Segmentacija i klasifikacija stanica u slikama s mikroskopa

Lendvaj, Mihael

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:426900

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-10**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij računarstva

SEGMENTACIJA I KLASIFIKACIJA STANICA U SLIKAMA S MIKROSKOPA

Završni rad

Mihael Lendvaj

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Mihael Lendvaj
Studij, smjer:	Programsko inženjerstvo
Mat. br. Pristupnika, godina upisa:	R 4381, 29.09.2020.
OIB Pristupnika:	52024633427
Mentor:	doc. dr. sc. Hrvoje Leventić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Segmentacija i klasifikacija stanica u slikama s mikroskopa
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rad:	Opisati primjenu i način brojanja stanica pomoću mikroskopa i hemocytometra. Istražiti literaturu i opisati razvijene metode za brojanje i klasifikaciju stanica sa slika mikroskopa. U praktičnom dijelu razviti, istrenirati i testirati vlastitu mrežu za klasifikaciju stanica sa slika mikroskopa. Rezervirano za: Mihael Lendvaj
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	15.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 26.09.2023.

Ime i prezime studenta:	Mihael Lendvaj
Studij:	Programsko inženjerstvo
Mat. br. studenta, godina upisa:	R 4381, 29.09.2020.
Turnitin podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom**: Segmentacija i klasifikacija stanica u slikama s mikroskopa**

izrađen pod vodstvom mentora doc. dr. sc. Hrvoje Leventić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD 4

2. PREGLED PODRUČJA TEME 5

2.1. Postupak ručnog brojanja stanica 5

2.2. Razvoj metoda za brojanje stanica 6

2.3. Segmentacija stanica 8

2.4. Klasifikacija stanica 9

3. KONVOLUCIJSKE NEURONSKE MREŽE 10

3.1. Neuron 10

3.2. Operacija konvolucije 10

3.3. U-net Arhitektura mreže 11

4. PROCES IZRADE NEURONSKE MREŽE 12

4.1. Označavanje slika 12

4.2. Organizacija datoteka 13

4.3. Skup podataka 14

4.4. You only look once 14

4.5. Treniranje modela. 15

5. REZULTATI PROGRAMSKOG RJEŠENJA 17

6. ZAKLJUČAK 21

Literatura. 22

Sažetak 23

Abstract 24

Životopis 25

1. UVOD

Od izuma mikroskopa do danas, mikroskopija kao znanost primjenjuje se u mnogim područjima, a ponajviše u medicini.

Izum hemocitometra u kombinaciji sa mikroskopom omogućio je identifikaciju i brojanje stanica u raznim uzorcima, a to je jedan od neophodnih postupaka u kliničkim i istraživačkim laboratorijima na području medicine.

Napredak u području znanosti donio je razne mogućnosti u pristupu dijagnostike i povećanju preciznosti prepoznavanja. Metode brojanja stanica razvile su se od ručnih do automatiziranih postupaka s mogućnostima većih brzina i točnosti.

Doba u kojem su brzina i točnost sve bitniji faktori analize uzoraka, bitno je imati što brže, ali opet točne i pouzdane rezultate.

Kroz ovaj rad istražuje se postupak analize stanica na slici mikroskopa i mogućnosti koje je omogućio razvoj tehnologije na području brojanja stanica. Važni segmenti rada govore o napretku korištenjem računalnog vida i dubokog učenja, te objašnjavaju njihovu implementaciju. U praktičnom dijelu rada prolazi se priprema podatataka, razvitak i testiranje neuronske mreže za brojanje stanica na slikama mikroskopa.

Kroz drugo poglavlje razmatra se ručni postupak, to jest način brojanja stanica pomoću mikroskopa i hemocitometra, te njihov razvoj kroz povijest i pojašnjavaju se segmentacija i klasifikacija. U trećem poglavlju objašnjavaju se pojmovi potrebni za razumijevanje koncepta umjetnih neuronskih mreža i osnovna arhitektura korištene mreže. Četvrtim poglavljem detaljno se prolazi kroz proces izrade skupa podataka i objašnjava se stvaranje konvolucijske neuronske mreže. U petom poglavlju opisuju se rezultati programskog rješenja kojim se ovaj rad bavi.

2. PREGLED PODRUČJA TEME

Algoritmi strojnog učenja postaju sve prisutniji u ljudskim životima, nalaze se u sve većem broju tehnologije koja se koristi. Omogućuju razne mogućnosti poput fotografiranja kvalitetnih slika s pametnim telefonima, detekciju objekata, ispravak grešaka u tipkanju, preporuke sljedećeg filma ili odjevnog predmeta.

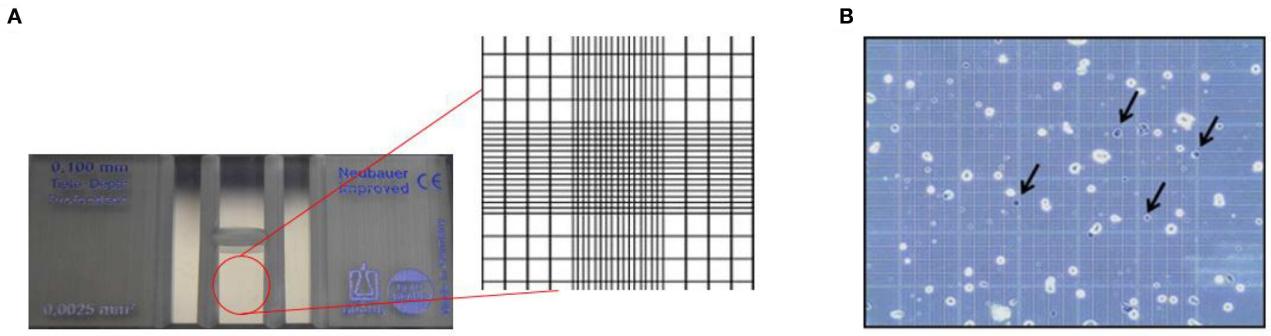
Razvitak strojnog učenja započeo je novu eru znanosti, te se strojno učenje uvuklo u svaku poru života, pa tako su i računala dobila "oči". [1] Područje računalnog vida nije ništa novo, no njegov razvitak u zadnjih 50 godina donio je razne mogućnosti. [2] Računalni vid postao je popularan zbog svoje ekonomske isplativosti, dobre pouzdanosti i široke primjenjivosti. Razni istraživači današnjice još uvijek ručno broje stanice u raznim tjelesnim tekućinama, taj proces postao je spor i skup, obzirom da je vrijeme postalo valuta koje nitko nema dovoljno.

