

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Vitjan Zavrtanik

**Segmentacijska konvolucijska
nevronska mreža za štetje polipov na
slikah**

MAGISTRSKO DELO
MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Matej Kristan

Ljubljana, 2018

AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

©2018 VITJAN ZAVRTANIK

ZAHVALA

Zahvaljujem se mentorju izr. prof. dr. Mateju Kristanu za odlične nasvete, ideje in napotke pri izdelavi magistrskega dela. Zahvaljujem se tudi družini in puncu Anji za podporo in vzpodbudo med študijem.

Vitjan Zavrtanik, 2018

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Cilji magistrskega dela	1
1.3	Sorodna dela	3
1.4	Prispevki	6
1.5	Struktura dela	7
2	Konvolucijske nevronske mreže	9
2.1	Nevronske mreže	9
2.2	Konvolucijske nevronske mreže	11
2.3	Drugi sloji in operacije konvolucijskih nevronskih mrež	20
2.4	Uporabljene arhitekture	27
3	Predlagana metoda	35
3.1	Arhitektura nevronske mreže	35
3.2	Priprava učne segmentacijske maske	36
3.3	Interpretacija segmentacijske maske	39
4	Razvoj aplikacije za detekcijo objektov	41
4.1	Uporabljena tehnologija	41
4.2	Struktura uporabniškega vmesnika	42

KAZALO

4.3	Orodja vmesnika	44
4.4	Dodatne funkcionalnosti	48
4.5	Shranjevanje podatkov	50
4.6	Učenje modela in inferenca	50
5	Eksperimentalna analiza	53
5.1	Podatkovna zbirka	53
5.2	Metoda evalvacije	59
5.3	Analiza rezultatov predlagane metode	61
5.4	Napake v testni množici	72
5.5	Primerjava z detektorjem PoCo	78
5.6	Primerjava z rezultati mreže RetinaNet	83
5.7	Popravek anotacij testne množice	86
6	Sklep	93
6.1	Nadaljnje delo	95

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
IOU	intersection over union	preseki čez unijo
AR	average recall	povprečni priklic
AP	average precision	povprečna natančnost
CNN	convolutional neural network	konvolucijska nevronska mreža

Povzetek

Naslov: Segmentacijska konvolucijska nevronska mreža za štetje polipov na slikah

V nalogi naslavljamo problem detekcije polipov meduz na slikah ostrig. Moderne metode detekcije objektov so pogosto sestavljene iz dveh faz. Najprej se na potencialnih lokacijah generirajo hipotetične regije, nato pa se posamezno hipotetično regijo klasificira v pripadajoči razred, glede na objekt, ki ga regija vsebuje. V delu se osredotočamo na drugačen pristop k detekciji objektov, saj najprej z uporabo konvolucijske nevronske mreže generiramo segmentacijsko masko objektov na sliki, nato pa z interpretacijo maske dobimo natančno lokacijo iskanih objektov. Razvito metodo SegCo uporabljamo za reševanje problema detekcije polipov meduz na slikah školjk. Rezultate predlagane metode smo primerjali z rezultati modernih metod detekcije objektov. Rezultate smo primerjali z modernim učljivim detektorjem RetinaNet [1] in specializirano metodo za detekcijo polipov PoCo [2]. V primerjavi z detektorjem RetinaNet, metoda SegCo dosega 2% izboljšavo mere F-1, v primerjavi z detektorjem PoCo pa 24% izboljšavo. V sklopu naloge je bil razvit tudi program, ki z uporabo predlagane metode omogoča avtomatsko detekcijo in štetje objektov na slikah. Delovanje programa ni omejeno le na detekcijo polipov, temveč z učinkovitim uporabniškim vmesnikom omogoča učenje novih modelov predlagane metode, zaradi česar lahko program detektira objekte tudi na drugih vrstah slik.

Ključne besede

Polipi, konvolucijska nevronska mreža, segmentacija slik, detekcija objektov

Abstract

Title: Convolutional neural network segmentation for counting polyps in images

We address the problem of jellyfish polyp detection on images of oysters. Modern methods of object detection often utilize convolutional neural networks for feature extraction and work in two stages. First, hypothetical regions are proposed at potential locations, the features of the regions are extracted and are later classified according to the object they contain. In this work we focus on an alternative approach to object detection in which we first use a convolutional neural network to obtain an image segmentation mask which we then interpret to extract the precise location and shape of the objects in the image. We use the proposed method SegCo to address the problem of jellyfish polyp detection on images of oysters. We compare the results of the proposed method with current state of the art object detection methods. We compare the results the state of the art learnable detector RetinaNet [1] and the specialized polyp detection method PoCo [2]. In comparison with RetinaNet, SegCo achieves a 2% improvement in F-1 score and in comparison with PoCo, the achieved improvement is 24%. In addition we developed a program, which utilizes the proposed method to enable the user to automatically detect and count objects in images. The application of our program is not limited to the detection of jellyfish polyps, as it contains an efficient user interface for training new models using the proposed method, which enables the user to easily apply our method to objects in other types of images.

Keywords

Polyps, convolutional neural network, image segmentation, object detection

Poglavje 1

Uvod

1.1 Motivacija

Meduze preživijo del svojega življenjskega cikla v obliki polipov. V tej fazi so pritrjene na trdne površine. S poznavanjem števila oziroma gostote polipov na površinah lahko biologi napovedujejo populacijo meduz v posameznih regijah. Poznavanje populacije meduz se lahko uporabi kot pomemben pokazatelj drugih dejavnikov sprememb v vodnem okolju. Pridobivanje podatkov o številu meduz je dolgotrajen proces, saj je potrebno ročno prešteti polipe na slikah ostrig, kjer se polipi pogosto nahajajo [3]. Primer slike ostrige s polipi lahko vidimo na Sliki 1.1. Polipi se med sabo razlikujejo po obliki, orientaciji, prosojnosti, posledično še po barvi, poleg tega pa se velikokrat medsebojno prekrivajo. Zaradi teh problematičnih vizualnih lastnosti polipov je zanesljivo ročno štetje oteženo, ker pa je na posamezni sliki lahko tudi več kot 1000 polipov, pa je štetje tudi dolgotrajno. Zaradi dolgotrajnosti in težavnosti se pri ročnem štetju pogosto pojavljajo napake [2].

1.2 Cilji magistrskega dela

Moderne metode detekcije objektov uporabljajo metode predlaganja regij, pogosto kar konvolucijsko nevronske mreže, ki predlaga regije na fiksnih lo-



Slika 1.1: Primer slike polipov pritrjenih na ostrigo.

kacijah [4, 5, 6, 7, 1]. Te metode sicer dosegajo odlične rezultate na tekmovanjih kot je [8] vendar pa ta večinoma preverjajo uspešnost detekcije objektov na primerih, kjer so iskani objekti relativno veliki in predstavljajo velik del slike. V primeru polipov in drugih primerov detekcije majhnih objektov [9] pa lahko zaradi premajhne ločljivosti značilnk, te metode delujejo manj uspešno.

Eden izmed ciljev dela je predlagati detekcijsko metodo, ki rešuje problem detekcije večjega števila majhnih objektov na slikah. Preverili smo več različnih pristopov, rezultate predlagane metode pa smo primerjali z rezultati modernih metod detekcije objektov. Cilj je tudi razvoj učinkovite aplikacije za anotacijo in detekcijo objektov, ki omogoča učenje modelov predlagane metode in bi lahko z avtomatizacijo štetja polipov in ostalih poljubnih objektov prihranila veliko količino časa uporabnikom, ki bi sicer štetje morali opraviti ročno.

1.3 Sorodna dela

Detekcija in segmentacija objektov sta popularni raziskovalni področji zaradi številnih možnosti uporabe razvitih algoritmov. Predstavljata pomemben del računalniške interpretacije slikovnega materiala. Uporabljata se na praktično vseh področjih kjer je potrebna obdelava slikovnega materiala saj je za veliko število aplikacij detekcija objektov nujna. Primeri področij, kjer se detekcija in segmentacija objektov uporabljata so štetje celic v mikroskopskih slikah [10], segmentacija in detekcija tumorjev v radioloških slikah [11, 12], detekcija ljudi [13] in analiza prometa [14].

V zadnjih letih pri razvoju segmentacijskih in detekcijskih algoritmov prevladujejo konvolucijske nevronske mreže [4, 5, 6, 7, 1, 15, 16, 17]. Naš pristop je baziran na mreži U-Net [15], ki uporablja simetrično polno konvolucijsko mrežo po vzorcu kodirnik-odkodirnik za segmentacijo slik. Kljub razvoju novejših metod [16, 17, 18] dosega U-Net odlične rezultate glede na svojo preprosto arhitekturo. Pristopi, ki uporabljajo arhitekturo U-Net pa dosegajo dobre rezultate tudi na tekmovanjih iz strojnega učenja [19].

Pri našem pristopu za detekcijo uporabimo segmentacijsko nevronske mrežo za generiranje segmentacijske maske, ki jo interpretiramo za pridobivanje detekcij. Pri interpretaciji segmentacijske maske lahko pride do težav pri objektih, ki se med seboj prekrivajo. Področje segmentacije instanc [13, 6, 20, 21] se ukvarja s semantično segmentacijo slik v kateri se pojavi več objektov istega razreda. Cilj je generirati segmentacijsko masko, ki ločuje med posameznimi instancami objektov. Med drugim se segmentacija instanc ukvarja tudi s problemom segmentacije prekrivajočih se objektov. Metode segmentacije instanc pogosto najprej generirajo potencialne regije nato pa znotraj teh regij generirajo segmentacijsko masko za vsebovan objekt. Mask-RCNN [6] razširja arhitekturo mreže Faster-RCNN [4] tako, da poleg klasifikacije objekta in pravokotne regije detekcije, kot izhod poda še segmentacijsko masko v detektirani regiji. Ena izmed slabosti takega pri-

stopa je, da če se dva objekta močno prekrivata, sta pravokotni regiji, ki naj bi bili rezultat pravilne detekcije, skoraj enaki. V takem primeru imajo pristopi, ki segmentacijsko masko generirajo znotraj detektirane regije težave pri razločevanju med prekrivajočimi se objekti. Pri pristopih, ki za generiranje segmentacijske maske instanc uporabljajo regije detektiranih objektov, se lahko pojavi težava tudi zaradi načina predlaganja regij, ki pogosto predlaga le nekaj tisoč potencialnih regij za objekte, vendar pa to v primeru večje količine objektov ni dovolj.

Alternativen pristop k segmentaciji instanc je učenje predstavitev posameznih pikslov v sliki [20, 21, 13]. Cilj je preslikati posamezen piksel v vektorski prostor, v katerem se piksli, ki pripadajo istemu objektu nahajajo blizu skupaj, hkrati pa so ločeni od pikslov, ki pripadajo ostalim objektom. Po preslikavi pikslov v prostor lahko na podlagi predstavitve piksle združimo v posamezne instance. V [13] se problema segmentacije instanc lotijo tako, da predstaviteni prostor pikslov definirajo kot koordinate pravokotnika segmentiranih objektov. Po pridobitvi predstavitev pikslov pridobijo detekcije vseh pikslov, ki v segmentacijski maski predstavljajo lokalne maksimume. Nad detekcijami nato izvedejo proces dušenja ne maksimalnih vrednosti (angl. non-maxima suppression), da pridobijo globalne detekcije posameznih instanc.

Detekcija polipov je sicer zelo ozko področje, tako da ji ni bilo posvečene veliko raziskovalne pozornosti, a vseeno se je z njo ukvarjalo že nekaj del [22, 23, 24, 2]. V [24, 2] pri svoji metodi za detekcijo detekcijo polipov najprej uporabijo metodo ACF [25] za predlaganje regij, nato pa se iz vsake regije pridobi vektor značilik z uporabo mreže AlexNet [26], ki je vnaprej naučena na podatkovni bazi ImageNet [27]. Vektorje, ki predstavljajo predlagane regije nato klasificirajo kot polip ali ozadje z uporabo klasifikatorja SVM [28]. Delo [2] vsebuje tudi analizo uspešnosti detekcije polipov pri ročnih označevalcih in predlaga metodo za evalvacijo detektorja, ki ni odvisna od mere IOU (angl. Intersection Over Union) kot je pogosto pri drugih detektor-

jih ali izzivih [8, 27], saj se osredotoča na štetje polipov, kjer pa je uspešnost prekrivanja anotacij manj pomembna. V [22] je uporabljena metoda ACF [25] za predlaganje regij, ki se jih nato kot v [24] spremeni v vektor značilk z uporabno konvolucijske nevronske mreže in klasificira kot polip ali ozadje. Čeprav se v [24] in [22] uporablja podoben pristop, je v [22] za pridobivanje značilk uporabljena nevronska mreža LeNet [29, 31], ki je manj kompleksna kot mreža AlexNet [26] poleg tega pa je bila mreža naučena na podatkovni bazi MNIST [30], ki vsebuje slike ročno napisanih znakov zaradi česar bi lahko bile naučene značilke manj informativne na danem problemu. Pristop v [22] dosega manjšo klasifikacijsko točnost kot v [24], vendar pa je pristope težko primerjati saj se deli razlikujeta v metodologiji evalvacije poleg tega pa se je uporabljena podatkovna baza slik polipov med njima razlikovala.

V delu [23] se avtorji osredotočajo na štetje polipov z odstranjevanjem ozadja iz slik z uporabo ročno oblikovanih značilk, ki jih uporabijo kot attribute pri učenju večnivojskega perceptrona. Po odstranitvi ozadja se iz preostalih regij generira približna segmentacijska maska nato pa se izvede metoda drsečega okna na različnih velikostih, kjer se kot polip klasificira okno, ki vsebuje manj kot 5% praznega prostora na vsaki lokaciji v segmentacijski maski.

Pogosto se pri ocenjevanju delovanja metod detekcije objektov uporablja mere in podatkovne zbirke različnih izzivov kot sta ImageNet [27] in COCO [8]. ImageNet za primarno mero pri svojem izzivu uporablja povprečno natančnost AP (angl. average precision). AP izračunajo tako, da spreminjajo mejo prekrivanja IOU (angl. intersection over union) pri kateri se detekcija še šteje za pravilno. Za vsako mejo IOU nato izvedejo povezovanje detekcij z anotacijami in izračunajo preciznost detektorja. Tudi izziv COCO uporablja mero povprečne natančnosti, pomembna mera pa je tudi povprečni priklic AR (angl. average recall) pri katerem se omeji število generiranih detekcij nato pa se izračuna priklic, ki ga detektor lahko doseže pri posamezni omejitvi. Povprečni priklic sicer bolje oceni sposobnost metode za reševanje problema predlaganja regij. Mero povprečne natančnosti in priklica izziv

COCO oceni še na objektih različne velikosti, s čimer se oceni še uspešnost metode pri detekciji majhnih, srednjih ali velikih objektov.

1.4 Prispevki

Glavni prispevek tega dela je konvolucijska nevronska mreža za segmentacijo slik polipov in metoda za detekcijo polipov, ki za uspešno detekcijo uporablja interpretacijo segmentacijske maske pridobljene s predlagano segmentacijsko nevronske mreže. Izvedena je bila tudi primerjava večih pristopov k problemu detekcije polipov, pri čemer je bil vsak pristop ovrednoten glede na uspešnost reševanja danega problema.

Drugi prispevek je razvoj aplikacije, ki nudi učinkovit in preprost uporabniški vmesnik za učenje modelov predlagane metode in detekcijo polipov. Aplikacija omogoča hitro anotiranje in shranjevanje anotacij na poljubnih slikah. Vmesnik za delo z anotacijami vključuje orodja za preprosto dodajanje in odstranjevanje pozitivnih in negativnih anotacij, omogočeno pa je tudi označevanje interesne regije znotraj katere se izvaja učenje in inferenca modela. Uporabnik lahko detekcijsko metodo z relativno dobro uspešnostjo nauči tudi na manjših množicah slik, morebitne napačne detekcije pa preprosto odstrani z uporabo funkcij vmesnika.

Kljub trudu človeških označevalcev je v množici anotacij nezanemarljiv del polipov neanotiranih. Zaradi pogoste slabe osvetljenosti, izostrenosti ali visoke stopnje prekrivanja z drugimi polipi je z ročnim pregledom zgrešene polipe težko odkriti, vendar pa jih je lahko prepoznati, če regijo kjer se tak polip nahaja, natančno pregledamo. Detekcijske metode lahko tako uporabimo kot metode predlaganja regij za ročno anotacijo, kjer lahko ročno preverimo vsako detekcijo, ki je množica anotacij ne vsebuje. Dodaten prispevek magistrskega dela je tako še izboljšana testna množica slik z anotacijami predlagana v [2], ki vsebuje bistveno manj napačnih ali zgrešenih anotacij in

omogoča bolj natančno evalvacijo algoritmov za detekcijo polipov na slikah.

1.5 Struktura dela

V Poglavju 2 je opisano delovanje uporabljenih metod. Opisano je delovanje konvolucijskih nevronske mreže na katerih temelji predlagana metoda. Predlagana metoda je natančneje opisana v Poglavju 3. V Poglavju 4 je opisan razviti program, ki predlagano metodo uporablja, v Poglavju 5 pa je podrobno opisana eksperimentalna analiza, s katero smo ovrednotili predlagano metodo in jo primerjali s sorodnimi deli.

Poglavje 2

Konvolucijske nevronske mreže

To poglavje opisuje delovanje nevronskih mrež, ki predstavljajo jedro predlagane metode in metod s katerimi naš pristop primerjamo. Opisuje postopek učenja nevronskih mrež s poudarkom na konvolucijskih nevronskih mrežah. Poglavje 2.1 opisuje splošno sestavo nevronskih mrež, Poglavje 2.2 opisuje delovanje in značilnosti konvolucijskih nevronskih mrež, Poglavje 2.3 pa opisuje še ostale uporabljene operacije, ki so potrebne za dobro delovanje konvolucijskih nevronskih mrež. V Poglavju 2.4 so opisane še uporabljene arhitekture konvolucijskih nevronskih mrež.

2.1 Nevronske mreže

Nevronske mreže so razred učnih algoritmov, ki se kot veliko drugih algoritmov strojnega učenja uporabljajo za učenje kompleksnih funkcij, ki preslikujejo vhodne podatke in v aproksimacijo željenih izhodnih podatkov. Konvolucijske nevronske mreže predstavljene v [31] se od klasičnih nevronskih mrež razlikujejo v načinu obdelave vhodnih podatkov in strukturi jedrnih učnih elementov, nevronov. V zadnjih letih dosegajo konvolucijske nevronske mreže odlične rezultate na problemih s kompleksnejšimi tipi podatkov kot so slike, video vsebine, tekst in zvok.

2.1.1 Delovanje nevronske mreže

Nevronske mreže so sestavljene iz nivojev nevronov. Vsak nevron sprejme poljuben vhodni signal x , izhodni signal y pa je odvisen od vhodnega signala, od povezav z nevroni iz katerih vhodni signal pride in od aktivacijske funkcije. Preprost primer nevronske mreže je predstavljen na Sliki 2.1. Nevron predstavljen z anotacijo $a_i^{(j)}$ je i -ti nevron v j -tem nivoju nevronske mreže. Vsak nevron je povezan z vsemi nevroni v naslednjem nivoju razen z nevronom $a_0^{(j)}$, ki predstavlja pristranskost (angl. bias.)

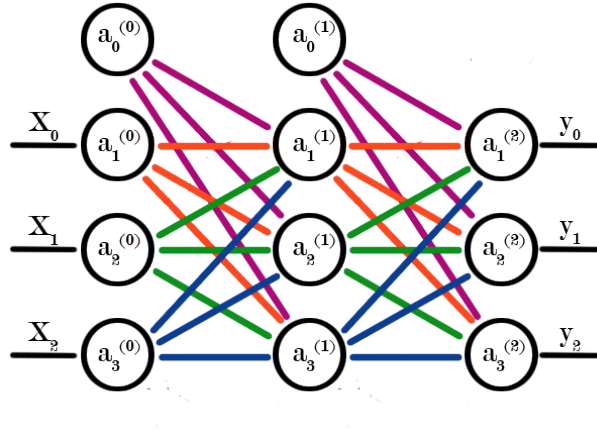
Vrednost posameznega nevrona a_k^j izračunamo kot linearno kombinacijo izhodov nevronov prejšnje stopnje:

$$a_k^j = \sum_i^N x_i^{j-1} w_{(i,k)}^{j-1}, \quad (2.1)$$

kjer je x_i^{j-1} izhodna vrednost vhodnega nevrona i na nivoju $j - 1$, $w_{(i,k)}^{(j-1)}$ pa utež s katero je določena povezava med nevronom k na nivoju (j) in nevronom i na nivoju ($j - 1$). a_n^j je vrednost k -tega nevrona na nivoju (j). Da dobimo izhodno vrednost nevrona, moramo vrednost transformirati s poljubno aktivacijsko funkcijo σ . Aktivacijska funkcija je zvezna in odsekoma odvedljiva nelinearna funkcija, ki v nevronske mreže vpelje nelinearnost kar omogoča modeliranje kompleksnejših funkcij. Končni izhod iz nevrona torej izračunamo tako, da vrednost iz (2.1) transformiramo s funkcijo $\sigma(\cdot)$

$$y_k^j = \sigma(a_k^j) = \sigma\left(\sum_i^N x_i^{j-1} w_{(i,k)}^{j-1}\right). \quad (2.2)$$

Nevroni $a_0^{(j)}$ so neodvisni od izhodov nevronov iz prejšnjega nivoja, kot lahko vidimo na Sliki 2.1. Imajo tudi konstantno aktivacijsko vrednost 1 in predstavljajo konstantni člen v linearni kombinaciji.



Slika 2.1: Preprost primer nevronske mreže. Neuron predstavljen z anotacijo $a_i^{(j)}$ je i -ti nevron v j -tem nivoju nevronske mreže.

2.2 Konvolucijske nevronske mreže

Konvolucijske nevronske mreže se od polno povezanih razlikujejo po načinu delovanja, saj za pridobivanje izhodov nevronov uporabljajo konvolucijo z naučenimi filtri.

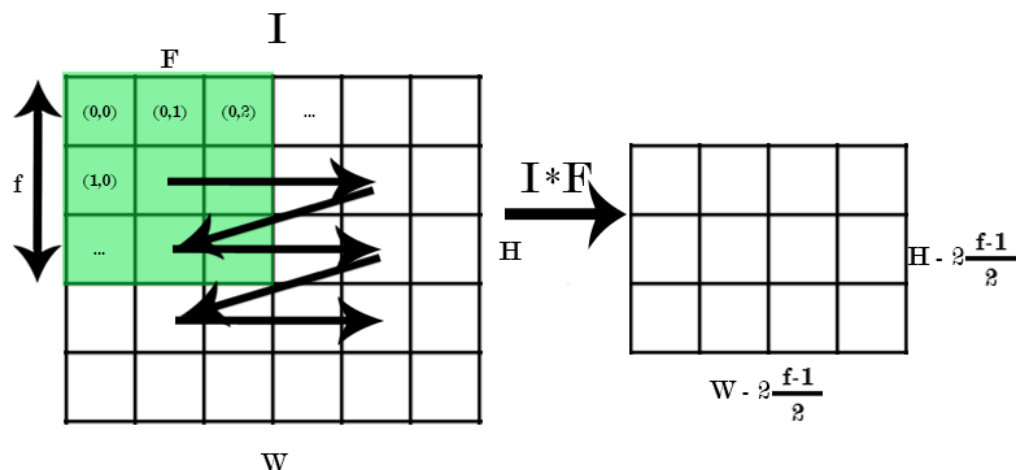
2.2.1 Konvolucija

Konvolucija je operacija med dvema poljubnimi signali. Rezultat konvolucije je definiran kot integral produkta obeh funkcij. Za poljubni fuknciji f in h je konvolucija definirana kot

$$(f * h)(x) = \int_{-\infty}^{\infty} f(y)h(x - y)dy, \quad (2.3)$$

kje je x točka na signalu. Pri uporabi diskretnih signalov seveda uporabljamo diskretno konvolucijo

$$(f * h)(x) = \sum_{y=-\infty}^{\infty} f[y]h[x - y]. \quad (2.4)$$



Slika 2.2: Grafični prikaz delovanja konvolucije na sliki I z višino H in širino W , s filtrom F velikosti $f = 3$.

Pri obdelavi slik uporabljamo konvolucijo s filtri poljubne velikosti. Konvolucijo med filtrom f z dimenzijami $M \times N$ in sliko I na poziciji (x, y) zapišemo kot

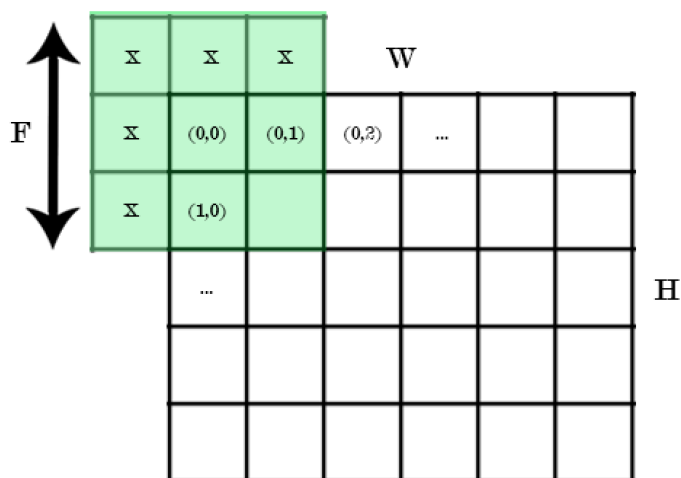
$$(I * f)(x, y) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I[x - k, y - l] f[k, l], \quad (2.5)$$

vendar imajo slikovni vhodi v konvolucijski sloj pogosto več dimenzij, tako da se enačba spremeni v:

$$(I * f)(x, y) = \sum_{c=0}^{C-1} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I_c[x - k, y - l] f_c[k, l], \quad (2.6)$$

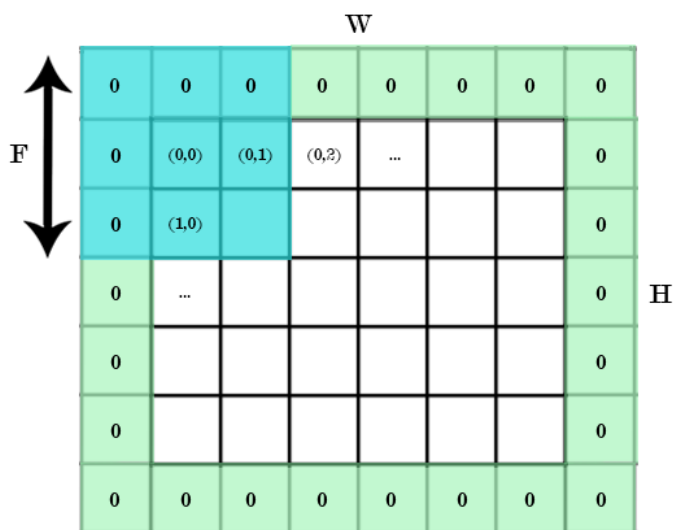
kjer je I_c c -ti kanal vhoda I , f_c pa c -ti kanal konvolucijskega filtra.

