranju

Veza: ur. broj: 3099-01/95-2170
od 03.05.1995.g.

1. Temeljem Zapovijedi zapovjednika VP-3099/1 izvršeno je zaore-
čavanje zemljišta PP minama između OT B-1 i C-9 (šuma uz rub
potoka Klobučak).

Prilikom miniranja nazočni su bili:

1. Đuro Ivanović,

2. Željko Paušković,

3. Danijel Gvozdić.

2. Borbeno osiguranje davala je izviđnička desetina 1./57. br "D" HV.

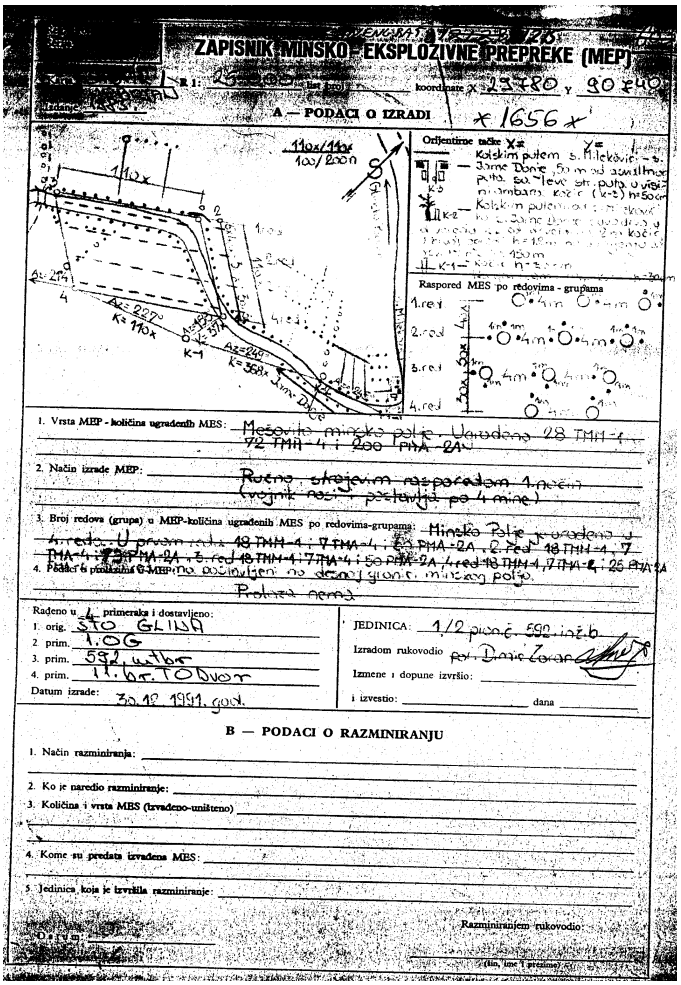
3. Sanitetsko osiguranje davala je sanitetska desetina 57. br "D" HV.

U prilogu: 2 x Dielovodnik miniranja.

Dostaviti: 1. Nač. inž. VP 3099
1 Zap. inž. voda VP 3099
1 Spisohran

Zap. inž. voda
razvodnik
Đuro Ivanović
Đuro Ivanović

Slika 1. Minski zapisnik 32975



ZAPISNIK MINSKO-EKSPLOZIVNE PREPREKE (MEP)

koordinate X 23480 Y 90240

A — PODACI O IZRADI

Orijentirne točke X= Y=

Raspored MES po redovima - grupama

1. Vrsta MEP - količina ugrađenih MES: Mešovito minsko polje. Uradena 28 TMM-4, 72 TMA-4 i 200 PMA-2A

2. Način izrade MEP: Ručno, strojevim raspoređem 4 reda (vojnik nosi i postavlja po 4 mine)

3. Broj redova (grupa) u MEP-količina ugrađenih MES po redovima-grupama: Minsko Polje je urađeno u 4. redu. U prvom redu 18 TMM-4 ; 7 TMA-4 ; 50 PMA-2A , 2. red 18 TMM-4 , 7 TMA-4 ; 25 PMA-2A , 3. red 18 TMM-4 ; 7 TMA-4 ; 50 PMA-2A , 4. red 18 TMM-4 , 7 TMA-4 ; 25 PMA-2A

4. Podaci o pružanju G-MEP: Na desnom glavn. minskom polju.

Poljska cesta

Rađeno u 4 primerka i dostavljeno:
1. orig. STO GLINA
2. prim. 1.OG
3. prim. 592. u/bor
4. prim. 11. br. TOPUSKO

Datum izrade: 30.12.1994. god.

JEDINICA: 1/2 pvač. 592. inž.b.

Izradom rukovodio por. Dragic Zoran

Izmene i dopune izvršio: _____

i izvestio: _____ dana _____

B — PODACI O RAZMINIRANJU

1. Način razminiranja: _____

2. Ko je naredio razminiranje: _____

3. Količina i vrsta MES (izvađeno-uništeno) _____

4. Kome su predata izvađena MES: _____

5. Jedinica koja je izvršila razminiranje: _____

Datum _____

Razminiranjem rukovodio: _____

(ime, ime i prezime)

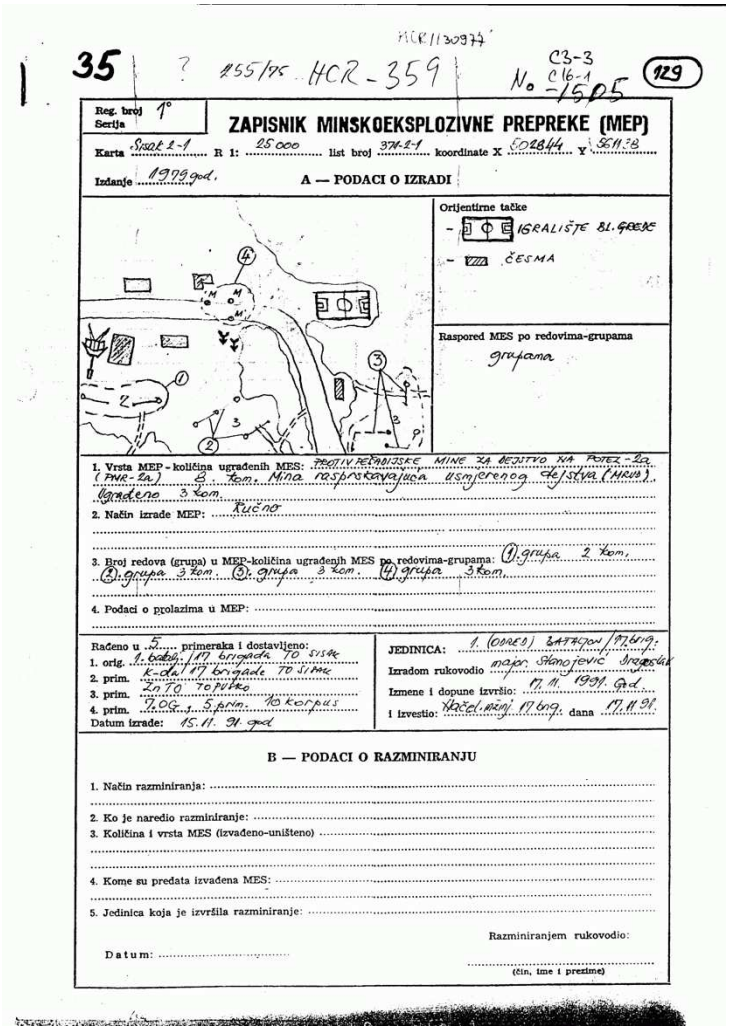
Slika 2. Minski zapisnik gp1656

2. Skup podataka

Kao što je već i prije navedeno, skup podataka čine minski zapisnici. To su službeni dokumenti koji se ispunjavaju podacima skupljenih iz različitih izvora, kako od službenih tako i od neslužbenih povijesnih izvora, razgovora sa stanovnicima susjednih područja, očitanja senzora i tako dalje. [5] Isto tako, s vremenom je moguće prikupiti nove informacije za određeno područje te je sve te promjene potrebno zabilježiti.

Za potrebe modela odabran je specifični obrazac dokumenta koji se koristi za ispunjavanje i zapisivanje informacija. Dokument može imati samo prednju ili prednju i stražnju stranicu. Na stražnjoj strani je uglavnom uvećana skica, ako nije stalo sve na prednju stranicu. Uz to mogu biti zapisane i neke informacije koje su bile dobivene kasnije ili iz drugog izvora. Primjer jednog takvog minskog zapisnika može se vidjeti u nastavku (Slika 3 i Slika 4).

U gornjem dijelu prednje strane dokumenta nalaze se osnovne informacije po kojima se bilježe i spremaju podaci o dokumentu, njegovo zaglavlje. Zatim slijedi skica područja na kojemu se nalaze minsko eksplozivne prepreke (MEP), kod ovog dokumenta je to primjerice dodatno i preciznije nacrtano na stražnjoj strani uz dodatni opis, moguće informacije koje su naknadno prikupljene. Pošto je sadržaj slike moguće razumjeti jedino u cijelosti, zasad nema potrebe da se svaki objekt na slici čita zasebno, nego će se slika u potpunosti spremati u rezultat. Nakon toga dolaze informacije o kakvim je MEP-ovima riječ; njihova vrsta, količina, pozicija redova i je li moguće proći između njih. Potom su informacije o broju kopija zapisnika i tko ih posjeduje. Desno od toga je dio dokumenta u kojem je zapisano tko je vodio razminiranje, kojom jedinicom i kada je izvijestio i koga. Na kraju dolaze podaci o razminiranje; što je izvađeno i u kojoj količini, kome je to predano i na koji način se razminiralo.



35 ? 155/95 HCR-359 HCR/130977 C3-3 No C16-1 1505 129

Reg. broj 1°
Serija

ZAPISNIK MINSKOEKSPLOZIVNE PREPREKE (MEP)

Karta Smat 1-1 R 1: 25 000 list broj 374-1-1 koordinate X 5028 44 Y 5611 38

Izdanje 1979 god.