Razvoj računala u smjeru pametnih telefona omogućio je fotografiranje digitalnih slika u djelićima sekundi, Telekomunikacijski napredci poput 5G infrastrukture omogućili su povezanost u stvarnom vremenu, te je lakše nego ikada prepoznavanje i brojanje stanica prepustiti računalu. Korištenjem računalnog vida ne smanjuje se broj koraka postupka analize jednog uzorka tjelesne tekućine, no time se postupak ubrzava i djelomično automatizira, te nije nužna ljudska pažnja prilikom svih koraka.

2.1. Postupak ručnog brojanja stanica

Postupak analize uzorka tekućine složen je proces, te zahtjeva više alata, jedan od njih je [3] hemocitometar, instrument koji se koristi za precizno brojanje stanica u uzorcima tekućine. Također se naziva i Neubauerova komora prema njemačkom znanstveniku Ernstu Neubaueru koji ju je razvio.

Instrument napravljen od stakla podijeljen je urezanim linijama u mrežu. Svaki kvadrat mreže ima poznatu duljinu stranica i površinu. Proces brojanja uključuje postavljanje tekućine uzorka na staklo, te hemocitometar omogućuje stanicama da se raspodjele u kvadrate. Stanice se onda ručno prebrojavaju gledajući kroz mikroskop i brojeći ih po kvadratima. Najčešći uzorci su krv, urin i slične prozirne tekućine.

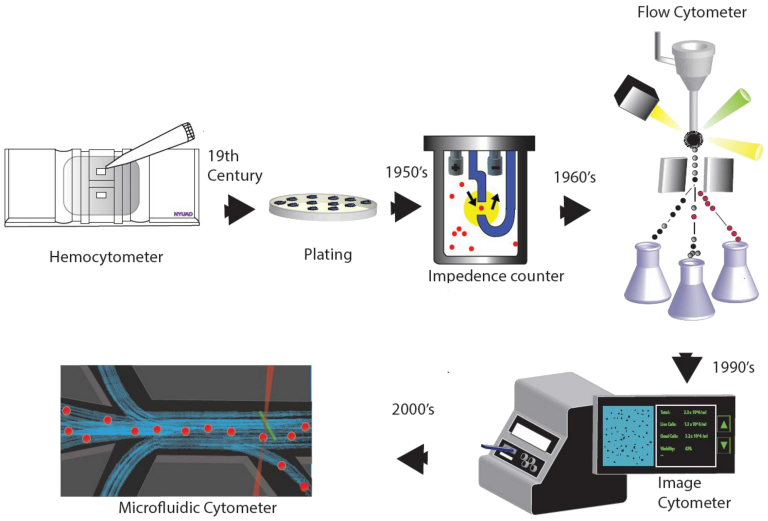


Sl. 2.1: *Slika postupka brojanja stanica pomoću hemocitometra.* ***A****- Povećano stakalce hemocitometra koje pokazuje Neubauerovu komoru s rešetkom* ***B****- Rešetka sa istaknutim mrtvim stanicama. Ova slika je licencirana pod Creative Commons Attribution 4.0 licencom [3]*

2.2. Razvoj metoda za brojanje stanica

Od samih početaka brojanje stanica se pokušava automatizirati kako bi se postigla veća brzina bez gubitka na točnosti.

Uz razvoj računala, optičkih i elektronskih komponenti, kamera i algoritma, [3] puno je jednostavnije eliminirati neke greške postojeće kod ručnog brojanja stanica. Automatizirani sustavi mogu vršiti pripremu uzoraka, te smanjiti proces i vrijeme obrade.



Sl. 2.2: *Razvoj metoda brojanja i detekcije stanica kroz povijest [3]*

Automatizirani sustavi razvijali su se kroz vrijeme, [3] te koristili razne tehnologije, spomenut ćemo se nekih od njih (prema slici 2.2):

- **Brojanje stanica pomoću električne impedancije** - jedna od najčešćih metoda, bazira se na "Coulterovom principu" koji je 1953. razvio Wallace H. Coulter, tvrdio je da može prebrojati preko 6.000 stanica u sekundi
- **Protočna citometrija** - način mjerenja volumena stanica u brzom toku tekućine dok svjetlost lasera prolazi kroz prozirni otvor. Kasnije inovacije rezultiraju povećanje brzine mjerenja, te je moguće prebrojiti preko 10.000 stanica u sekundi

- **Citometrija zasnovana na slikama** - razvojem računala i porastom mogućnosti procesuiranja zadnjih par desetljeća, uz razvoj cijenovno prihvatljih foto senzora, pojavili su se razni sustavi za analizu uzoraka koji koriste strojno učenja i razne algoritme obrade slike. [3] Ovaj oblik omogućava dodatne podatke koje ljudskom oku nisu vidljive, kao što su povećanje nukleusa unutar stanice.

- **Mikrokfluidna citometrija** - metoda zasnovana na integriranom skupu struktura poput mreže mikrokanala, električnih senzora i optičkih komponenti. Princip djelovanja sličan je kao kod protočne citometrije, no ovaj pristup koristi integrirane sklopove te su sustavi puno manji.

2.3. Segmentacija stanica

Segmentacija stanica sa slike mikroskopa označava postupak podjele slike na više dijelova. Razdvajanje od pozadine ljudi rade automatski, te o njemu ne razmišljamo, no on je vrlo bitan kod računalnog vida. Segmentacija slike (primjer slika 2.3) govori o poziciji objekata na slici.

Sl. 2.3: *Segmentacija aviona (izvor: PASCAL VOC2011)*

Govoreći o računalnom vidu, [4] postoje:

1. **Amplitudna segmentacija**
   (a) Amplitudna segmentacija s manualnim postavljanjem praga
   (b) Amplitudna segmentacija Otsu metodom
   (c) Adaptivna amplitudna segmentacija
   (d) Amplitudna segmentacija digitalnih slika u boji
2. **Segmentacija na temelju rubova**
   (a) Detekcija rubova
   (b) *Canny* algoritam za detekciju rubova
   (c) Segmentacija slike pomoću *Houghove* transformacije
3. **Segmentacija na temelju rasta područja**
4. **Segmentacija grupiranjem**

2.4. Klasifikacija stanica

Klasifikacija stanica sa slike mikroskopa predstavlja razvrstavanje raznih objekata u pripadajuće grupe na temelju njihovog izgleda. Primjeri mogu biti klasifikacija živih, mrtvih stanica, pozadine, bakterija ili nepoznatih materijala. Klasifikacija se postiže postupkom koji se sastoji od prikupljanja podataka s uzorcima različitih klasa. Primjer klasificiranih slika (slika 2.4).