Kot lahko vidimo na Sliki 2.2 je rezultat konvolucije s filtrom dimenzij



Slika 2.3: Ponazoritev težave pri izračunu robnih vrednosti. Z x so označeni piksli, ki se nahajajo izven slike, filter pa jih še vedno prekriva.

$(N_f \times N_f)$ manjši od vhodne slike za $2 \frac{(N_f-1)}{2}$ po višini in širini, saj na robovih slike ne moremo izračunati vrednosti po (2.5), saj se v teh primerih konvolucijsko jedro ne prekriva z vhodno sliko, in bi za izračun izhodne vrednosti na teh pozicijah potrebovali vrednosti izven slike, ki pa seveda ne obstajajo. Rešitev za ta problem je dodajanje vrednosti pikslom, ki jih filter prekriva a se nahajajo izven slike. Prikaz problema lahko vidimo na Sliki 2.3, kjer so z x označeni piksli, ki jih prekriva filter, slika pa ne. Za nadomeščanje manjkajočih vrednosti obstaja več strategij, vendar pa je v vsakem primeru vrednost konvolucije na robovih slik slabše definirana. Pogosto se za potrebe nadomeščanja vrednosti sliko podloži z ničlami, kot lahko vidimo na Sliki 2.4, kjer so z zeleno barvo označene podložene vrednosti na sliki, ki nam omogočajo izvajanje konvolucije tako, da je izhodna slika enake velikosti kot vhodna. Takemu nadomeščanju vrednosti pravimo podloga slike (angl. padding).

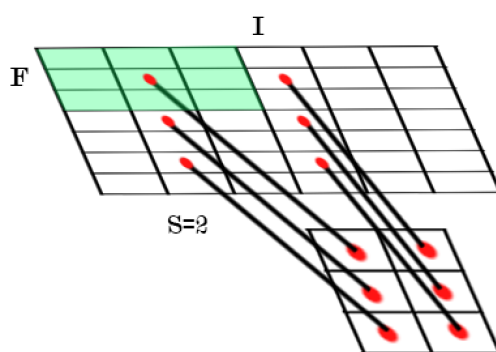


Slika 2.4: Primer podloge slike z ničlami. Z zeleno barvo so označene podložene vrednosti, z modro barvo pa je označeno še konvolucijsko jedro z velikostjo $F = 3$ na poziciji $(0, 0)$ oziroma na zgornjem levem kotu slike.

2.2.2 Konvolucijski sloj

Pri konvolucijskih slojih v nevronske mrežah se konvolucija izvaja po (2.6). Izhod iz konvolucijskega sloja je poleg vhoda odvisna tudi od filtrov sloja in večih parametrov, ki določajo kako se konvolucija izvaja:

- **Dodajanje obrobe** (angl. padding) Izhod konvolucije se za pozicijo v sliki (x, y) izvede po Enačbi 2.6, kjer filter f postavimo na pozicijo tako, da je njegov center na poziciji (x, y) . Kot lahko vidimo na Sliki 2.3 je za robne pozicije v sliki izračun konvolucije s filtrom neizvedljiv, saj bi potrebovali vrednosti slike na pozicijah, ki ne obstajajo. Za reševanje tega problema lahko robne vrednosti slike preprosto zanema-



Slika 2.5: Primer delovanja konvolucije z vhodno sliko I in s filtrom F velikosti 3, ki deluje s korakom $S = 2$. Z rdečo barvo so označene pozicije na katerih se izvede korak konvolucije.

rino in konvolucijo izvedemo na lokacijah kjer je izračun mogoč. V tem primeru je zaradi izpuščenih robnih pikselov rezultat konvolucije manjši od vhodne slike.

- **Korak** (angl. stride), ki določa način premikanja konvolucijskega filtra po sliki, kar vpliva na dimenzije izhoda konvolucije. Pogosto se uporablja korak velikosti 1, kar pomeni, da se po (2.6) obdela vsako lokacijo na sliki (x, y) . V primeru, da bi se pri konvoluciji uporabil korak velikosti 2, bi se (2.6) uporabila le na vsaki drugi lokaciji torej na

$$(x + 2k, y + 2h), k \in [0, \lfloor \frac{W}{2} \rfloor], h \in [0, \lfloor \frac{H}{2} \rfloor], \quad (2.7)$$

kjer sta W in H širina in višina vhodne slike. Seveda to tudi spremeni izhod operacije saj pri koraku velikosti K dobimo izhod s širino

$$W_{izhod} = \lfloor \frac{W}{K} \rfloor \quad (2.8)$$

in višino:

$$H_{izhod} = \lfloor \frac{H}{K} \rfloor. \quad (2.9)$$

- **Velikost filtrov:** Konvolucijski filtri so lahko različne velikosti. Večji filtri lahko sicer pripomorajo pri odkritju večjih in kompleksnejših struktur a to pomeni, da vsebujejo filtri bistveno več vrednosti. Število

vrednosti v filtru raste s kvadratom velikosti filtra torej filter z velikostjo $N_f = 3$ vsebuje 9 parametrov, z velikostjo $N_f = 4$ pa že 16. Zaradi računske kompleksnosti konvolucije z uporabo večjih filtrov se v modernih konvolucijskih nevronske mrežah uporablja zaporedna vezava konvolucijskih slojev z manjšimi filtri, kar omogoča posnemanje konvolucije z večjimi filtri z manjšo dodano kompleksnostjo [32, 33].

- **Število filtrov** določa izhod iz konvolucijskega sloja. Z vsakim od K filtrov se izvede konvolucija z vhom zaradi česar ima izhod iz konvolucijskega sloja obliko (W, H, K) , kjer sta W in H širina in višina vhoda v primeru, da uporabljamo ustrezno podlogo slike. Število filtrov je hiperparameter, ki ga prilagodimo glede na problem, ki ga rešujemo, ter strojno opremo na kateri se bo nevronska mreža izvajala. Preveliko število filtrov lahko pripomore k prekomernemu prilagajanju podatkov (angl. overfitting) in zvišuje kompleksnost modela, premajhno število filtrov pa lahko rezultira v pretirano preprostem modelu, ki ne uspe dovolj dobro modelirati funkcije, ki povezuje vhodne in izhodne podatke in problem zadovoljivo rešuje.

2.2.3 Parametri v konvolucijskih nevronske mrežah

V polno povezanih nevronske mrežah je vsak nevron povezan z vsemi nevroni v naslednjem sloju, torej sta 2 nivoja, ki sta sestavljena vsak iz N nevronov, med seboj povezana z N^2 povezav. V tem primeru se mora mreža za vsak nivo naučiti N^2 uteži. V konvolucijskih nevronske mrežah za obdelavo slik pa je izhod vsakega nivoja predstavitev slike, ki je pogosto odvisna od dimenzij vhodne slike. Na vhodnem nivoju konvolucijske nevronske mreže je pogosto slika v *RGB* prostoru dimenzije $(W, H, 3)$, kjer je W širina, H višina, 3 pa je število barvnih kanalov. V primeru, da bi želeli tak vhod obdelati v polno povezani nevronske in bi želeli da sledeči nivo vsebuje enako število nevronov kot vhodni nivo, bi potrebovali $(3WH)^2$ povezav. To seveda ni sprejemljivo zato se za obdelovanje slik uporablja operacija konvolucije, parametri, ki

se jih mora mreža naučiti pa so neodvisni od dimenzij slike (H, W) . Tako ima prvi nivo, ki uporablja F konvolucijskih filtrov velikosti 5×5 in na vhod dobi sliko s tremi barvnimi kanali, število uteži $N = F \times 5 \times 5 \times 3$. Ob tem uporabimo predpostavko, da naučeni filter prispeva k informativni reprezentaciji slike ne glede na pozicijo v sliki, saj ga s konvolucijo apliciramo na celotno sliko.

2.2.4 Kriterijska funkcija

Pri ocenjevanju uspešnosti med učenjem se nevronske mreže ne razlikujejo bistveno od drugih algoritmov strojnega učenja, ki pri učenju uporabljajo gradientni spust. Za ocenjevanje napake moramo definirati kriterijsko funkcijo $\lambda(\cdot)$, ki ocenjuje razliko med izhodno vrednostjo nevronske mreže in željeno izhodno vrednostjo. Zaradi različnih nalog, ki jih nevronske mreže lahko opravljajo, obstaja več kriterijskih funkcij, ki ocenjujejo napako pri opravljanju določene naloge. Kriterijsko funkcijo pri učenju s postopkom optimizacije poizkušamo minimizirati.

2.2.5 Učenje konvolucijske nevronske mreže

Učenje nevronskih mrež se izvaja s postopkom vzratnega prehoda (angl. backpropagation) [34, 35], pri katerem se trenutna napaka oziroma vrednost kriterijske funkcije vzvratno razširja po nivojih nevronske mreže.

- **Prehod naprej** Za izvajanje inference s konvolucijskimi nevronskimi mrežami za obdelavo slik moramo najprej izvesti konvolucijo z konvolucijskimi jedri prvega nivoja, nato pa značilke, ki nastanejo kot rezultat konvolucije, uporabiti kot vhod v naslednji nivo. Postopek ponavljamo do zadnjega nivoja, katerega izhod je končni rezultat nevronske mreže za vhodno sliko.

Izhod konvolucije nivoja l z^l je odvisen od izhoda prejšnjega nivoja o^{l-1}

in filtrov nivoja w^l

$$z_{(x,y,c')}^l = o_{(x,y)}^{l-1} * w_{c'}^l + b_{c'}^l, \quad (2.10)$$

kjer je o^l izhod nivoja l , $w_{c'}^l$ pa označuje filter, ki se nahaja na nivoju l , vpliva pa na izhodni kanal c' , $b_{x,y}^l$ označuje konstanten člen.

Enačbo lahko zapišemo po vzoru (2.6)

$$z_{(x,y,c')}^l = \sum_{c=0}^{C-1} \sum_{k=0}^{M-1} \sum_{h=0}^{N-1} o_{(x-k,y-h,c)}^{l-1} w_{c'}^l(k,h,c) + b_{c'}^l, \quad (2.11)$$

kjer z $w_{c'}^l$ označimo enega izmed filtrov nivoja l . Z $w_{c'}^l(k,h,c)$ označimo element filtra $w_{c'}^l$ na poziciji (k,h,c) , kjer indeksa k in h označujeta prostorske dimenzije, indeks c pa označuje kanal posameznega filtra. Število kanalov C je odvisno od števila kanalov izhoda prejšnjega nivoja o^{l-1} .

Za pridobitev izhoda nivoja o^l nad rezultatom konvolucije uporabimo še aktivacijsko funkcijo

$$o_{(x,y,c')}^l = \sigma(z_{(x,y,c')}^l), \quad (2.12)$$

kjer $\sigma(\cdot)$ označuje aktivacijsko funkcijo nivoja.

- **Vzratni prehod** Za optimiziranje uteži filtrov mreže moramo napako vzvratno propagirati po nivojih nevronske mreže. Pri uporabi gradientnega spusta, uteži spreminjamo glede na smer gradienta funkcije napake E . Posodabljanje uteži s preprostim gradientnim spustom lahko zapišemo kot

$$W_k = W_{k-1} - \eta \frac{\delta E}{\delta W}, \quad (2.13)$$

kjer η označuje stopnjo učenja, $\frac{\delta E}{\delta W}$ parcialni odvod funkcije napake po utežeh, W_k pa označuje vrednosti uteži v iteraciji k [35]. Zahtevani

parcialni odvod v konvolucijskih nevronske mrežah na nivoju l lahko za posamezen element posameznega filtra $w_{c'}^l$ izračunamo z

$$\frac{\delta E}{\delta w_{c'}^l(k,h,c)} = \sum_{x=0}^{H-f_1} \sum_{y=0}^{W-f_2} \frac{\delta E}{\delta o_{(x,y,c')}^l} \frac{\delta o_{(x,y,c')}^l}{\delta z_{(x,y,c')}^l} \frac{\delta z_{(x,y,c')}^l}{\delta w_{c'}^l(k,h,c)}, \quad (2.14)$$

kjer je $\frac{\delta o_{(x,y,c')}^l}{\delta z_{(x,y,c')}^l}$ odvod (2.12) in je odvisen od odvoda aktivacijske funkcije σ , $\frac{\delta z_{(x,y,c')}^l}{\delta w_{c'}^l(k,h,c)}$ pa je preprost odvod Enačbe 2.11. Indeks c' označuje številko filtra v množici filtrov w^l na nivoju l . Za izračun potrebujemo še parcialni odvod napake E po izhodu nivoja $\frac{\delta E}{\delta o_{(x,y,c')}^l}$, ki pa ga pridobimo po enačbi

$$\frac{\delta E}{\delta o_{x,y,c'}^l} = \sum_{c=0}^{C-1} \sum_{k=0}^{f_1} \sum_{h=0}^{f_2} \frac{\delta E}{\delta o_{(x-k,y-h,c)}^{l+1}} \frac{\delta o_{(x-k,y-h,c)}^{l+1}}{\delta z_{(x-k,y-h,c)}^{l+1}} \frac{\delta z_{(x-k,y-h,c)}^{l+1}}{\delta o_{(x,y,c')}^l}. \quad (2.15)$$

Vidimo lahko, da je odvod $\frac{\delta o_{(x-k,y-h,c)}^{l+1}}{\delta z_{(x-k,y-h,c)}^{l+1}}$ odvisen od aktivacijske funkcije $\sigma(\cdot)$

$$\frac{\delta o_{(x-k,y-h,c)}^{l+1}}{\delta z_{(x-k,y-h,c)}^{l+1}} = \sigma'(z_{(x-k,y-h,c)}^{l+1}). \quad (2.16)$$

Odvod $\frac{\delta z_{(x-k,y-h,c)}^{l+1}}{\delta o_{(x,y,c')}^l}$ pa lahko pridobimo s preprostim odvajanjem (2.11)

$$\begin{aligned} \frac{\delta z_{(x-k',y-h',c'')}^{l+1}}{\delta o_{(x,y,c')}^l} &= \frac{\delta}{\delta o_{(x,y,c')}^l} \left(\sum_{c=0}^{C-1} \sum_{k=0}^{M-1} \sum_{h=0}^{N-1} o_{(x-k,y-h,c)}^l w_{c''}^{l+1}(k,h,c) + b_{c''}^{l+1} \right) \\ &= w_{c''}^{l+1}(k',h',c''), \end{aligned} \quad (2.17)$$

kjer za lažje ločevanje med oznakami v razširjeni enačbi indekse k , h in c' zapišemo kot k' , h' in c'' . Tako odvod $\frac{\delta z_{(x-k,y-h,c)}^{l+1}}{\delta o_{(x,y,c')}^l}$ zapišemo kot $\frac{\delta z_{(x-k',y-h',c'')}^{l+1}}{\delta o_{(x,y,c')}^l}$.

Za izračun (2.15) na nivoju l potrebujemo vrednost istega odvoda na nivoju $l + 1$. Gradient lahko tako propagiramo skozi posamezne nivoje z iterativnim izračunom vrednosti $\frac{\delta E}{\delta w^l}$ in $\frac{\delta E}{\delta o^l}$ od zadnjega do prvega nivoja $l = \{1, \dots, n\}$.

Izračun se razlikuje le ob prvi iteraciji, kjer $\frac{\delta E}{\delta o^n}$ določa odvod po izhodu zadnjega nivoja o^n . Vrednost odvodov zadnjega nivoja je odvisna od uporabljene kriterijske funkcije.

2.3 Drugi sloji in operacije konvolucijskih nevronske mreže

Poleg konvolucijskih nivojev se v konvolucijskih mrežah uporabljajo tudi številne druge operacije, ki pripomorejo k boljšemu učenju in delovanju nevronske mreže.

2.3.1 Združevalni sloji

Pogosto želimo s konvolucijo zajeti informacije v širšem območju. Pridobiti želimo specifične značilke, ki prispevajo k opisu večjega dela objekta, na primer očesa, pri detekciji obraza, kar pa težko dosežemo s konvolucijskimi filtri velikosti 3×3 , ki pokrivajo bistveno manjše območje kot ga zavzema iskani del. Eden izmed načinov za povečavo vidnega polja konvolucije je zaporedna vezava večje količine konvolucijskih slojev, vendar pa to povečuje kompleksnost modela. Za povečavo vidnega polja konvolucije se pogosto uporabljajo združevalni sloji, ki svojemu vhodu spremenijo velikost, kar povzroči, da značilke, ki opisujejo iskani del objekta zavzemajo bistveno manjši del slike zaradi česar lahko z manjšimi konvolucijskimi filtri dosežemo boljši odziv na lokaciji iskanega dela. Zaradi vpliva na velikost reprezentacije slike se zmanjša tudi računski kompleksnost ob inferenci in učenju [36].

Pri združevalnih slojih se uporablja več metod združevanja med popularnejšimi sta predvsem združevanje maksimumov in povprečno združevanje:

- **Združevanje maksimumov** (angl. max pooling) vhodno sliko razdeli na podregije enake velikosti, vsaka regija pa predstavlja en piksel v izhodni sliki, ki mu pripišemo maksimalno vrednost v podregiji. Z združevanjem maksimumov se bistveno zmanjša velikost vhodne predstavitve slike, tako so pri združevanju maksimumov, ki uporablja podregije z velikostjo 2×2 , dimenzije vhodne slike (W, H, C) , izhodna slika pa ima dimenzije $(\frac{W}{2}, \frac{H}{2}, C)$. Vidimo lahko, da se spremenita dimenziji W in H , število kanalov C pa ostane nespremenjeno saj se združevanje maksimumov izračuna za posamezne vhodne kanale. Pri zmanjšanju velikost predstavitve slike je skoraj neizogibna izguba majhnega deleža informacije o lokaciji posameznih značilk, hkrati pa to doda robustnost na majhne spremembe v poziciji značilk.
- **Povprečno združevanje** (angl. average pooling) je alternativa združevanju maksimumov. Namesto maksimuma v posamezni podregiji vrednost izhodnega piksla izračuna tako, da povpreči vrednosti pikselov v pripadajoči podregiji. V praksi se večkrat uporablja združevanje z maksimumi saj pogosto doseže boljše rezultate od združevanja s povprečjem [36].

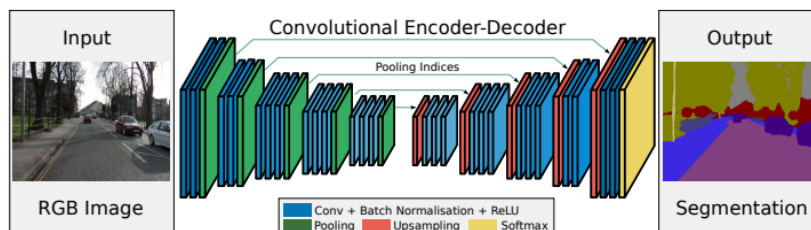
Več modernih arhitektur nevronske mreže se izogiba uporabi združevalnih slojev [37, 38]. Številni pristopi jih nadomeščajo s konvolucijskimi sloji, ki uporabljajo korak večji od 1. Pogosto uporabljeno združevanje s podregijami velikosti (2×2) lahko nadomestimo s konvolucijskim slojem s korakom velikosti 2. Pristop združevanja s konvolucijskim slojem doda kompleksnost v model saj se mora naučiti še funkcije združevanja. V praksi so rezultati združevanja maksimumov in združevanja s konvolucijskim slojem primerljivi [39].

2.3.2 Povečevanje resolucije značilnic

Pri modernih arhitekturah uporabljamo sloje združevanja za boljše učenje kompleksnejših značilnic, vendar pa globlje konvolucijske nevronske mreže pogosto uporabljajo več slojev združevanja, ki velikost predstavitev vhodne slike bistveno zmanjšajo. Pri problemih kot so semantična segmentacija, kjer je cilj generiranje segmentacijske maske katere velikost je enaka velikosti vhoda, moramo manjšo predstavitev slike povečati do primerne velikosti. Sloji povečevanja resolucije značilnic skrbijo za povečevanje predstavitev prejšnjega nivoja. Za povečevanje resolucije se uporabljajo različne funkcije povečevanja, ki pa so lahko ročno definirane ali pa naučene.

- **Bilinearna interpolacija** je povprečje vrednosti sosednjih pikslov, uteženo z razdaljo interpolirane točke do ostalih. Za izvedbo razširitve slike velikosti $(N \times N)$ na $(2N \times 2N)$ najprej izvedemo linearno interpolacijo za izračun vrednosti vmesnih pikslov po eni izmed osi torej po vrsticah ali stolpcih, nato pa te vrednosti uporabimo še za interpolacijo po drugi osi.
- **Interpolacija najbližjih sosedov** je pristop k povečevanju predstavitev, ki prepíše vrednost iz sosednjih pikslov. V primeru, da želimo sliko velikosti $(N \times N)$ povečati na velikost $(2N \times 2N)$ to z interpolacijo najbližjih sosedov dosežemo tako, da vsako vrstico in stolpec v sliki podvojimo.
- **Razširitev z indeksi** (angl. max unpooling): Pristop uporabimo v paru z operacijo združevanja z maksimumi, kjer si za vsako podregijo v predstavitvi zapomnimo na kateri lokaciji se je nahajal maksimum, ki smo ga uporabili pri združevanju. Predstavitev razširimo tako, da vsak piksel predstavitve prepíšemo na lokacijo maksimuma v pripadajoči podregiji, ostale vrednosti v razširjeni predstavitvi so enake 0. Ta pristop se uporablja v konvolucijskih mrežah z arhitekturo koder dekodek (angl. encoder-decoder), kjer se operacije združevanja in razširitev pojavljajo

v paru. Primer take mreže je segmentacijska mreža Segnet [16], arhitekturo katere lahko vidimo na Sliki 2.6.

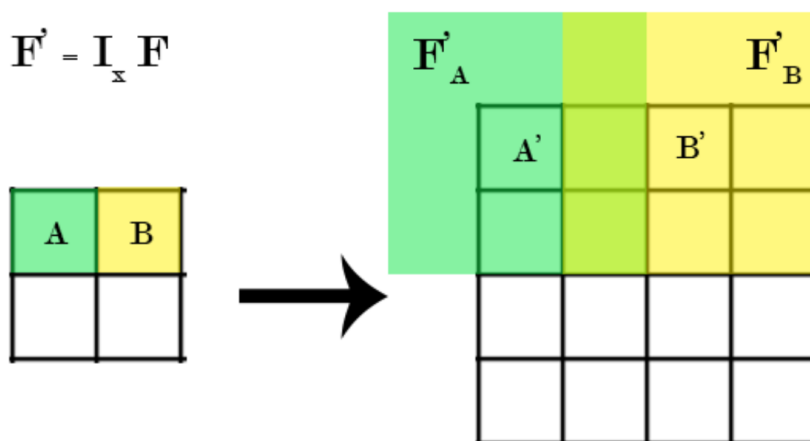


Slika 2.6: Arhitektura segmentacijske konvolucijske nevronske mreže Segnet, ki za operacije razširitve (angl. upsampling) uporablja razširitev z indeksi. Slika povzeta po [16].

- **Transponirana konvolucija** (angl. transposed convolution) Metodo transponirane konvolucije si lahko predstavljamo kot neke vrste konvolucijo s korakom manjšim od 1, če si korak predstavljamo kot razmerje v velikosti med vhodno in izhodno matriko. Tako bi v primeru, da bi nad vhodno sliko izvedli konvolucijo s korakom $\frac{1}{2}$, kot rezultat operacije dobili izhodno sliko, ki bi bila dvakrat večja od vhodne.

Primer transponirane konvolucije lahko vidimo na Sliki 2.7. Kot v konvolucijskem sloju se tudi v sloju transponirane konvolucije mreža nauči konvolucijskih filtrov. Vzemimo, da je filter F naučeni filter v sloju transponirane konvolucije. Preprosto razložen postopek izvajanja je tak, da z metodo drsečih oken s korakom 1 obdelamo manjšo vhodno sliko, hkrati pa na pripadajočih lokacijah v izhodni sliki vpisujemo vrednosti filtra F pomnoženega s trenutno obdelovano vrednostjo v vhodni sliki. V primeru na Sliki 2.7 lahko za lokacijo A na vhodni sliki I in njeno pripadajočo lokacijo na izhodni sliki A' določimo filter F'_A , ki ga dobimo tako, da vrednost na lokaciji A I_A pomnožimo z naučenim filtrom F . Tako je $F'_A = I_A F$. Ob izvajanju transponirane konvolucije tako na lokaciji A' vpišemo vse vrednosti filtra F'_A v piksele,

ki jih filter postavljen na lokaciji A' pokriva. Na Sliki 2.7 je območje, ki ga pokriva filter F'_A na lokaciji A' ponazorjen z zeleno barvo. Enak postopek ponovimo za vsak korak, vendar pa lahko včasih pride do prekrivanja vrednosti kot lahko vidimo tudi na našem primeru, saj piksli katerih vrednost bi vpisal filter na lokaciji B' prekrivajo s tistimi, ki bi jih vpisal filter na lokaciji A' . Na regijah kjer pride do prekrivanja se vrednosti seštevajo, tako imajo v našem primeru piksli, kjer se F'_A in F'_B prekrivata, vrednost enako vsoti pripadajočih vrednosti obeh filtrov.



Slika 2.7: Primer operacije transponirane konvolucije. Vhodno matriko velikosti 2×2 lahko razširimo v matriko velikosti 4×4 . F je naučeni filter v transponiranem konvolucijskem sloju. A in B sta lokaciji v vhodni matriki, A' in B' pa pripadajoči lokaciji v izhodni matriki.

2.3.3 Normalizacija skupin

Pogosto se vhodno podatkovno bazo normalizira tako, da se od vhodnih podatkov odšteje povprečno vrednost, nato pa rezultat deli s standardnim odklonom. Normalizacijo vhodnih podatkov lahko tako opišemo z

$$x' = \frac{x - \mu}{\sigma}, \quad (2.18)$$

kjer je μ povprečna vrednost, σ pa standardni odklon podatkovne baze za vsako vhodno spremenljivko. Vhodne spremenljivke se lahko po vrednosti bistveno razlikujejo med sabo, namen normalizacije pa je preoblikovanje vhodnih spremenljivk v približno enako območje vrednosti kar bistveno pospeši učenje nevronske mreže.

V prikritih nivojih nevronske mreže se zaradi spreminjajočih se uteži na prejšnjih nivojih distribucija vhodov v posamezen nivo konstantno spreminja, distribucija ciljnih izhodov pa ostaja ista. To otežuje učenje saj lahko povzroča nasičenje nevronov in s tem problem izginjajočega gradienta, kar pa se ponavadi rešuje s primerno metodo inicializacije [40], aktivacijsko funkcijo ReLU [41] in z nizko stopnjo učenja (angl. learning rate), ki pa poveča čas učenja mreže [42]. Z uporabo sloja normalizacije skupin se distribucija vhodov v skrite nivoje stabilizira v zaporednih korakih učenja.