A — PODACI O IZRADI

Orijentirne tačke
- IGRALIŠTE BL. GREBE
- ČESMA

Raspored MES po redovima-grupama
grupama

1. Vrsta MEP - količina ugrađenih MES: PROTIVTENKOVSKE MINE ZA BOJITVO NA POTEZ - Za (PMR - Za) 8 kom. Mina raspskavajuca usmerenog dejstva (MRUD) ugrađeno 3 kom
2. Način izrade MEP: Ručno
3. Broj redova (grupa) u MEP-količina ugrađenih MES po redovima-grupama: (1). grupa 2 kom. (2). grupa 3 kom. (3). grupa 3 kom. (4) grupa 3 kom.
4. Podaci o prolazima u MEP:

Rađeno u 5 primeraka i dostavljeno:
1. orig. 1. bataj. 17 brigade 70 Sisak
2. prim. k-dr 17 brigade 70 Sisak
3. prim. 2n TO 70 Petrinja
4. prim. 7. OG 5. arm. 10 korpus
Datum izrade: 15.11. 91. god

JEDINICA: 1. (ODRED) BATRON / 17 brig.
Izradom rukovodio major Stanojević Dragoslav
Izmene i dopune izvršio: 17. 11. 1991. God.
i Izvestio: Načel. štaba 17 brig. dana 17.11.91.

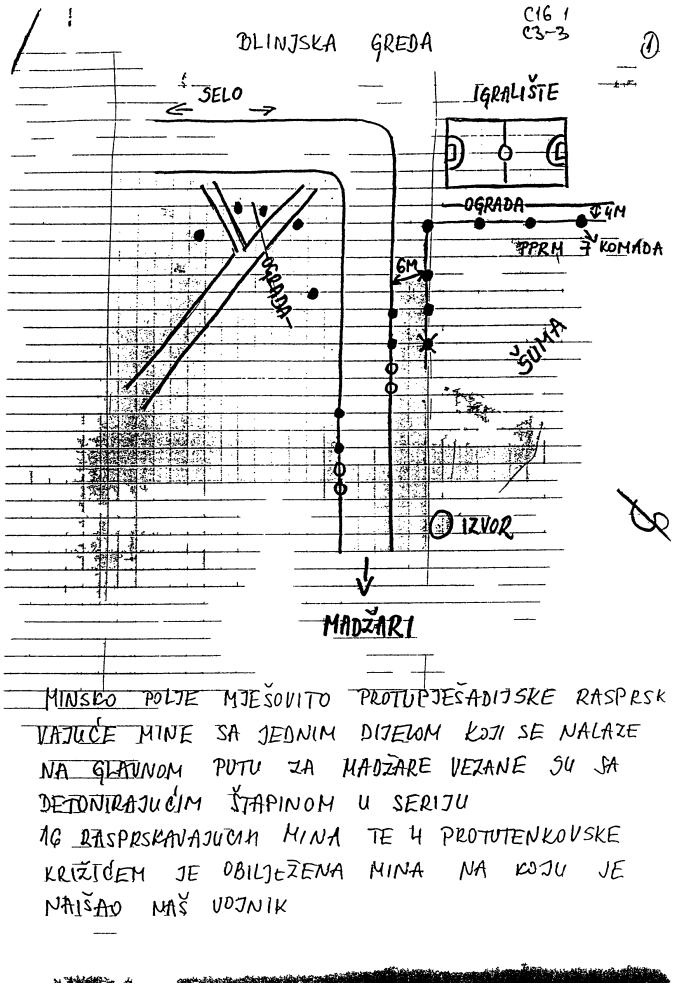
B — PODACI O RAZMINIRANJU

1. Način razminiranja:
2. Ko je naredio razminiranje:
3. Količina i vrsta MES (izvađeno-uništeno)
4. Kome su predata izvađena MES:
5. Jedinica koja je izvršila razminiranje:

Razminiranjem rukovodio:
(čin, ime i prezime)

Datum:

Slika 3. Minski zapisnik 30977a



Slika 4. Minski zapisnik 31144b

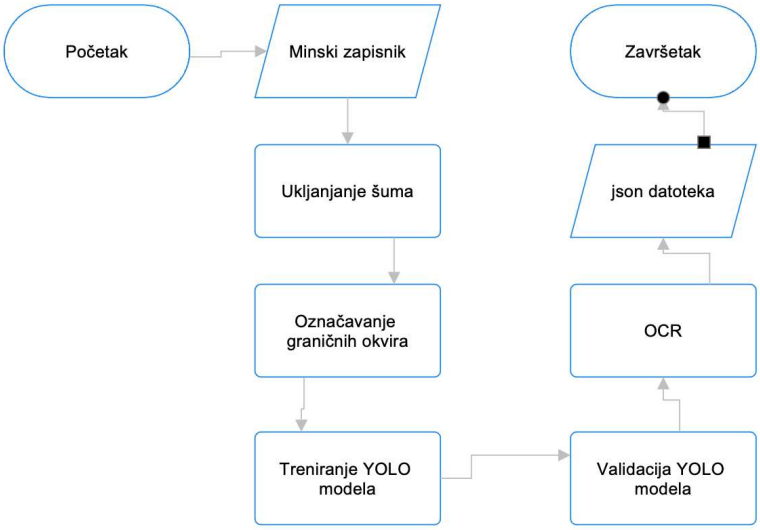
Kao što se na slici iznad može vidjeti i kao što je navedeno u poglavlju prije, postoje manja crna područja, moguće od razmrljane tinte ili nečega drugoga, ta područja se nazivaju šumom. Da bi model i optičko prepoznavanje znakova moglo lakše pročitati i dati što bolji rezultat, potrebno je te mrlje ukloniti. To je učinjeno programski koristeći Pythonovu biblioteku otvorenog koda, OpenCV. Prvo se učitana slika pretvara u sivu sliku jer je to kasnije potrebno za konture. Zatim se određuje granična vrijednost (engl. *threshold*) koja se koristi za stvaranje binarne slike i postavlja se fokus na određena područja interesa. Rezultat toga je binarna slika gdje je pozadina bijela a tekst je crn. Zatim se na tu sliku primjenjuje jezgru (engl. *kernel*), matrica dimenzija `4 x 4`, provodi se konvolucija nad slikom, kojom se mijenja struktura slika. Matrica je odabrana u obliku elipse najviše iz razloga da se dobije prirodniji efekt zaglađivanja. Radi se u principu da se izbacuju pa dodaju pikseli na rubove objekata. Cilj ovoga postupka je izgladiti konture objekata i eliminirati šum oko njih, na primjer mrlje od tinte. Zatim se prolazi kroz svaku tu konturu i gleda se njezina površina, tu

je određen prag od 150 piksela. U slučaju da je površina manja od praga, smatra se da je riječ o šumu i obojat će ga se u crno čime će se uklopit u pozadinu i izbrisati na originalnoj slici. Za kraj je još potrebno invertirati binarnu sliku te je rezultat slika bez šuma.

Kod nekih slika u skupu podataka je uklanjanje šuma bilo uspješno, dok je za neke bilo manje uspješno. Iz tog razloga, se ručno izabralo hoće li se u skup podataka za treniranje iskoristiti originalna slika ili slika s koje se probao ukloniti šum.

Skup podataka je podijeljen u tri podskupa: skup za treniranje, skup za validaciju, skup za testiranje. Važno je da su skupovi međusobno disjunktni. Model će se učiti na skupu za učenje, zatim će se njegovu pogrešku validacije procijeniti na skupu za validaciju. Skup za testiranje koristimo kao skup primjere koje model nije vidio te će se na njima gledati koliko je model zapravo točan. U skupu za treniranje nalazi se 60 posto cjelokupnog skupa podataka, dok je u skupu za validaciju i testiranje podjednako raspodijeljeno ostalih 40 posto dokumenata. [6] Raspored slika po skupovima je nasumično odabran, samo je bilo važno da ako postoji stražnja strana zapisnika, da ona bude u istom skupu kao i njezina prednja stranica jer bi inače model krivo učio. Prednja stranica je bila označena dodatnom oznakom „a“ nakon brojčane oznake zapisnika, a stražnja stranica je imala oznaku „b“, da se i njihov redoslijed ne bi pomiješao.

Slika u nastavku prikazuje korake sustava, koji se koristio u ovome radu. Prva dva procesa, uklanjanje šuma i označavanje graničnih okvira, predstavljaju korake pred-procesiranja podataka. Nakon toga slijedi treniranje i evaluacija YOLO modela koristeći predefinirane konfiguracije. I na kraju, koristeći rezultate dobivene modelom, minski zapisnici prolaze OCR-om, točnije OCR se primjenjuje na svaki granični okvir koji je definiran za taj zapisnik.