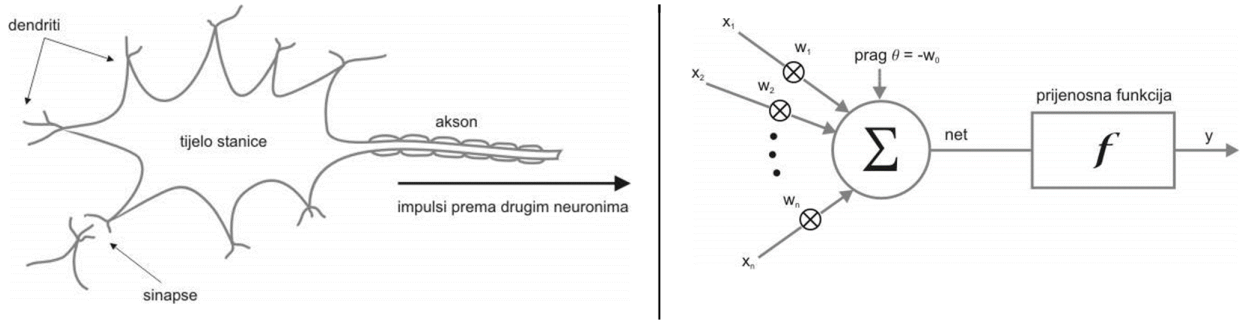
Sl. 2.4: *Klasificirane slike sa pripadajućim klasama navedenim iznad svake slike (izvor: MNIST)*

3. KONVOLUCIJSKE NEURONSKE MREŽE

Konvolucijske neuronske mreže [5] podvrsta su koja služi za procesuiranje nestrukturiranih podataka, većinom su to slike(pod njih spada i video), zvuk, tekst i govor. Arhitektura im se bazira na višeslojnim perceptronima, te se pokazala kao najbolji model za izvlačenje bitnih svojstava iz nestrukturiranih podataka. David H. Hubel i Torsten N. Wiesel dobili su 1981. godine Nobelovu nagradu za otkriće djelovanja neurona, te pokazali njihovo djelovanje. Eksperiment su izvršili na mačkama i otkrili kako se različite skupine neurona u mozgu aktiviraju ovisno o kutu pod kojim se objekt promatra. Pokazali su kako neki neuroni prepoznaju rubove, nevezano uz kut gledanja. Njihov rad bio je teorijska pozadina pri razvoju konvolucijskih neuronskih mreža. Ime "konvolucijske neuronske mreže" govori da koristimo neurone i konvoluciju, stoga je bitno pojma objasniti.

3.1. Neuron

Neuron je osnovna stanica živčanog sustava. Prirodni neuron bio je inspiracija za osnovnu gradivnu jedinicu neuronske mreže po kojem je i dobila ime. Slika 3.5 pokazuje izgled i građu neurona, te izgled koji računalo pokušava oponašati, Umjetni neuron (desna strana slike) pokazuje na početku ulazne podatke označene sa $x_1, x_2, ..., x_n$, nakon toga i slobodne koeficijente (engl. *bias*) kao $w_1, w_2, ..., w_3$, na posljetku podatci se spajaju u sumator i izlaze preko prijenosne funkcije u izlaz iz neurona.



Sl. 3.5: *Karikatura prirodnog neurona s lijeve strane i umjetnog s desne [6]*

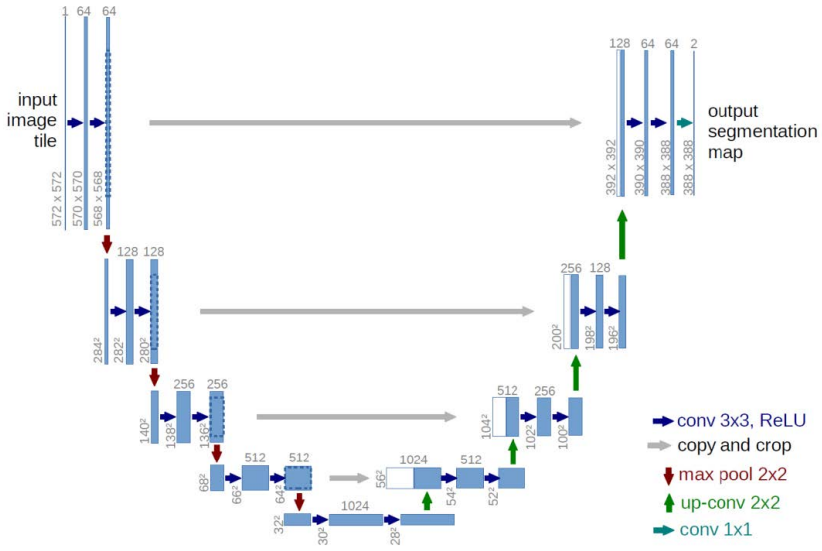
Izlazni signal prenosi se do aktivacijske funkcije, nakon koje se spajaju na ulaze drugih neurona, te dobivamo mrežu čiji cilj je glumiti živčani sustav koji reagira na različite podražaje poput neurona u našem živčanom sustavu.

3.2. Operacija konvolucije

Konvolucija je matematička operacija [7] skalarnog produkta jedne funkcije s obzirom na translatiranu i reflektiranu drugu funkciju. Ova funkcija omogućuje mreži da lokalno detektira značajke poput rubova, tekstura ili oblika

3.3. U-net Arhitektura mreže

U-Net je [8] arhitektura konvolucijskih neuronskih mreža koja se koristi u području segmentacije slika. Ova arhitektura je posebno dizajnirana za rješavanje problema segmentacije slika gdje je potrebno identificirati i razlučiti pojedinačne objekte ili regije unutar slike. Nazvan je prema svom obliku koji sliči na slovo "U". Sastoji se od dvije glavne komponente: "kodera" (encoder) i "dekodera" (decoder). Prikaz položaja kodera i dekodera, te njihovih ulaza i izlaza na slici 3.6.