Kot vhod v nevronske mreže pogosto uporabimo več primerov vhodnih podatkov iz baze, v primeru konvolucijskih nevronske mreže, ki na vhod sprejmejo slike lahko kot vhod uporabimo matriko v obliki $(M \times W \times H \times C)$, kjer M predstavlja število slik, ki jih hkrati vstavimo v konvolucijsko nevronske mreže, C pa predstavlja število kanalov vsake od M predstavitev slike. Vhodu take oblike pravimo skupina (angl. batch). Pri postopku normalizacije skupin (angl. batch normalization) [42] izhod nivoja pred aktivacijsko funkcijo normalizira po enačbi

$$\hat{x}_i = \frac{x_i - \mu_S}{\sigma_S}, \quad (2.19)$$

kjer je x_i i -ti član vhodne skupine v nivo normalizacije, μ_S in σ_S pa sta povprečna vrednost in standardni odklon značilnik skupine S . μ_S se izračuna po

$$\mu_S = \frac{1}{M} \sum_{i=1}^M x_i, \quad (2.20)$$

kjer je M število članov skupine. σ_S pa izračunamo po

$$\sigma_S = \sqrt{\frac{1}{M} \sum_{i=1}^M (x_i - \mu_S)^2}. \quad (2.21)$$

Sloj normalizacije skupin ima parametra γ in β , ki se jih mora naučiti in vplivata na izhod iz sloja

$$y_i = \gamma \hat{x}_i + \beta. \quad (2.22)$$

S slojem normalizacije skupine torej izhod iz prejšnjega nivoja najprej normaliziramo glede na povprečno vrednost in standardnim odklonom posameznih značilnk nato pa se izračunan izhod \hat{x}_i skalira s parametrom γ in zamakne s parametrom β . S parametroma γ in β lahko operacijo normalizacije tudi izničimo, če se mreža nauči parametrov vrednosti $\gamma = \sigma_S$ in $\beta = \mu_S$ in se s tem nauči funkcije identitete. Sprememba sloja normalizacije v funkcijo identitete da mreži možnost, da vpliv normalizacije odstrani če to prispeva k minimizaciji cenilne funkcije.

Sloj normalizacije skupine ima več prednosti zaradi česar se ga uporablja v številnih modernih arhitekturah konvolucijskih nevronske mrež [37, 38, 42, 43]:

- Omogoča uspešno učenje z bistveno višjo stopnjo učenja in s tem postopek učenja bistveno pospeši [42].
- Vhodne podatke x_i normalizira z μ_S in σ_S , ki ju izračuna iz celotne skupine S in zaradi nedeterminističnega izbora članov skupine ob učenju v mrežo vpelje majhno stopnjo šuma, ki lahko pomaga pri regularizaciji [42].
- Poveča robustnost mreže na način inicializacije.

2.3.4 Nivo izločevanja

Nivo izločevanja (angl. dropout) [44] je pomembna metoda za regularizacijo nevronske mreže. V nivoju izločevanja se aktivacije nekaterih vhodnih nevronov nastavi na 0 s čemer jih izločimo iz mreže. Verjetnost, da se izhod posameznega nevrona nastavi na 0 je odvisna od parametra ρ , ki ga nastavimo ob kreiranju mreže. Ob vsaki iteraciji učenja izberemo ρ delež nevronov, ki jih izločimo in s tem odstranimo njihov vpliv na vhod v sledeči nivo v prehodu naprej. Izločitev nevronov upoštevamo tudi v vzratnem prehodu zaradi česar so gradienti izločenih nevronov enaki 0. Med fazo testiranja se spremembe uteži, ki nastanejo zaradi učenja nivojev z manj nevroni, aproksimira z množenjem aktivacij vhodnega nivoja z ρ [44]. Operacija nivoja izločevanja je med fazo testiranja le:

$$y = \rho x, \quad (2.23)$$

kjer je y izhod, x pa vhod nivoja izločanja. Tako je izhod iz posameznega nevrona k skupaj z nivojem izločevanja:

$$y_k^j = \rho \sigma \left(\sum_{i=0}^n x_i^{j-1} w_{i,k}^{j-1} \right), \quad (2.24)$$

kjer je σ aktivacijska funkcija, w^{j-1} in x^{j-1} uteži in aktivacije nevronov prejšnjega nivoja $j - 1$, ρ pa delež izločenih nevronov v nivoju j .

2.4 Uporabljene arhitekture

2.4.1 Arhitektura U-Net

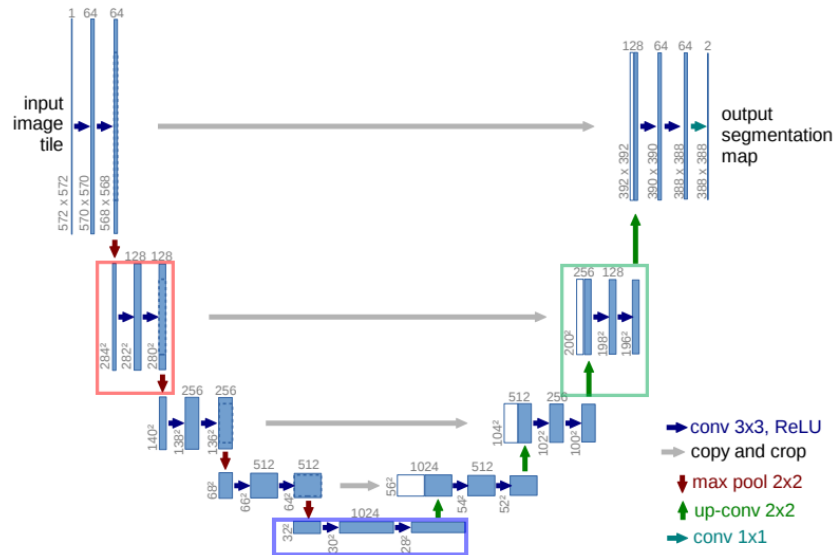
Mreža U-net je bila razvita in uporabljena v [15] za semantično segmentacijo biomedicinskih slik. Njeno arhitekturo lahko vidimo na Sliki 2.8, kjer vidimo, da je sestavljena iz združevalne in razširjevalne poti. Združevalna in razširjevalna pot sta sestavljeni iz zaporedno vezanih konvolucijskih blokov:

- **Združevalni blok** je sestavljen iz dveh konvolucijskih nivojev in nivoja združevanja maksimumov. Združevalna pot je sestavljena iz štirih

združevalnih blokov. Na Sliki 2.8 je primer združevalnega bloka označen z rdečim pravokotnikom. V vsakem sledečem združevalnem bloku se število filtrov, ki jih uporabljajo konvolucijski nivoji bloka, podvoji. Konvolucijski nivoji v prvem bloku imajo tako 64 filtrov, v drugem 128, tretjem 256 v četrtem pa 512. Tak način združevanja bistveno zmanjša prostorske dimenzije predstavitev slik na globljih nivojih in s tem omogoča boljše predstavitev kompleksnejših značilk. Večanje števila filtrov v slojih je pomembno zaradi postopnega povečevanja kompleksnosti značilk, ki jih nivoji predstavljajo.

- **Zgoščena predstavitev** je izhod konvolucijskega nivoja, ki sledi združevalni poti in ponovno vsebuje dvakrat večje število filtrov, kot konvolucijski sloji zadnjega združevalnega bloka. Omenjeni nivo je na Sliki 2.8 označen z modrim pravokotnikom.
- **Razširitveni blok** je sestavljen iz nivoja razširitve in dveh konvolucijskih nivojev. Kot v združevalnih blokih se tudi v zaporednih razširitvenih blokih spreminja število filtrov, le da se tokrat število filtrov v vsakem sledečem bloku razpolovi. Na Sliki 2.8 je primer razširitvenega bloka označen z zelenim pravokotnikom.

Mreže FCN [45] se po podobni združevalni poti oziroma po sloju zgoščene predstavitve doda še en konvolucijski nivo s filtri velikosti 1×1 , ki iz zgoščene predstavitve generira segmentacijsko masko z nizko resolucijo. Generirano segmentacijsko masko se nato uporabi kot vhod v razširitveni nivo, ki jo preoblikuje v velikost enako ciljni segmentacijski maski. Zaradi velike stopnje razširitve z enim razširitvenim nivojem je segmentacijska maska groba in posledično relativno nenatančna. Za izboljšanje kvalitete segmentacijske maske [45] predlaga še preskočne povezave (angl. skip connections), ki za napovedovanje segmentacijske maske uporabijo značilke v višjih nivojih, kjer so dimenzije predstavitev višje, in jih seštejejo s preoblikovanimi segmentacijskimi maskami, ki so bile generirane z uporabo značilk nižjih nivojev.



Slika 2.8: Arhitektura mreže U-net. Na desni strani nivojev so označene prostorske dimenzije označenih nivojev v obliki $W \times H$ za vhodno sliko dimenzij 572×572 . Na vrhu nivojev je označeno število konvolucijskih filtrov za nivo. Z rdečimi puščicami so označeni nivoji združevanja maksimumov, z zelenimi puščicami pa nivoji razširjanja. Sive puščice prikazujejo operacijo kopiranja kanalov značilnik na posameznem nivoju. Slika je bila prilagojena po delu [15].

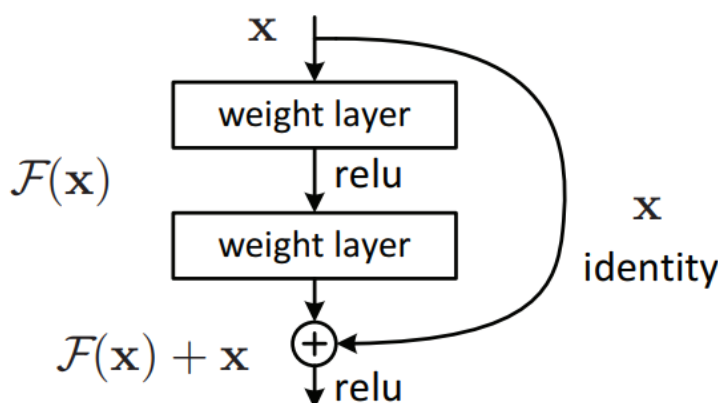
V mreži U-Net [15] se po pridobitvi zgoščene predstavitve uporabljajo razširitveni bloki. Arhitektura mreže U-Net je simetrična in število razširitvenih blokov je enako številu združevalnih blokov.

Kot FCN [45] tudi mreža U-Net [15] uporablja preskočne povezave. Na Sliki 2.8 so preskočne povezave označene s sivimi puščicami. Vidimo lahko, da preskočne povezave povezujejo združevalne bloke s pripadajočimi razširitvenimi bloki. V vsakem razširitvenem bloku se izhod razširitvenega sloja združi z značilkami 2. konvolucijskega sloja pripadajočega združevalnega bloka. Značilke se združi po dimenziji kanalov, kot lahko vidimo na Sliki 2.8, kjer se 2. konvolucijski sloj združevalnega bloka označenega z rdečim pravokotnikom

združi z izhodom razširitvenega nivoja razširitvenega bloka označenega z zelenim pravokotnikom. Združene značilke obeh slojev uporabimo kot vhod v naslednji konvolucijski sloj razširitvenega bloka. Postopen način prostorskega razširjanja predstavitve in kombiniranje značilk iz prejšnjih nivojev preko preskočnih povezav omogoča boljši pretok informacij o kontekstu objektov, ki so pomembne za generiranje natančne segmentacijske maske z pri višjih resolucijah [15].

2.4.2 Arhitektura ResNet

Število nivojev pri zmagovalnih mrežah klasifikacijskega izziva ILSVRC [27] je skozi čas naraščalo. Zmagovalne mreže izziva kot so [26], [46], [32] so bile sestavljene iz 8, 19 in 22 nivojev. Z rastočim številom nivojev se sicer povečuje zmožnost modeliranja kompleksnejših funkcij hkrati pa je pri globljih mrežah problem izginjajočih gradientov bolj izrazit zaradi težjega propagiranja napake skozi mrežo pri koraku vzratnega prehoda. Glavni prispevek arhitekture mreže ResNet [37] je učinkovito obvladovanje problema izginjajočih gradientov pri učenju mreže.



Slika 2.9: Blok v mreži ResNet. Slika povzeta po [37].

Na Sliki 2.9 lahko vidimo sestavni blok mreže ResNet. Sestavljen je iz konvolucijskega nivoja z aktivacijsko funkcijo ReLu, drugega konvolucijskega

nivoja brez aktivacije, seštevalne funkcije, ki sešteje vhod x in izhod iz drugega konvolucijskega nivoja, ter aktivacijske funkcije ReLu [41], ki na vhod prejme izhod seštevalne funkcije. Vidimo lahko, da je med vhom bloka ter seštevalne funkcije preskočna povezava. Cilj bloka brez preskočne povezave je preslikati vhod x v neko informativno predstavitev y z uporabo naučene preslikave \mathcal{H}

$$y = \mathcal{H}(x). \quad (2.25)$$

Izkaže se, da je v globokih mrežah nivoje težko dovolj dobro optimizirati, da zadovoljivo modelirajo \mathcal{H} . Rešitev, ki jo za problem predlaga ResNet [37] je torej dodatek preskočne povezave, ki funkcijo \mathcal{H} spremeni v

$$\mathcal{H}(x) = \mathcal{F}(x) + x. \quad (2.26)$$

Zaradi preskočne povezave in operacije seštevanja, se mora blok naučiti funkcije \mathcal{F} . V primeru, da je naučena preslikava ničelna $\mathcal{F}(x) = 0$, je celotna preslikava bloka identiteta vhoda $\mathcal{H}(x) = x$. Izkaže se, da je v praksi parametre nivojev bistveno lažje optimizirati tako, da aproksimirajo preslikavo $\mathcal{F}(x)$, iz katere je trivialno pridobiti operacijo identitete, kot pa, da aproksimirajo celotno preslikavo $\mathcal{H}(x)$ [37]. Z zaporedno vezavo 152 blokov prikazanih na Sliki 2.9, ResNet-152 dosega odlične rezultate na izzivu ILSVRC [27]. Mreža ResNet je sestavljena iz bistveno več nivojev kot takratni zmagovalci predhodnih ILSVRC [27] izzivov [26, 46, 32], dosega pa tudi boljše rezultate.

2.4.3 Arhitektura RetinaNet

Glavni prispevek arhitekture za detekcijo objektov RetinaNet [1] je uvedba nove kriterijske funkcije, ki omili problem neuravnoteženih podatkov (angl. unbalanced data). Eden izmed pomembnejših problemov pri učenju mrež za detekcijo objektov je neuravnoteženost podatkov, saj ozadje in objekti, ki nas ne zanimajo pokrivajo velik delež slik na katerih učimo svoj model. Uspešne arhitekture mrež detekcije objektov se s problemom neuravnoteženosti podatkov pogosto spopadajo z metodo predlaganja regij [47, 5, 4, 6], ki bistveno

zmanjša število potencialnih regij v katerih se objekti nahajajo, kar odstrani veliko število regij, ki vsebujejo ozadje ali neželjene objekte. Enostopenjski detektorji kot je [48] za reševanje problema pogosto uporabijo izbor izbor težkih primerov (angl. hard example mining), kjer izberejo le primere z najvišjo z najvišjo izgubo glede na kriterijsko funkcijo, hkrati pa te primere izbirajo tako, da ohranijo uravnoteženo razmerje med primeri ozadja in objektov.

Problem detekcije objektov si lahko predstavljamo kot klasifikacijski problem, kjer moramo vsako regijo klasificirati kot objekt ali kot ozadje. Pri problemih binarne klasifikacije se pogosto uporablja kriterijska funkcija binarne križne entropije (angl. binary cross entropy)

$$CE(p, y) = \begin{cases} -\log(p), & \text{if } y = 1 \\ -\log(1 - p), & \text{drugače} \end{cases}, \quad (2.27)$$

kjer je p napovedana verjetnost razreda, y pa pripadnost razredu $y \in \{\pm 1\}$. Za lažje pisanje enačb definiramo napovedano verjetnost posameznemu razredu p_t kot

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{drugače} \end{cases}. \quad (2.28)$$

Križno entropijo lahko nato zapišemo kot

$$CE(p, y) = CE(p_t) = -\log(p_t). \quad (2.29)$$

Mreža RetinaNet za kriterijsko funkcijo uporablja izgubno funkcijo FL (angl. focal loss), ki se osredotoča na primere, ki jih mreža težko klasificira. Kriterijska funkcija FL je definirana kot

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t), \quad (2.30)$$

kjer je p_t enak napovedani verjetnosti definirani v (2.28). γ je nastavljen parameter, ki določa kako velik vpliv mreža pripisuje težkim primerom, v delu

[1] pa predlagajo uporabo vrednosti $\gamma = 2$.

Problem, ki ga avtorji [1] izpostavijo je, da v primeru izbora križne entropije kot kriterijske funkcije, lahki primeri še vedno prispevajo nezanemarljiv del celotne vrednosti funkcije. V primeru regije, ki ne vsebuje objekta $y = 0$, kjer je napovedana verjetnost, da se v regiji nahaja objekt relativno nizka a dovolj dobra za klasifikacijo $p_t = 0,8$ je vrednost FL pri $\gamma = 2$ enaka

$$FL(0,51) = -(1 - 0,8)^2 \log(0,8) = 0,019, \quad (2.31)$$

vrednost križne entropije pa enaka

$$CE(0,51) = -\log(0,8) = 0,09. \quad (2.32)$$

Vidimo lahko, da je vrednost CE pri istem primeru bistveno višja od vrednosti FL . Čeprav je mreža z dokaj visokim prepričanjem pravilno klasificirala učni primer, je dodatek izgubi od tega primera pri CE relativno visok. V primeru visoke stopnje neuravnoteženosti razredov, ki je pri detekciji objektov pogosta, lahko veliko število preprosto klasificiranih negativnih primerov povzroči težave pri učenju zaradi prevelikega prispevka k izgubi.

Za generiranje predstavitve slike RetinaNet uporablja mrežo ResNet-50 ali ResNet-101 z značilkami FPN [49], vendar pa se lahko arhitektura mreže za pridobivanje značilk poljubno spremeni, saj je glavni prispevek dela [1] obvladovanje neuravnoteženosti podatkov. Vseeno je mreža ResNet-101 z značilkami FPN trenutno med boljšimi mrežami za ekstrakcijo značilk [1].

Poglavje 3

Predlagana metoda

V tem poglavju je opisana predlagana metoda SegCo za detekcijo in štetje polipov na slikah. V Poglavju 3.1 je opisana arhitektura uporabljene konvolucijske nevronske mreže za segmentacijo slik, v Poglavju 3.2 je opisan način modificiranja učnih podatkov in kriterijske funkcije za uspešno učenje metode, v Poglavju 3.3 pa je opisan postopek interpretacije generirane segmentacijske maske za detekcijo polipov.

3.1 Arhitektura nevronske mreže

Za metodo SegCo implementirano v uporabniškem vmesniku smo uporabili prirejeno verzijo mreže U-net [15]. Ohranili smo simetričnost arhitekture *U-Net*, vendar pa smo preverjali uspešnost segmentacijske mreže z različnim številom združevalnih in razširitvenih blokov ter z različnim številom filtrov na vsakem nivoju. V Poglavju 2.4.1 smo omenili, da se število filtrov podvoji v vsakem sledečem združevalnem bloku. Pri originalni mreži *U-Net* [15] ima prvi blok 64 filtrov, nato pa se v vsakem sledečem bloku podvoji, tako da je zaporedje števila filtrov v blokih *U-Net* brez upoštevanja združevanja zaradi preskočnih povezav enako (64, 128, 256, 512, 1024, 512, 256, 128, 64), kjer ima 1024 filtrov le latentna predstavitev, ostala števila pa predstavljajo število filtrov v združevalnih in razširitvenih blokih. Zaporedje filtrov združevalnih

in pripadajočih razširitvenih blokov bi lahko zapisali tudi kot:

$$(2^i k)_{i=0}^{n-1}, \quad (3.1)$$

kjer je k število filtrov v prvem bloku, n pa število parov združevalnih in razširitvenih blokov. Sloj latentne predstavitve bi tako vseboval $2^n k$ filtrov oziroma pri originalni *U-Net* mreži, kjer je $k = 64$ in $n = 4$, bi vseboval 1024 filtrov.

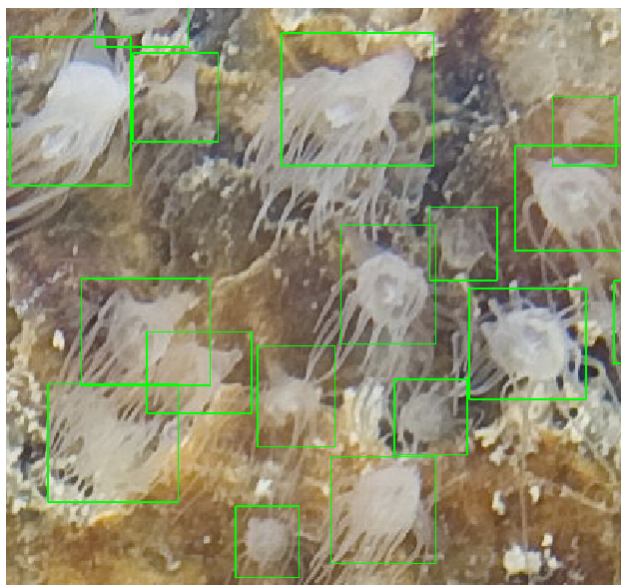
V nadaljevanju mrežo vrste *U-Net* z n združevalnimi in razširitvenimi bloki ter k filtri v nivojih prvega bloka imenujemo kar mreža z n bloki in k filtri.

Mreži, ki smo jo uporabili v razviti aplikaciji smo zmanjšali število filtrov, saj se je v eksperimentih opisanih v Poglavju 5 izkazalo, da višje število filtrov bistveno ne izboljša rezultatov pri reševanju problema detekcije polipov. Uporabljena mreža ima tako 4 bloke in le 16 filtrov, kar odstrani veliko večino parametrov glede na originalno mrežo *U-Net*. To nam sicer zmanjša sposobnost modeliranja kompleksnejših funkcij in bi morda poslabšalo uspešnost pri nekaterih drugih problemih segmentacije, vendar mreža dobro deluje na našem problemu.

3.2 Priprava učne segmentacijske maske

Podatkovna baza anotacij polipov ne vsebuje podatkov o natančni segmentaciji polipov, vsebuje pa podatke o višini, širini in lokaciji posameznih anotacij. Iz danih podatkov moramo za učenje segmentacijske mreže generirati zadovoljivo segmentacijsko masko. Segmentacijska maska s pravokotnimi anotacijami polipov ni najbolj natančna, saj pravokotne anotacije zavzemajo večjo regijo od pripadajočega polipa. Na Sliki 3.1 lahko vidimo pravokotne ročne anotacije polipov, ki poleg posameznih polipov pokrivajo še del njihove okolice. Če bi za regijo polipa v segmentacijski maski vzeli celotno pravokotno

regijo anotacije, bi torej del regije, ki polipa ne vsebuje, napačno označili kot polip. V primeru, da bi hoteli regijo zmanjšati, tako da v segmentacijski maski ne bi bilo pikslov, ki bi bili napačno klasificirani kot polip, bi se problem obrnil, saj bi nato piksle polipa napačno klasificirali kot ozadje.



Slika 3.1: Ročne anotacije polipov.

Za reševanje tega problema definiramo prirejeno kriterijsko funkcijo, ki bazira na binarni križni entropiji, ki prispevek izgube na določenih regijah odstrani

$$CE(p, y) = \begin{cases} -\log(p), & \text{if } y = 1 \\ 0, & \text{if } y = 0,5 \\ -\log(1 - p), & \text{otherwise} \end{cases} \quad (3.2)$$

Na učni segmentacijski maski lahko regije, na katerih iz danih anotacij ne moremo določiti pripadnosti razredu, označimo z $y = 0,5$ in jih s tem odstranimo iz učenja. To dosežemo tako, da na poziciji pravokotne anotacije

polipa generiramo okroglo regijo $R_{y=1}$ z radijem enakim

$$r = \frac{\min(W_a, H_a)}{2}, \quad (3.3)$$

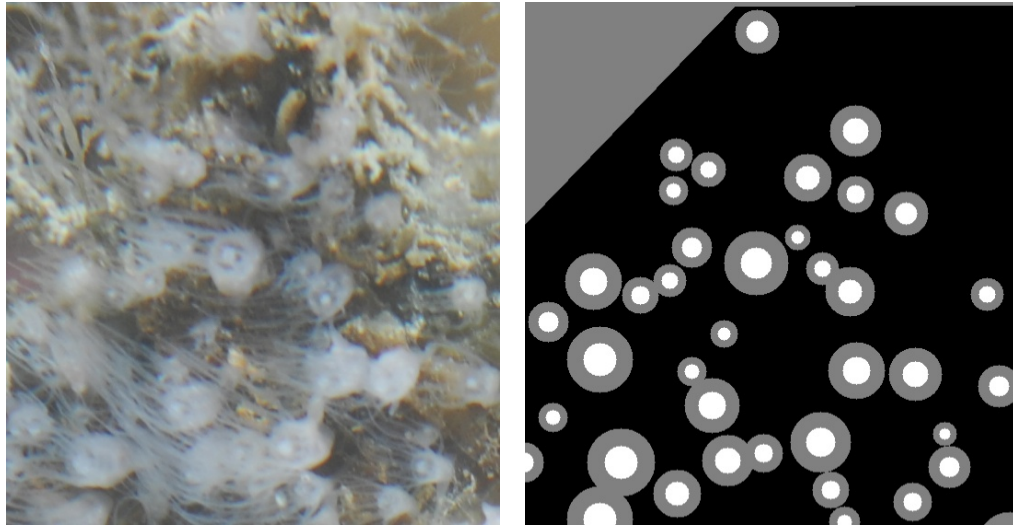
kjer sta W_a in H_a širina in višina anotacije. Na lokacijo anotacije polipa dodamo še okroglo regijo $R_{y=0.5}$ z radijem enakim

$$r = \max(W_a, H_a). \quad (3.4)$$

Regija $R_{y=0.5}$ iz učenja odstranjuje slabo definirane regije, $R_{y=1}$ pa označuje le regije za katere smo prepričani, da pripadajo polipom. Mreža se kljub grobim oznakam nauči dovolj informativne predstavitve konteksta polipa, da generira relativno natančno segmentacijsko masko.

V podatkovni bazi imamo na vsaki sliki označeno tudi interesno regijo znotraj katere naj bi bili anotirani vsi polipi. Z vrednostjo $y = 0,5$ označimo tudi okolico izven interesne regije v kateri se polipi vseeno nahajajo a niso bili anotirani. Primer slike polipov in generirane maske za to sliko lahko vidimo na Sliki 3.2b, kjer so z belo barvo označene centralne regije polipov, ki se nahajajo na Sliki 3.2a, s sivo pa neupoštevane regije. S sivo barvo je označen tudi zgornji levi kot maske na Sliki 3.2b, saj se ta regija nahaja izven območja anotacije polipov.