Slika 5. Dijagram sustava

3. YOLO model

3.1 YOLO v1 do v7

YOLO model je prvi put najavljen 2016. godine. Kod detekcije objekata postoje jednoprolazni i dvoprolazni detektori. YOLO model je jednoprolazni detektor, što znači da pokušava detektirati objekte u samo jednom prolazu, kao što i sam naziv kaže, što osjetno smanjuje vrijeme potrebno za treniranje modela. Razlika između YOLO modela i ostalih modela u to doba, je da je ideja iza YOLO modela bila da se problem rješava na regresijski način, znači predviđanje numeričkih vrijednosti (rezultati su brojevi), umjesto s klasifikacijom, pojedinačnim primjerima se predviđan diskretna oznaka, kao što je implementirano kod tada poznatih modela. [7]

Prva verzija, YOLOv1, je primijenila jedinstveni pristup tako što je koristila (1×1) konvoluciju nakon koje je slijedio (3×3) konvolucijski filter. Slika se podijeli na određeni broj ćelija, koje su jednakih veličina. Cilj je gledati je li središte objekta koji se nalazi na slici, pao unutar neke ćelije mreže, zatim ta ćelija identificira objekt. Svaka ćelija se sastoji od okvira te se kao rezultat dobiva vjerojatnost da se objekt nalazi unutar naznačenog okvira. [8]

YOLOv2, za razliku od svog prethodnika YOLOv1, predviđa dimenzije objekta u različitim kvadratnim rasponima veličina, od 320 x 320 do 608 x 608, odbacivanjem potpuno povezanih slojeva koji su prije bili prisutni. Ova verzija isto tako uvodi različite metode rastresanja (*engl. data augmentation*) I optimizacijske strategije. Sastoji se od 19 konvolucijskih slojeva i 5 slojeva sažimanja maksimumom (engl. *max pooling*). [9]

YOLOv3 je napravio korak naprijed u detektiranju manjih objekata na slikama. Fokus je bio ispravku grešaka lokalizacije i poboljšanju učinkovitosti optimizacijske detekcije. Izgrađen je na temelju Darknet-53 okvira, što znači da se sastojao od 53 konvolucijska sloja. Ovom arhitekturom, brzina YOLO modela se gotovo udvostručila. YOLOv3 je umjesto softmax-a i binarne unakrsne entropije, koristio logističku regresiju za izračunavanje rezultata za svaki granični okvir. [10]

YOLOv4 donosi značajni napredak u arhitekturi modela, povećavajući brzinu i točnost u detekciji objekata. Ova verzija uključuje integraciju CSPDarknet53 (*Cross-Stage Partial*),

SPP i PANet. Još uz to koristi i više graničnih okvira za točniju detekciju te se koristi drugačija funkcija gubitka. Ova verzija je pojednostavnila učenje te učinila YOLO model dostupnijim korisnicima, Koristi sofisticirane metodologije poput CSP veze i napredne tehnike skaliranja značajki za optimizaciju performansi uz održavanje računalne učinkovitosti.

YOLOv5 je prva verzija gdje se uvodi PyTorch, umjesto Darknet-a koji se u prijašnjih nekoliko verzija koristio, prvenstveno zbog njegove prilagođenosti korisniku. Tu je predloženu u arhitekturi da se u konvolucijskom sloju koristi prozor (što je zapravo matrica), kojim se onda kliže po slici, određenim korakom (engl. *stride*). Cilj je bio smanjiti potrošnju memorije i vrijeme računanja.

YOLOv6 optimiziran za otkrivanje u stvarnom vremenu na CPU-u i GPU-u. Sve u svemu, predstavlja značajan napredak u evoluciji YOLO arhitektura, s poboljšanjima u brzini, točnosti i učinkovitosti. Za razliku od verzija prije, YOLOv6 koristi EfficientREP kao novu okosnicu (engl. *backbone*), dopušta više paralelnih pokretanja, čime prestiže prethodne verzije u učinkovitosti i brzini.

YOLOv7 je bio treniran na MS COCO (The Common Objects in Context) skupu podataka. Ova verzija je predložila predložio proširenu verziju učinkovite mreže agregacije slojeva (ELAN), mehanizam za učinkovitije učenje i konvergenciju u dubokim modelima tako što kontrolira najkraćeg najdužeg gradijentnog spusta.

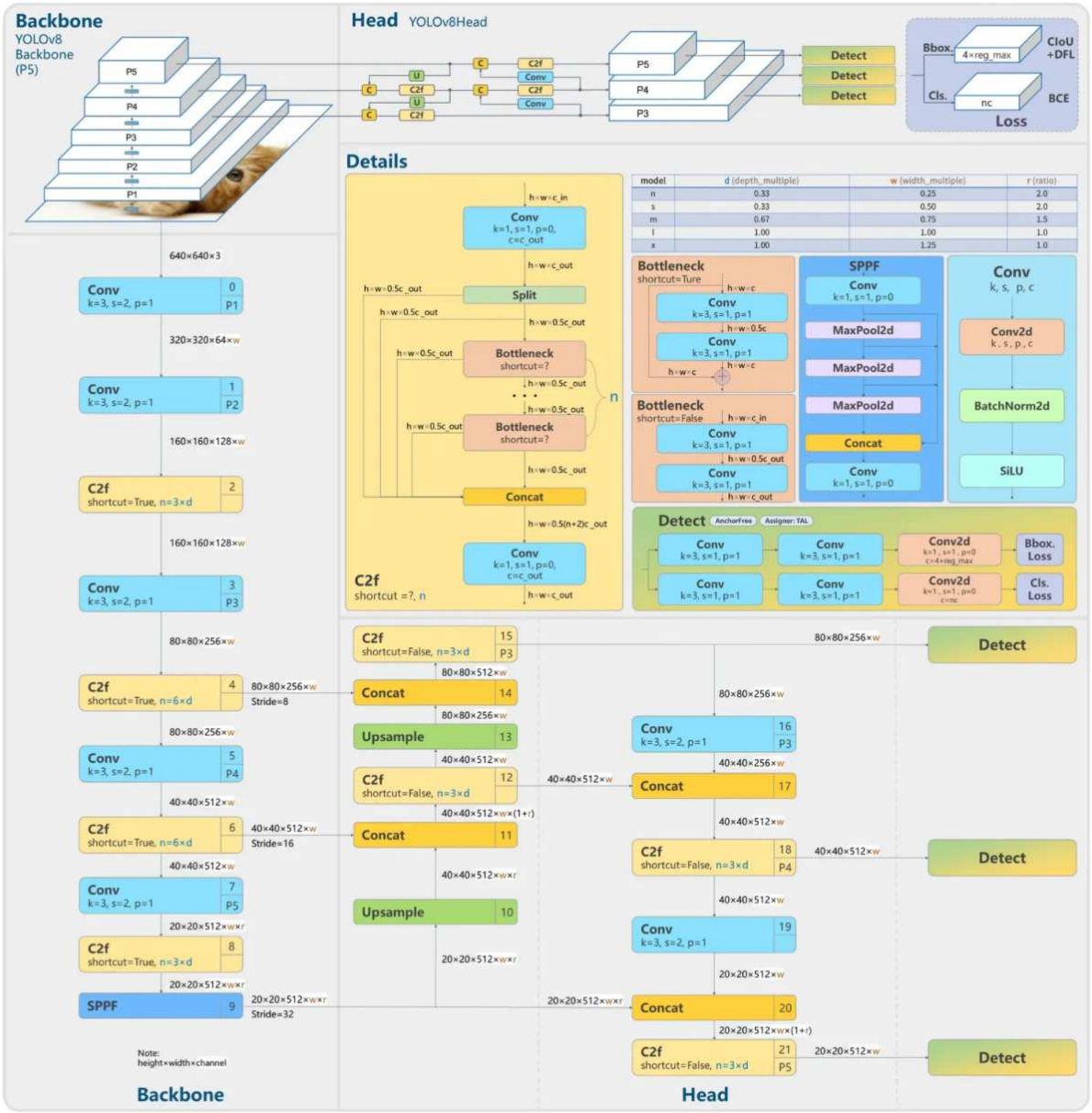
3.2 YOLOv8

Time dolazimo do YOLOv8, koji je korišten u ovome radu. U siječnju 2023., Ultralytics ga je predstavio te je označio veliki pomak u seriji YOLO modela jer je korisnicima mogao pružiti inovativan raspon poboljšanja i svestranih mogućnosti. Ovisno o potrebama aplikacije, predstavljeno je pet različitih pod verzija: YOLOv8n (nano), YOLOv8s (mala), YOLOv8m (srednja), YOLOv8l (velika) i YOLOv8x (ekstra velika).

YOLOv8 je izgrađen na temelju YOLOv5 verzije, iako u arhitekturi mreže koristi Darknet53 strukturu, određeni dijelovi strukture su poboljšani. Tako je, na primjer, modul C3 u mreži za ekstrakciju značajki zamijenjen je modulom C2f s rezidualnom vezom, koja uključuje dva djelomična uska grla kroz konvolucijske slojeve. [11] Kao rezultat ovih promjena, znatno se

smanjio broj potrebnih parametara i tenzora, što znači da je i zauzeće memorije bilo znatno manje.

Ukratko, model je podijeljen u tri glavne komponente: okosnicu (kralježnicu), vrat i glavu. Okosnica je odgovorna za izdvajanje značajki iz ulazne slike, a YOLOv8 koristi različite okosnice, neke od njih su CSPDarknet53 i EfficientDet, kao što je i bilo spomenuto. [12] Prikaz arhitekture se može vidjeti na Slika 6. Na jednostavniji način; okosnica iz slike izdvaja značajke tako što u ulaznom sloju (slici) traži uzorke, kao na primjer rubove ili promjene u teksturi, i kreira mapu značajki. Vrat međusobno povezuje okosnicu i glavu. On služi da slaže piramidu značajki, prikazanu u gornjem lijevom kutu slike ispod, kao slojeve ima mape značajki koje dobije od okosnice. Također, vrat je zaslužan da mreža može detektirati objekte različitih veličina. Uz to još i pazi da se uz objekte gleda i područje oko njih te se i to uzima u obzir kada se računa točnost detekcije. Iz tih razloga, ovaj dio modela je odgovoran za poboljšanje u brzini modela, što u nekim slučajevima može rezultirati lošijom generalizacijom. Zadnja komponenta je glava. U ovom koraku se zapravo generiraju informacije koje će definirati granične okvire. Tu se određuje ocjena pouzdanosti za svaki definirani okvir i kolika je vjerojatnost da je objekt unutar okvira. [13]



Slika 6. YOLOv8 arhitektura[1]

Arhitektura YOLO model ase sastoji od 225 slojeva (engl. *layers*). Od toga sloj može biti konvolucijski, C2f, brzo prostorno piramidalno sažimanje (engl. *Spatial Pyramid Pooling Fast*, SPPF) I sloj detekcije (engl. *Detect*).