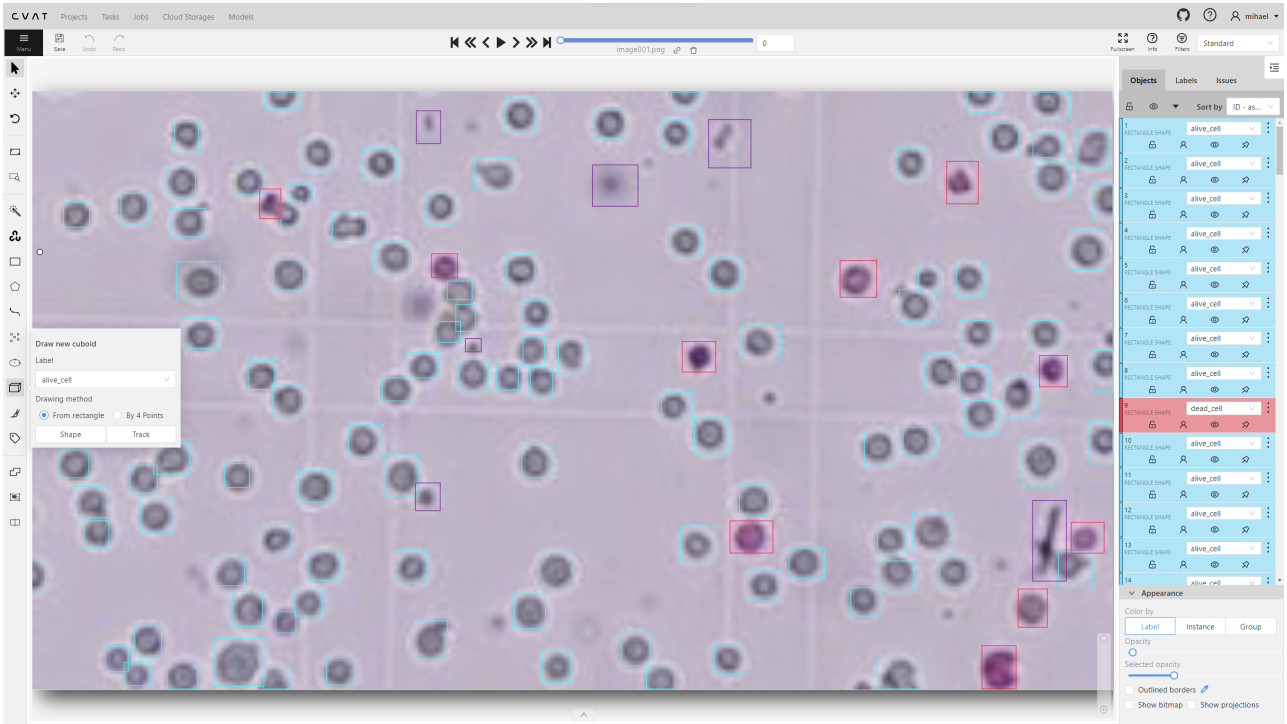
Sl. 3.6: *Prikaz U-net arhitekture [8]*

4. PROCES IZRADE NEURONSKE MREŽE

Kako bi izgradio model neuronske mreže, potrebni su neki podatci, kako bi mreža mogla naučiti prepoznavati i brojati stanice. Za potrebe ovog rada korišteni su isječci slika sa mikroskopa. Slike mikroskopa izrezane su na isječke zbog ograničenosti sustava, obzirom da treniranje modela zahtjeva velike količine radne memorije. Prema [9] najbolji način za istrenirati model je korištenjem yolo [10] "**You only look once**" modela. YOLO je model napravljen od strane Ultralytics kompanije. Model je napravljen kao generalni model koji je potrebno istrenirati te se on prilagođava podatcima. Za istrenirati model na bilo kakve podatke bilo je potrebno podatke označiti (engl. *annotate*). Taj proces izvršio se koristeći *online* aplikaciju CVAT.

4.1. Označavanje slika

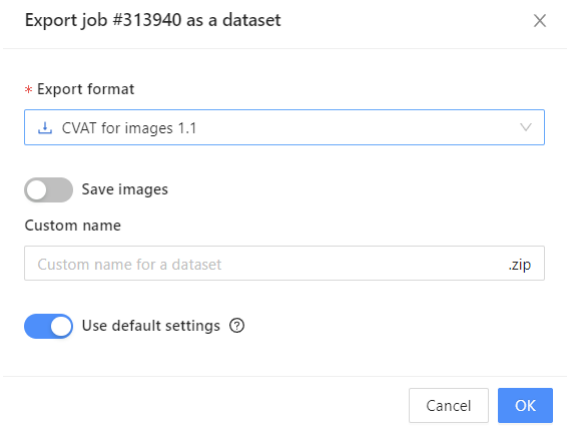
Na aplikaciji CVAT potrebno je napraviti projekt i u njega dodati slike. Kada se slike prenesu na oblak aplikacije, potrebno je napisati klase i označavanje može početi.



Sl. 4.7: *Prikaz sučelja aplikacije CVAT.ai*

Na slici 4.7 prikazano je sučelje sa označenom slikom. U označavanju korišteni su pravokutnici, zbog posebne vrste obilježavanja koje koristi model yolo. Kako bi svaki objekt obilježili, potrebno je pokazivač pozicionirati sa lijeve strane na zatamnjenu ikonu 3d kocke ta odabrati vrstu objekta koji želimo označici, nakon toga potrebno je kliknuti na gumb pod nazivom "*Shape*". Tim postupkom se odabire alat za "crtanje" pravokutnika na slici, pokazivač dobiva dodatne linije za povećanje preciznosti, te je potrebno kliknuti na dva kuta stanice i napraviti time pravokutnik oko svake žive ili mrtve stanice i oko raznih nečistoća kojih će sigurno biti, odabrane su te tri klase objekata obzirom da nije moguće izbjeći pojavu raznih nečistoća, a

cilj je prepoznati žive i mrtve stanice, također se označavanjem nečistoća povećava vjerojatnost točne klasifikacije stanica. Kada se označi sliku, potrebno je klikom na strelicu prema desno koja se nalazi na gornjem dijelu zaslona prijeći na iduću sliku te označiti sve objekte na njoj. Nakon završetka označavanja svih objekata na svim slikama oznake je potrebno preuzeti u odgovarajućem formatu. Preuzimanje oznaka vrši se klikom na *Menu* → *Export job dataset*. Otvoriti će se prozor (prema slici 4.8)



Sl. 4.8: *Prikaz izbornika za preuzimanje oznaka*

U prozoru potrebno je odabrati pod "*Export format*" opciju "CVAT for images 1.1" kao na slici 4.8

4.2. Organizacija datoteka

Kako bi se model istrenirao, potrebno je datoteke postaviti u odgovarajuće direktorije i napraviti nove datoteke postavki. Obzirom da će se model trenirati, potrebno je slike podjeliti na skupove, u procesu treninga koristit će se setovi "*train*" i "*val*". Slike i njihove oznake potrebno je postaviti u pripadajuće direktorije prema stablu na slici 4.9. Slike je potrebno odvojiti te većinu staviti u "test/images/", a manjinu slika u "val/images/". U direktorije labels je potrebno staviti pripadajuće oznake.