Vhodne slike z visoko resolucijo so neprimerne za učenje globokih konvolucijskih nevronske mreže zaradi omejitev spomina grafične kartice. Vhodne slike tako najprej razdelimo na manjše dele, ki jih grafična kartica bistveno lažje obdela. Uporabljena segmentacijska mreža je polno konvolucijska zato nima omejitev glede dimenzij vhodnih slik, le da morajo biti dimenzije deljive z 2^n , kjer je n število uporabljenih združevalnih blokov. Zaradi njihove velikosti smo pri učenju slike razdelili na dele dimenzij 512×512 kar je primerna dimenzija za mrežo, ki vsebuje do 9 blokov, hkrati pa predstavlja manjše breme za spomin grafične kartice. To seveda ponovimo tudi v postopku inference na testnih slikah. Za pridobivanje natančne segmentacijske



(a) Slika polipov

(b) Učna segmentacijska maska

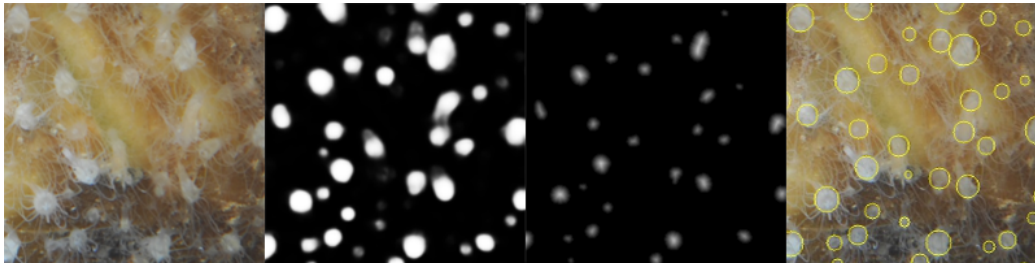
Slika 3.2: Slika polipov in pripadajoča učna segmentacijska maska. Z belo barvo so označene centralne regije polipov, s sivo barvo pa regije, ki jih pri učenju ne upoštevamo.

maske moramo sliko na manjše dele razdeliti tako, da se sosednji deli med seboj prekrivajo. Tako se izognemo napakam v segmentaciji na robovih, ki se pojavljajo zaradi podloženih robnih vrednostih ob izvajanju konvolucije in zaradi polipov, ki se nahajajo na robu dela slike in se tako na delu ne nahajajo v celoti. Prekrivajoče se regije združujemo tako, da za končno segmentacijsko masko celotne slike vzamemo maksimalno vrednost posameznih pikselov segmentacijskih mask prekrivajočih se regij. Segmentacijsko masko moramo tako generirati večkrat, kar zvišuje računsko kompleksnost inference.

3.3 Interpretacija segmentacijske maske

Po generiranju segmentacijske maske moramo iz nje pridobiti detekcije. Izhod iz segmentacijske nevronske mreže je predstavitev slike, ki za vsak piksel napove verjetnost, da piksel pripada enemu izmed polipov. Najprej

segmentacijsko masko binariziramo glede na napovedano verjetnost v vsakem pikslu. Na binarizirani sliki nato z metodo oddaljenosti (angl. distance transform) [50] izračunamo oddaljenost vsakega piksla polipa do najbližjega piksla ozadja. Lokacijo posameznih detekcij poiščemo tako, da na maski oddaljenosti najdemo lokalne maksimume. Za iskanje maksimumov uporabimo metodo zatiranja nemaksimalnih vrednosti (angl. non-maxima suppression), ki vrednosti vseh pikslov, ki so strogo večji od pikslov v njihovi okolici, nastavi na vrednost 1, ostali piksli pa imajo vrednost 0.



Slika 3.3: Koraki metode detekcije polipov.

Slika 3.3 prikazuje rezultate posameznih korakov metode detekcije. V prvem koraku generiramo segmentacijsko masko, v drugem koraku masko binariziramo in jo obdelamo z metodo oddaljenosti, v tretjem koraku pa iz maske oddaljenosti pridobimo lokalne maksimume in na njihovi lokaciji na sliki označimo detekcije polipov. Radij detekcije polipa je enak vrednosti pripadajočega maksimuma na maski oddaljenosti. Mejno vrednost Θ_{th} pri binarizaciji maske določimo eksperimentalno, postopek pa je opisan v Poglavju 5.3.2.

Poglavje 4

Razvoj aplikacije za detekcijo objektov

V sklopu dela smo razvili aplikacijo za preprosto anotacijo in detekcijo objektov na slikah s poudarkom na detekciji polipov. Aplikacija omogoča avtomatsko detekcijo in štetje objektov na slikah z naučenim modelom, omogoča pa tudi učenje lastnih modelov z uporabo ročnih anotacij. V Poglavju 4.1 so našteje knjižnice in okolja, ki so bila uporabljena pri razvoju aplikacije, Poglavje 4.2 opisuje strukturo posameznih komponent uporabniškega vmesnika aplikacije, v Poglavju 4.3 so opisane glavne funkcionalnosti za delo z anotacijami, Poglavje 4.4 našteva dodatne funkcionalnosti aplikacije in opisuje njihovo delovanje, Poglavje 4.5 pa obravnava shranjevanje anotacij in podatkov potrebnih za učenje metode. Komponente aplikacije za učenje modela in inferenco sta opisani v Poglavju 4.6.

4.1 Uporabljena tehnologija

Razvoj aplikacije bi lahko razdelili na razvoj uporabniškega vmesnika in razvoj metod strojnega vida, ki predstavljajo glavne funkcionalnosti aplikacije. Pri razvoju uporabniškega vmesnika smo uporabili knjižnico za razvoj uporabniških vmesnikov PyQt [51]. Detekcijska metoda in pomožne metode za

učenje modelov so bile razvite z uporabo okolj Keras [52] in Tensorflow [53]. Obdelava slik za interpretacijo segmentacijske maske za pridobivanje detekcij je bila razvita z uporabo knjižnice OpenCV [54].

4.2 Struktura uporabniškega vmesnika

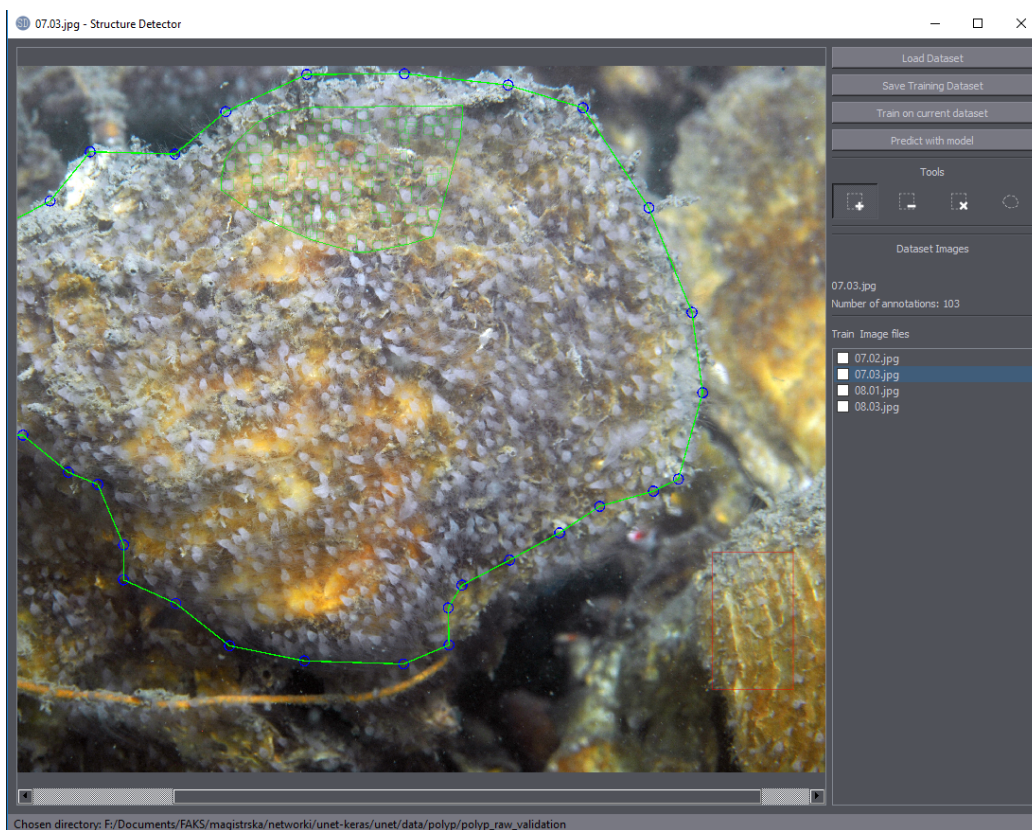
Aplikacija je sestavljena iz glavnega pogleda, ki prikazuje trenutno izbrano sliko in orodne vrstice, ki omogoča izbor orodij in zagon funkcij aplikacije. Sestavo aplikacije lahko vidimo na Sliki 4.1. Na levi strani lahko vidimo sliko z anotiranimi polipi, ki so označeni z zelenimi pravokotniki in z označeno interesno regijo, ki je prikazana v obliki poligona, ki ima na ogliščih modre kroge, ki so med seboj povezani z debelejšimi zelenimi črtami za boljšo vidljivost.

4.2.1 Orodna vrstica

Na levi strani uporabniškega vmesnika se nahaja orodna vrstica, kjer uporabnik izbira med orodji, zaganja različne funkcionalnosti aplikacije in ima pregled nad trenutnimi anotacijami in datotekami v obdelavi. Orodno vrstico lahko razdelimo na vrstico za delo z zbirkami, vrstico z orodji in pregled datotek. Na Sliki 4.2 je prikazan uporabniški vmesnik razvite aplikacije z označenimi komponentami.

Vrstica za delo z zbirkami vsebuje več gumbov in omogoča zagon funkcionalnosti nalaganja podatkovnih zbirk, shranjevanja anotacij izbrane zbirke, učenja metode na izbranih slikah in izvajanja detekcije na trenutni množici slik. Uporabnik lahko funkcionalnosti, ki jih vrstica za delo z zbirkami nadzoruje, zažene preko primerno poimenovanih gumbov, ki so prikazani na Sliki 4.3.

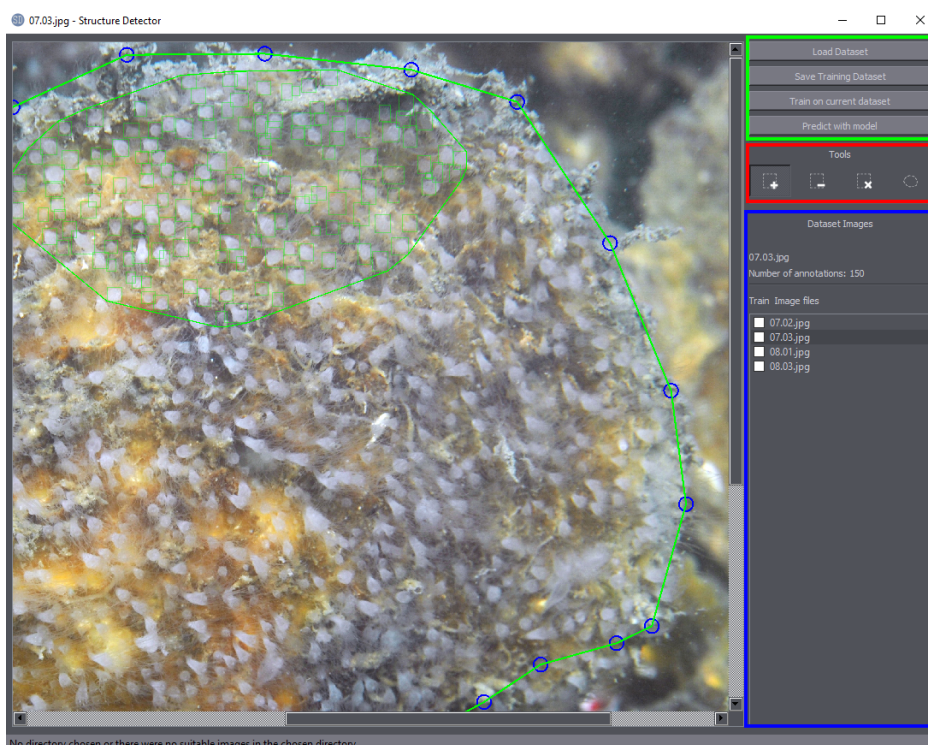
Pregled datotek, ki je prikazan na Sliki 4.4 prikazuje podatke o anotacijah na trenutno izbrani množici slik. Prikazuje število anotacij na trenutno



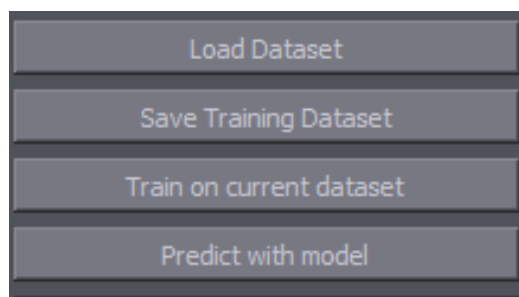
Slika 4.1: Uporabniški vmesnik razvite aplikacije.

izbrani sliki, ki jo lahko izberemo iz seznama slik, ki se prav tako nahaja znotraj pregleda datotek. V seznamu slik lahko uporabnik obkljuka slike, ki jih želi dodati v učno množico pri učenju metode.

Vrstica z orodji vsebuje orodja za anotacijo objektov na slikah. Prikazana je na Sliki 4.5 in vsebuje orodje za dodajanje in brisanje anotacij ter za označevanje interesne regije.



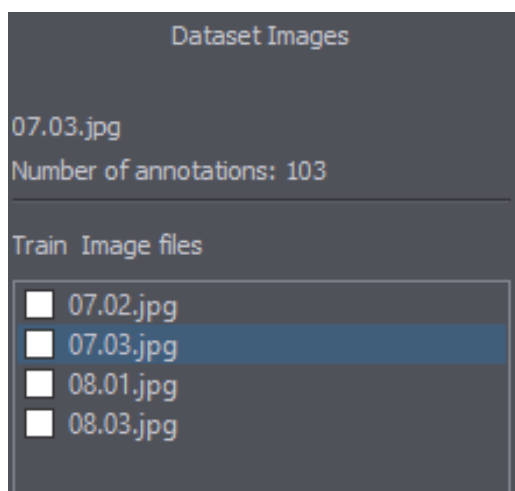
Slika 4.2: Uporabniški vmesnik razvite aplikacije z označenimi komponentami. Vrstica za delo z zbirkami je označena z zelenim, vrstica z orodji z rdečim, pregled datotek pa z modrim pravokotnikom.



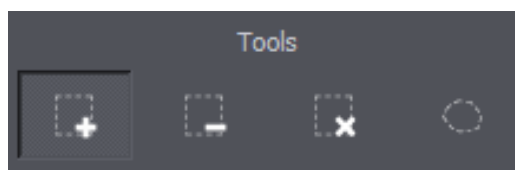
Slika 4.3: Vrstica za delo z zbirkami.

4.3 Orodja vmesnika

Za potrebe anotacije, učenja in detekcije so bila razvita različna orodja, ki te stvari omogočajo. Razvita so bila orodja za anotacijo pozitivnih primerov,



Slika 4.4: Pregled datotek.



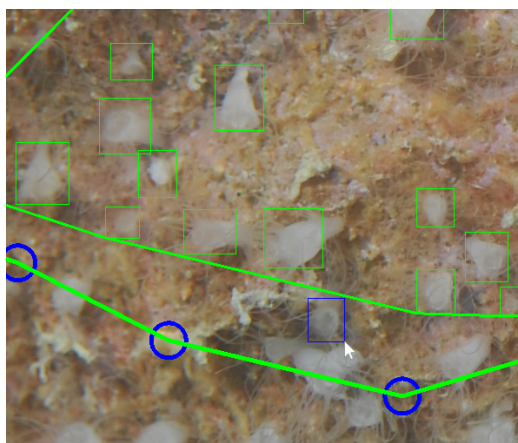
Slika 4.5: Izbor orodij. Od leve proti desne si sledijo: ikona s kvadratom s plusom predstavlja orodje za dodajanje anotacij, kvadrat z minusom predstavlja orodje za dodajanje negativnih anotacij, kvadrat z iksom predstavlja orodje za brisanje anotacij, ikona poligona predstavlja orodje za anotacijo interesne regije.

anotacijo negativnih primerov, brisanje anotacij in označevanje interesne regije. Uporabniku bolj skriti sta še metodi za učenje detekcijske metode in inferenco. Na Sliki 4.5 so prikazane ikone za izbor orodij na orodni vrstici. Od leve proti desni si sledijo ikone za: pozitivno anotacijo, negativno anotacijo, brisanje anotacij in označevanje interesne regije.

4.3.1 Anotacija objektov

Orodje za anotacijo objektov omogoča označevanje posameznih objektov na slikah. Uporabnik na glavnem pogledu aplikacije objekt označi tako, da z

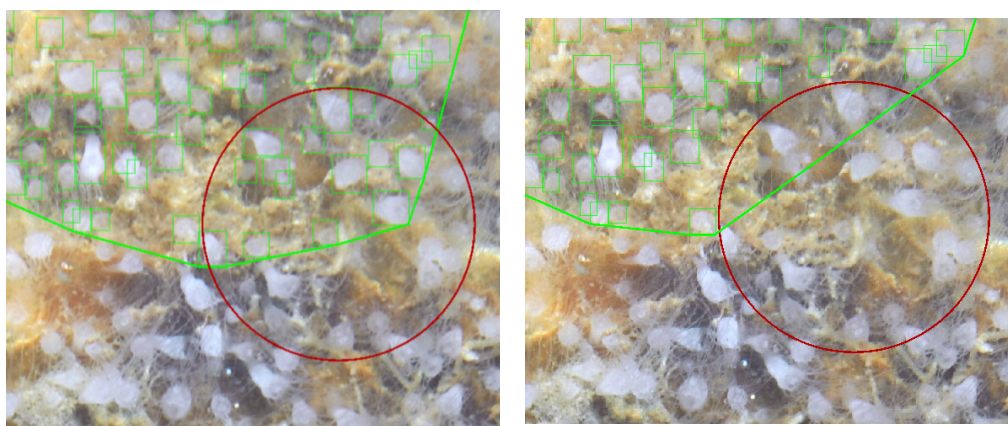
miško označi lokacijo in velikost pravokotnika, ki objekt obdaja. Na Sliki 4.6 so z zelenimi pravokotniki prikazane že določene detekcije, z modro barvo pa je prikazan pravokotnik, ki mu uporabnik še določa velikost. V primeru modrega pravokotnika ima uporabnikova miška pritisnjen levi gumb, ko pa uporabnik gumb spusti se oblikovanje pravokotnika konča, pravokotnik se obarva zeleno, anotacija pa se shrani k ostalim. Po končanem dodajanju anotacije, se nad posodobljeno množico anotacij ponovno izračuna konveksna ovojnica, ki je označena z zelenim poligonom brez modrih krogov.



Slika 4.6: Označevanje polipov. Z modrim pravokotnikom je označena anotacija, ki jo uporabnik še določa.

4.3.2 Brisanje

Ob izboru orodja za brisanje se na lokaciji uporabnikovega kurzorja pojavi rdeč krog, ki označuje regijo znotraj katere bodo anotacije izbrisane. Primer uporabe orodja za brisanje lahko vidimo na Sliki 4.7, kjer je z rdečim krogom označena regija brisanja, ki je locirana na uporabnikovem kurzorju. Preden uporabnik s klikom uporabi orodje za brisanje se v regiji nahajajo anotacije kot vidimo na Sliki 4.7a, po uporabi pa se anotacije izbrišejo, popravi pa se tudi konveksna ovojnica množice anotacij, kot lahko vidimo na Sliki 4.7b.

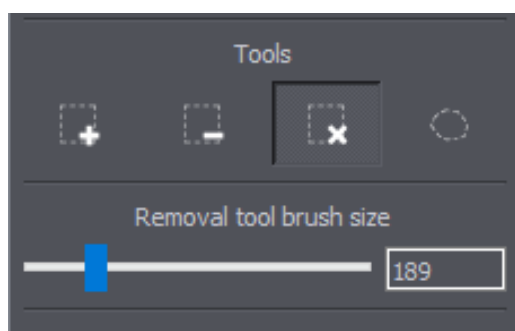


(a) Anotacije pred brisanjem

(b) Anotacije po brisanju

Slika 4.7: Anotacije pred in po koraku brisanja. Ob izboru orodja za brisanje se na poziciji kurzorja pojavi rdeč krog, ki označuje regijo brisanja anotacij.

Pod vrstico z orodji se po izboru orodja za brisanje pojavi nastavitev za radij rdečega kroga s katerim lahko uporabnik anotacije briše. To uporabniku omogoča bolj natančno brisanje z majhnim radijem oziroma bolj grobo a bistveno hitrejše brisanje večje količine anotacij z velikim radijem. Nastavitev radija je prikazana na Sliki 4.8.



Slika 4.8: Nastavitev radija kroga za brisanje.

4.3.3 Interesna regija

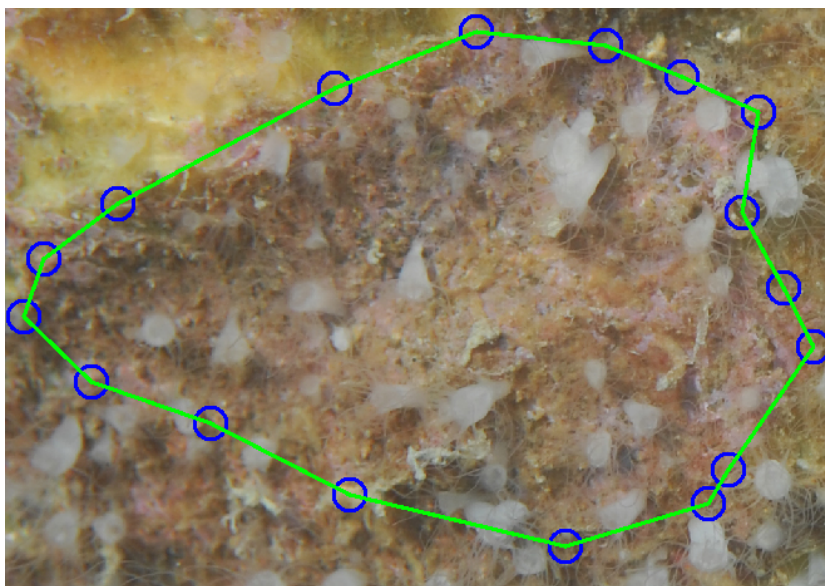
Na slikah školjk na katerih se nahajajo polipi želimo detektirati polipe le na območju školjke, ki pa predstavlja le določen del celotne slike. Velik del slike predstavlja ozadje, ki lahko prav tako vsebuje polipe, vendar pa so ti velikokrat bolj zamegljeni zaradi spremembe v izostrenosti fotoaparata ali pa so slabše osvetljeni. V primeru, da bi detekcijsko metodo izvedli na celotni sliki bi metoda detektirala tudi veliko polipov v ozadju, poleg tega pa bi se pojavilo tudi nekaj napačnih detekcij, saj metode nismo učili na primerih polipov, ki se nahajajo v ozadju. Detekcije polipov v ozadju nas torej ne zanimajo, saj je zaradi slabše kvalitete slik in različnih kotov površin polipe pogosto tudi nemogoče uspešno prešteti. Opisanemu problemu se izognemo tako, da na vsaki sliki določimo interesno regijo, ki določa območje na katerem se izvedeta učenje in inferenca detekcijske metode.

Orodje za označevanje interesne regije deluje tako, da uporabnik s kliki označi zaporedje točk na sliki, ki predstavljajo oglišča poligona. Točke se med sabo povežejo v poligon glede na zaporedje po katerem so bile dodane, zadnja točka pa se vedno poveže s prvo. Rezultat je poligon interesne regije, ki ga lahko vidimo na Sliki 4.9. Za označevanje interesne regije na posamezni sliki uporabnik z našim orodjem potrebuje nekaj sekund, čas označevanja pa se lahko tudi podaljša v primeru kompleksnejše oblike regije.

4.4 Dodatne funkcionalnosti

4.4.1 Približevanje slike

Na glavnem pogledu lahko z miškinim kolescem prikazano sliko približamo ali oddaljimo kar omogoča bistveno boljši pregled nad poljubno regijo. Zmožnost približevanja omogoča uporabniku, da si določen del slike bolje ogleda in z višjo natančnostjo določi če se na neki lokaciji nahaja objekt. To je še posebej



Slika 4.9: Označen poligon interesne regije. Modri krogi predstavljajo oglišča poligona.

pomembno pri anotaciji polipov, saj so te pogosto prosojni, zamegljeni ali pa jih delno prekrivajo drugi polipi, zato je anotiranje takih polipov brez funkcije približevanja bistveno težje.

4.4.2 Preklic prejšnjih akcij

Z uporabo bližnjice *Ctrl+Z* lahko prekličemo pretekle akcije anotacije in brisanja anotacij. Vsako brisanje ali dodajanje anotacij se shrani na sklad uporabnikovih preteklih akcij. Ob uporabi bližnjice za preklic prejšnjih akcij se s sklada vzame uporabnikova zadnja akcija, nato pa se izvede njeno nasprotje. V primeru, da je zadnja akcija anotacija, se nazadnje dodana anotacija izbriše, v primeru, da zadnja akcija brisanje pa se povrnejo vse anotacije, ki so bile izbrisane v zadnji operaciji brisanja. Po izvedbi obratne operacije se ta akcija izbriše s sklada.

4.5 Shranjevanje podatkov

Ob zagonu programa mora uporabnik izbrati direktorij kjer se nahajajo slike, ki vsebujejo objekte, ki jih uporabnik želi avtomatsko detektirati. Po anotiranju objektov in označevanju interesnih regij lahko uporabnik uporabi funkcionalnost shranjevanja, ki anotacije shrani v datoteko *annotations.csv* v direktorij v katerem se nahajajo slike. Negativne anotacije se shranijo v *negative_annotacions.csv*, interesne regije pa v *roi_points.csv*.

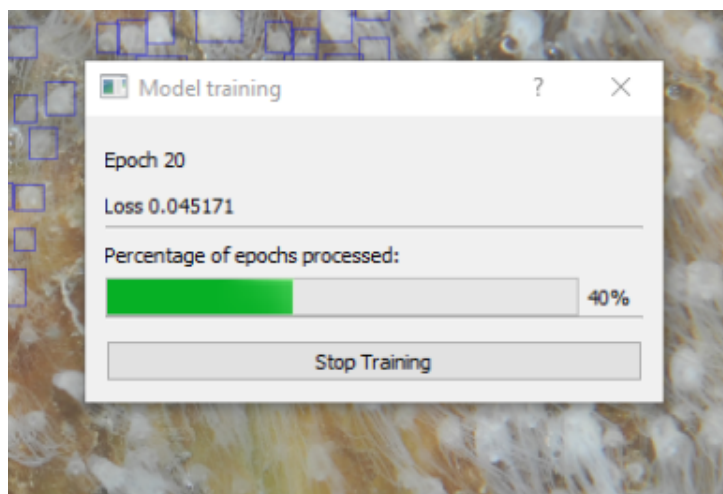
Poleg anotacij se pripravijo in shranijo tudi podatki potrebni za učenje mreže. V izbranem direktoriju se ustvari direktorij *training_data*, ki vsebuje direktorija *images* in *masks*, ki vsebujeta slike in segmentacijske maske. Pri shranjevanju podatkov torej shranimo tudi učno množico slik in segmentacijskih mask. Najprej za vsako sliko iz anotacij generiramo segmentacijsko masko, nato pa slike in maske razdelimo na manjše dele. V učno množico shranimo le dele slik in mask, ki vsebujejo anotacije saj bi sicer vključili tudi dele ozadja, ki pa ne prispevajo k učenju saj se nahajajo izven interesne regije in so posledično na segmentacijskih maskah označeni z vrednostjo 0,5, zaradi česar so izločeni iz izračuna kriterijske funkcije.