Kod izgradnje konvolucijske mreže konvolucijski sloj je najvažniji. Konvoluciju, kao matematički izraz, je operacija definirana nad pikselima slike i matrica koja se naziva jezgra. Izlaz konvolucije je mapa značajki (engl. *feature map*). Što je bitno naglasiti je da kod konvolucijskog sloja, interakcija se između samo jednog dijela ulaznog sloja. Na primjer ne

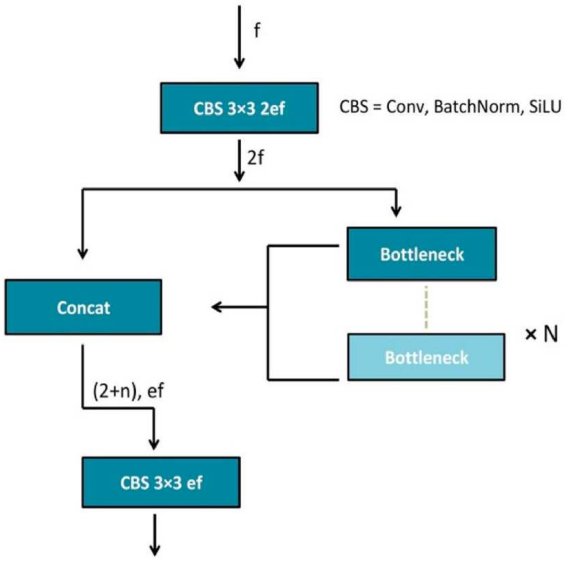
[1] Izvor: https://blog.roboflow.com/whats-new-in-yolov8/

uzimaju se u obzir svi pikseli ulazne slike nego samo dio njih koji se nalaze „ispod“ jezgre. Matrica se kliže po ulaznim podacima sloja, pomiče se za određeni korak (engl. *stride*). [15]

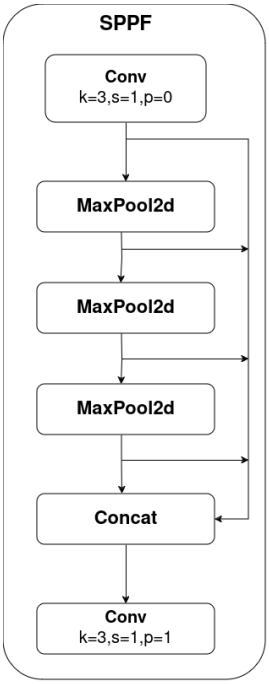
Sljedeći sloj koji se pojavljuje u arhitekturi YOLO modela je C2f sloj. Struktura C2f sastoji se od dva bloka koji provode konvoluciju (time se smanjuje broj kanala u mapi značajki) i tri konvolucijska bloka uskog grla. Strukturni blokovi uskih grla, sastoje se od dva konvolucijska bloka i kontakta koji ih povezuje. C2f sloj optimizira mrežnu strukturu reguliranjem gradijentnih ruta, poboljšavajući sposobnost obuke. Na Slici 8. se može vidjeti prikaz strukture C2f sloja. [21] Razlika C3 sloja, koji je bio u prethodnim verzijama, i C2f sloja je da C2f koristi `3 x 3` konvoluciju umjesto `6 x 6`.

Desni dio, Slika 7., prikazuje arhitekturu SPPF sloja. Dakle, ovaj sloj se sastoji do jednog konvolucijskog sloja te zatim slijede tri sloja maksimalnog sažimanja (engl. *Maxpool*). Iznlazna mapa značajki iz svakog od slojeva maksimalnog sažimanja se povezuje I prolazi kroz još jedan konvolucijski sloj. SPPF sloj omogućuje neuronskim mrežama rad sa slikama različitih rezolucija jer slike prolaze kroz slojeve različitih razinama granularnosti. [22]

Zadnji sloj koji je potrebno navesti je sloj detekcije. Ovaj sloj se nalazi u komponenti glave u strukturi YOLO modela. Postoje dva puta, prvi je za predviđanje graničnih okvira, a druga je za predviđanje razreda objekta unutar okvira. Obje putanje se sastoje od dva konvolucijska bloka nakon kojih slijedi jedan sloj konvolucije koji daje gubitak graničnog okvira i gubitak klase. [22]



Slika 7. Arhitektura C2f sloja[2]



Slika 8. Arhitektura SPPF sloja[3]

Dakle, YOLOv8 podijeli ulaznu sliku u mrežu ćelija, zatim za svaku ćeliju, predviđa granični okvir i vjerojatnost svake klase. Glavna razlika u pod verzijama YOLOv8 je veličina i kompleksnost modela. Što je model veći, time i kompleksniji, to je točniji, no u isto vrijeme je i sporiji. Dok su manji modeli manje kompleksni i manje točni, ali su znatno brži.

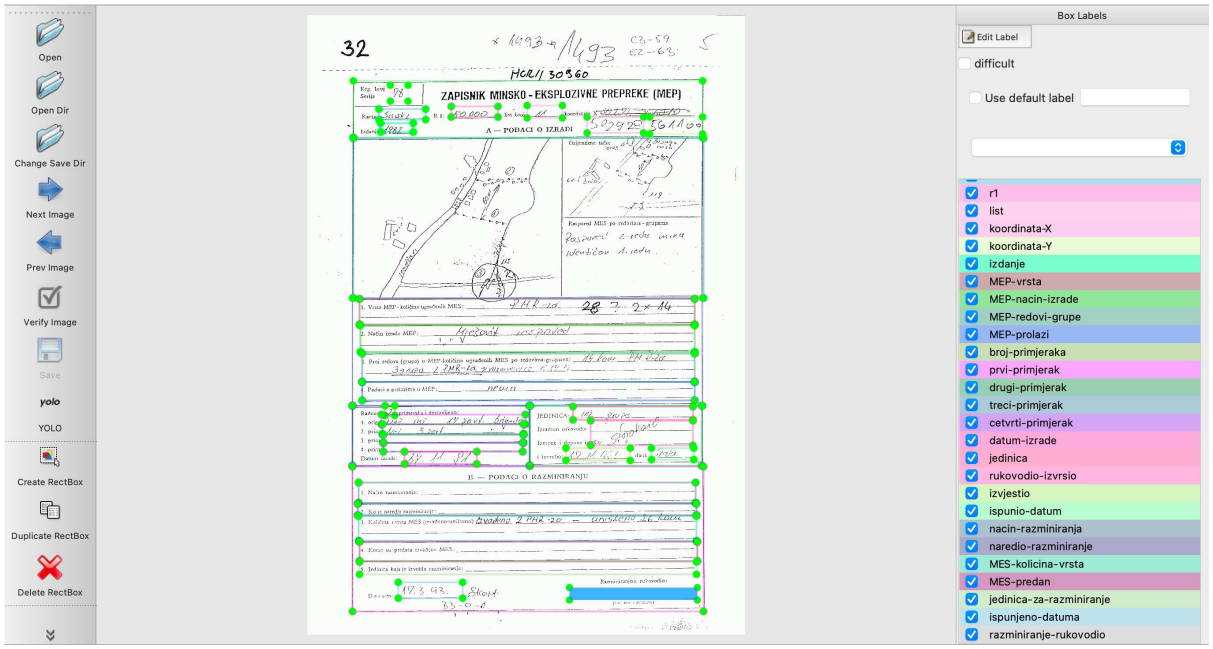
Kada se skup podataka priprema za korištenje u YOLO modelu, potrebno je za svaki podatak/sliku napraviti njezinu istoimenu datoteku (u ovom slučaju to će biti tekstualna datoteka) u kojoj su definiraju granični okviri koji su na toj slici označeni. Struktura je pokazana na Slika 10. Kako se označavaju granični okviri i njihovo povezivanje sa slikom je opisano u nastavku.

[2] Izvor: Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. *TeleStroke: real-time stroke detection with federated learning and YOLOv8 on edge devices*. Journal of Real-Time Image Processing, *21*(4), (2024). str. 121.
[3] Izvor: YOLOv8 architecture explained (2024., ožujak). Poveznica: https://abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5 . Pristupljeno: 29. Kolovoza 2024

3.3 LabelImg i granični okviri

Kao što je već prije spomenuto, YOLO koristi granične okvire pomoću kojih izvršava detekciju objekata. Svaki okvir sastoji se od pet komponenti: x, y, w, h i pouzdanost. Uređeni par (x,y) predstavlja centar tog okvira s obzirom na granice ćelija. Širina i visina se pak odnose na dimenzije slike u potpunosti. Također uz svaki okvir stoji i oznaka razreda, oznaka (*engl. label*), kojoj pripada.

Označavanje podataka je proces u kojemu se podacima dodaju meta podaci, da bi ih algoritmi za strojno učenje mogli razumjeti. Ovo se tipično radi ručno, tako da se za svaku sliku dodaju oznake, anotacije ili okviri. Pošto ovaj proces zahtjeva da se prođe cijeli skup podataka, što može biti naporan posao jer su skupovi uglavnom veliki, postoje posebni alati koji se koriste. Jedan takav alat je program `labelImg`. To je grafički program otvorenog koda, pomoću kojega je moguće označavati okvire.



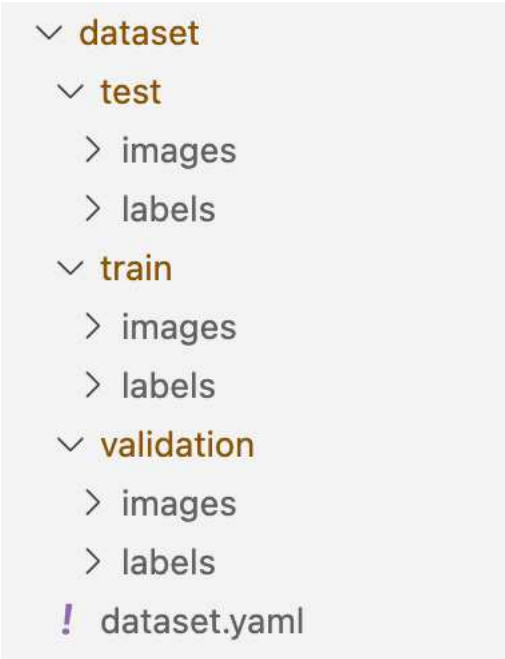
Slika 9. labelImg sučelje s primjerom

Slika iznad prikazuje kako se u `labelImg` označavaju okviri te njihove oznake, sa desne strane su nabrojane od prije definirane oznake u tom skupu podataka. Moguće je odabrati

format za izlaznu datoteku, no za potrebe YOLO modela to je bila samo tekstualna datoteka u kojoj su popisani okviri u formatu definiranom na početku poglavlja.