Sl. 4.9: *Prikaz izbornika za preuzimanje oznaka*

Oznake su pohranjene u "ime slike".txt formatu, te svaka slika ima pripadajuću istoimenu datoteku .txt formata u kojoj se nalaze njene oznake. Oznake su napisane u 4 stupca

unutar svake .txt datoteke, prvi stupac označava klasu, drugi prvu kordinatu centra, treći visinu pravokutnika, četvrti drugu kordinatu centra, peti širinu pravokutnika (prikazano u slici 4.10)

```
0 0.549532 0.718698 0.052164 0.049822
0 0.069181 0.924586 0.043977 0.046805
0 0.276754 0.083698 0.043216 0.043728
0 0.199591 0.560680 0.038012 0.041479
0 0.063187 0.608195 0.042515 0.047515
0 0.008684 0.693817 0.017368 0.039231
0 0.064678 0.825414 0.038012 0.047515
0 0.021082 0.843136 0.040292 0.042249
1 0.172018 0.814083 0.051404 0.050533
0 0.287544 0.852929 0.049942 0.054260
0 0.274503 0.881982 0.035789 0.033905
```

Sl. 4.10: *Prikaz formata datoteke za oznake*

Datoteke su time postavljene u skup podataka raspodjelom prikazanom na slici 4.9.

4.3. Skup podataka

Skup podataka sastoji se od dijela za treniranje, validaciju i testiranje, podijeljen je u tri direktorija. Dio za treniranje sadrži 157 isječaka, 1933 živih i 650 mrtvih stanica. Validacijski dio sastoji se od 32 isječka, 159 živih i 42 mrtvih stanica, a testni skup sastoji se od 49 isječaka, 155 mrtvih i 244 živih stanica.

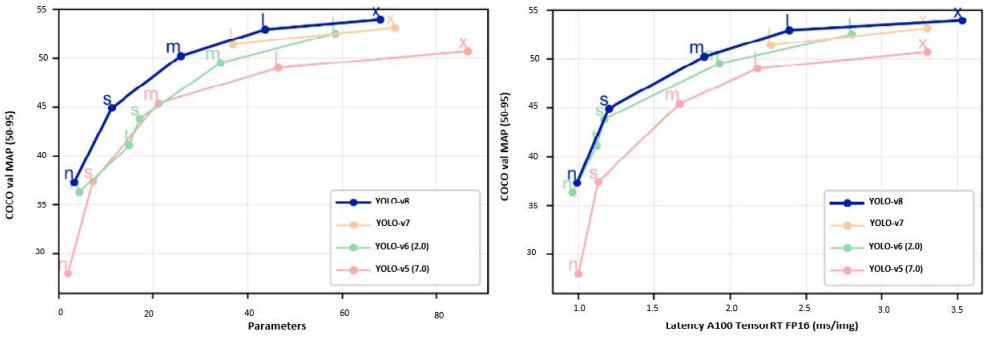
4.4. You only look once

Za potrebe kreiranja modela korištena je [10] YOLO arhitektura. Prva verzija izdana je 2015. godine, a zadnja pod brojem 8 najavljena je u siječnju 2023. godine. Arhitektura je optimizirana za izvođenje u stvarnom vremenu s visokom točnošću, što je postao glavni cilj svake verzije do sada. Svaka inačica točnija je i sve brža, time se omogućuje korištenje na računalima s manje resursa. U ovom radu korištena je verzija 8, obzirom da je biblioteka još u svojim ranim fazama, moguće su velike promjene na njoj, stoga napominjem da je za potrebe rada korištena biblioteka *ultralytics == 8.0.16*.

Uz biblioteku *ultralytics* korištene su i biblioteke *torch == 2.0.1+cu118*, *torchaudio == 2.0.2+cu118* i *torchvision == 0.15.2+cu118*. Biblioteke je potrebno instalirati u razvojno okruženje jer ne dolaze sa standardnom instalacijom *python* programskog jezika.

Za potrebe rada koristit će se metoda modela *"object detection"*, te vrsta modela "*yolov8m.pt*", yolov8 ima razne veličine modela koje je moguće preuzeti i istrenirati, ovisno o potrebama brzine i kompleksnosti modela. Računalo na kojem radim u mogućnosti je pokrenuti i istrenirati "*medium*" model, odnosno *"yolov8m"*.

Model verzije 8 usporedivši ga s prethodnim verzijama arhitekture je [11] točniji i brži na istom broju parametara (prikaz slika 4.11).



Sl. 4.11: *Prikaz performansi različitih verzija YOLO arhitekture i svih njihovih vrsta modela[11]*

4.5. Treniranje modela

Model se trenira korištenjem *python* programskog jezika ili iz "*Command Line Interface*" sučelja. Obzirom na veće performanse, koristila se CLI metoda za treniranje modela nad podatcima. Prije samog testiranja potrebno je dodati jednu datoteku sa imenima klasa, putanju do validacijskog i trening direktorija, te broj klasa. U ovom radu korištena je datoteka imena *"data_custom.yaml"* (prikaz slika 4.12)

```
1  train: C:\Users\mihael\Documents\Fakultet\Semestar6\ZR\train
2  val: C:\Users\mihael\OneDrive\Documents\Fakultet\Semestar6\ZR\val
3
4  nc: 3
5
6  names: ["alive", "dead", "trash"]
```

Sl. 4.12: *Prikaz sadržaja datoteke data_custom.yaml*

Nakon imenovanja datoteke i postavljenih direktoraija, potrebno je u sučelje unijeti parametre i pokrenuti treniranje. (prema slici 4.13)

```
1  yolo task=detect mode=train epochs=100 data=data_custom.yaml
     ↪ model=yolov8m.pt imgsz=640 batch=8 patience=20
```

Sl. 4.13: *Prikaz naredbe za treniranje modela*

Argumenti ukazuju da se koristi object detection, te da se model trenira na 100 epoha, a radi se o *medium* modelu sa slikama veličine 640. Model "gleda" 8 slika u jednom trenutku. Značajke modela prikazane su slikom 4.14.