4.6 Učenje modela in inferenca

4.6.1 Učenje modela

Po generiranju učne množice iz slik in anotacij lahko uporabnik zažene učenje segmentacijske mreže. Z namenom poenostavitve postopka učenja za uporabnika so število iteracij, stopnja hitrosti učenja in postopek obogatitja učne množice vnaprej nastavljeni znotraj metode. Seveda je postopek učenja nevronske mreže občutljiv na te parametre, zaradi česar lahko učenje z nastavljenimi parametri pripelje do slabših rezultatov. Fiksna nastavitve parametrov tako potencialno žrtvuje natančnost metode za preprostost uporabe.

Na Sliki 4.10 lahko vidimo izgled komponente uporabniškega vmesnika, ki uporabnika obvešča o poteku učenja segmentacijske mreže. Uporabniku je na voljo trenutna vrednost izgube glede na kriterijsko funkcijo in število procesiranih prehodov čez podatke. Z vrednostjo izgube lahko uporabnik spremlja kako uspešno segmentacijska mreža modelira učne podatke.

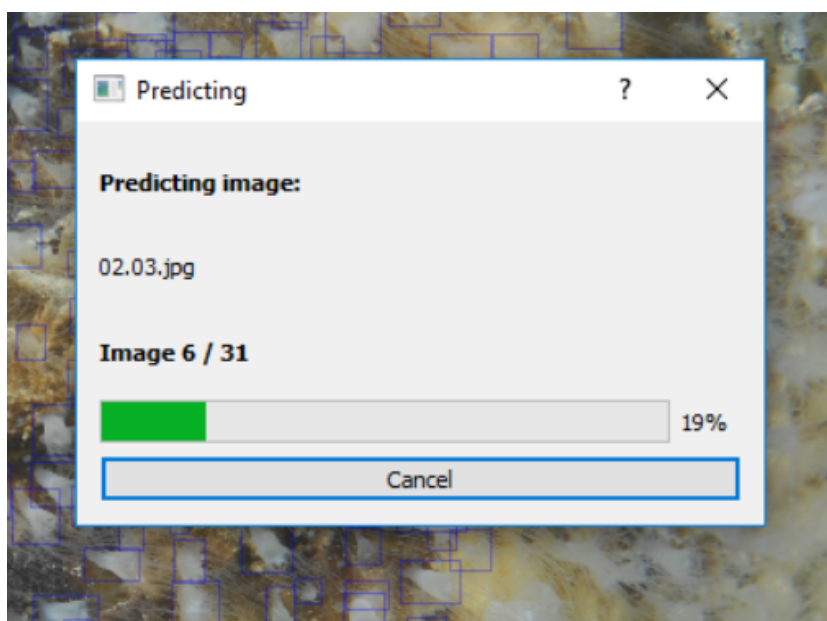


Slika 4.10: Komponenta uporabniškega vmesnika za učenje modela. Prikaže trenutni napredek pri postopku učenja z informacijo o številu procesiranih prehodov čez podatke (angl. epoch) in o trenutni vrednosti kriterijske funkcije na učnih podatkih.

4.6.2 Inferenca

Uporabnik lahko zažene postopek inference nad slikami v izbrani množici. Aplikacija upošteva le detekcije, ki se nahajajo znotraj interesne regije vsake slike. Na Sliki 4.11 je prikazana komponenta, ki med inferenco uporabnika obvešča o napredku. Prikaže ime slike nad katero v tistem trenutku izvaja postopek detekcije, število slik nad katerimi je bila detekcija že izvedena in število vseh slik nad katerimi se bo detekcija izvajala. Postopek lahko uporabnik tudi predčasno prekine s klikom na gumb. Ob končanju postopka se komponenta odstrani, na slikah pa se znotraj interesnih regij pojavijo

detekcije. V primeru, da interesna regija na določeni sliki ni označena se detekcije pojavijo na celotnem območju slike. Po izvedeni inferenci lahko rezultate shranimo ali pa popravimo morebitne napačne detekcije in iz njih generiramo novo učno množico na kateri lahko model ponovno naučimo, a z večjo količino podatkov.



Slika 4.11: Komponenta uporabniškega vmesnika za detekcijo objektov z naučenim modelom.

Poglavje 5

Eksperimentalna analiza

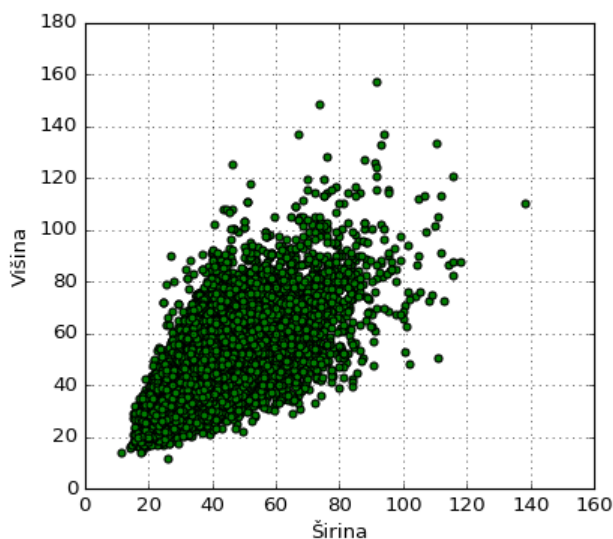
Vsi eksperimenti so izvedeni na računalniku s procesorjem *Intel(R) Core(TM) i7-6700K @ 4.00 GHz (8 CPUs)*, 32 GB pomnilnika in grafično kartico *NVIDIA GeForce GTX 1070*.

5.1 Podatkovna zbirka

5.1.1 Sestava podatkovne zbirke

Učenje ter evalvacija vseh metod sta izvedena na podatkovni zbirki sestavljeni iz 37 slik ostrig na katerih so pritrjeni polipi [2, 22]. Vsaki sliki pripadata še množica anotacij polipov in množica točk, ki predstavlja mnogokotnik, ki označuje interesno regijo, kjer se anotacije polipov nahajajo. Skupaj podatkovna zbirka vsebuje 39886 anotacij polipov v obliki (x, y, w, h) , kjer x in y določata stolpec in vrstico slike na katerih se nahaja center polipa, w in h pa širino in višino označenega polipa. Vsaka slika v povprečju vsebuje 1030 anotacij, povprečna širina in višina pa sta 38 in 43 pikslov. Na Sliki 5.1 sta prikazani višina in širina posameznih anotacij. Vidimo lahko, da sta višina in širina močno korelirani, vendar je variabilnost dimenzij anotacij polipov znotraj podatkovne množice vseeno velika. Slike 5.2a, 5.2b in 5.2 prikazujejo histograme višine, širine in površine anotacij, Tabela 5.1 pa podatke o višini, širini in površini anotacij polipov iz katerih je razvidno, da se posa-

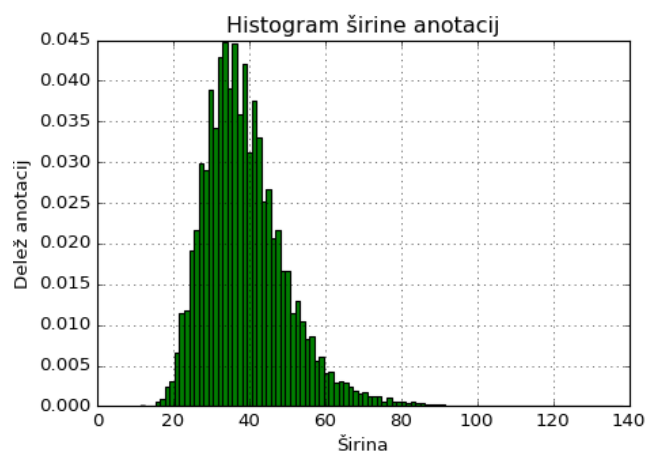
mezne anotacije med sabo močno razlikujejo. Število iskanih objektov, ter njihove dimenzije so sicer pomembne informacije za razvoj sistema za detekcijo objektov. Podatkovno zbirko smo razdelili na učno in testno množico. Uporabili smo testno množico sedmih slik, ki jo za svojo evalvacijo uporablja [2]. Uporabljeno testno množico imenujemo PCD. Preostale slike smo uporabili v učni množici, pri preverjanju vpliva velikosti učne množice na rezultat pa smo uporabljali tudi različne podmnožice nabora učnih slik.



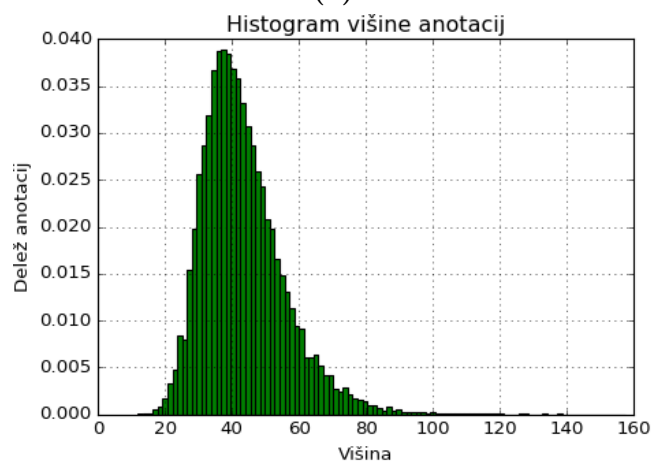
Slika 5.1: Širina in višina ročnih anotacij polipov.

	Širina	Višina	Površina
μ	39,12	43,81	1798,77
σ	11,07	12,68	1006,81
<i>min</i>	11	12	156
<i>max</i>	138	157	15276

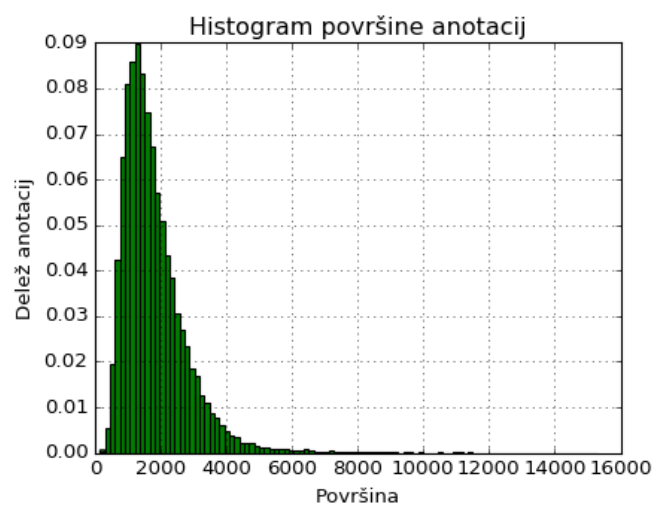
Tabela 5.1: Povprečna vrednost μ , standardni odklon σ , minimum in maksimum vrednosti širine, višine in površine v zbirki anotacij polipov.



(a)



(b)

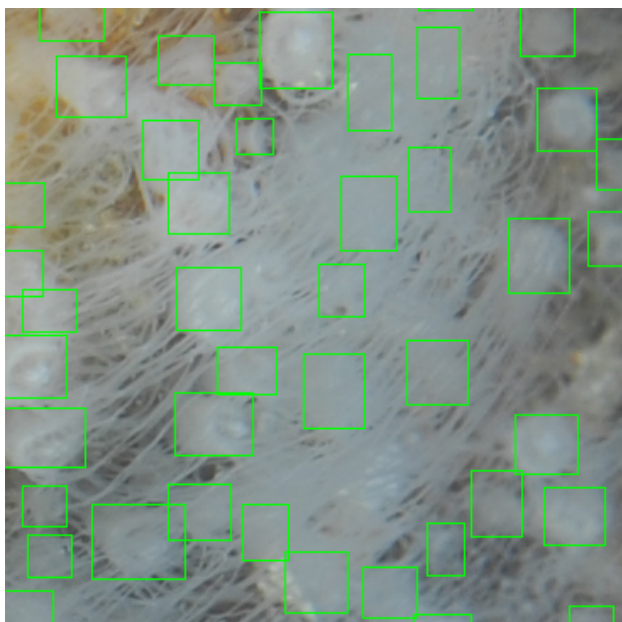


(c)

Slika 5.2: Histograma širine (a), višine (b) in površine (c) anotacij.

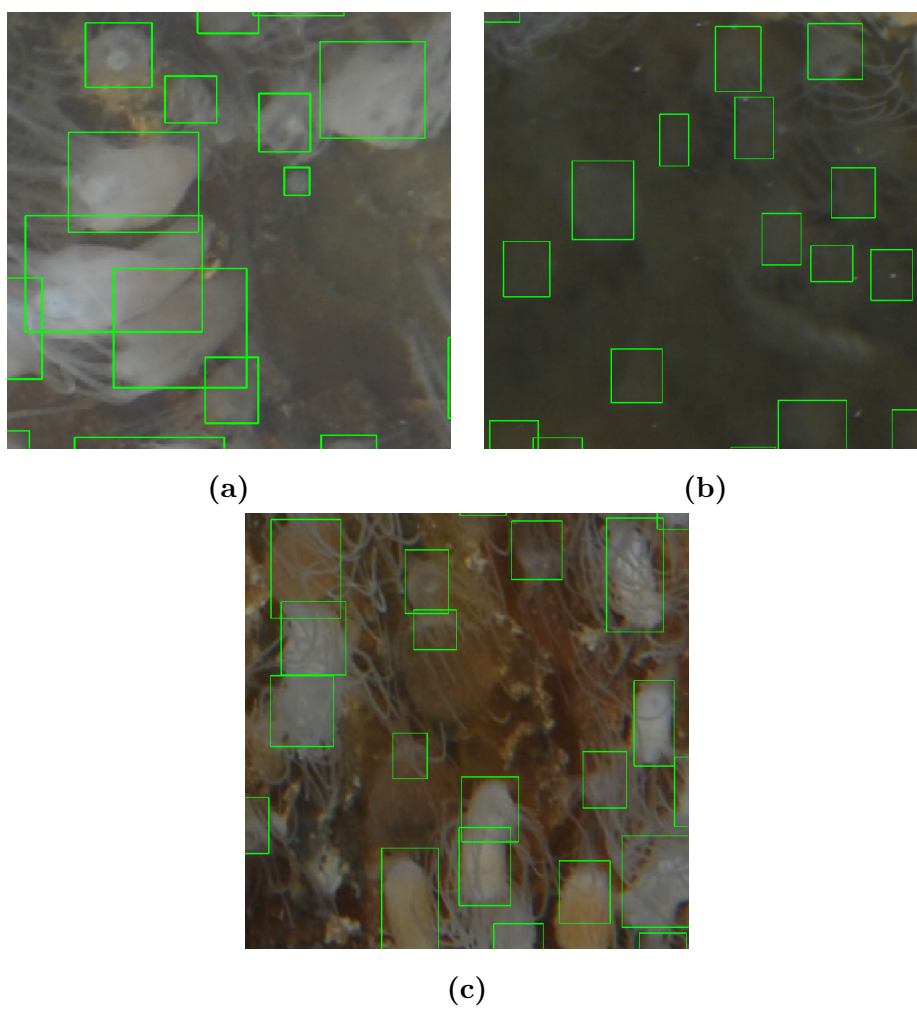
5.1.2 Izgled polipov

Polipi na slikah se ne razlikujejo le po velikosti, temveč tudi po orientaciji, barvi, obliki in osvetljenosti. Določeni polipi so prosojni zaradi česar na njihov izgled močno vpliva barva ozadja na katerem se nahajajo. Polipi se med seboj delno prekrivajo, pogosto pa posamezne polipe prekrivajo lovke drugih polipov. Primer prekrivanja lahko vidimo na Sliki 5.3. Prekrivanje oteži anotacijo že človeškim označevalcem, saj je kljub visoki resoluciji slik težko določiti meje med sosednjimi polipi, zaradi česar je težko določiti tudi pravilnost anotacij na regiji prikazani na Sliki 5.3 in na regijah, ki so ji podobne.



Slika 5.3: Primer regije, kjer se polipi med seboj pokrivajo.

Slika 5.4 prikazuje visoko stopnjo variabilnosti med posameznimi polipi. Na Sliki 5.4a vidimo primere velikih in majhnih polipov, Slika 5.4b prikazuje anotacije polipov, ki so slabo izostreni in slabše osvetljeni, Slika 5.4c pa prikazuje spremembe v barvi polipov zaradi osvetljenosti, prosojnosti ali razvitosti polipa.



Slika 5.4: Primeri vizualne variabilnosti med polipi.

5.1.3 Pomanjkljivosti slik in anotacij

Celotno podatkovno zbirko [22, 2] je anotiralo več anotatorjev v sklopu nastanka različnih del, zato prihaja do razlik v anotacijah zaradi različnih predstav o dobrih anotacijah. Na nekaterih slikah so tako anotacije večje kot na ostalih, na nekaterih pa so označeni tudi polipi v ozadju, ki so izven fokusa kamere. Spremenljiva velikost anotacij polipov bi lahko otežila učenje zaradi slabše kvalitete generirane maske, ki bi lahko kot dele polipa označevala tudi okolico polipov v primerih bistveno prevelike anotacije.

Slike se razlikujejo tudi po natančnosti označevanja. V delu [2] so svojo metodo detekcije polipov ocenjevali na manjši množici slik, katere anotacije je z namenom minimiziranja napake, preverilo več označevalcev. V delu so izvedli tudi oceno napake označevalcev in ocenili, da je povprečna maksimalna relativna napaka posameznega označevalca 18%. Večinoma se napaka kaže v zgrešenih in ne napačno označenih polipih, saj je nekatere manjše in prosojne polipe zelo težko opaziti, le redko pa se zgodi, da bi označevalec del ozadja označil kot polip. Razlog za visoko napako označevalcev je dolgotrajnost označevanja, število polipov na sliki in vizualne lastnosti polipov, ki anotacijo otežujejo.

Posledica težavnosti označevanja je torej nenatančna podatkovna množica, ki pa lahko povzroča omejuje uspešnost učenja segmentacijske mreže, saj pomanjkanje anotacij pri zgrešenih polipih povzroča napake v učni segmentacijski maski in s tem dodaja nezanemarljivo količino napačno označenih regij v podatkovno zbirko. Zaradi nenatančnih anotacij v [2] uporabljajo množico 7 anotiranih slik *PCD*, ki so bile anotirane in preverjene večkrat. Velika količina napak je bila na ta način odstranjena, je pa tako anotiranje zelo dolgotrajno in neprimerno za anotiranje celotne podatkovne zbirke, poleg tega pa se tudi na tako anotirani množici pojavljajo napake.

5.2 Metoda evalvacije

Problem detekcije lahko prevedemo na problem binarne klasifikacije. Standardna mera za evalvacijo binarnega klasifikatorja je mera F_1 [55], ki je definirana kot

$$F_1 = 2 \frac{Pr \text{ } Rec}{Pr + Rec}, \quad (5.1)$$

kjer je Pr preciznost klasifikatorja, Rec pa njegov priklic (angl. recall). Preciznost je definirana kot razmerje med pravilnimi detekcijami in vsemi detekcijami, ki jih metoda označi. Pove nam torej kakšen delež detekcij metode ja pravilnih. Zapišemo jo kot

$$Pr = \frac{TP}{TP + FP}, \quad (5.2)$$

kjer so TP pravilne detekcije (angl. true positive), FP pa napačne detekcije (angl. false positive). Priklic je definiran kot delež pravilnih detekcij, ki jih detektor odkrije oziroma jih klasificira kot pravilne. Priklic zapišemo kot

$$Pr = \frac{TP}{TP + FN}, \quad (5.3)$$

kjer so FN (angl. false negative) zgrešene detekcije, ki so bile napačno klasificirane kot ozadje (angl. false negative).

Za obravnavo problema detekcije polipov kot klasifikacijski problem moramo definirati kako določimo pravilnost detekcije. Kvaliteto detekcije se pogosto ocenjuje na podlagi prekrivanja pravokotne detekcije ter ročne anotacije. Za mero prekrivanja se uporablja Jaccardov index oziroma mero preseka nad unijo IOU (angl. intersection over union), ki je definirana kot

$$IOU_{AD} = \frac{D \cap A}{D \cup A}, \quad (5.4)$$

kjer je $D \cap A$ površina preseka med detekcijo D in anotacijo A , $D \cup B$ pa je skupna površina, ki jo zasedata D in A .

Po izvajanju detektorja objektov nad sliko nam ta vrne množico detekcij, pogosto v obliki pravokotnih anotacij. Uspešnost detektorja določimo tako, da pridobljene detekcije primerjamo z ročnimi anotacijami. Detekcije moramo z ročnimi anotacijami povezati glede na mero pravilnosti detekcije. Pogosto je uporabljena pohlepna metoda za povezovanje, kjer v poljubnem vrstnem redu povezujemo detekcije z anotacijami, če ustrezajo kriteriju pravilnosti. V primeru uporabe mere pravilnosti IOU, vsako detekcijo D povežemo ročno anotacijo A , pri kateri je $IOU_{AD} > T$, kjer je T mejna vrednost za upoštevanje detekcije kot pravilne. Po povezavi A in D , D označimo kot pravilno detekcijo, A pa označimo kot povezano in je ne povezujemo z ostalimi detekcijami [27].

5.2.1 Metoda ocenjevanja detekcije polipov

Metode evalvacije izzivov ImageNet [27] in COCO [8] so uporabne za evalvacijo metod za splošno detekcijo objektov, vendar pa je glavni cilj naše metodo predvsem detekcija polipov z namenom ugotavljanja števila polipov na slikah. Mera IOU ni najbolj učinkovito merilo za kvaliteto anotacije, če je glavni cilj štetje polipov, saj ni tako pomembno ali detekcija dobro pokriva polip temveč samo ali je metoda posamezen polip zaznala. V delu [24] pri evalvaciji njihovega detektorja polipov za mero pravilnosti detekcije uporabijo razdaljo med centroma anotacije in detekcije. Za povezovanje anotacij in detekcij uporabljajo algoritem LAPJV [56], ki rešuje problem povezovanja vozlišč uteženega bipartitnega grafa tako, da minimizira uteži izbora povezav. V primeru detekcije objektov sta disjunktni množici polno povezanega bipartitnega grafa množica anotacij in množica detekcij, uteži med množicama pa so razdalje med centri anotacij in detekcij. Kot pri meri IOU, tudi pri razdalji med centri določimo mejno vrednost τ_d pri kateri se detekcija še šteje kot pravilna. Ta meja je v [2] definirana kot mediana diagonal anotacij testne množice. Če za izračun mediane diagonal uporabimo anotacije množice PCD je mejna vrednost $\tau_d \approx 100$ pikslov [2]. Algoritem vrednotenja uspešnosti detektorja torej ne upošteva mere IOU med detekcijami in anotacijami temveč

se osredotoča na povezovanje tistih, ki so si med sabo blizu. Da bi poudarili točnost lokalizacije naše metode smo mejo τ_d pri postopku evalvacije nastavili na bistveno nižjo vrednost $\tau_d = 30$ pikslov. To vrednost τ_d smo uporabili pri vseh eksperimentih razen pri primerjavi metode SegCo z detektorjem PoCo [2], kjer smo se držali evalvacijske metode opisane v [2].

Za evalvacijo naše detekcijske metode uporabljamo kriterij razdalje saj je glavni cilj metode štetje polipov in ne njihova natančna segmentacija in določanje dimenzij. Vseeno pa je kriterij IOU pomemben v primeru razširitve aplikacije za pridobivanje podatkov o obliki objektov na slikah. V tem primeru bi nas poleg števila utegnile zanimati še širina in višina detektiranih objektov, za kar pa so potrebne detekcije, ki objekte dobro pokrivajo kar pa se odraža v dobri meri IOU glede na anotacije.

5.3 Analiza rezultatov predlagane metode

5.3.1 Vpliv parametrov mreže

Segmentacijska mreža predlagane metode SegCo je sestavljena po arhitekturi U-Net [15], vendar pa je njena kompleksnost odvisna od števila uporabljenih U-Net blokov in števila filtrov v konvolucijskih nivojih mreže. Število filtrov smo definirali kot število filtrov v konvolucijskih nivojih v prvem bloku mreže, v vsakem sledečem bloku pa se število filtrov v nivojih podvoji, kot je značilno za arhitekturo U-Net. Segmentacijsko mrežo smo naučili večkrat z različnimi kombinacijami parametrov in rezultate ovrednotili na testni množici. Pri eksperimentih smo uporabili števila filtrov $n \in \{8, 16, 32, 64\}$ in število blokov $k \in \{2, 3, 4\}$. Manjše število blokov vpliva na kvaliteto predstavitve širšega kontesкта z zmanjševanjem vidnega polja značilnk v predstavitvi, vendar pa so objekti, ki jih detektiramo manjši zato je tudi uporabnost značilnk, ki jih pridobimo z visokim številom blokov, potencialno manjša. Pogosto se izkaže, da so značilke pridobljene na nižjih nivojih pomembne za detekcijo manjših objektov [49]. Podobno lahko večje število filtrov modelira kompleksnejše

funkcije, hkrati pa prispeva k računski zahtevnosti inference.

Učenje mreže je probabilističen postopek, saj je uspešnost učenja odvisna postopkov kot so inicializacija uteži, regularizacija z izpustom in generiranje učnih primerov z obogatitvijo učne množice. Zaradi komponente naključnosti pri učenju lahko metoda dosega različne rezultate, če učenje zaženemo večkrat na isti množici, vendar pa je postopek učenja dovolj robusten, da je pri dobro izbranih parametrih razlika majhna.

Segmentacijsko mrežo smo večkrat naučili na isti podatkovni množici, vendar pa smo spreminjali globino ter začetno število filtrov pri konstruiranju mreže. Za boljšo primerjavo vpliva števila blokov in filtrov smo učenje izvajali brez obogatitve učne množice. Tabela 5.2 prikazuje rezultate na testni množici v obliki mere F-1, AR in AP. Vidimo, da uspešnost glede na mero F-1 praviloma narašča skupaj s številom blokov in številom filtrov, vendar pa v nekaterih primerih to ne drži:

- Pri številu blokov $k = 2$ mreža slabše segmentira slike testne množice, saj se verjetno težje nauči kontekstualno pomembnejših značilnik, ki opisujejo prostorsko večje značilnosti slike. Manjše število blokov lahko do neke mere nadomestimo z večjim številom filtrov, saj je očitno kombinacija večje količine značilnik z manjšim vidnim poljem dovolj informativna za predstavitev polipov.
- Mreža s številom blokov $k = 3$ se po uspešnosti na testni množici po meri F-1 ne razlikuje bistveno glede na število filtrov, vendar pa je lahko to rezultat naključnosti učenja.
- Mreža s $k = 4$ že pri nizkem številu filtrov $n = 8$ dosega boljše rezultate od mreže s $k = 2$, vendar pa se z rastočim številom filtrov rezultate

glede na mero $F-1$ ne izboljša bistveno, dosega pa nezanemarljivo boljše rezultate po meri AR . Razlog za to bi lahko bile manjkajoče anotacije v testni množici, saj bi se ob detekciji neoznačenih polipov na nekaterih slikah natančnost zmanjšala zaradi česar bi kljub višjemu priklicu, povprečna mera $F-1$ ostala enaka. V Poglavju 5.4 so napake testne množice bolj podrobno opisane.