Skup podataka je podijeljen na skup za treniranje, skup za validaciju I skup za testiranje. Točna struktura je opisana na Slika 10. U direktorijima imena `images` se nalaze originalne slike minskih zapisnika u PNG formatu, dok se u `labels` direktorijima nalaze rezultati označavanja okvira pomoću `labelImg`.

Uz te direktorije se nalazi još i `dataset.yaml` datoteke. To je datoteka u kojoj je zapisana konfiguracija skupa podataka. U njoj su napisane putanje do sva tri podskupa podataka te popis svih oznaka i njihova imena. Ova datoteka je potreba YOLO modelu da može povezati informacije iz tekstualnih datoteka u `labels` direktoriju s nazivima oznaka.



Slika 10. Raspored skupa podataka

4. Treniranje i validacija modela

Značajna prepreka kod treniranja modela je računalna složenost jer svako pokretanje zahtijeva značajne računalne resurse, poput snage, memorije i vremenske složenosti. Kompleksnost arhitekture modela koji se koriste u detekciji objekata je složena, kao što je u poglavlju prije napisano, može sadržavati i nekoliko stotina konvolucijskih slojeva.

Kako je u radu riječ o slikama s kojih se pokušava očitati tekst, jako je važna njihova veličina jer ako se previše spljošte onda će se i njihov tekst izobličiti, što bi moglo dovesti do krivih rezultata. A opet s druge strane, za model je bitno da su sve slike istih veličina, što u skupu podataka nije bilo inicijalno zadovoljeno.

Tako je bilo potrebno definirati veličinu slike koju će model prihvaćati. Napravljen je kompromis između veličine slike i brzine modela. Istraživanje po internetu za već prije trenirane modele je pokazalo da je 800 piksela optimalna veličina, dok su slike iz skupa podataka bile većih dimenzija. Iz tog razloga je model treniran na tri različite konfiguracije; s veličino slika `800 x 800` (kao postojeći optimum), `1024 x 1024` (što se i dalje smatra optimalnim) te `1280 x 1280` (što je najviše što m2 čip može podnijeti, to jest da se do kraja izvrši).

Treniranje se isprobalo na više modela te se uspoređivalo kako će ulazna veličina slike, utjecati na preciznost OCR-a. U nastavku su opisane isprobane konfiguracije modela.

4.1 Treniranje modela

Zajedničke komponente modela koji je bio treniran su

1. Python verziju 3.11.5
2. Ultralytics YOLOv8.2.78
3. M2 čip
4. Optimizator: Adam

Primjer koda za treniranje, jedne od verzija modela, je naveden na slici Kod 1. Kao parametri metode `train` se postavlja putanja do skupova podataka koja je definirana u yaml (Yet Another markdown language) formatu, `dataset.yaml` datoteke, u kojoj su napisane svi razredi i njihove oznake te putanje to skupova za treniranje, validaciju i testiranje, kao što je u poglavlju prije bilo definirano. Zatim, pošto je model treniran na M2 čipu parametar `device` se postavlja na vrijednost `mps`. Parametar `batch` definira veličinu grupe, to jest koliko uzoraka iz skupa podataka se zajedno obrađuje odjednom. To je postavljeno na 8, s obzirom na memoriju i čip. Epoha predstavlja jedan cijeli prolaz podataka za treniranje kroz model te u kontekstu ovoga rada je odabrano 100 epoha. Sljedeći parametar je veličina slike (`imgsz`) koja će biti na ulazu u model, to je parametar za koji se testiralo više vrijednosti. Postavljena vrijednost u YOLO modelu je `800 x 800`, to je prva isprobana inačica. Sljedeća vrijednost s kojom je model bio treniran je `1024 x 1024`, što je preporučena vrijednost za M2 jer se postiže optimalna brzina. Zadnja isprobana vrijednost je `1280 x 1280`, to je najveća vrijednost koja se može postaviti kada je riječ o M2 čipu.

Parametar koji nije specifično definiran, nego se koristila unaprijed postavljena vrijednost je optimizator i korišten je Adam (Adaptive Moment Estimation). Koristi stohastički gradijent, što znači da se unutar epohe za svaki primjer težine ažuriraju zasebno.

```
from ultralytics import YOLO
import argparse


model = YOLO("yolov8n.pt")

parser = argparse.ArgumentParser()
parser.add_argument('--batch', type=int, default=8)
parser.add_argument('--epochs', type=int, default=100)
parser.add_argument('--imgsz', type=int, default=800)

args = parser.parse_args()
results = model.train(
      data="./dataset/dataset.yaml",
      device="mps",
      batch=args.batch,
```

```
        epochs=args.epochs,
        imgsz=args.imgsz,
        verbose=True,
        visualize=True
)

metrics = model.val()
```

Kod 1. yolo_training.py

4.2 Validacija YOLO modela

Nakon što je treniranje bilo završeno, odrađena je evaluacija modela. To je napravljeno pomoću poziva metode `.val()`, što se može vidjeti u zadnjoj liniji koda na slici Kod 1.

Naravno, u svakom prolazu epohe unutar procesa treniranja se odrađuje validacija na validacijskom skupu, na temelju toga se ažuriraju težine, no to nije isto kao i evaluacija modela.

Validacijom modela se provjera je li model koji je treniran dovoljno točan za sustav koji predstavlja. [14] Kod validacije modela riječ je o mjeri pouzdanosti, vjerojatnosti da okvir sadrži objekt. Kod detekcije objekata najviše zastupljena mjera je srednja prosječna preciznost (engl. *mean Average Precision (mAP)*). To je uprosječena vrijednost srednjih preciznosti izračunatih za sve klase. [15] Validacijska metoda vraća sljedeće vrijednosti: mAP50 i mAP50-95 te još prosjeke nekih drugim mjera, što će biti spomenuto i objašnjeno kasnije. [14]

Mjera mAP50 pokazuje prosječnu preciznost, koristeći Jaccardov indeks (IoU) od 0.5. Prvenstveno, ova mjera točnosti uzima u obzir samo one okvire koje je lagano detektirati. Ovdje su to recimo okviri koji su na sredini papira, tamo je najmanja zastupljenost šuma, uz to su i većih dimenzija. Mjera mAP50-95 je općenitija od mAP50. Ova mjera računa preciznost ali za više pragova, točnije u rasponu od 0.5 do 0.95, otkud i naziv mjere. Time pokazuje sveobuhvatan pregled uspješnosti modela na različitim detekcijama okvira, bili oni teži ili lakši za pronaći i odrediti.

Kod validacije modela je izričito važno napomenuti i Jaccardov indeks (eng. Jaccard index, Intersection over Union). On mjeri koliko je preklapanje između predviđenog graničnog

okvira, dakle koji je model izračunao, I okvira koji je bio definiran na ulazu modela. [16] Jaccardov indeks se koristi kod mAP mjera, to je zapravo ovaj prag koji se postavlja I od kojega se mjeri.

5. OCR

Optičko prepoznavanje znakova (engl. *Optical Character Recognition*, OCR) je područje računalnog vida koje je namijenjeno, kao što i sam naziv kaže, klasifikacijom znakova na slikama. [17] S obzirom da je područje još u razvoju, postoji više OCR softvera, neki od njih su Google Cloud Vision OCR, Microsoft Azure OCR, Amazon Textract i tako dalje. Uz njih postoji I Tesseract OCR. To je besplatni softvere otvorenog koda, što ga čini izrazito pristupačnim I jedan je od razloga zašto se baš taj softvere koristi u ovome radu. Također, Tesseract OCR je jedan od najčešće korištenih OCR softvera I poznat je po tome što identificira različite jezike s velikom točnošću. [18] Python-tesseract (`pytesseract`) je omotač za Googleov Tesseract-OCR-a.

YOLO model, opisan gore, će se koristiti da odredi granične okvire i napiše njihove koordinate i dimenzije (četiri od pet parametara šta su izlaz modela) te će se onda iz tih okvira čitati tekst koji će pohraniti u json datoteku.

Na slici Kod 2 je prikazan poziv metode `image_to_string` koji se koristi za čitanje tekst iz `cropped_image` što je dio originalne slike samo izrezan po dimenzijama određenog okvira za taj tekst. Uz to kao drugi parametar šalje se jezik na kojem je tekst napisan da bi ga OCR znao pročitati, ovaj parametar se definira kod poziva skripte, u narednoj liniji (engl. *command line*). Za slučaj ovoga modela, taj parametar je definiran za hrvatski jezik (vrijednost mu je `hrv`) jer je to jezik koji je korišten u zapisnicima.

```
text = pytesseract.image_to_string(cropped_img,
lang=args.lang)
```

Kod 2. pytesseract poziv

U izlaznoj datoteci rezultati će biti pohranjeni u formatu razred-tekst, gdje je razred ime za određeni granični okvir, a tekst je vrijednost koju je OCR dobio kada je primijenjen na taj specifični okvir. Potrebno je napomenuti da se za razred „skica“ sadržaj ne čita nego se u cjelokupnosti pohranjuje u base64 formatu u izlazni json. Base64 je metoda kodiranja binarnih podataka kao ASCII teksta. Primjer jednog dijela izlazne datoteke može se vidjeti na

slici Kod 3. Vidi se da dijelovi tekst nisu točno, to jest napisane su riječi koje ne postoje ili nemaju smisla, to bi se riješilo odgovarajućom post obradom podataka, no o tome kasnije.


```
{
        "class": "MEP-prolazi",
        "text": "4. Podaci o prolazima u MEP:"
    },
    {
        "class": "prvi-primjerak",
        "text": "LAP. IR. VODA"
    },
    {
        "class": "koordinata-X",
        "text": "5020,"
    },
    {
        "class": "nacin-razminiranja",
        "text": "3\n\nNa\u010din razminiranja\n\n#4. 2 midi
ATA. FRANA lira a\u201c"
    },
    {
        "class": "karta",
        "text": "SIZAK - ISTok .."
    },
    {
        "class": "datum-izrade",
        "text": "2%. os. 4441"
    }
```