Program će izvesti epohu po epohu i stati u jednom trenutku i validirati model s najboljim preformansama (vidi sliku 4.15).

```
                   from  n    params  module                                       arguments
  0                  -1  1      1392  ultralytics.nn.modules.Conv                  [3, 48, 3, 2]
  1                  -1  1     41664  ultralytics.nn.modules.Conv                  [48, 96, 3, 2]
  2                  -1  2    111360  ultralytics.nn.modules.C2f                   [96, 96, 2, True]
  3                  -1  1    166272  ultralytics.nn.modules.Conv                  [96, 192, 3, 2]
  4                  -1  4    813312  ultralytics.nn.modules.C2f                   [192, 192, 4, True]
  5                  -1  1    664320  ultralytics.nn.modules.Conv                  [192, 384, 3, 2]
  6                  -1  4   3248640  ultralytics.nn.modules.C2f                   [384, 384, 4, True]
  7                  -1  1   1991808  ultralytics.nn.modules.Conv                  [384, 576, 3, 2]
  8                  -1  2   3985920  ultralytics.nn.modules.C2f                   [576, 576, 2, True]
  9                  -1  1    831168  ultralytics.nn.modules.SPPF                  [576, 576, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11            [-1, 6]  1         0  ultralytics.nn.modules.Concat                [1]
 12                  -1  2   1993728  ultralytics.nn.modules.C2f                   [960, 384, 2]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14            [-1, 4]  1         0  ultralytics.nn.modules.Concat                [1]
 15                  -1  2    517632  ultralytics.nn.modules.C2f                   [576, 192, 2]
 16                  -1  1    332160  ultralytics.nn.modules.Conv                  [192, 192, 3, 2]
 17           [-1, 12]  1         0  ultralytics.nn.modules.Concat                [1]
 18                  -1  2   1846272  ultralytics.nn.modules.C2f                   [576, 384, 2]
 19                  -1  1   1327872  ultralytics.nn.modules.Conv                  [384, 384, 3, 2]
 20            [-1, 9]  1         0  ultralytics.nn.modules.Concat                [1]
 21                  -1  2   4207104  ultralytics.nn.modules.C2f                   [960, 576, 2]
 22        [15, 18, 21]  1   3777433  ultralytics.nn.modules.Detect                [3, [192, 384, 576]]
Model summary: 295 layers, 25858057 parameters, 25858041 gradients, 79.1 GFLOPs
```

Sl. 4.14: *Prikaz značajki modela*

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     37/100      7.16G      1.281      0.569      1.007        226        640: 100%|██████████| 20/20 [01:22<00:00,  4.14s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:11<00:00,  5.89s/it]
                   all         31        201      0.978      0.974      0.984      0.665
Stopping training early as no improvement observed in last 20 epochs. Best results observed at epoch 17, best model saved as best.pt.
To update EarlyStopping(patience=20) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

37 epochs completed in 1.491 hours.
Optimizer stripped from runs\detect\train\weights\last.pt, 52.0MB
Optimizer stripped from runs\detect\train\weights\best.pt, 52.0MB

Validating runs\detect\train\weights\best.pt...
Ultralytics YOLOv8.0.16  Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (NVIDIA GeForce RTX 3050 Ti Laptop GPU, 4096MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:11<00:00,  5.93s/it]
                   all         31        201      0.974      0.989      0.993      0.699
                 alive         31        159          1      0.978      0.994      0.698
                  dead         31         42      0.948          1      0.992      0.701
Speed: 1.9ms pre-process, 171.3ms inference, 0.0ms loss, 1.0ms post-process per image
Saving runs\detect\train\predictions.json...
Results saved to runs\detect\train
```

Sl. 4.15: *Prikaz izlaza programa na kraju modela*

Program će nakon treniranja spremiti najbolji model i statističke podatke u pripadajuće direktorije (vidi sliku 4.15).

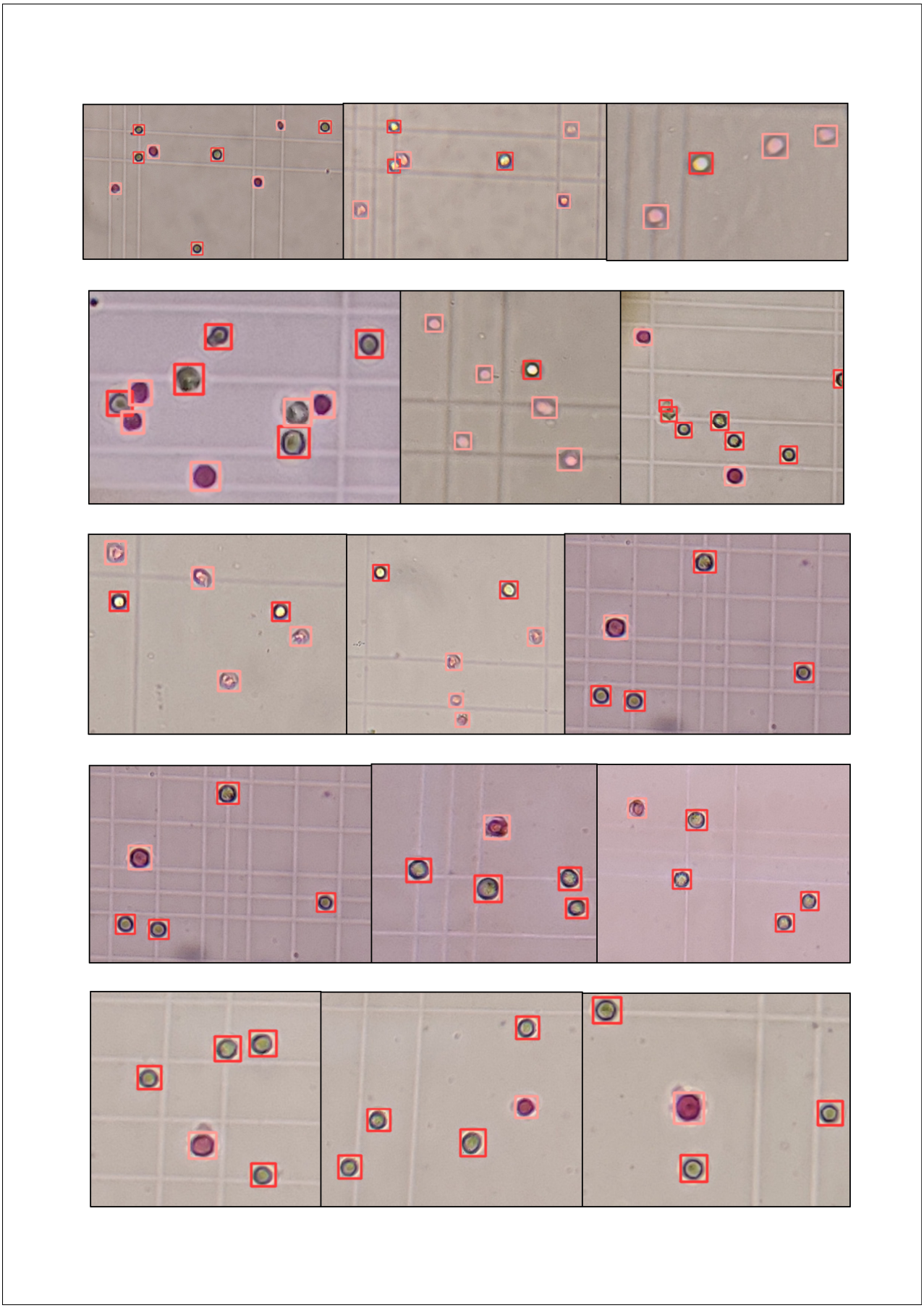
5. REZULTATI PROGRAMSKOG RJEŠENJA

Programsko rješenje prikazano u prethodnom poglavlju rada prikazat će rezultate korištenjem metode "*predict*". Obzirom da je biblioteka u razvoju, kreirao se*python* program koji će pokrenuti yolo model da pronađe i označi objekte na novim slikama, a nove slike postavljene su u novi direktoriji "*test/*"