		Število filtrov (n)				
		8	16	32	64	
Število blokov (k)	2	AR	$0,84 \pm 0,06$	$0,82 \pm 0,13$	$0,83 \pm 0,12$	$0,91 \pm 0,05$
		AP	$0,77 \pm 0,06$	$0,81 \pm 0,05$	$0,83 \pm 0,04$	$0,82 \pm 0,04$
		F1	$0,80 \pm 0,07$	$0,81 \pm 0,07$	$0,83 \pm 0,07$	$0,86 \pm 0,04$
	3	AR	$0,89 \pm 0,04$	$0,92 \pm 0,06$	$0,90 \pm 0,06$	$0,90 \pm 0,06$
		AP	$0,86 \pm 0,02$	$0,86 \pm 0,02$	$0,89 \pm 0,02$	$0,88 \pm 0,02$
		F1	$0,87 \pm 0,01$	$0,89 \pm 0,02$	$0,89 \pm 0,03$	$0,89 \pm 0,03$
	4	AR	$0,88 \pm 0,06$	$0,91 \pm 0,05$	$0,90 \pm 0,06$	$0,93 \pm 0,03$
		AP	$0,87 \pm 0,02$	$0,85 \pm 0,02$	$0,85 \pm 0,02$	$0,87 \pm 0,02$
		F1	$0,88 \pm 0,02$	$0,89 \pm 0,02$	$0,88 \pm 0,04$	$0,89 \pm 0,02$

Tabela 5.2: Povprečni priklic, natančnost in mera $F-1$ na testni množici pri metodi SegCo s segmentacijsko mrežo, ki je sestavljena iz k blokov U-Net z osnovnim številom filtrov n .

Tabela 5.3 prikazuje čas potreben za segmentacijo slike testne množice v odvisnosti od blokov in filtrov. Resolucija posamezne testne slike je (4288×2844). Za minimiziranje možne napake zaradi stanja sistema smo detekcijo in merjenje časa ponovili 10-krat za vsako verzijo naučene segmentacijske mreže. Čas, ki ga metoda porabi je seveda odvisen od števila operacij potrebnih za inferenco na vsaki sliki, ki pa narašča s številom blokov in številom filtrov. Vrednosti števila parametrov v Tabeli 5.4 in porabljenega časa v Tabeli 5.3 so sicer povezane, vendar pa vseeno mreže s podobnim številom parametrov

		Število filtrov (n)			
		8	16	32	64
Število blokov (k)	2	$3,43 \pm 0,32s$	$4,12 \pm 0,47s$	$5,86 \pm 0,51s$	$10,05 \pm 0,89s$
	3	$3,47 \pm 0,36s$	$4,41 \pm 0,49s$	$6,30 \pm 0,60s$	$12,30 \pm 1,04s$
	4	$3,54 \pm 0,43s$	$4,38 \pm 0,53s$	$6,66 \pm 0,63s$	$14,60 \pm 1,23s$

Tabela 5.3: Čas, ki ga metoda, ki uporablja segmentacijsko mrežo s k bloki in n fitri porabi za detekcijo na sliki testne množice.

porabijo več časa, če uporabljajo manjše število blokov. Kot primer vzemimo mrežo s 4 bloki in 16 filtri, ki vsebuje $1,94 \times 10^6$ parametrov in mrežo s 3 bloki in 32 filtri, ki vsebuje $1,95 \times 10^6$ filtrov. Kljub relativno majhni razliki v številu parametrov $\Delta p \approx 10^4$ porabi mreža s 3 bloki več časa $\Delta t \approx 2s$. Razlog za to je večje število združevalnih operacij v mreži, ki vsebuje več blokov. To bistveno zmanjša prostorske dimenzije predstavitve slike, kar omogoča hitrejšo izvršbo konvolucije.

Porabljen čas prikazan v Tabeli 5.3 je torej sestavljen iz inference segmentacijske mreže in interpretacije segmentacijske maske za posamezno sliko dimenzije 4288×2488 . Zaradi svoje velikosti so slike razdeljene na manjše dele, ki pa se morajo med seboj še prekrivati, saj je na robovih neprekrivajočih se delov nemogoče generirati zanesljive segmentacijske maske zaradi manjkajoče informacije o objektih na robovih, ki jih z razdelitvijo slike na manjše dele ločimo. Deli slike, ki se med seboj prekrivajo v veliki meri rešijo to težavo, vendar pa moramo dele slike tako obdelovati večkrat kar posledično zvišuje računsko kompleksnost metode.

5.3.2 Interpretacija segmentacijske maske

Generirane detekcije so odvisne od mejne vrednosti Θ_{th} , ki jo uporabimo za binarizacijo generirane segmentacijske maske. Zmanjševanje Θ_{th} pogosto

Število blokov (k)	Število filtrov (n)			
	8	16	32	64
2	$0,029 \times 10^6$	$0,117 \times 10^6$	$0,466 \times 10^6$	$1,88 \times 10^6$
3	$0,121 \times 10^6$	$0,486 \times 10^6$	$1,95 \times 10^6$	$7,69 \times 10^6$
4	$0,485 \times 10^6$	$1,94 \times 10^6$	$7,76 \times 10^6$	$31,03 \times 10^6$

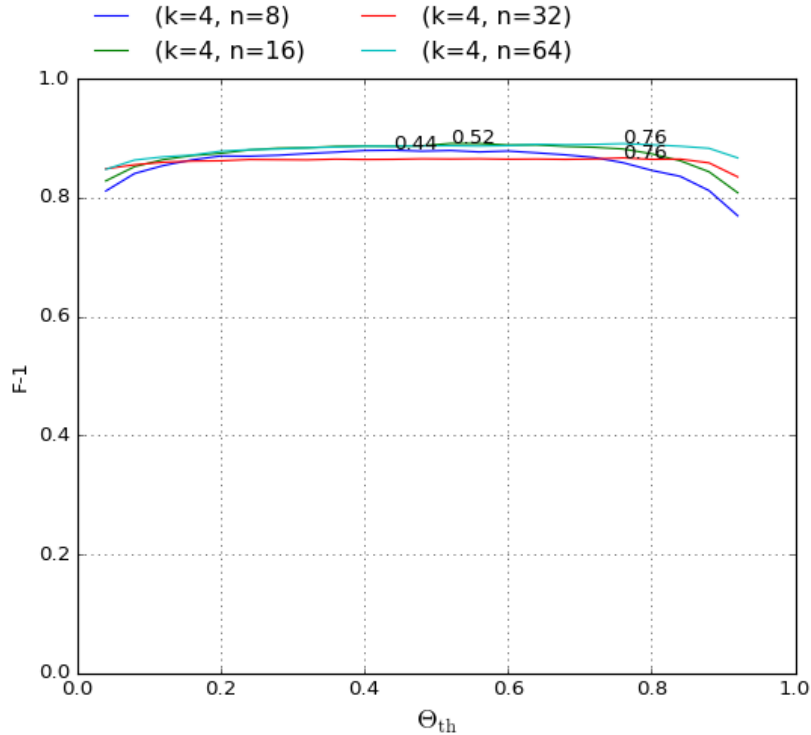
Tabela 5.4: Število parametrov mreže.

poviša priklic metode, a zmanjša njeno natančnost. Analizirali smo robustnost metode na parameter binarizacije Θ_{th} in ugotovili, da je metoda robustna na široko območje vrednosti Θ_{th} . Vpliv parametra Θ_{th} smo preverjali na območju intervala $\Theta_{th} \in (0,04; 0,96)$.

Na Sliki 5.5 je prikazana odvisnost mere $F-1$ od uporabljene mejne vrednosti za različne globine mrež, Slika 5.6 pa prikazuje vrednosti AR in AP pri različnih mejnih vrednostih. Mejna vrednosti pri kateri je mera $F-1$ maksimalna se med mrežami sicer razlikuje, vendar pa so deviacije vrednosti s spreminjajočo se mejno vrednostjo dokaj majhne. Mreža s 4 bloki in 64 filtri ima maksimalno povprečno vrednost $F-1 = 0,89$ pri $\Theta_{th} = 0,76$ z $AP = 0,89$ in $AR = 0,90$, vendar pa ima pri $\Theta_{th} = 0,44$ povprečno vrednost $F-1 = 0,89$ z $AP = 0,86$ in $AR = 0,93$. Povprečna mera $F-1$ se s spremembo mejne vrednosti praktično ne spremeni v kolikor je ta v primernem intervalu, ki pa je za vsako metodo drugačen. Na Sliki 5.5 lahko vidimo, da so na intervalu $\Theta_{th} \in (0,40; 0,60)$ mere $F-1$ blizu maksimumov za vse mreže.

Na Sliki 5.6 lahko vidimo, da se po pričakovanjih z naraščajočo mejo zmanjšuje AR in viša AP , vendar pa hitrost padanja AR in AP ni enaka. Vrednost AR z naraščajočo mejo hitro pada, vrednost AP pa se spreminja bistveno počasneje. Pri uporabi nizke vrednosti $\Theta_{th} = 0,04$ je AP pri vseh mrežah še vedno nad 0,75, vrednost AR pa lahko pade pod 0,65 pri visoko postavljenih mejah. Zaradi počasnejšega padanja AR pri nizkih vrednostih lahko za izvajanje metode izberemo nižjo mejno vrednost. Primerna vrednost

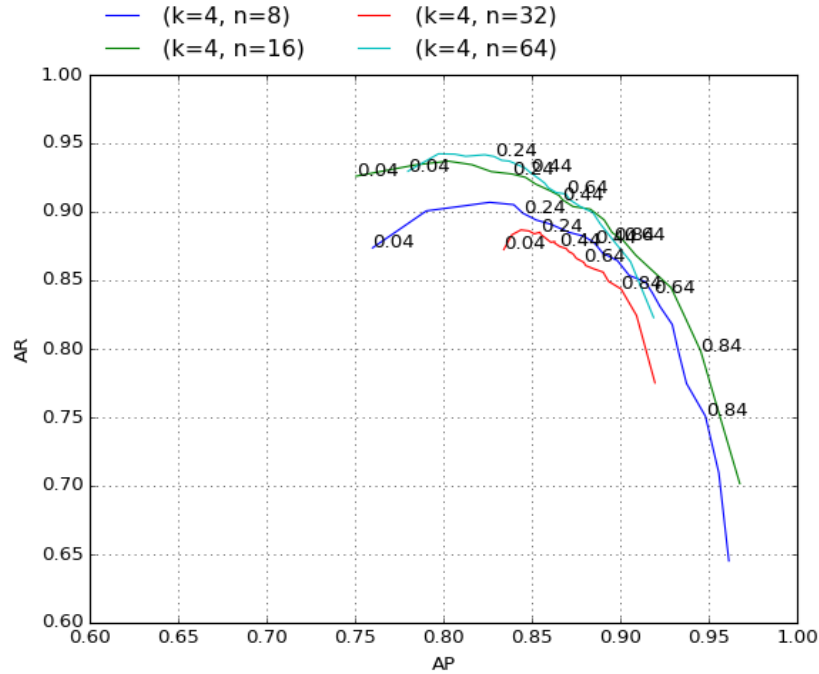
Θ_{th} bi tako lahko bila katerakoli vrednost iz intervala $\Theta_{th} \in (0,40; 0,60)$. V naših nadaljnjih eksperimentih smo uporabili pragovno vrednost $\Theta_{th} = 0,44$.



Slika 5.5: Vrednost F-1 pri različnih pragovnih vrednostih Θ_{th} za binarizacijo segmentacijske maske za mreže s 4 bloki in različnih številom filtrov. Oznaka v legendi ($k = 4, n = j$) označuje uporabo mreže s štirimi bloki in j filtri.

5.3.3 Obogatenje podatkov

Zaradi relativno majhne učne množice smo preizkusili uporabo obogatitve učnih slik, kar lahko pomaga pri učinkovitosti segmentacijske mreže pri posploševanju [26]. Deluje tudi kot regularizacija saj se dovolj globoke mreže lahko dobro prilagodijo učni množici, kar se lahko do neke mere zgodi tudi na naši učni množici, kljub uporabi izpusta kot metodo regularizacije. Slike smo obogatili z uporabo splošnih postopkov obogatitve kot so naključne horizon-



Slika 5.6: Vrednost AR in AP pri različnih mejah prepričanja za binarizacijo segmentacijske maske za mreže s 4 bloki in različnim številom filtrov. Uporabili smo mejne vrednosti Θ_{th} iz intervala $(0.04, 0.96)$.

talne in vertikalne preslikave in povečava in zmanjševanje slik. Preizkusili smo še obogatitve z glajenjem zaradi relativno pogostega pojava slabe izostrenosti polipov a hkrati slabšega priklica na polipih v teh regijah.

Za spreminjanje učne množice smo uporabili različne metode obogatitja podatkov:

1. **Splošno obogatitje:** Spreminjanje s horizontalno in vertikalno preslikavo in s povečavo in zmanjševanjem slike. Vsako sliko učne množice smo spremenili tako, da smo jo s 50% verjetnostjo preslikali vertikalno ali horizontalno. Po preslikavi smo sliko še povečali ali zmanjšali za 20% njene velikosti. Spremembo velikosti smo izvedli tako, da smo pri povečavi vzeli del slike in ga povečali na originalno velikost, pri pomanjšanju pa smo sliko pomanjšali in jo podložili s piksli konstantne

vrednosti do originalne velikosti. Zaradi modificirane kriterijske funkcije smo lahko podložene piksele nastavili na vrednost 0,5 in jih tako izločili iz učenja. S transformacijami smo želeli izboljšati robustnost na rotacije in spremembe v velikosti polipov.

2. **Glajenje:** Vhodne slike učne množice smo z 10% verjetnostjo zgladili z Gaussovimi jedrom naključne velikosti. Velikost jedra je bila enakomerno vzorčena iz intervala (12; 70) oziroma $(0,2\mu_{diag}; 1,2\mu_{diag})$, kjer je μ_{diag} povprečna dolžina diagonale anotacij učne množice. Cilj spremembe je bil izboljšava robustnosti na spremembe v izostrenosti slike. Pri obogatitvi z glajenjem smo uporabili tudi postopke splošnega obogatitve.

V Tabeli 5.5 lahko vidimo rezultate pri uporabi splošnega obogatitve S , obogatitve z glajenjem G in brez uporabe postopkov obogatitve B . Vidimo lahko, da uporaba splošnega obogatitve pomaga pri posploševanju mreže, kar rahlo zviša povprečni priklic na testni množici, saj se mreža nauči bolj robustnih značilnik. Natančnost se rahlo zniža kar bi lahko pomenilo, da se naučene značilke pogosteje izražajo na ozadju, ali pa je manjša natančnost rezultat detekcije novih manjkajočih anotacij.

Tudi z uporabo glajenja za simuliranje slabe izostritve smo dosegli izboljšavo v priklicu še posebej na sliki #6, kjer je izostrenost slabša kot pri ostalih slikah. Kot pri splošnem obogatitvi, je tudi pri glajenju povprečna natančnost rahlo padla. Tudi tu je težko določiti ali je bil padec povzročen zaradi večjega števila napačnih detekcij ali zaradi detekcij pravih polipov, ki zaradi manjkajočih anotacij niso bili označeni.

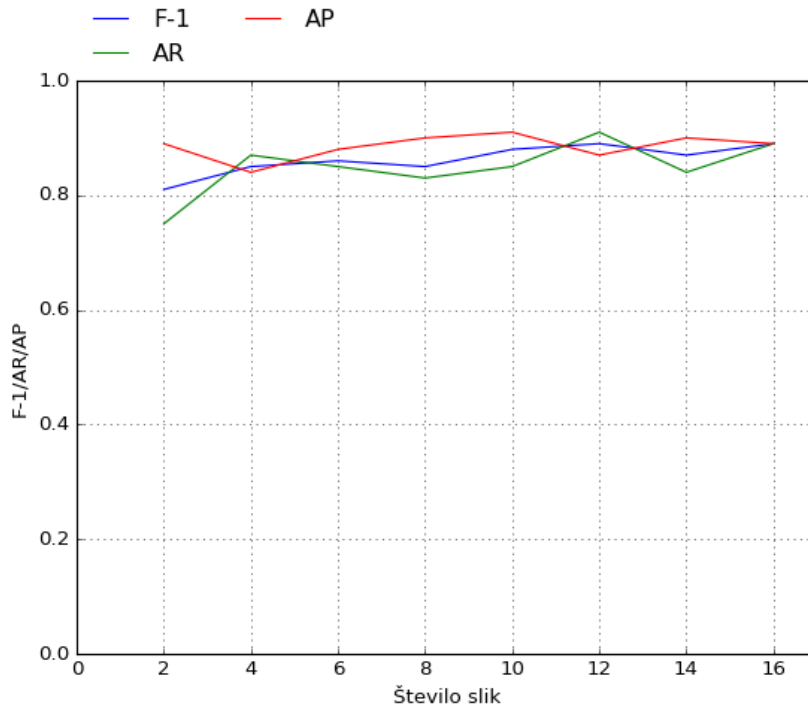
5.3.4 Vpliv količine podatkov

Možnost doseganja dobrih rezultatov na manjši količini slik je pomembna zaradi visoke zahtevnosti ročne anotacije. Na Sliki 5.7 so prikazane mere F -

Slika	S		G		B	
	rec.	prec.	rec.	prec.	rec.	prec.
# 1	0,93	0,85	0,95	0,83	0,92	0,86
# 2	0,95	0,88	0,95	0,86	0,92	0,89
# 3	0,93	0,86	0,94	0,84	0,91	0,87
# 4	0,92	0,89	0,93	0,88	0,87	0,91
# 5	0,94	0,87	0,92	0,87	0,93	0,87
# 6	0,86	0,87	0,91	0,86	0,75	0,92
# 7	0,91	0,88	0,91	0,89	0,85	0,88
μ	$92 \pm 3\%$	$87 \pm 1\%$	$93 \pm 2\%$	$86 \pm 2\%$	$88 \pm 6\%$	$88 \pm 2\%$

Tabela 5.5: Vrednosti natančnost in priklic predlagane metode na slikah testne množice z uporabo obogatitve učnih podatkov. **S** je označeno obogatitve z preslikavami in spremembo velikosti, **G** je označeno obogatitve z glajenjem, **B** pa so označeni rezultati brez obogatitve.

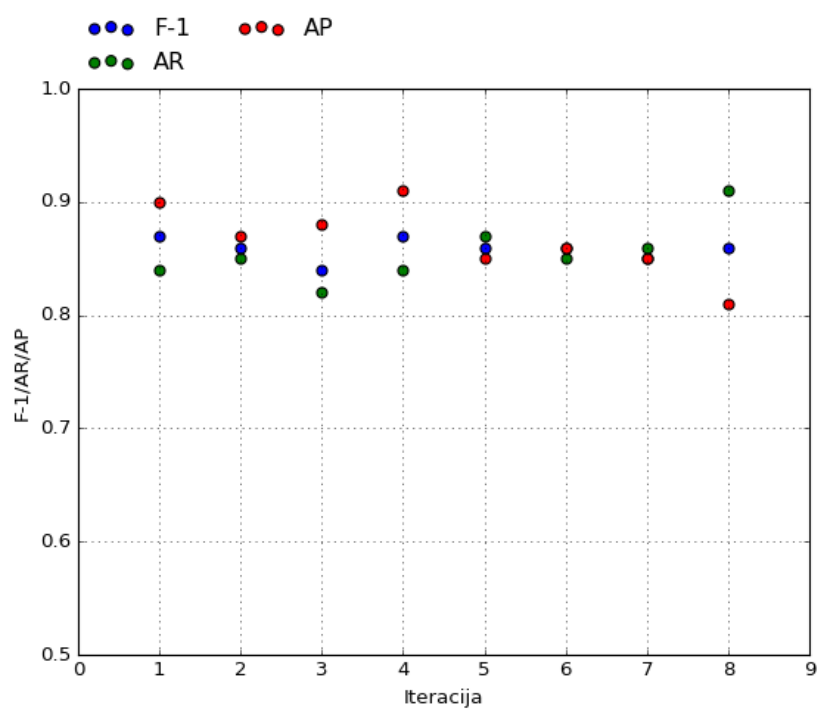
1, *AR* in *AP* za predlagano metodo SegCo, ki je bila naučena na različnem številu anotiranih slik iz učne množice. Večje število lahko poveča robustnost detektorja na vizualne spremembe polipov, vendar pa lahko dosežemo dobre rezultate že z manjšo količino slik, saj vsaka slika vsebuje vsaj nekaj 100 anotacij, ki lahko dobro predstavljajo spremenljivost izgleda polipov. Bolj kot število slik je pomembna dobra porazdelitev vizualnih lastnosti, zato mora učna množica vsebovati polipe, ki se razlikujejo po lastnostih kot so velikost, prosojnost, ozadje, barva, orientacija ali prekrivanje. Za ugotavljanje vpliva izbora slik na rezultate smo metodo večkrat naučili z uporabo naključno izbrane učne množice sestavljene iz 4 slik. Rezultate vidimo na Sliki 5.8. Opazimo, da so povprečne vrednosti $\mu_{F-1} = 0,86$, $\mu_{AR} = 0,85$ in $\mu_{AP} = 0,86$ relativno visoke, posamezne vrednosti pa se od povprečja lahko nezanemarljivo odmikajo. Vrednost *AR* se tako nahaja na intervalu od $AR = 0,82$ do $AR = 0,91$. Z manjšo količino slik lahko dosegamo dobre rezultate, vendar pa je uspešnost odvisna od izbora slik.



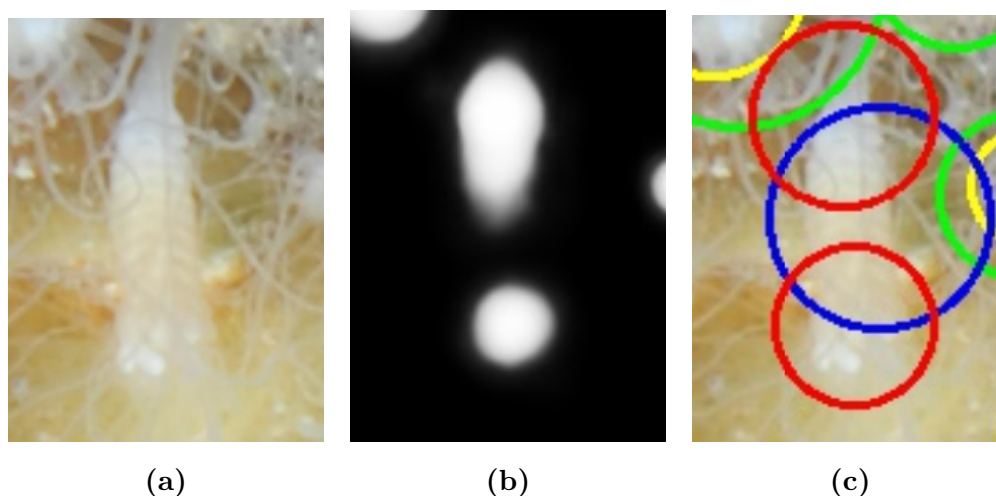
Slika 5.7: Vrednost F-1, AR in AP na testni množici pri različnih velikostih učne množice.

5.3.5 Napake pri detekciji s predlagano metodo

Napake se pri metodi SegCo najpogosteje pojavijo pri primerih polipov z atipičnim izgledom, ki so v učni množici slabše predstavljeni. Primer takega polipa in odziv mreže na njega lahko vidimo na Sliki 5.9. Vidimo lahko, da segmentacijska mreža slabo določi meje polipa in namesto ene detekcije metoda za interpretacijo maske določi dve, ki se nahajata na delih polipa, ki so bolj podobni tipičnim polipom. Uspešno detekcijo in verjetno tudi učenje predstavitev takih polipov onemogoča tudi podolgovata oblika polipa in anotacije zaradi katere je v takih primerih generirana učna maska še posebej nenatančna. Na Sliki 5.9c je anotacija polipa označena z modrim krogom.



Slika 5.8: Vrednost F-1, AR in AP na testni množici metode naučene na učni množici sestavljeni iz 4 naključno izbranih slik. Izbor in učenje smo ponovili večkrat.

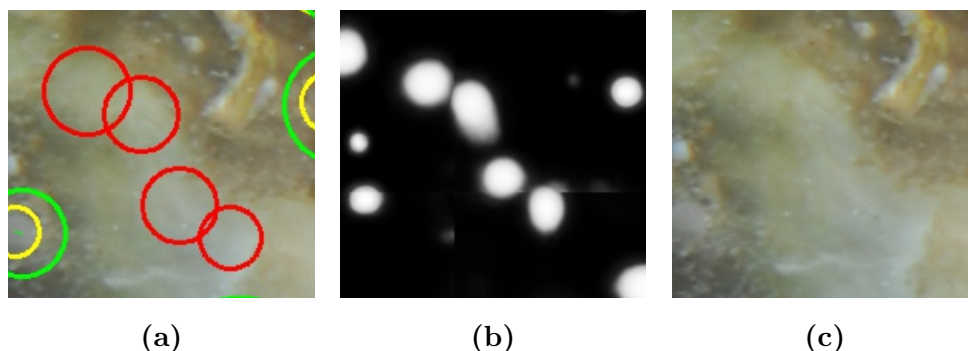


Slika 5.9: Primer slabe detekcije polipa atipičnega izgleda. (a) Podolgovat polip s prosojnim sredinskim delom zaradi česar pride do napačne segmentacije. (b) Segmentacijska mreža polip segmentira v dve regiji. (c) Detekcije pridobljene z interpretacijo segmentacijske maske so locirane na obeh koncih polipa, evalvacija pa jih zaradi razdalje do anotacije označi za napačne, anotacijo pa označi kot zgrešeno.

Na Sliki 5.10 vidimo zgrešene detekcije na svetlejšem delu ozadja. Na ozadju se lahko pojavljajo svetlejše strukture, ki iz ozadja rahlo izstopajo in jih lahko metoda prepozna kot polipe. Take strukture so težavne tudi za anotatorje, saj je težko zagotoviti, da taka struktura ni polip, še posebej če je slika na tem območju zamegljena, ali pa če so te strukture majhne.

5.4 Napake v testni množici

Evalvacijske eksperimente smo izvajali na podatkovni množici PCD, ki je bila definirana v delu [2], kar nam omogoča dobro primerjavo metode SegCo z detektorjem PoCo [2], poleg tega pa testna množica dobro predstavlja možne variacije v izgledu polipov. Zbirka vsebuje 3894 ročnih anotacij polipov, ki se nahajajo na 7 slikah. Slike so bile ročno anotirane s strani večih



Slika 5.10: Primer napačnih detekcij na delu ozadja. (a) Z rdečimi krogi so prikazane napačne detekcije metode SegCo na ozadju. (b) Segmentacijska maska iz katere metoda pridobi napačne detekcije. (c) Del ozadja, ki vsebuje strukture, ki jih metoda prepozna kot polipe.

anotatorjev z namenom minimiziranja anotacijskih napak, vendar so napake kljub trudu anotatorjev še vedno prisotne. Napake se pojavljajo predvsem v obliki zgrešenih anotacij in nam otežujejo evalvacijo in primerjavo metod, saj je zaradi njih težko oceniti pravilne natančnosti detekcijskih metod, ki pa polipe, ki niso bili anotirani, vseeno detektirajo, njihove detekcije pa zaradi manjkajoče anotacije označimo kot napačne.