Kod 3. json izlaz zapisnika 30520

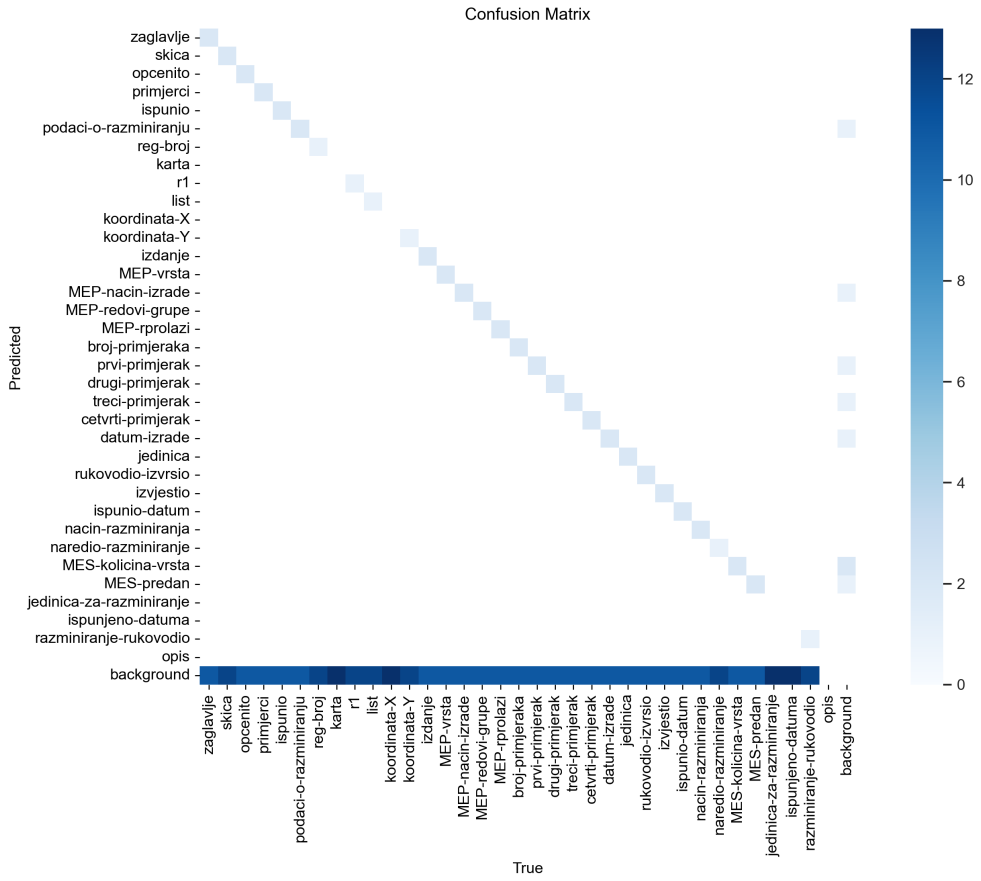
6. Rezultati

Nakon što je model istreniran, potrebno je odrediti jesu li dobiveni rezultati vjerodostojni za skup podataka, to se izvršava pomoću mjera vrednovanja. Mjere vrednovanja je metoda pomoću koje se može kvantificira točnost (ili pogrešku) modela. Tim mjerama se provjerava je li izlaz dobiven modelom (engl. *predicted value*) u skladu s očekivanim izlazom (engl. *true value*).

Matrica zabune (engl. *confusion matrix*) je kvadratna matrica koja prikazuje koliko puta je predviđena oznaka bila ista kao u očekivana, a koliko puta nije. Kada se predviđena I očekivana vrijednost podudaraju onda je to stvarno pozitivan primjer (engl. *true positive*), TP. Druga moguća situacija je da je očekivana vrijednost jedna, a model procijeni da je na tom mjestu okvir za drugi razred, koji se ne nalazi u tom području, onda je riječ o lažno negativnom (engl. *false negative*), FN. Također, je moguće I obrnuto, da je riječ o okviru za drugi razred, a model procijeni da je za prvi, to je onda lažno pozitivan (engl. *false positive*), FP, ako se promatra iz perspektive prvoga razreda. Zadnja mogućnost je da primjer ne pripada prvom razredu I model procijeni da to zaista nije prvi razred, onda je riječ o stvarno negativnom (engl. *true negative*), TN.

S obzirom da u modelu koji je opisan u ovome radu, postoji 33 razreda za koje su se predviđali granični okviri, matrica zabune će biti dimenzije `33 x 33`. Primjer matrice zabune za model sa slikama veličine 1280 piksela je prikazana na Slika 11. Iz prikazane slike može se vidjeti da za svaki razred drugo najviše podudaranja ima za točno pozitivne (to su kvadrati na glavnoj dijagonali). Za razred pozadina (engl. *background*), koji model samo inicijativno dodaje, ima najviše podudaranja po svim razredima. Prvenstveno razredi koji su se nalazi na vrhu ili dnu papira, imaju više primjeraka gdje je okvir zamijenjen za pozadinu. Jedan od glavnih razloga tome je što su oštećenja na tim mjestima papira bila najvjerojatnija pa je time I teže bilo za procijeniti granice okvira.

Za sve tri konfiguracije modela, matrica zabune je poprilično ista, uvijek je glavna dijagonala najzastupljenija, samo je njezina zasićenost promijenjena.



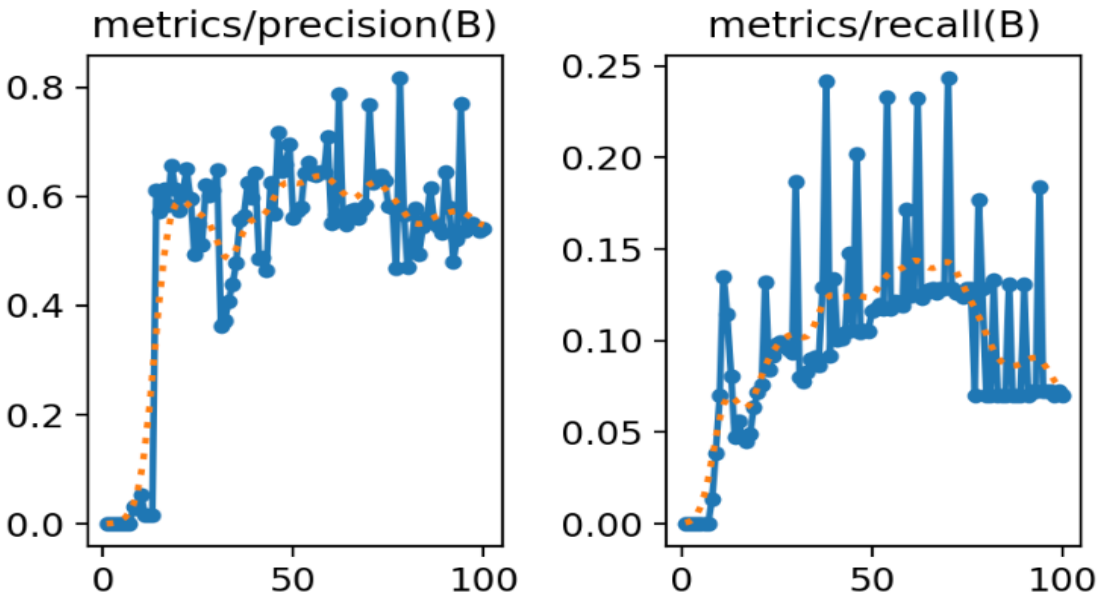
Slika 11. Matrica zabune za model sa slikama veličine 1280 piksela

Sljedeća mjera koja je izračunata i važna napomenuti je preciznost (engl. *precision*). Preciznost je definirana kao udio stvarno pozitivno, TP, određenih okvira u sklopu svih označenih okvira koje je model označio pozitivno, TP I FP, u smislu da odgovaraju kom razredu. U idealnoj situaciji je preciznost jednaka 1, točnije svi primjeri koje je model označio pozitivnima doista i jesu pozitivni, pripadaju tom razredu.

Još jedna mjera je odziv (engl. *recall*), poznato još i kao stopa stvarno pozitivnih. Ova mjera govori o udjelu stvarno pozitivno određenih okvira, TP, u skupu svih primjera koji su određeni kao taj okvir, dakle TP i FN zajedno.

Ove dvije mjere su pokazane kroz grafove (Slika 12), to jest kako se njihova vrijednost mijenjala kroz sve epohe. Prvo za preciznost, vidljivo je da preciznost raste, što je i bilo za očekivati jer što model više epoha prođe to je više informacija prikupio i točnije može određivati granične okvire. Najveća postignuta vrijednost za preciznost je 0.81789 u 78. epohi. Također se može vidjeti da preciznost, u kasnijim epohama, varira između 0.5 i 0.6.

Što se odziva tiče, vidi se da je njegova vrijednost dosta niža od vrijednosti preciznosti. Najveća vrijednost za odziv je 0.24363. u 70. epohi. Odziv, u početnim epohama raste, usporeno ali postepeno, dok onda pri kraju vrijednost mu opada.



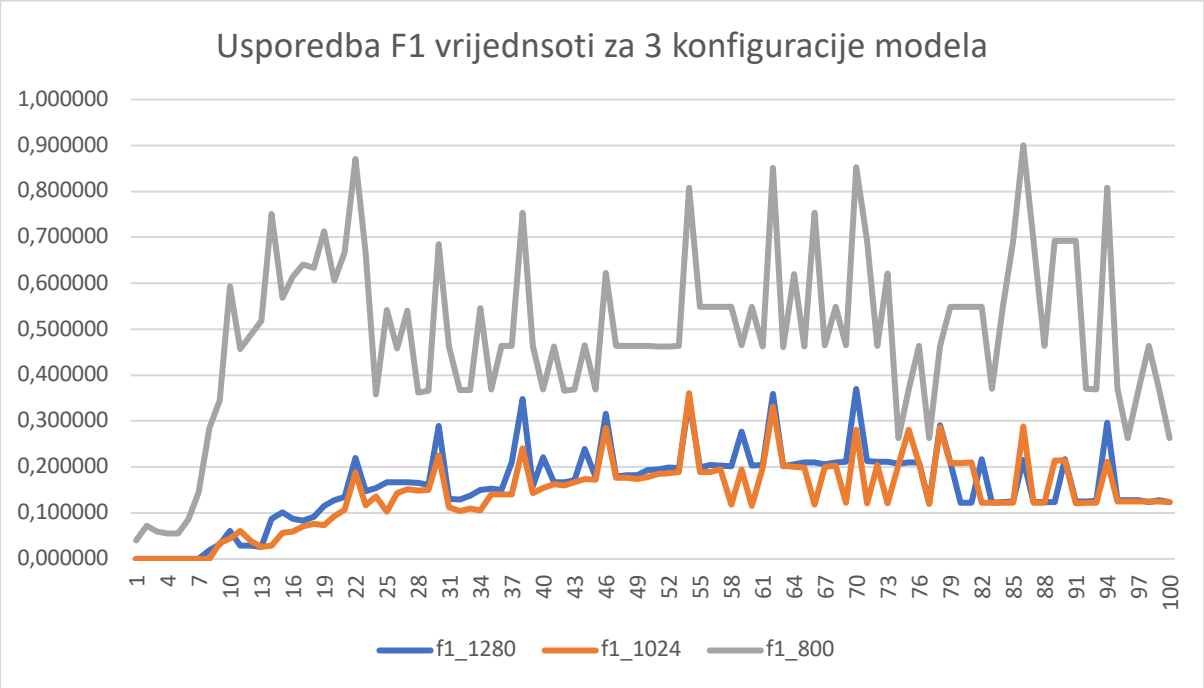
Slika 12. Preciznost i odziv modela sa slikama veličine 1280 piksela

U praksi su ove dvije mjere često izravno suprotstavljene: ako model postavimo tako da je odziv velik, onda je tipično da će to rezultirati nižom preciznošću, i obrnuto, model s visokom preciznošću će u većini slučajeva imati niži odziv. Zbog toga, kada je riječ o rezultatima modela i njihovoj točnosti, svakako je potrebno navesti vrijednosti obje mjere. Međutim, u većini slučajeva se želi izraziti samo jedan broj kojim će se odrediti točnost modela. Mjera vrednovanja koja radi upravo to je mjere F1 (engl. *F1 score*). [19] Ova mjera se računa po formuli

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R}$$

Uz pomoć te formule i već od prije izračunatih preciznosti i odziva za sve tri konfiguracije modela kroz svih 100 epoha, napravljen je sljedeći graf. Na grafu se vidi da je F1 mjera kroz sve epohe za modele sa slikama veličine 1280 i 1024 piksela, gotovo jednaka. Za razliku od toga F1 mjera za model sa slikama veličine 800 piksela je znatno veća, dapače teži prema vrijednosti 1 što je vrijednost koja se želi postići. Razlog tome je da su se modeli sa slikama

većih dimenzija te još kao posljedica maloga skupa podataka i postojanja šuma u podacima, prenaučio. Isto tako za sve tri konfiguracije može se primijetiti da F1 vrijednost u početku raste, onda malo stagnira, a zatim u jednom trenutku počinje padati. To znači da je model postao prenaučen. To je još dodatno podržano time da su i preciznost i odziv svoj maksimum postigli prije 80. epohe. Može se zaključiti da je to neki optimalni broj epoha koji je bilo potrebno koristiti.



Slika 13. Mjera vrijednosti F1 za modele

Ovo su bili rezultati treniranja modela, što se tiče validacije modela, u poglavljima prije je bilo spomenuto da je kod validacije modela, najzastupljenija mjera srednja prosječna preciznost (mAPI). U nastavku na to, pozivom metode `.val()` su one i izračunate. U Tablica 1 može se vidjeti da je mAP50 najveći za slike dimenzija 800 piksela. Dakle model je tamo detektirao najviše „lakih“ okvira. Ista situacija je i za vrijednosti mjere mAP50 -95. Isto tako, može se vidjeti da je u oba slučaja vrijednost mjere kod slika s dimenzijama 800 piksela duplo i više puta veća nego kod slika dimenzije 1024 piksela.

Još jedna mjera koja je spomenuta u tablici u nastavku i koja je izračunata u modelu je mjera fitnesa (engl. *fitness*). Fitnes je vrijednost koju želimo da je što veća, jer na temelju nje se određuje koje težine će se koristiti za model. Fitnes se računa u svakoj epohi. Zaključno po

rezultatima iz tablice, konfiguracija modela gdje su veličine slika `800 x 800`, pokazuje najbolje rezultate.

mjera	Imgsz=1280	Imgsz=1024	Imgsz=800
preciznost	0.590058	0.471522	0.710803
odziv	0.128441	0.065449	0.294332
mAP50	0.202980	0.134802	0.307544
mAP50-95	0.138106	0.096502	0.262607
fitnes	0.144593	0.100332	0.267100

Tablica 1. Usporedba mjera za tri konfiguracije modela

Ovo su bili rezultati YOLO modela. Za rezultate OCR modela nije bilo izračunatih mjera već se kroz rezultate u json datoteci može vidjeti je li model točno pročitao riječ ili nije. Kao što se na slici Kod 3 već moglo vidjeti, većina teksta je približno pročitana. Možda riječi nemaju skroz smisla ali iz konteksta i logičkim zaključivanjem se može naslutiti koja bi vrijednost trebala pisati. Isto tako, vidi se u rezultatu razreda `MEP-prolazi` da je model uspješno i potpuno točno pročitao tekst koji je bio natiskan prije ispunjavanja. Da su zapisnici čitljivije pisani i da su nedavno ispunjavani, vjerojatnost da OCR ispravno pročita riječi bi bila znatno veća.

7. Diskusija

Što su slike modela veće, parametar `imgsz` na ulazu modela, time je model složeniji. Tako je na primjer za treniranje najjednostavnijeg modela sa slikama veličine `800 x 800` trajalo 2 sata i 30 minuta, dok je treniranje sa slikama veličine `1280 x 1280` trajalo 5 sati. U omjerima vremena treniranja modela, ovo se smatra i brzim treniranjem, ali isto jedan od razloga tome je da je skup podataka bio relativno mali, samo 83 minska zapisnika, dok u većini slučajeva skupovi podataka su i do nekoliko tisuća primjeraka u skupu.

Kao što je u rezultatima bilo raspravljano, pokazalo se da je model gdje su slike bile najmanje (800 piksela), po mjerama točnosti imao najtočnije rezultate, sve mjere su bile veće od vrijednosti istih mjera za ostale dvije konfiguracije. Što je ispalo kontradiktornim originalnim očekivanjima i predikcijama. Ali je pokazatelj da je došlo do prenaučenosti modela te bi bilo potrebno smanjiti broj epoha, na primjer do trenutka prije nego vrijednost od F1 krene ponovo podati.

Isto tako iako su za konfiguraciju modela sa slikama veličine 800 piksela, mjere točnosti bile najbolje, kod rezultata OCR situacija je bila obrnuta. Što je i bilo za očekivati jer što je veća slika to tekst na njoj više čitljiv pa bi i OCR-u trebalo biti lakše za pročitati. OCR je na rijetko kojem mjestu u okviru uspio pročitati tekst sa potpunom točnošću.

7.1 Buduća istraživanja

Za daljnji nastavak razvoja i poboljšanja ovoga modela je mogućnost post procesiranje podataka. Dakle da se napravi rječnik sa riječima koje se prvenstveno koriste kod humanitarnog razminiranja. Da kada se dobije rezultat iz OCR-a u json formatu da se napiše i zatim pokrene program koji bi pregledao json i riječi koje nemaju smisla pretvorio u nešto što je smisleno i više vjerojatno da je pisalo u tom polju. Primjerice na slici Kod 4 poznato je da je riječ o zapisniku s područja Siska te se može pretpostaviti da bi prva riječ u polju `text` trebala biti Sisak. Kada bi se to ispravilo, daljnja obrada ovih podataka bila bi lakša i točnija, što bi osiguralo još veću sigurnost i pouzdanost u model.

```
{
        "class": "karta",
        "text": "SIZAK - ISTok .."
}
```


Kod 4. json izlaz za zapisnik 30706

Kao što je i u uvodnom poglavlju napisano, u ovom radu se specifično fokusirano na jedan tip minskih zapisnika, a napomenuto je da postoje više tipova formulara koji se mogu ispunjavati ili koji su iz nekih drugih izvora. Kada bi se ovaj model primijenio na te „drugačije“ zapisnike, rezultati bi bili poprilično loši. U toj situaciji, potrebno bi bilo trenirati novi model koji će se učiti specifično na tom formatu, no onda naravno treba ponovo označavati granične okvire i ponoviti ostale korake koji su slijedili iza toga.

Iako je pytesseract jedan od najzastupljenijih OCR-ova i ima izrazito dobre rezultate kod čitanja skeniranih dokumenata, uglavnom je uvijek riječ o ne ručnom pisanim dokumentima. Ovaj rad je još jedan pokazatelj da je pytesseract lošiji izbor za ručno pisane dokumente. U drugu ruku, na primjer EasyOCR vraća vjerojatnost za svaku riječ koju pročita. To bi zajedno s idejom o post procesiranju podataka dalo vjerojatnije bolje rezultate, kada bi se još koristio i rječnik koje je specificiran za temu humanitarnog razminiranja.

Jednom kada na izlazu dođe json datoteka, informacije iz nje je potrebno dalje pohraniti da bi mogle biti dostupne ostalim korisnicima. To se može napraviti pomoću baze podataka u koju bi se direktno zapisivala polja iz teksta svakoga razreda. Time bi se olakšala razmjena informacija između različitih izvora. Isto tako, dio baze mogao bi biti dostupan i javnosti, dok bi neki podaci ostali tajni, ovisno o njihovoj osjetljivosti. Uz to baza bi se mogla povezati na Internet stranice gdje bi mogle postojati karte širih područja pa bi se na njih nadodale informacije pročitane pomoću modela iz minskih zapisnika. Lako bi se nadopunile i ažurirale te stranice kada bi došle nove informacije.

Alternativa, za bazu podataka je ideja da se zapisnici i njihove informacije pohranjuju na lance blokova (engl. *blockchain*). Korištenje blockchain tehnologije za pohranjivanje informacija u evidenciju minskih polja može značajno poboljšati učinkovitost, dostupnost i sigurnost operacija humanitarnog razminiranja, Ovaj pristup bi ponudio brojne prednosti kao što su decentralizirana i sigurna pohrana, zaštićenost podataka, povjerenje i transparentan pristup podacima implementacijom pouzdane blockchain tehnologije u području

razminiranja. Skalabilnost, transparentnost, pouzdanost, integracija s postojećim sustavima, zakonska usklađenost i isplativost pitanja su koja treba dodatno istražiti. [20]

Još jedan napredak u budućim istraživanjima je da se razred `skica` koji je trenutno samo pohranjen u base64 formatu, mogao isto koristiti za detekciju objekata, to jest da se točno detektiraju objekti koji su u slici (drveće, kuće, vinogradi i tako dalje). To bi otvorilo mogućnost da se zapisnici međusobno mogu povezivati u slučaju da je riječ o istome području, a možda do sada nisu bili spojeni.

Zaključak

U humanitarnom razminiranju veliku ulogu igraju dobri i precizni senzori, pouzdana oprema i učinkovite tehnike obrade podataka da bi se osigurala sigurnost i učinkovitost. Evidencija o minskom području, koja dokumentira lokaciju i vrste minsko-eksplozivne prepreke, MEP, mora se dobro održavati i u potpunosti popuniti za buduću upotrebu. Tradicionalno su ti zapisi bili ručno ispunjavani i pohranjeni na različitim lokacijama, ali napredak računalnog vida sada omogućuje automatsko prepoznavanje i digitalizaciju rukom pisanih tekstova, čineći te zapise pristupačnijim i lakšim za upravljanje. Model YOLO, korišten u ovom radu, treniran je za otkrivanje područja interesa na skeniranim dokumentima i označen graničnim okvirima, koji se zatim čitaju pomoću tehnologije optičkog prepoznavanja znakova, OCR. Izdvojeni tekst sprema se u JSON formatu, modernizirajući i poboljšavajući upravljanje zapisima o minskom području.

Nakon treniranja modela, napravljena je evaluacija, koristeći adekvatne mjere pouzdanosti, da bi se pokazalo je li model ispravan i točan za zadani skup podataka. Rezultati su pokazali matricom zabune da s 33 razredima, da je većina predviđenih okvira za razrede istovjetna očekivanim područjima gdje su okviri i trebali biti, ali problemi nastaju s razredima koje se nalaze na vrhu ili dnu dokumenata zbog mogućeg oštećenja papira. Preciznost i prisjećanje ključne su metrike, s vrhuncem preciznosti u 78. epohi, a odziv sveukupno nižim, što ukazuje na kompromis između ovih metrika. Rezultat mjere F1, koja je harmonijska sredina preciznošću i odziva, pokazao je da je model treniran sa slikama od 800 piksela imao najbolje rezultate, sugerirajući da veće slike i mali skupovi podataka mogu dovesti do pretjeranog prilagođavanja. Konačno, rezultati fitnessa, korišteni za odabir težine modela, potvrdili su da je konfiguracija slike od 800 piksela optimalna.

Neka od budućih unaprjeđenja su na primjer post procesiranje podataka, da se definira rječnik u kojemu su napisane sve riječi koje se često koriste u komunikaciji kod humanitarnog razminiranja te se rezultat json datoteke pregleda uz pomoć tog rječnika I isprave rezultati neispravno pročitanog teksta. Isto tako primjena tehnologije lanaca blokova gdje bi se zapisnici I njihovi podaci mogli sigurno I transaparento pohraniti.

Literatura

[1] Habib, Maki K. "Humanitarian demining: reality and the challenge of technology–the state of the arts." *International Journal of Advanced Robotic Systems* 4.2 (2007): str. 19.

[2] Bhagtani, A., Computer Vision: Intelligent Automationthat Sees, Automation Anywhere, (2021, ožujak), Poveznica: https://www.automationanywhere.com/company/blog/rpa-thought-leadership/computer-vision-intelligent-automation-that-sees , pristupljeno: 15. Srpnja 2024.

[3] Hassan, M. I., Ahmat, D., & Ouya, S. (2024, May). *Technologies Behind the Humanitarian Demining: A Review*. International Conference on Smart Applications, Communications and Networking (2024), pp. 1-7.

[4] Horvat, M., Krmpotić, K., Krtalić, A., Akagić, A. *Bridging Blockchain Technology and Humanitarian Demining: A Novel Concept for Decentralized Storage of Landmine and UXO Locations*. Central European Conference on Information and Intelligent Systems; Faculty of Organization and Informatics: Varaždin (2023). str. 369-375.

[5] Horvat, M., Krtalić, A., Akagić, A., Krmpotić, K., & Skender, S.. *Humanitarian Demining Using Data Observatories and Data Lakes*. CEIA HUMANITARIAN CLEARANCE TEAMWORK, Vodice, (2023), str. 31.

[6] Šnajder, J., *Osnovni koncepti*. Predavanje. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2022.

[7] Marijan, K., *Vrednovanje metoda za detekciju osoba u slikama*. Diplomski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2022.

[8] Alif, M. A. R., & Hussain, M, *YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain*. arXiv preprint arXiv:2406.10139 (2024).

[9] Hussain, M., *Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo*. IEEE Access, 12, (2024), 42816-42833.

[10] Seikavandi, M. J., Nasrollahi, K., & Moeslund, T. B. *Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth*, . (2021, January) Thirteenth International Conference on Machine Vision str. 609-618. SPIE.

[11] Liu, Q., Huang, W., Duan, X., Wei, J., Hu, T., Yu, J., & Huang, J. *DSW-YOLOv8n: A new underwater target detection algorithm based on improved YOLOv8n*. *Electronics*, (2023). str. 12.

[12] Torres, J., *The best YOLOv8 Architecture Explained: Exploring the YOLOv8 Architecture*, (2024, ožujak). Poveznica: https://yolov8.org/yolov8-architecture-explained/ . Pristupljeno: 15. Srpnja 2024.

[13] Pedro J., *Detailed Explanation of YOLOv8 Architecture – Part 1*, (2023, prosinac). Poveznica: https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-part-1-6da9296b954e . Pristupljeno: 15. Srpnja 2024.

[14] *Model Validation with Ultralytics YOLO,* (2024). Poveznica: https://docs.ultralytics.com/modes/val/#how-do-i-validate-my-yolov8-model-with-ultralytics . Pristupljeno 28. Kolovoza 2024.

[15] Hanžek, V., Fino ugađanje konvolucijskih modela za lokalizaciju objekata . Diplomski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2020.

[16] Performance metrics deep dive (2024). Poveznica: https://docs.ultralytics.com/guides/yolo-performance-metrics/#introduction . Pristupljeno: 28. Kolovoza 2024.

[17] Ilijaš, M., Duboke konvolucijske neuronske mreže za raspoznavanje znakova. Diplomska radionica. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2015.

[18] Lozano, L. Tesseract OCR Review: Is it a good OCR software? (2024, siječanj). Poveznica: https://updf.com/ocr/tesseract-ocr-review/ . Pristupljeno: 28. Kolovoza 2024.

[19] Šnajder, J., *Vrednovanje modela*. Predavanje. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2022.

[20] Horvat, M., Krmpotić, K., Krtalić, A., & Akagić, A. (2023). *Bridging Blockchain Technology and Humanitarian Demining: A Novel Concept for Decentralized Storage of Landmine and UXO Locations*. Central European Conference on Information and Intelligent Systems; Faculty of Organization and Informatics: Varazdin, (2023). str. 369-375

[21] Zayani, H. M., Ammar, I., Ghodhbani, R., Maqbool, A., Saidani, T., Slimane, J. B., & Alenezi, S. M.. *Deep Learning for Tomato Disease Detection with YOLOv8*, Engineering, Technology & Applied Science Research, 14(2), (2024) str. 13584-13591.

[22] YOLOv8 architecture explained (2024., ožujak). Poveznica: https://abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5 . Pristupljeno: 29. Kolovoza 2024.

Sažetak

Primjena modela dubokog učenja za prepoznavanje teksta u minskim zapisnicima

Humanitarno razminiranje je proces uklanjanja minsko-eksplozivnih prepreka iz poslijeratnih područja. Prikupljanje informacija i podataka u njima se zapisuje u strukturno definirane zapisnike koji su rukom ispunjavani. Razvojem područja računalnog vida, omogućilo se da se te zapisnike digitalizira i učiniti ih dostupnima u digitalnom formatu, time se postiže brža obrade i analize tih podataka.

U ovome radu s namjerom da se pročita tekst s minskih zapisnika. To se postiglo treniranjem modela za detekciju objekata, YOLO modela, koji je postavio okvire oko teksta na zapisniku te se zatim pomoću OCR pročitao napisani tekst. Model je treniran više puta, koristeći različite veličine ulaznih slika, da se mogu usporediti rezultati. Koristeći određene mjere pouzdanosti, provjerilo se jesu li dobiveni rezultate dobri predstavnici skupa podataka. Također su navedena i moguća poboljšanja koja se u budućim istraživanjima mogu primijeniti.

Ključne riječi: YOLO model, OCR, minski zapisnik, mjera pouzdanosti, računalni vid, Python, granični okvir

Summary

Application of deep learning models for text recognition in mine records

Humanitarian demining is the process of detecting and removing mine-explosive devices from post-war areas. Gathered information and data is recorded in structurally defined records that are filled in by hand. With the development of the field of computer vision, it was possible to digitize these records and make them available in a digital format, thereby achieving faster processing and analysis of data gathered.

Purpose of this thesis is to read the text from these mine records. This was achieved by training an object detection model, the YOLO model, which placed bounding boxes around the text on the log and then read the written text using OCR. The model was trained multiple times, using different sizes of input images, to compare the results. Using certain reliability measures, it was checked whether the obtained results were good representatives of the data set. Possible improvements that can be applied in future research are also discussed.

Keywords: YOLO model, OCR, minefield records, confidence score, computer vision, Python, bounding boxes