```
1   import os
2   from ultralytics import YOLO
3
4   model = YOLO("yolov8m_custom.pt")
5   model.predict(
6           source=os.path.join('test', 'images'),
7           show=False,
8           conf=0.5,
9           save=True,
10          line_thickness=2,
11          hide_labels=True,
12          save_txt=True)
```

Sl. 5.16: *Prikaz programa za predikciju nad novim slikama*

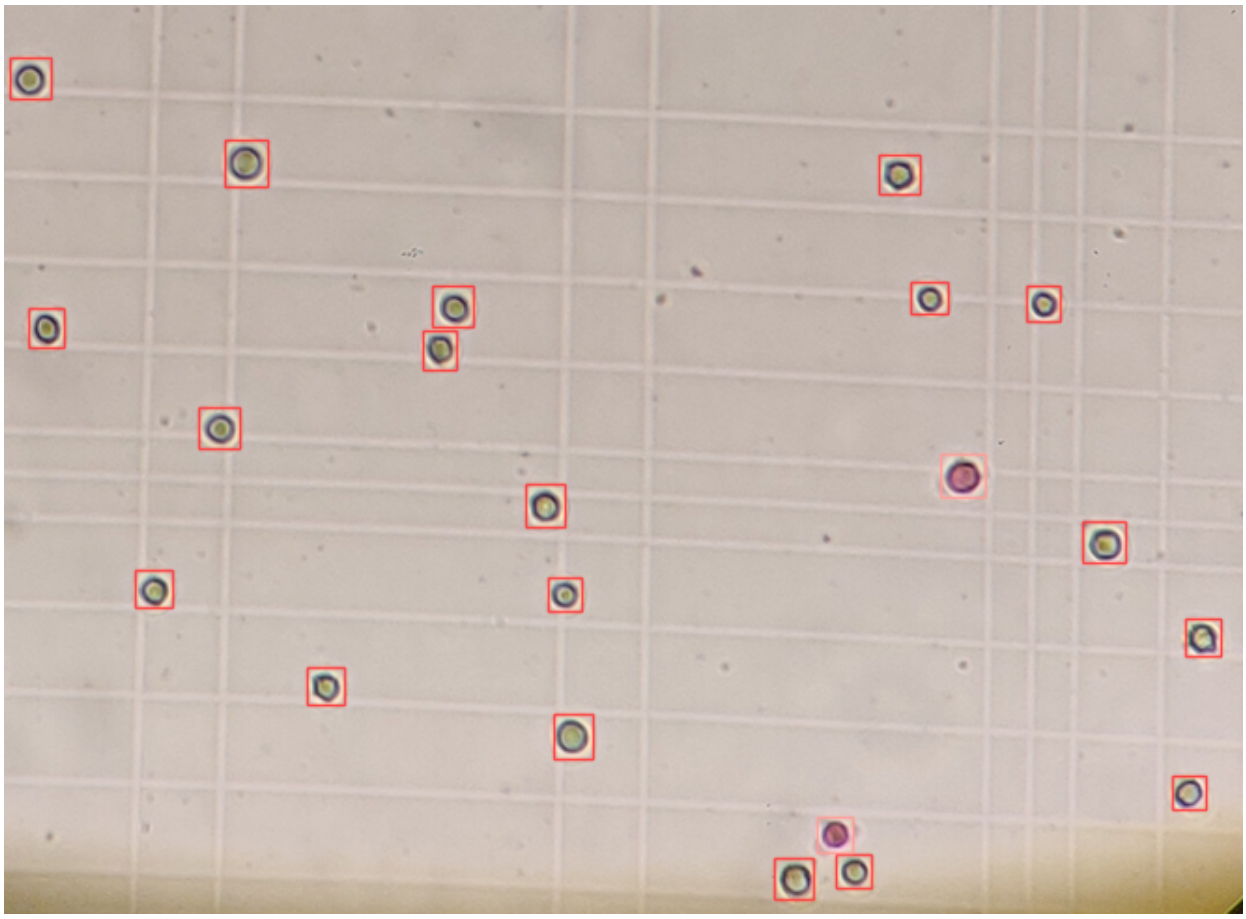
Program (slika 5.16) koristi najbolji model koji je spremljen iz programa za trening podataka (slika 4.13), a model je potom pronađen, preimenovan i premješten u engl. "*root*" direktorij. Program za predikciju novih slika obradit će slike koje su stavljene u direktoriji *test/images/*. Slike u direktoriju model prije nije "vidio", stoga su dobre za provjeru modela.



Sl. 5.17: *Prikaz označenih slika koje je model pretpostavio*

Slike sa označenim stanicama program izbacuje u formatu novih slika, te nigdje ne ispisuje broj objekata ni koji su objekti pronađeni. Biblioteka [10] ima mogućnost osim slika sa iscrtanim oznakama također za svaku sliku spremiti sve pronađene objekte u obliku istom kao i oznake prikazane slikom 4.10. Ta mogućnost nudi rješenje problema nedostatka ispisa vrsta i količina pronađenih objekata na slici, stoga je napisan program koji iščitava oznake iz datoteke koje su spremljene zajedno sa novom označenom slikom. Program funkcionira na način da mu se predaje putanja do izvora slika koja se prosljeđuje u program 5.16. Nakon izvođenja programa za predikciju, otvara se i izlistava direktoriji u kojem je pohranjen ispis predikcija sa datotekama oznaka. Datoteke se potom čitaju kako bi se prebrojile i spremile vrste i broj oznaka. Na posljetku se ispisuje vrsta i tip oznaka koje je model pronašao zajedno sa pripadajućom količinom takvih oznaka.

Rezultati se ispisuju u programu u obliku prikazanom na slici 5.19 za pripadajuću sliku 5.18.



Sl. 5.18: *Prikaz slike koju je model označio*

```
Ultralytics YOLOv8.0.16  Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (NVIDIA GeForce RTX 3050 Ti Laptop GPU, 4096MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs
Class           Count
    Alive cell  19
    Dead cell   2
Results saved to runs\detect\predict15
1 labels saved to runs\detect\predict15\labels