Pridobljene detekcije smo ovrednotili tako, da smo ročno preverili vse detekcije, ki smo jih z evalvacijsko metodo označili za napačne. Kot pozitivne smo na novo označili le detekcije, za katere smo bili brez dvoma prepričani, da vsebujejo polip, veliko pa je tudi dvoumnih detekcij, ki morda vsebujejo polip, vendar pa je to težko z gotovostjo trditi. V Tabeli 5.6 je označeno število detekcij na posamezni sliki, ter števila pozitivnih in negativnih detekcij. V stolpcu n_t so označena števila pravilnih detekcij, ki v testni množici niso označene. Od detekcij, ki so bile označene kot nepravilne je na posamezni sliki v povprečju kar $72\% \pm 19\%$ takih, ki vsebujejo polipe. To kaže na težavnost ročne anotacije saj testna množica kljub večim anotatorjem vsebuje nezanemarljivo število napak. Zgrešenim polipom so skupne neželjene vizualne lastnosti kot so majhnost, prekrivanje, slaba osvetlitev, zamegljena

slika ali prosojnost. Glede na več kot 90% povprečni priklic, obstaja verjetnost, da je zgrešenih polipov še nekoliko več, vendar jih naša metoda ni odkrila.

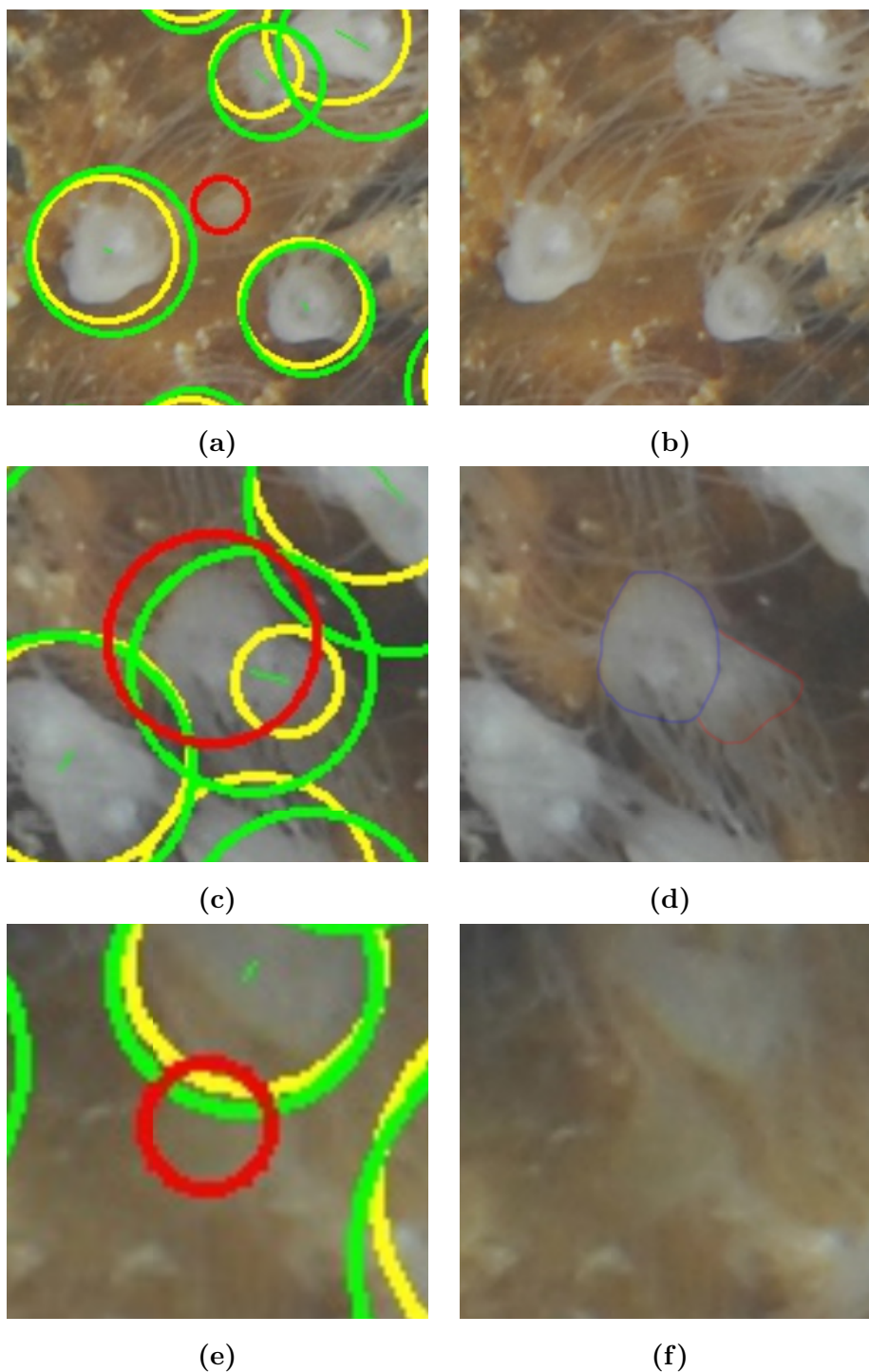
Dodatni primeri polipov z manjkajočimi anotacijami so prikazani na Sliki 5.12. Brez dovolj natančne metode, ki bi zgrešene anotacije uspela detektirati in jih izpostaviti je take napake zelo težko opaziti.

Slika	n_{gt}	n_{det}	n_{TP}	n_{FN}	n_{FP}	n_T	$\frac{n_T}{n_{FP}}$
# 1	455	487	416	39	71	36	0,51
# 2	655	692	612	43	80	69	0,86
# 3	543	580	498	45	82	55	0,67
# 4	770	787	706	64	81	79	0,98
# 5	723	729	653	70	76	71	0,93
# 6	350	365	315	35	50	31	0,62
# 7	398	384	354	44	30	14	0,46

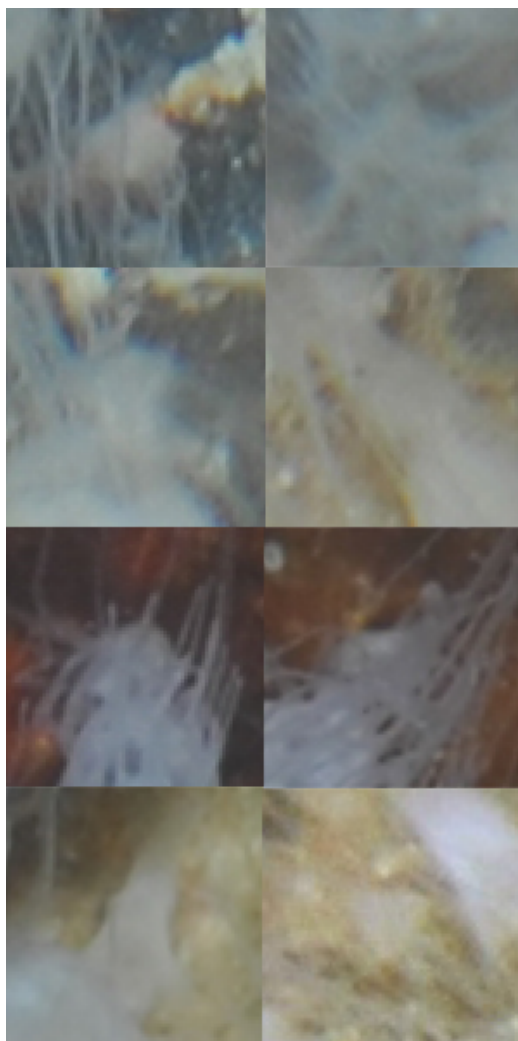
Tabela 5.6: Detekcije pridobljene s predlagano metodo. n_{gt} je število anotacij, n_{det} je število detekcij, n_{TP} je število detekcij, ki so bile označene kot pravilne, n_{FN} je število anotacij, ki jih je metoda zgrešila, n_{FP} je število detekcij, ki so bile označene kot nepravilne, n_T pa je število detekcij, ki so bile označene kot nepravilne, vendar so bile po ročnem pregledu označene kot pravilne.

5.4.1 Manjkajoče anotacije

Na Sliki 5.11 lahko vidimo primere polipov, ki niso bili vključeni v testno množico saj so jih anotatorji spregledali zaradi njihovih negativnih vizualnih lastnosti. Sliki 5.11b vsebuje zelo majhen polip, Slika 5.11d vsebuje dva prekrivajoča se polipa, ki ju je anotator označil kot en polip, na Sliki 5.11f pa se nahaja slabo osvetljen, prosojen polip, ki je posledično podobne barve kot ozadje.



Slika 5.11: Primeri zgrešenih ročnih anotacij na testni množici. **(a,b)** Zgrešena anotacija zelo majhnega polipa, ki ga predlagana metoda detektira. **(c,d)** Na Sliki **c** je z zelenim krogom prikazana anotacija, ki pokriva 2 polipa, saj je anotator spregledal delno prekrit polip v ozadju. Kroga rdeče in rumene barve pa označujeta detekciji metode. Na Sliki **d** sta z rdečo in modro barvo označeni obrobi obeh polipov za boljšo vidljivost. **(e,f)** Sliki prikazujeta detekcijo slabo osvetljenega prosojnega polipa, ki ga je anotator zgrešil.



Slika 5.12: Primeri pravih detekcij, ki jih zaradi manjkajočih anotacij označimo kot napačne.

5.4.2 Napačne anotacije

Anotatorji dosegajo zelo visoko natančnost, zato na slikah ni skoraj nobene anotacije za katero bi z gotovostjo lahko trdili, da ne vsebuje polipa. Nekaterim anotacijah na območjih z nizkim kontrastom med polipi in ozadjem, slabo osvetljenostjo ali slabo zbitrenostjo kamere je zelo težko določiti pravilnost. Primer take regije je na Sliki 5.13, ki prikazuje svetlo območje, ki je sicer enake barve in oblike kot veliko polipov, vendar je težko razločiti

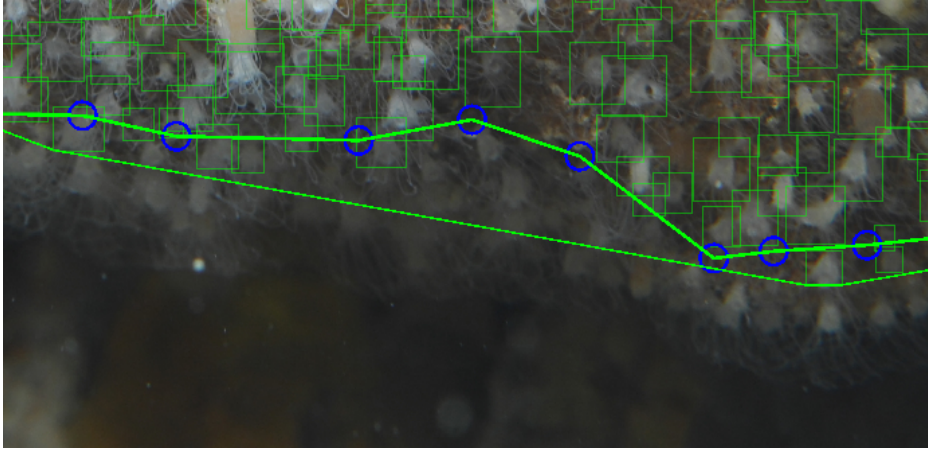
lovke ali natančne robove, ki regijo ločujejo od ozadja. Anotatorji in naša metoda so regijo označili kot polip, a se ta ne razlikuje bistveno od primerov prikazanih na Sliki 5.10 kar predstavlja dvoumnost nekaterih regij.



Slika 5.13: Primer regije, ki je bila anotirana kot polip, a so tipične značilnosti polipa slabo vidne.

5.4.3 Interesna regija

Na testni množici so interesne regije pogosto postavljene tako, da se nekateri anotirani polipi nahajajo na njihovem robu. Primer postavitve regije je prikazan na Sliki 5.14, kjer so črte interesnega poligona prikazane z zelenimi črtami, ki imajo na ogljiščih modre kroge. Zaradi tako postavljene interesne regije se pojavljajo primeri kjer se anotacija in detekcija nahajata na različnih straneh meje zaradi česar eno od njiju izločimo iz postopka evalvacije. Taki primeri so v evalvacijskem postopku označeni kot napačne detekcije ali pa zgrešene anotacije, odvisno od tega ali je iz procesa izločena anotacija ali detekcija. V nekaterih primerih je lahko interesna regija postavljena tako, da vsebuje neanotirane polipe kar ponovno pripelje do oznake napačnih detekcij v evalvacijskem postopku. Na Sliki 5.15 lahko vidimo primere napačnih oznak zaradi slabo postavljene interesne regije. Na Sliki 5.15a vidimo več pravih detekcij, ki so bile označene kot napačne zaradi manjkajočih ano-



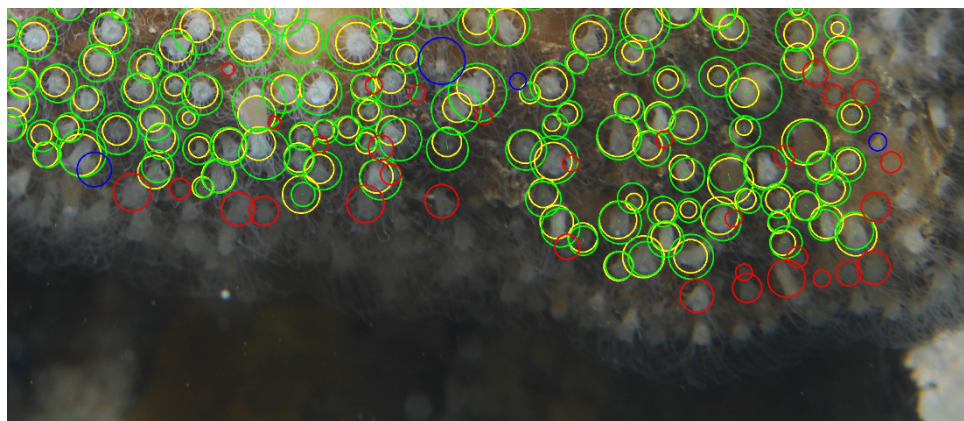
Slika 5.14: Anotacije testne množice, ki se nahajajo izven oziroma na robu interesne regije.

tacij ali pa slabše postavljene interesne regije. Obraten primer pa vidimo na Slikah 5.15b in 5.15c, kjer vidimo, da je metoda pravilno segmentirala polipe, ki pripadajo anotacijam, vendar se detekcije nahajajo izven interesne regije in jih zato evalvacijski postopek izloči.

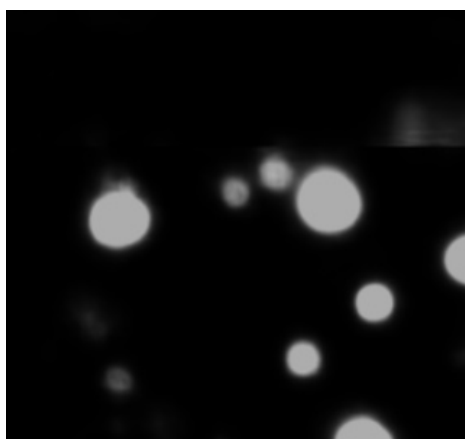
5.5 Primerjava z detektorjem PoCo

V delu [2] je predstavljena metoda PoCo, ki rešuje problem detekcije polipov. PoCo dosega zelo visoko stopnjo natančnosti na testni množici *PCD*, vendar pa na nekaterih slikah dosega relativno nizek priklic. Pri evalvaciji v [2] uporabljajo mejno vrednost τ_d enako mediani diagonal anotacij. Za boljšo primerjavo z metodo PoCo v tem poglavju tudi pri evalvaciji metode SegCo uporabimo τ_d enak mediani diagonal anotacij. Te rezultate nato primerjamo še z rezultati metode pri mejni vrednosti $\tau_d = 30$, ki jo uporabimo v ostalih eksperimentih.

V Tabelah 5.7 in 5.9 lahko vidimo rezultate obeh metod na podatkovni zbirki *PCD*. Metodi dosežeta zelo različni natančnost in priklic. Vidimo lahko, da ima SegCo v povprečju boljši priklic $AR = 0,99 \pm 0,01$, a slabšo



(a)



(b)



(c)

Slika 5.15: Problem uporabe interesne regije za evalvacijo. **(a)** Z rdečimi krogi označeni robni polipi so klasificirani kot napačne detekcije, čeprav vidimo, da pravilno detektirajo polipe. V postopku evalvacije se detekcije izven interesnega območja odstranijo, vendar pa se te detekcije še vedno nahajajo znotraj interesne regije, le da na tem območju polipi niso bili anotirani. **(b,c)** Sliki vsebujeta primera generirane segmentacijske maske in zgrešenih ročnih anotacij označenih z z modrimi krogi. Na Sliki **(b)** vidimo, da ima segmentacijska mreža močan odziv na lokacijah zgrešenih anotacij, kar pomeni, da se na lokacijah nahajajo detekcije, vendar pa se nahajajo izven interesne regije, saj se njena meja nahaja ravno med središčem zaznane regije in anotacije. Taki primeri so sicer redki, a vseeno vplivajo na rezultat evalvacije.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
#1	455	498	1,09	0,09	0,91	1,00	0,95
#2	655	726	1,11	0,11	0,90	1,00	0,94
#3	543	603	1,11	0,11	0,90	1,00	0,95
#4	770	835	1,08	0,08	0,92	1,00	0,96
#5	723	784	1,08	0,08	0,92	1,00	0,96
#6	350	383	1,09	0,09	0,89	0,98	0,93
#7	398	417	1,05	0,05	0,92	0,97	0,94
μ			$1,09 \pm 0,02$	$0,09 \pm 0,02$	$0,91 \pm 0,01$	$0,99 \pm 0,01$	$0,95 \pm 0,01$

Tabela 5.7: Rezultati metode SegCo na testni množici PCD uporabljeni v [2]. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 64 filtri na prvem nivoju. Vrednost τ_d je enaka mediani diagonal anotacij.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
#1	455	493	1,08	0,08	0,92	1,00	0,96
#2	655	718	1,10	0,10	0,91	1,00	0,95
#3	543	588	1,08	0,08	0,92	1,00	0,96
#4	770	802	1,04	0,04	0,96	1,00	0,98
#5	723	738	1,02	0,02	0,97	0,99	0,98
#6	350	371	1,06	0,06	0,92	0,97	0,94
#7	398	391	0,98	0,02	0,96	0,94	0,95
μ			$1,05 \pm 0,04$	$0,06 \pm 0,03$	$0,94 \pm 0,02$	$0,98 \pm 0,02$	$0,96 \pm 0,01$

Tabela 5.8: Rezultati metode SegCo na testni množici PCD uporabljeni v [2]. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 16 filtri na prvem nivoju. Vrednost τ_d je enaka mediani diagonal anotacij.

natančnost $AP = 0,91 \pm 0,01$. Kot smo že omenili, lahko del izgubljene natančnosti pripišemo anotacijskim napakam. Ocena priklica metode SegCo je

natančnejša saj se zaradi visoke natančnosti človeških anotatorjev le redko zgodi, da anotacija ne bi vsebovala polipa. Glavni vzrok napak pri oceni priklica, so anotirani polipi blizu roba interesne regije. Metoda te polipe občasno lokalizira izven interesne regije zaradi česar jih med evalvacijo zavržemo, anotacije pa obdržimo. Preprostejša mreža metode SegCo s 16 filtri dosega primerljive rezultate, ki so prikazani v Tabeli 5.8. Dosega marginalno slabši priklic $AR = 0,98 \pm 0,02$ in rahlo boljšo natančnost $AP = 0,94 \pm 0,02$ kot SegCo s 64 filtri.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	451	0,99	0,01	0,94	0,94	0,94
# 2	655	494	0,75	0,25	1,00	0,75	0,86
# 3	543	441	0,81	0,19	0,97	0,78	0,87
# 4	770	571	0,74	0,26	1,00	0,74	0,85
# 5	723	647	0,89	0,11	0,98	0,88	0,92
# 6	350	274	0,78	0,22	0,99	0,78	0,87
# 7	398	464	1,17	0,17	0,82	0,95	0,88
μ			$0,87 \pm 0,15$	$0,17 \pm 0,08$	$0,96 \pm 0,06$	$0,83 \pm 0,08$	$0,88 \pm 0,03$

Tabela 5.9: Rezultati metode PoCo na testni množici. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Vrednost τ_d je enaka mediani diagonal anotacij. Rezultati pridobljeni iz [2].

5.5.1 Uporaba strožje evalvacijske pragovne vrednosti

Metoda SegCo dosega dobre rezultate tudi pri bistveno strožji mejni vrednosti $\tau_d = 30$. Tabeli 5.10 in 5.11 prikazujeta rezultate pri vrednosti $\tau_d = 30$. Sprememba na strožjo mejno vrednost τ_d zniža natančnost in priklic metode, saj morajo biti centri detekcij bistveno bližje pripadajočim anotacijam, da jih evalvacijska metoda sprejme kot pravilne, kar pa povzroči, da se več slabo lokaliziranih detekcij označi kot napačne, posledično pa se teh detekcij ne poveže z anotacijami, ki jih tako označimo kot zgrešene. Kljub strogi meji τ_d

metoda SegCo dosega dobre rezultate. Vrednosti priklica in natančnosti rahlo padeta na $AR = 0,95$ in $AP = 0,87$, zaradi nekaterih slabše lokaliziranih detekcij in povezav med napačnimi detekcijami in blizu ležečimi anotacijami, vendar so to še vedno dobri rezultati, ki kažejo na to, da metoda SegCo dobro lokalizira detektirane polipe.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	492	1,08	0,08	0,85	0,95	0,89
# 2	655	704	1,07	0,07	0,87	0,95	0,91
# 3	543	592	1,09	0,09	0,84	0,94	0,89
# 4	770	816	1,06	0,06	0,88	0,95	0,92
# 5	723	771	1,07	0,07	0,88	0,96	0,91
# 6	350	375	1,07	0,07	0,86	0,94	0,90
# 7	398	409	1,03	0,03	0,88	0,92	0,90
μ			$1,07 \pm 0,02$	$0,07 \pm 0,02$	$0,87 \pm 0,02$	$0,95 \pm 0,02$	$0,90 \pm 0,01$

Tabela 5.10: Rezultati metode SegCo na testni množici *PCD* uporabljeni v [2]. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 64 filtri na prvem nivoju. Uporabili smo vrednost $\tau_d = 30$ za natančnejšo oceno lokalizacije polipov.

Z uporabo implementacije detektorja PoCo [57], ki je priložena delu [2], smo reproducirali rezultate dela [2]. V Tabeli 5.12 so prikazani reproducirani rezultati na množici *PCD*. Vidimo lahko, da so rezultati primerljivi z rezultati dela [2], ki so prikazani v Tabeli 5.9. V Tabeli 5.13 so prikazani reproducirani rezultati z uporabo mejne vrednosti $\tau_d = 30$ pri evalvaciji. Priklic in natančnost se ob spremembi τ_d poslabšata. To prikazuje dobro sposobnost lokalizacije metode SegCo v primerjavi z detektorjem PoCo.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	487	1,07	0,07	0,85	0,95	0,90
# 2	655	699	1,07	0,07	0,88	0,95	0,91
# 3	543	580	1,07	0,07	0,86	0,94	0,90
# 4	770	787	1,02	0,02	0,90	0,93	0,92
# 5	723	729	1,01	0,01	0,90	0,93	0,91
# 6	350	365	1,04	0,04	0,86	0,91	0,89
# 7	398	384	0,96	0,04	0,92	0,91	0,91
μ			$1,03 \pm 0,04$	$0,05 \pm 0,02$	$0,88 \pm 0,02$	$0,93 \pm 0,02$	$0,91 \pm 0,01$

Tabela 5.11: Rezultati metode SegCo na testni množici *PCD* uporabljeni v [2]. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 16 filtri na prvem nivoju. Uporabili smo vrednost $\tau_d = 30$ za natančnejšo oceno lokalizacije polipov.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	458	1,01	0,01	0,92	0,93	0,93
# 2	655	545	0,83	0,17	0,98	0,82	0,89
# 3	543	435	0,80	0,20	0,96	0,77	0,86
# 4	770	573	0,74	0,26	1,00	0,74	0,85
# 5	723	646	0,89	0,11	0,98	0,87	0,92
# 6	350	278	0,79	0,21	0,99	0,78	0,87
# 7	398	482	1,21	0,21	0,79	0,95	0,86
μ			$0,90 \pm 0,15$	$0,16 \pm 0,07$	$0,94 \pm 0,07$	$0,84 \pm 0,07$	$0,88 \pm 0,03$

Tabela 5.12: Reproducirani rezultati metode PoCo na testni množici *PCD*. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Vrednost τ_d je enaka mediani diagonal anotacij.

5.6 Primerjava z rezultati mreže RetinaNet

Metodo SegCo smo primerjali z metodo RetinaNet [1], ker je trenutno ena izmed najmodernejših metod učljivih detektorjev. Za učenje detekcijske mreže RetinaNet [1] smo uporabili isto učno množico kot za učenje metode SegCo.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	458	1,01	0,01	0,72	0,73	0,72
# 2	655	545	0,83	0,17	0,73	0,61	0,67
# 3	543	435	0,80	0,20	0,78	0,62	0,69
# 4	770	573	0,74	0,26	0,87	0,65	0,74
# 5	723	646	0,89	0,11	0,80	0,71	0,76
# 6	350	278	0,79	0,21	0,84	0,67	0,74
# 7	398	482	1,21	0,21	0,61	0,73	0,67
μ			$0,90 \pm 0,15$	$0,16 \pm 0,07$	$0,76 \pm 0,08$	$0,67 \pm 0,05$	$0,71 \pm 0,03$

Tabela 5.13: Reproducirani rezultati metode PoCo na testni množici *PCD*. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Uporabili smo vrednost $\tau_d = 30$ za natančnejšo oceno lokalizacije polipov.

Za ekstrakcijo značilnk smo uporabili mrežo ResNet-50 [37], ki je bila predhodno naučena za razpoznavanje objektov na slikah množice ImageNet [27].

V Tabeli 5.14 so prikazani rezultati metode RetinaNet na testni množici, ki jih lahko primerjamo z rezultati metode SegCo in detektorja PoCo [2] v Tabelah 5.9 in 5.10. Vidimo lahko, da RetinaNet dosega odlične rezultate na testni množici. Priklic je enakovreden predlagani metodi, vendar pa RetinaNet dosega višjo stopnjo natančnosti $AP = 0,90 \pm 0,03$, ki pa je zaradi manjkajočih anotacij v testni množici, potencialno manj zanesljiva. Metoda RetinaNet dosega dobre rezultate tudi na slabo izostrenih delih slike. Kot primer lahko vzamemo rezultate na sliki #6, kjer je slaba izostrenost zelo pogosta, RetinaNet pa vseeno dosega visoko stopnjo priklica $rec. = 0,89$ v primerjavi z detektorjem PoCo [2]. Za pridobljene rezultate smo v metodi RetinaNet nastavili mejno vrednost prepričanja na 0,4, saj je metoda pri tej vrednosti dosegla najboljše rezultate, seveda pa bi brez evalvacije vpliva mejne vrednosti na uspešnost metode težko zatrdili, da bi detektor s tako postavljeno mejo dobro deloval tudi na drugih množicah slik.

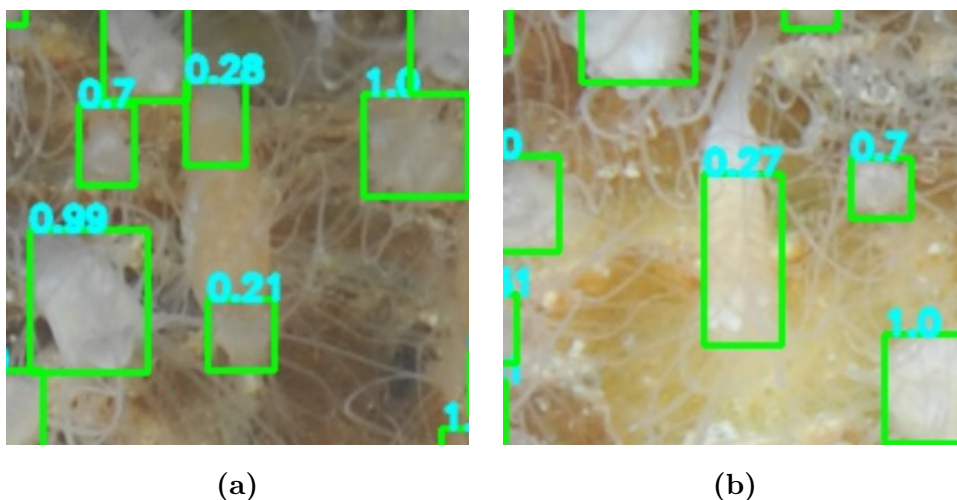
Detektor RetinaNet za detekcijo na posamezni sliki testne množice *PCD* potrebuje $12,9 \pm 1,4$ s, kar je rahlo hitreje od predlagane metode SegCo s 4 bloki in 64 filtri, a bistveno počasneje od metode s 16 filtri.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	455	476	1,05	0,05	0,86	0,92	0,89
# 2	655	692	1,06	0,06	0,89	0,95	0,92
# 3	543	580	1,07	0,07	0,87	0,95	0,91
# 4	770	706	0,92	0,08	0,96	0,89	0,92
# 5	723	734	1,02	0,02	0,91	0,93	0,92
# 6	350	335	0,96	0,04	0,92	0,89	0,91
# 7	398	375	0,94	0,06	0,92	0,87	0,89
μ			$1,00 \pm 0,05$	$0,05 \pm 0,02$	$0,90 \pm 0,03$	$0,91 \pm 0,03$	$0,91 \pm 0,01$

Tabela 5.14: Rezultati metode RetinaNet na testni množici *PCD*. n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Mejna vrednost prepričanja za označevanje pravih detekcij je bila nastavljena na 0,4. Pri evalvaciji smo uporabili vrednost $\tau_d = 30$.

5.6.1 Napake pri detekciji z metodo RetinaNet

Pri detekciji z metodo RetinaNet se pojavljajo podobne napake, kot pri predlagani metodi. Slika 5.16 prikazuje dva primera atipičnih polipov, ki jih metoda RetinaNet slabo detektira. Slika 5.16b prikazuje zgrešeno detekcijo istega polipa, ki je na Sliki 5.9 podan kot primer slabe detekcije predlagane metode. V tem primeru metoda RetinaNet polip sicer pravilno zazna, vendar pa detekcija nima visoke vrednosti prepričanja, zaradi česar bi morali mejno vrednost prepričanja bistveno spustiti, če bi želeli detektirati tudi ta polip. Z zmanjševanjem mejne vrednosti prepričanja za pravilne detekcije pa velikokrat žrtvujemo natančnost za višji priklic, saj zaradi tega kot pravilne označimo tudi nepravilne detekcije, ki imajo seveda nižjo vrednost prepričanja. Za uspešno detekcijo takih polipov, bi morali biti bistveno boljše



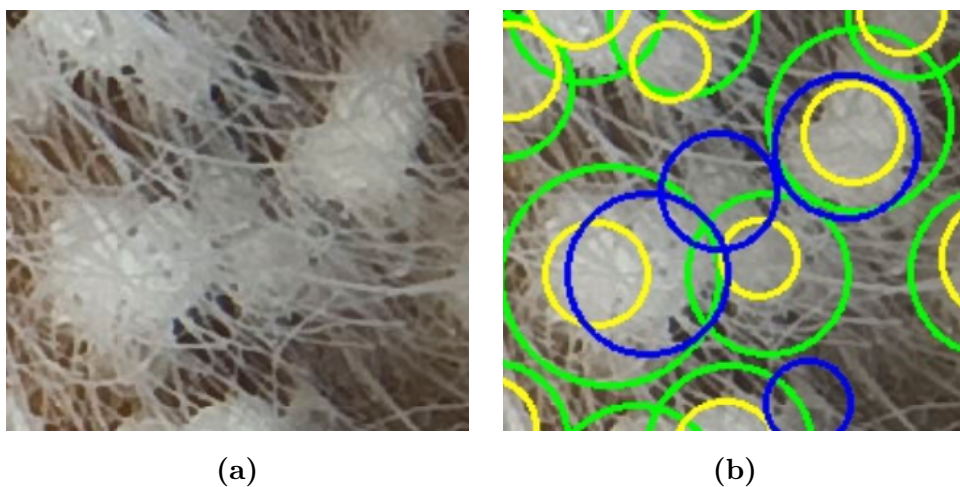
Slika 5.16: Slabo detektirani polipi z metodo RetinaNet

predstavljeni v učni množici, saj je polipov takega izgleda relativno malo.

Metoda RetinaNet ima težave pri detekciji prekrivajočih ali dotikajočih se polipov. Na Sliki 5.17 so prikazane detekcije na regiji, kjer se polipi močno prekrivajo. Z modrimi krogi so označeni zgrešeni polipi, z zelenimi krogi so označene ročne anotacije, ki so bile povezane z uspešnimi detekcijami, z rumenimi krogi pa uspešne detekcije. Opazimo, da so zgrešeni polipi močno prekriti z deli sosednjih polipov. Razlog za slabšo detekcijo takih polipov bi lahko bil postopek zatiranja slabše ocenjenih prekrivajočih se detekcij, ki je za tako vrsto detektorja nujen, saj predlaga veliko količino potencialnih regij, ki jih moramo večinoma odstraniti. Z odstranitvijo prekrivajočih se regij seveda omejimo sposobnost detekcije prekrivajočih se objektov.

5.7 Popravek anotacij testne množice

Težave z anotacijami v testni množici *PCD* opisane v Poglavju 5.4 nam otežujejo natančno evalvacijo metod. Manjkajoče detekcije in slabše postavljene interesne regije povzročajo napake pri oceni natančnosti. Z ročnim pre-



Slika 5.17: Zgrešeni polipi pri detekciji z metodo RetinaNet na regiji z visoko stopnjo prekrivanja polipov. Z zelenimi krogi so označene anotacije, ki so bile uspešno povezane z detekcijami, ki so označene z rumenimi krogi. Modri krogi označujejo zgrešene anotacije, ki niso bile povezane z nobeno detekcijo.

gledom detekcij, ki so bile v evalvacijskem postopku označene kot napačne smo testno množico dopolnili in metodo še enkrat ocenili na popravljeni množici slik. Novo testno množico imenujemo PCD_d . Da bi minimizirali prilaganje testne množice naučeni metodi smo za dopolnitev anotacij uporabili detekcije različnih metod. Uporabili smo metodo SegCo s segmentacijsko mrežo s 4 nivoji in s 16 in 64 filtri in naučeno mrežo RetinaNet. Dodatek manjkajočih anotacij lahko pripomore k boljši oceni uspešnosti metod, vendar je zelo verjetno, da popravljena testna množica ne vsebuje vseh potrebnih anotacij za idealno oceno metod, saj uporabljeni detektorji ne zaznajo vseh polipov, ki se na slikah nahajajo. Množico bi lahko dodatno popravljali še ročno, vendar zaradi očitne relativno visoke napake človeških anotatorjev, ne moremo zanesljivo dodati vseh manjkajočih polipov.

Tabela 5.15 prikazuje število dodanih anotacij za vsako sliko. Največji delež polipov smo dodali na sliki #4, kjer je stopnja prekrivanja zelo visoka.

Slika	n_{gt}	n_{tot}	n_{add}	$\frac{n_{tot}}{n_{gt}}$
# 1	455	492	37	1,08
# 2	655	732	77	1,12
# 3	543	600	57	1,10
# 4	770	879	109	1,14
# 5	723	820	97	1,13
# 6	350	382	32	1,09
# 7	398	412	14	1,03

Tabela 5.15: Dodane anotacije v dopoljnjeni testni množici. n_{gt} je število anotacij originalne testne množice, n_{tot} je število anotacij v dopoljnjeni množici, n_{add} pa je število detekcij, ki so bile dodane v množico.

To kaže na težavnost ročne anotacije prekrivajočih se polipov. Zaradi dvo-umnih lastnosti detektiranih regij je v nekaterih primerih težko določiti ali detekcija vsebuje polip ali ne. Testno množico smo dopolnili le z detekcijami, kjer lahko brez dvoma določimo vsebnost polipov, kar pa pomeni, da verjetno nismo dodali anotacij vseh polipov, ki jih metode pravilno detektirajo. Najmanj anotacij pa smo dodali na sliki #7 predvsem zaradi slabe izostrenosti, ki onemogoča natančno določevanje vsebnosti polipov v detekcijah.

5.7.1 Rezultati na dopoljnjeni testni množici

Na dopoljnjeni množici PCD_d smo ocenili metode SegCo, PoCo in RetinaNet. V Tabeli 5.16 vidimo rezultate metode SegCo na dopoljnjeni testni množici PCD_d . Zaradi dodanih manjkajočih anotacij je natančnost bistveno višja v primerjavi z rezultati v Tabeli 5.10. Priklic se z dodanimi anotacijami ne spremeni bistveno, saj metoda zgreši približno enak delež dodanih anotacij. V Tabeli 5.17 vidimo rezultate metode SegCo s 16 filtri. Manj kompleksna metoda dosega rahlo nižji priklic saj ne zazna vseh dodanih anotacij, vendar pa so rezultati vseeno primerljivi z metodo s 64 filtri po natančnosti in pov-

prečni meri $F-1$, saj je ta pri obeh metodah enaka $F-1 = 0,94 \pm 0,01$.

Tabela 5.18 prikazuje reproducirane rezultate metode PoCo na množici PCD_d z uporabo evalvacijske meje $\tau_d = 30$. V primerjavi z rezultati na množici PCD , ki jih prikazuje Tabela 5.13, je natančnost rahlo višja, saj del dodanih anotacij metoda PoCo pravilno zazna, vendar pa so bile te detekcije pri evalvaciji na množici PCD označene kot napačne. Priklic je pri evalvaciji na množici PCD_d nižji kot pri množici PCD , saj metoda PoCo ne zazna nekaterih novo anotiranih polipov, ki imajo pogosto atipične vizualne lastnosti.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	492	500	1,02	0,02	0,92	0,94	0,93
# 2	732	729	1,00	0,00	0,94	0,94	0,94
# 3	600	605	1,01	0,01	0,94	0,94	0,94
# 4	879	838	0,95	0,05	0,99	0,94	0,97
# 5	820	790	0,96	0,04	0,97	0,93	0,95
# 6	382	384	1,01	0,01	0,94	0,95	0,94
# 7	412	422	1,02	0,02	0,91	0,94	0,93
μ			$0,99 \pm 0,02$	$0,01 \pm 0,02$	$0,95 \pm 0,02$	$0,94 \pm 0,01$	$0,94 \pm 0,01$

Tabela 5.16: Rezultati metode SegCo na testni množici PCD_d . n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 64 filtri. Pri evalvaciji smo uporabili vrednost $\tau_d = 30$.

Izboljšava testne množice izboljša tudi oceno natančnosti metode RetinaNet. Tabela 5.19 prikazuje rezultate RetinaNet na izboljšani testni množici. Natančnost se kot pri predlagani metodi bistveno izboljša, saj obe metodi detektirata velik delež istih manjkajočih polipov. Priklic je pri uporabi metode RetinaNet slabši, saj del dodanih anotacij zgreši. Razlog za to bi lahko bilo visoko prekrivanje med polipi, saj se na nekaterih slikah del dodanih polipov delno prekriva z ostalimi polipi, zaradi česar bi bile lahko detekcije izločene v postopku zatiranja ne maksimalnih vrednosti. Kot primer lahko vzamemo

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	492	496	1,01	0,01	0,93	0,94	0,93
# 2	732	722	0,99	0,01	0,96	0,95	0,96
# 3	600	590	0,98	0,02	0,94	0,93	0,93
# 4	879	806	0,92	0,08	0,99	0,91	0,95
# 5	820	749	0,91	0,09	0,99	0,90	0,95
# 6	382	374	0,98	0,02	0,94	0,93	0,94
# 7	412	395	0,96	0,04	0,95	0,91	0,93
μ			$0,96 \pm 0,03$	$0,04 \pm 0,03$	$0,96 \pm 0,02$	$0,92 \pm 0,03$	$0,94 \pm 0,01$

Tabela 5.17: Rezultati metode SegCo na testni množici PCD_d . n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Za pridobivanje rezultatov se je uporabila segmentacijska mreža s 4 bloki in 16 filtri. Pri evalvaciji smo uporabili vrednost $\tau_d = 30$.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	492	458	0,93	0,07	0,75	0,70	0,72
# 2	732	548	0,75	0,25	0,74	0,56	0,64
# 3	600	435	0,72	0,28	0,80	0,58	0,67
# 4	879	573	0,65	0,35	0,90	0,58	0,71
# 5	820	646	0,79	0,21	0,85	0,67	0,74
# 6	382	278	0,73	0,27	0,85	0,62	0,72
# 7	412	485	1,18	0,18	0,62	0,74	0,68
μ			$0,82 \pm 0,16$	$0,23 \pm 0,08$	$0,79 \pm 0,08$	$0,63 \pm 0,06$	$0,70 \pm 0,03$

Tabela 5.18: Reproducirani rezultati metode PoCo na dopoljnjeni testni množici PCD_d . n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Pri evalvaciji smo uporabili vrednost $\tau_d = 30$.

slika #4 na kateri metoda RetinaNet dosega nižji priklic $rec. = 0.81$, velik del slike pa predstavljajo regije, kjer se polipi močno prekrivajo. Vidimo lahko, da imajo pri anotaciji prekrivajočih se polipov težave tudi človeški anotatorji, saj je pri dopolnitvi anotacij slike #4 bilo dodanih 97 anotacij. Dodatna razlaga za slabši priklic bi lahko bilo pretirano prileganje učni množici, kar bi

poslabšalo zmožnost učenja bolj splošnih značilk in s tem robustnost na vizualne spremembe polipov, saj se dodani polipi pogosto vizualno razlikujejo od predhodno anotiranih.

Slika	n_{gt}	n_{det}	Razmerje	Rel. napaka	prec.	rec.	$F-1$
# 1	492	478	0,97	0,03	0,93	0,91	0,92
# 2	732	706	0,96	0,04	0,96	0,92	0,94
# 3	600	584	0,97	0,03	0,95	0,93	0,94
# 4	879	715	0,81	0,19	1,00	0,81	0,90
# 5	820	750	0,91	0,09	0,98	0,90	0,94
# 6	382	339	0,89	0,11	0,97	0,86	0,91
# 7	412	379	0,92	0,08	0,95	0,88	0,91
μ			$0,92 \pm 0,05$	$0,08 \pm 0,05$	$0,96 \pm 0,02$	$0,89 \pm 0,04$	$0,92 \pm 0,01$

Tabela 5.19: Rezultati metode RetinaNet na testni množici PCD_d . n_{gt} je število ročnih anotacij na posamezni sliki, n_{det} pa število detektiranih polipov. Pri evalvaciji smo uporabili vrednost $\tau_d = 30$.

Poglavje 6

Sklep

V delu smo se ukvarjali s problemom detekcije polipov na slikah ostrig. Predlagali smo metodo SegCo, ki uporablja konvolucijsko nevronske mreže za semantično segmentacijo slike, nato pa z interpretacijo segmentacijske maske generira detekcije. Parametre segmentacijske mreže smo prilagodili danemu problemu in poiskali arhitekturo, ki dosega dobre rezultate hkrati pa je dovolj majhna za relativno hitro in robustno učenje, saj je bil cilj uporaba detekcijske metode v razviti aplikaciji. Preizkusili smo različne pristope obogatitve učne množice in z uporabo glajenja izboljšali uspešnost metode na slabše izostrenih regijah.

Odkrili smo pomanjkljivosti testne množice *PCD*, ki je bila s strani človeških anotatorjev označena večkrat [2] z namenom minimizacije napak. Testno množico smo dopolnili z ročnim pregledovanjem detekcij, ki so bile v evalvacijskem postopku označene kot napačne. S tem smo dodali velik del manjkajočih detekcij, kar nam omogoča bolj natančno evalvacijo metod na danem problemu.

Uspešnost metode SegCo smo ocenili in rezultate primerjali z metodo za detekcijo polipov PoCo [2] in metodo RetinaNet [1]. Metoda PoCo je na originalni testni množici *PCD* iz dela [2] dosegala slabši povprečni pri-

klic od naše metode, a je imela rahlo višjo natančnost pri evalvaciji z mejno vrednostjo τ_d , ki je bila nastavljena na vrednost mediane diagonal anotacij. Razlog za slabšo natančnost naše metode lahko v veliki meri pripišemo manjkajočim anotacijam originalne testne množice *PCD*. Pri evalvaciji z mejo $\tau_d = 30$, ki preverja sposobnost lokalizacije polipov, dosega metoda SegCo boljše rezultate od metode PoCo. Parametre metode RetinaNet smo prilagodili za uspešno delovanje na problemu detekcije polipov, vendar smo ugotovili, da metoda slabše detektira prekrivajoče se polipe, ki pa so v slikah pogost pojav. Poleg tega je metoda RetinaNet bistveno počasnejša od predlagane zaradi večje globine nevronske mreže in postopkov potrebnih za pridobivanje detekcij. Na dopoljnjeni testni množici PCD_d smo metode ocenjevali z uporabo mejne vrednosti $\tau_d = 30$. Metoda SegCo na množici PCD_d dosega višjo natančnost in priklic v primerjavi z metodo PoCo. Dopolnjevanje testne množice vpliva tudi na rezultate evalvacije metode RetinaNet, kjer se povprečna natančnost bistveno poveša, priklic pa se rahlo zmanjša, saj metoda RetinaNet slabše zaznava polipe, ki jih označujejo dodane anotacije. Rezultati bolj podrobne analize potrjujejo, da predlagana metoda SegCo dosega boljše rezultate kot trenutno najboljša specializirana metoda za štetje polipov Poco in trenutno najboljši učljiv detektor RetinaNet.

Razvili smo aplikacijo, ki omogoča preprosto upravljanje z anotacijami, učenje razvite metode na anotiranih slikah in detekcijo na poljubnih slikovnih množicah. Cilj razvite aplikacije je omogočiti preprosto uporabo sicer zahtevnih metod strojnega vida na problemih detekcije objektov. Aplikacija avtomatizira naloge štetja in označevanja objektov, ki bi sicer uporabniku vzele večjo količino časa.

6.1 Nadaljnje delo

Predlagana metoda dosega odlične rezultate na dani podatkovni množici, vendar pa bi lahko metodo aplicirali na številne druge vrste slik, kjer bi avtomatska detekcija objektov lahko poenostavila in minimizirala potrebno človeško interakcijo za štetje objektov. Zaradi načina interpretacije segmentacijske maske obstaja možnost, da je predlagana metoda omejena na detekcijo objektov, ki so približno okrogle oblike. Za oceno splošnosti delovanja metode bi bila potrebna še izvedba učenja in evalvacije na drugih podatkovnih množicah.

Veliko potencialnega dela je še na področju funkcionalnosti razvite aplikacije. Aplikacijo bi lahko dopolnili z dodatnimi funkcionalnostmi pri anotaciji, učenju, detekciji ali delu z podatkovnimi množicami. Uporabnikom bi lahko nudili bolj informativni prikaz napredka med učenjem mreže z uporabo manjše validacijske množice ali pa vizualizacijo detekcij med učenjem kar na sliki v uporabniškem vmesniku. Razvili bi lahko lažje ločevanje med učno in testno množico, bolj intuitivno izbiro posameznih naučenih modelov in druge izboljšave uporabniške izkušnje detekcije od učenja do inference.

Obstaja tudi možnost uporabe drugih arhitektur za detekcijo objektov. Detektorji kot so RetinaNet ali Mask-RCNN [1, 6] dosega odlične rezultate na različnih problemih detekcije objektov, RetinaNet pa dosega dobre rezultate tudi na problemu detekcije polipov in bi se morda na drugih vrstah slik odrezal bolje od predlagane metode. Pri naši metodi smo opazili, da lahko na problemu detekcije polipov mreže z manjšo kompleksnostjo dosežejo primerljive rezultate kot globlje mreže, kar pa bi morda lahko uporabili tudi pri detektorjih kot so RetinaNet, kjer bi lahko globoke mreže za pridobivanje značilnik kot je ResNet-101 [37] nadomestili z manj kompleksnimi mrežami.

Predlagano metodo bi morda lahko izboljšali z uporabno novejših arhitektur segmentacijskih mrež [33, 17], ki dosega dobre rezultate na se-

gmentacijskih izzivih [8]. Naslovili bi lahko tudi pomankljivosti metode za interpretacijo segmentacijske maske. Vsekakor obstaja še veliko možnosti za izboljšavo tako pri detekciji polipov, kot pri splošnih metodah detekcije objektov.

Literatura

- [1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2999–3007.
- [2] M. Vodopivec, R. Mandeljc, T. Makovec, A. Malej, M. Kristan, Towards automated scyphistoma census in underwater imagery: useful research and monitoring tool, *Journal of Sea Research* 142 (2018) 147 – 156.
- [3] A. Malej, T. Kogovšek, A. Ramšak, L. Catenacci, Blooms and population dynamics of moon jellyfish in the northern adriatic, *Cahiers de Biologie Marine* 53 (3) (2011) 337–342.
- [4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: *Advances in neural information processing systems*, 2015, pp. 91–99.
- [5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [6] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: *Computer Vision (ICCV), 2017 IEEE International Conference on*, IEEE, 2017, pp. 2980–2988.
- [7] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

-
- [8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.
- [9] G. Priyankara, O. Seneviratne, R. Silva, W. Soysa, C. De Silva, An extensible computer vision application for blood cell recognition and analysis, Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.
- [10] W. Xie, J. A. Noble, A. Zisserman, Microscopy cell counting and detection with fully convolutional regression networks, *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization* 6 (3) (2018) 283–292.
- [11] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp. 565–571.
- [12] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. Van Ginneken, C. I. Sánchez, A survey on deep learning in medical image analysis, *Medical image analysis* 42 (2017) 60–88.
- [13] J. Richeimer, J. Mitchell, Bounding box embedding for single shot person instance segmentation, arXiv preprint arXiv:1807.07674.
- [14] Y.-K. Ki, Accident detection system using image processing and mdr, *International Journal of Computer Science and Network Security* 7 (3) (2007) 35–39.
- [15] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

-
- [16] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, arXiv preprint arXiv:1511.00561.
- [17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE transactions on pattern analysis and machine intelligence* 40 (4) (2018) 834–848.
- [18] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.
- [19] Opis zmagovalne rešitve data science bowl 2018, <https://www.kaggle.com/c/data-science-bowl-2018/discussion/54741>, dostopano: 20.8.2018.
- [20] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, K. P. Murphy, Semantic instance segmentation via deep metric learning, arXiv preprint arXiv:1703.10277.
- [21] B. De Brabandere, D. Neven, L. Van Gool, Semantic instance segmentation with a discriminative loss function, *Deep Learning for Robotic Vision*, workshop at CVPR 2017.
- [22] K. Shirgoski, Computer-vision based polyp detection in underwater images, B.s. thesis, University of Ljubljana, Faculty of computer science (September 2015).
- [23] B. Bohte, Sistem za delno avtomatsko štetje polipov na slikah, B.s. thesis, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko (Februar 2015).
- [24] M. Vodopivec, R. Mandeljc, A. Malej, M. Kristan, Polyp counting made easy: two stage scyphistoma detection for a computer-assisted census in

- underwater imagery, Fifth International Jellyfish Bloom Symposium: Abstract book, Barcelona.
- [25] P. Dollár, S. Belongie, P. Perona, The fastest pedestrian detector in the west., *Proceedings of the British Machine Vision Conference 2* (3) (2010) 7–18.
- [26] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision (IJCV)* 115 (3) (2015) 211–252. doi: 10.1007/s11263-015-0816-y.
- [28] C. Cortes, V. Vapnik, Support-vector networks, *Machine learning* 20 (3) (1995) 273–297.
- [29] Y. LeCun, et al., Lenet-5, convolutional neural networks (2015).
URL <http://yann.lecun.com/exdb/lenet/>
- [30] Y. LeCun, C. Cortes, MNIST handwritten digit database.
URL <http://yann.lecun.com/exdb/mnist/>
- [31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324.
- [32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
- [33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: *Proceedings of the*

- IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
- [34] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, *nature* 323 (6088) (1986) 533.
- [35] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324.
- [36] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in: *Artificial Neural Networks–ICANN 2010*, Springer, 2010, pp. 92–101.
- [37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [38] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 5987–5995.
- [39] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, *arXiv preprint arXiv:1412.6806*.
- [40] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [41] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

-
- [42] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International Conference on Machine Learning* 37 (2015) 448–456.
- [43] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, J. Sun, Megdet: A large mini-batch object detector, *arXiv preprint arXiv:1711.07240*.
- [44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research* 15 (1) (2014) 1929–1958.
- [45] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: *Computer Vision and Pattern Recognition (CVPR)*, 2015.
URL <http://arxiv.org/abs/1409.4842>
- [47] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [48] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: *European conference on computer vision*, Springer, 2016, pp. 21–37.
- [49] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: *Computer Vision and Pattern Recognition (CVPR)*, 2017 IEEE Conference on, Vol. 1, IEEE, 2017, pp. 936–944.

-
- [50] A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, *Pattern recognition* 1 (1) (1968) 33–61.
- [51] R. Computing, Pyqt library, <https://riverbankcomputing.com/software/pyqt/intro>.
- [52] F. Chollet, keras, <https://github.com/fchollet/keras> (2015).
- [53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattemberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL <https://www.tensorflow.org/>
- [54] G. Bradski, The OpenCV Library, *Dr. Dobb's Journal of Software Tools*.
- [55] T. Fawcett, An introduction to roc analysis, *Pattern recognition letters* 27 (8) (2006) 861–874.
- [56] R. Jonker, A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing* 38 (4) (1987) 325–340.
- [57] R. Mandeljc, <https://github.com/rokml/polyp-detector> (2016).