Process finished with exit code 0
```

Sl. 5.19: *Prikaz ispisa programa za predikciju nad novim slikama*

6. ZAKLJUČAK

Klasifikacija i segmentacija stanica na slikama mikroskopa složen je postupak koji zahtjeva kombinaciju raznih tehnika za rješavanje. Do sada mnogi su istraživači korištenjem novih tehnologija došli do složenih rješenja ovog problema. Zadatak rada spada u područje strojnog učenja koji prepoznaje stanice, klasificira ih i prebrojava. Literatura u području bavi se usporedbom raznih programskih rješenja, te ističe brzinu i točnost kao dvije glavne značajke svakog modela. Također ističu se broj slojeva i parametara kao bitne značajke. Konvolucijske neuronske mreže nameću se u mnogim usporednim istraživanjima kao najbolji način rješavanja ovakve vrste problema. Sukladno time u radu se koristi YOLO model istreniran na novom skupu podataka isječaka slika sa mikroskopa. Zbog svoje brzine i specijaliziranosti prepoznavanja na medicinskim podatcima, vrsta modela je pogodna za implementaciju vezanu uz prepoznavanje objekata malih dimenzija poput stanica. Početni ulazni skup podataka slika sa mikroskopa izrezan je u isječke zbog ograničenosti dostupnih resursa uz zadržavanje brzine i točnosti. Značajke skupa podataka isječaka označene su ručno, te je teško odrediti njihovu točnost obzirom na male dimenzije živih i mrtvih stanica. Na točnost također utječu i male razlike u količini pigmenta i raznim drugim utjecajima poput prljavštine i automatskih poboljšavanja slika prisutnih na većini današnjih pametnih telefona kojima su slike fotografirane. Nastavno na to slike su označene u formatu koji odgovara tipu modela prema dokumentaciji, a vrsta zapisa oznaka koristi sintaksu pravokutnika korištenu u *openCV* biblioteci, te je uz nju dodan broj klase. Model je razvijen nad skupom podataka i osposobljen da prepozna objekte koje klasificira na žive i mrtve stanice te ih označava iscrtavanjem pravokutnicima u bojama pripadajućih klasa. Model uspješno rješava probleme klasifikacije i segmentacije uz problem nedostatka podataka o broju stanica na svakoj slici. Uz to teško se može utvrditi velika preciznost modela. Problem nedostatka broja tipa stanica nakon pregleda slike riješen je iščitavanjem izlazne datoteke modela. Manjak preciznosti modela i njegovu osjetljivost na promjenu boja moguće je riješiti u nekom idućem radu sa većim modelom i složenijim sklopovljem ili sa drugim izvorom fotografija koji je konzistentniji u postavkama svjetline i količine boje na fotografijama. Cijele slike mikroskopa također je moguće analizirati većim modelom i s kvalitetnijim sklopovljem.

Literatura

[1] Richard Szeliski. *Computer Vision: Algorithms and Applications, 2nd Edition*. Springer, September 2021.

[2] Milan Hrga. Računalni vid. *Zbornik radova Veleučilišta u Šibeniku*, 12(1-2):207–216, April 2018. Publisher: Veleučilište u Šibeniku.

[3] Abhishek Vembadi, Anoop Menachery, and Mohammad A. Qasaimeh. Cell Cytometry: Review and Perspective on Biotechnological Advances. *Frontiers in Bioengineering and Biotechnology*, 7:147, June 2019.

[4] Nikola Danilović. *Metode segmentacije objekata i područja u digitalnoj slici*. info:eu-repo/semantics/bachelorThesis, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Communications. Chair of Multimedia Systems and Digital Television, September 2020.

[5] Kristina Džomba. *Konvolucijske neuronske mreže*. info:eu-repo/semantics/masterThesis, University of Zagreb. Faculty of Science. Department of Mathematics, Zagreb, July 2018.

[6] Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva, Strojno učenje - nastavni materijali, 2001.

[7] Siniša Šegvić. Duboko učenje 1 - Konvolucijski modeli.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015. Series Title: Lecture Notes in Computer Science.

[9] Mohammad Mahmudul Alam and Mohammad Tariqul Islam. Machine learning approach of automatic identification and counting of blood cells. *Healthcare Technology Letters*, 6(4):103–108, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/htl.2018.5098.

[10] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023. original-date: 2022-09-11T16:39:45Z.

[11] Muhammad Hussain. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7):677, July 2023. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Sažetak

Postupak segmentacije i klasifikacije stanica u uzorcima neizostavan je postupak pri otkrivanju mnogih bolesti i analize raznih uzoraka u mnogim istraživanjima. Razvoj računala i sve veći broj otkrića na području tehničkih znanosti omogućio je kroz povijest razvitak raznih metoda analize uzoraka različitih tjelesnih tekućina.

U ovom radu opisao se proces brojanja stanica pomoću mikroskopa i hemocitometra, te način korištenja hemocitometra. Pokazalo se kako su brzina i točnost bitni parametri kod analize uzoraka.

Bavilo se analizom slike, te mogućnostima segmentacije i klasifikacije, opisalo koliko je priprema podataka važna pri strojnom učenju.

Na posljetku se sve to znanje i istraživanje spojio u kreiranje YOLO modela koji prepoznaje stanice na novim slikama u realnom vremenu i broji ih, iscrtava i označava, te prikazuje.

Ključne riječi:: detekcija objekata, hemocitometar, klasifikacija slike, segmentacija slike, strojno učenje, yolo

Abstract

Title: Segmentation and classification of cells in microscope images

The process of segmentation and classification of cells in samples is an indispensable procedure in the detection of many diseases and the analysis of various samples in many researches. The development of computers and the increasing number of discoveries in the field of technical sciences have brought us throughout the history of developing various methods of different body fluids sample analasys.

In this paper, I described the process of counting cells using a microscope and a hemocytometer, and how to use the hemocytometer. I studied how speed and accuracy are important parameters when analyzing samples.

I dealt with image analysis, segmentation and classification possibilities, described how important data preparation is to us in machine learning.

In the end, I combined all this knowledge and research to create a YOLO model that recognizes cells in new images in real time and counts them, plots and labels them, and displays them.

Keywords: blood cells, hemocytometer, image classification, image segmentation,machine learning, object detection, yolo

Životopis

Mihael Lendvaj rođen je 8.10.2000. godine u Zagrebu. Pohađao je Osnovnu školu Silvija Strahimira Kranjčevića u Zagrebu, te nakon toga svoje je školovanje nastavio kroz program Tehničke gimnazije u Tehničkoj školi u Palmotićevoj ulici u Zagrebu. Godine 2019. završava srednju školu i upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku