

ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ БЕОГРАД

Катедра за сигнале и системе

**Детекција, класификација и сегментација
рециклажног отпада методама машинског учења**

Мастер рад

Кандидат

Јанко Шуштершић

3076/2022

Ментор

Проф. др Вељко Папић

Београд, 2024. године



Универзитет у Београду
Електротехнички факултет



Назив студијског програма: Електротехника и рачунарство

Ниво студија: Мастер академске студије

Модул: Сигнали и системи

Предмет: Компјутерска визија

Број индекса: 2022/3076

Јанко И. Шуштершић

**Детекција, класификација и сегментација
рециклажног отпада методама машинског учења**

Мастер рад

Комисија за преглед и одбрану:

1. Проф. др Вељко Папић – ментор
2. Проф. др Томислав Шекара

Датум одбране: _____

Оцена: _____

РЕЗИМЕ РАДА

Рециклажа је процес којим се материјали из отпада прикупљају, обрађују и претварају у нове производе или сировине са циљем очувања природних ресурса и смањења загађења. У постројењима за прераду отпада аутоматизација процеса сортирања је кључна. Често се у ту сврху користе покретне траке, опремљене пневматским роботима и видео камерама, док су се методе компјутерске визије на бази неуралних мрежа показале као ефикасан начин надгледања и доношења одлука. У овом раду представљена је софтверска реализација и оптимизација низа алгоритама машинског учења за детекцију, класификацију и сегментацију рециклажног отпада из птичије перспективе, а затим је анализирана њихова ефикасност. Методологија је обухватала детекцију отпада помоћу два различита модела, Faster R-CNN и YoloV5, класификацију уз помоћ DenseNet неуралне мреже и сегментацију користећи се U-Net дубоком конвулуционом мрежом. Дискутовани су њихов распоред и применљивост у индустрији кроз параметре тачности, брзине извршавања и искоришћених ресурса. Коначни резултати детектованих објеката за рециклажу приказани су на оригиналној слици помоћу маркера у боји, тако да кориснику олакшају употребу софтвера. Уз мало модификација, софтверско решење представљено у овом раду може бити примењено при асистенцији или потпуној аутоматизацији процеса сортирања отпада у рециклажним центрима.

Кључне речи: рециклажа, детекција, класификација, сегментација, Faster R-CNN, YoloV5, DenseNet, U-Net, конвулуционе неуралне мреже, обрада слике, компјутерска визија

ABSTRACT

Recycling is a process of waste collection and conversion into new objects or materials, with a goal of reducing pollution and saving energy. Automating the waste sorting stage in recycling plants is key. For that purpose, conveyor belts, video cameras and pneumatic robot arms are often used, while computer vision methods based on neural networks have proven to be efficient decision makers. This paper presents a software implementation and optimization of a few machine learning algorithms for detection, classification and segmentation of recycling waste photos taken from a bird view. Methodology includes two models for object detection – Faster R-CNN and Yolov5, one machine learning model called DenseNet for classification purposes, and a deep convolutional neural network U-Net for fine segmentation. Their layout and efficiency in realistic conditions are analyzed through accuracy, latency and resource consumption and cost. Final results are shown as colored regions on the original photo for easier software usage. Solution presented in this paper could be used as an assistance or autonomous sorting method in recycling plants with minimal modifications.

Keywords: recycling, detection, classification, segmentation, Faster R-CNN, YoloV5, U-Net, DenseNet, convolutional neural networks, image processing, computer vision

ЗАХВАЛНИЦА

Захваљујем се ментору проф. др Вељку Папићу на помоћи, саветима, стрпљењу и подршци током израде мастер рада. Такође се захваљујем члану комисије проф. др Томиславу Шекари на пажљивом читању и коментарима на тезу.

Јанко Шуштершић

У Београду, август 2024.

САДРЖАЈ

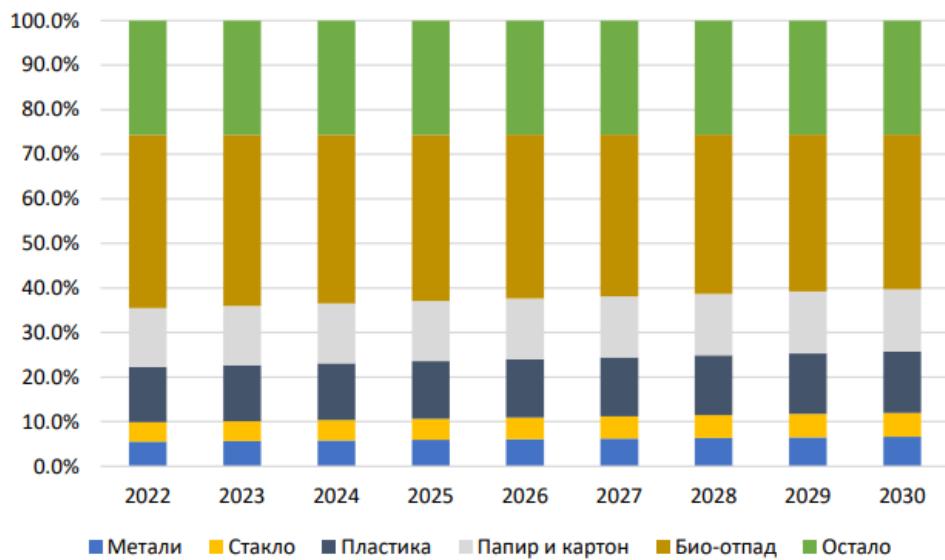
1. УВОД	1
1.1 Преглед релевантних појмова	2
2. СКУП ПОДАТКА	6
2.1 Скуп података за детекцију	9
2.2 Скуп података за класификацију	11
2.3 Скуп података за сегментацију	12
3. МЕТОДОЛОГИЈА РАДА	16
3.1 Детекција	17
3.2 Класификација	21
3.3 Семантичка сегментација	25
4. РЕЗУЛТАТИ	29
4.1 Резултати детекције	29
4.2 Резултати класификације	38
4.3 Резултати семантичке сегментације	44
4.4 Резултати сегментације инстанци	49
5. ДИСКУСИЈА И ЗАКЉУЧАК	53
6. ЛИТЕРАТУРА	55
7. ПРИЛОГ А	57
8. ПРИЛОГ Б	58
9. ПРИЛОГ В	61

1 УВОД

Рециклирање је процес којим се материјали из отпада прикупљају, обрађују и претварају у нове производе или сировине, уместо да заврше на депонији или буду спаљени. Циљ рециклирања је смањење количине отпада, очување природних ресурса, смањење загађења и подршка одрживијем начину живота. Раздвајање отпада по категоријама не обухвата само раздвајање у домаћинствима, при самом настанку, већ и сортирање у постројењима где се отпад раздваја и по мањим, корисним целинама, за даљу употребу.

Према програму управљања отпада у Републици Србији, стопа рециклирања комуналног отпада износила је 15.5% у 2020. години, што није испунило претходно постављене циљеве. Главни проблеми и изазови у вези са управљањем отпадом у Србији за период 2022-2031. године су:

- одвојено сакупљање рециклиабилног отпада, осим амбалажног отпада из домаћинства, још увек није адекватно спроведено;
- већи део комуналног отпада нетретиран се одлаже на депоније, а знатан део комуналног отпада још увек се одлаже на несанитарне депоније;
- недовољно је изграђених регионалних центара за управљање комуналним отпадом (што укључује регионалне депоније, трансфер станице, постројења за сепарацију рециклиабилног отпада, постројења за компостирање) [1].



Слика 1. Пројектовани састав генерисаног отпада за Србију у периоду 2022-2030. године
[1]

У постројењима за прераду отпада, аутоматизација сортирања отпада погодног за рециклирање је кључна. Методе компјутерске визије на бази неуралних мрежа су се показале као

ефикасан начин надгледања технолошких процеса у индустрији. Покретне траке, опремљене пневматским роботима и видео камерама, често се користе у ту сврху. Њихова способност да локализују жељене категорије отпада и издвоје га, заснива се на софтверском алгоритму који може поуздано да изврши детекцију, класификацију и сегментацију над датим сликама. Прецизна детекција и сегментација су уско повезани проблеми који су неопходни за одређивање тачне позиције предмета како би робот могао да изведе његово издавање и премештање [2].

У првом поглављу овог рада дати су кратак преглед појмова потребних за разумевање проблема, као и мотивација за израду софтверског решења које врши аутоматско издавање и визуализацију отпада у индустријском постројењу. Поред тога, описани су проблеми са којима се можемо сусрести у пракси и њихова најчешћа решења. У делу Скуп података детаљно су описане расположиве фотографије и ознаке коришћене у овом мастер раду, као и начини на који се извршило побољшање истих. Метеодологија рада је дата у трећем поглављу, заједно са теоријском основом иза сваког коришћеног модела. Испитани параметри модела, њихове тачности, перформансе и цена обучавања и употребе описани су полављу Резултати. На крају четвртог поглавља приказани су свеукупни резултати од улазне фотографије до излазне визуализације која се може применити у рециклажном центру. Последње поглавље се бави закључцима и потенцијалним побољшањима која су превазилазила обим овог мастер рада.

1.1 Преглед релевантних појмова

Детекција објекта подразумева налажење минималног правоугаоника који обухвата регион од интереса (слика 2.а). За разлику од сегментације, детекцијом не добијамо никакву информацију о облику сегмента, већ само координате обухватајућег правоугаоника, па је иницијално овај проблем лакше решити. За успешно детектовање објекта на слици потребно је направити обухватајуће правоугаонике за сваку инстанцу појављивања објекта, а затим и одредити њихову класу припадности. Већина алгоритама детекције користи почетне, предложене правоугаонике и покушава да класификује слику унутар сваког од њих. Нажалост, већи део алгоритама не узима у обзир целокупан контекст слике, већ покушава да класификује само мали део информација, па из тог разлога нема потпуну представу о околини.

Проблем који нам даје много више информација о самом региону јесте сегментација објекта. Уопштено говорећи, сегментација подразумева издавање карактеристичних делова слике или поделу слике на регионе сличног атрибута за њихову даљу детаљну анализу и дискусију. Сегментација се обично користи да се на слици лоцирају објекти и њихове границе. Сходно овој карактеристици, сегментацију можемо дефинисати као процес доделе ознаке сваком пикселу у слици тако да пиксели са истом ознаком деле одређене морфолошке карактеристике (слика 2.б). Врсте сегментације су семантичка сегментација и сегментација инстанци. Семантичка сегментација нам даје информацију само о припадности пиксела региону од интереса, док сегментација инстанци за сваку инстанцу објекта даје другачију вредност лабеле, и на тај начин прави разлику између другачијих објекта.



Слика 2. Разлика између детекције (а) и сегментације инстанци (б) објеката на слици

У овом раду, разматраће се примена дубоких неуралних мрежа као софтверског решења сегментације инстанци већег броја најзаступљенијих категорија рециклажног отпада, као и њихова ефикасност у реалном времену. Проблем се може поделити на низ изолованих, али једноставнијих проблема детекције, класификације и семантичке сегментације слике. У ту сврху, јавно доступни скуп података WaRP на сајту kaggle.com коришћен је као тренирајући скуп [3]. Овај скуп података чине фотографије са покретних трака рециклажних постројења у реалним, лоше осветљеним условима, које обухватају 28 различитих категорија отпада, у које спадају: пластичне и стаклене флашне, картонске амбалаже, алуминијумске конзерве, детерџенти и канистери.

Међутим, развијање кода машинског учења у индустрији чини свега 5-10% целокупне инфраструктуре која представља решење једног оваквог проблема [4]. У пракси, потребно је решити низ изазова како би се осигурало да модел функционише ефикасно и одрживо. Процес примене модела машинског учења у индустрији се углавном састоји из четири етапе: почиње са дефинисањем проблема (најчешће у виду пресликовања $X \rightarrow Y$) и захтева, затим се прелази на аквизицију и припрему скупа података, након чега се бирају и оптимизују модели који решавају представљени проблем, и на крају модел се користи и одржава у постројењу. Ови стадијуми нису потпуно независни, наиме проблем унутар једног од њих често захтева враћање уназад и решавање узрока на нивоу испод.



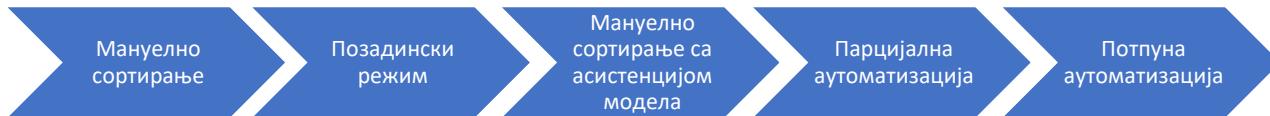
Слика 3. Етапе приликом пројектовања и примене модела машинског учења у индустрији [5]

Највећи део овог рада биће посвећен моделовању, који укључује одабир модела машинског учења за одговарајући проблем и оптимизацију његових перформанси. Приликом анализе разматрана је тачност и робусност модела, као и њихова брзина извршавања и скалабилност у реалним условима за потребе пословних захтева. У овом случају, различити модели и њихови параметри се оптимизују у односу на фиксиран, основни (енг. *benchmark*) скуп података тако да се достигну најбоље могуће метрике. Ово је модел-центричан приступ који је најзаступљенији при научном истраживању.

При прешавању реалних проблема међутим, скуп података никада није фиксиран. Скуп података може да се унапреди кроз разне облике претпроцесирања тако да зависност пресликавања улаза у излаз постане једноставнија, чиме би се проблем решио моделом са мање параметара. Један од примера било би неконзистентно обележавање скupa података од стране различитих лабелара који може једноставно да се реши, а притом омогућава побољшање метрика модела за ред величине више него коришћење бољег модела. Ово је основа за датацентричан приступ решавања проблема машинског учења који даје посебно добре резултате за мање обимне скупове података [5].

Други ефекат променљивог скupa података су померај података и померај концепта (енг. *data drift, concept drift*). Наиме, током времена улазни подаци или мапа пресликавања из улаза у излаз се мењају (нпр. промена осветљења у постројењу, промена типа или положаја камере...) који могу да утичу на перформансе модела. Решење овог проблема лежи у константном надгледању модела, прикупљањем аналитичких података и формирању петље повратних информација тако да се модел може ажурирати када до промене дође. Неопходно је покрити развијени софтвер тестовима чија је пролазност обавезна и која служи као први ниво заштите од случајних грешака приликом ажурирања модела.

Увођење модела у систем се врши постепено, а стадијуми замене мануелног сортирања отпада са аутоматизованим моделом дати су на слици 4. Првобитно софтвер ради у режиму позадинске евалуације, када је потпуна одлука о сортирању на човеку, док се ефикасност модела евалуира у поређењу. Насупрот томе, парцијална аутоматизација подразумева људску интервенцију када модел није сигуран. Хибридно сортирање и парцијална аутоматизација представљају међу кораке до коначног резултата потпуне аутоматизације када преостаје само одржавање система. Битно је да у сваком тренутку постоји опција промене верзије модела или повратка на претходни ниво уколико нису испуњени прописани захтеви у датом тренутку.



Слика 4. Нивои самосталности модела приликом аутоматизације процеса

Модел се мора неприметно интегрисати са ширим софтверским системом, било преко АПИ-ја, микросервиса или других механизама. Одлучивање о томе где ће модел да се извршава (нпр. у облаку, локално...) зависи од кашњења, трошкова, захтева за безбедност и контролу. Неке апликације захтевају тренутна предвиђања појединачних одбира. Код њих је изазов осигурати да модел може да рукује подацима у реалном времену и да пружи брзе одговоре. Други сценарији могу укључивати периодично генерирање предвиђања на групама одбира, што захтева од система да ефикасно рукује великим величинама података. У овом раду, поред одабира и оптимизације модела, дотаћи ћемо се неких од ових проблема у индустрији: обраде података на улазу и излазу, излазног решења за евалуацију и тестирања модела.

2 СКУП ПОДАТАКА

У рециклажним центрима, отпад се аутоматски раздваја по мањим целинама на покретним тракама, те је за то потребно разликовати више под-категорија:

1. пластичне флашe
2. стаклене флашe
3. картонске амбалажe
4. детерценти
5. канистери
6. алуминијумске конзерве

За прве три категорије, корисно је знати њихову величину, боју и употребну сврху, зато што се технологије за рециклажу разликују у зависности од ових параметара. У ту сврху, развијен је WaRP (скраћено од енг. *Waste Recycling Plant*) скуп података, који садржи ручно обележене слике сортиране у 28 категорија. Ове фотографије су добијене са покретних трака индустријских центара у реалним, неидеално осветљеним условима, објеката који могу бити тешко деформисани или да се преклапају, и покривају све горе наведене скупове (слика 5) [2].

Пластичне флашe су подељене у 17 мањих категорија (префикс *bottle-*) у зависности од величине, боје и употребне сврхе. Суфикс пун (-*full*) означава да ли је флашa пуна ваздуха или спљоштена, што је битно за успешно маневрисање роботом који их премешта. Стаклене флашe (префикс *glass-*) се према боји могу поделити у тамне, зелене и провидне, док се картонске амбалажe (префикс *cardboard-*) према употреби деле на оне од сокова или млека. Конзерве (*cans*) и канистери (*canister*) немају под-категорије, док се кутије од детерцената могу сврстати у четири класе према боји (префикс *detergent-*).

Скуп је подељен на три дела: WaRP-D, WaRP-C, и WaRP-S, за обучавање модела детекције, класификације и сегментације, редом (слика 6). У табели 1 дат је детаљни преглед броја примера у сваком од скупова. У претходном поглављу поменули смо да је систем вештачке интелигенције сачињен из два дела, кода машинског учења и скупа података. Развијање модела је итеративни процес који оптимизује модел, хиперпараметре и податке кроз фазу тренирања, а уз помоћ анализе грешке. Приликом пројектовања неуралних мрежа врши се подела скупа података на обучавајући (~70%), валидациони (~15%) и тестирајући скуп (~15%).

Главни изазови приликом оптимизације су:

1. Успех на тренирајућем скупу, који се мери просечном грешком,
2. Успех на валидационом и тестирајућем скупу,
3. Успех у виду бизнис метрика/циљева пројекта.



Слика 5. Примери отпадног материјала подељених по категоријама [2]



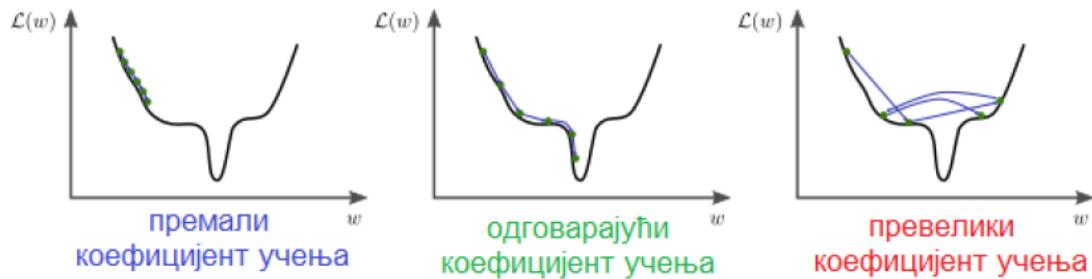
Слика 6. Визуелна репрезентација скупова WaRP-D, WaRP-C, и WaRP-S са одговарајућим лабелама [3]

Категорија	WaRP-D трен.	WaRP-D вал.	WaRP-D тест.	WaRP-C трен.	WaRP-C вал.	WaRP-C тест	WaRP-S трен.	WaRP-S тест
bottle-blue	453	82	87	520	114	106	3	1
bottle-green	342	61	65	382	84	75	3	1
bottle-dark	376	75	80	473	60	96	3	1
bottle-milk	273	51	54	284	63	60	3	1
bottle-transp	793	154	164	1174	258	235	3	1
bottle-multicolor	99	26	28	104	23	31	3	1
bottle-yogurt	223	38	41	227	50	42	3	1
bottle-blue-full	225	38	40	234	51	45	3	1
bottle-transp-full	383	74	79	432	96	93	3	1
bottle-dark-full	144	29	31	152	33	36	3	1
bottle-green-full	198	31	33	195	43	35	3	1
bottle-multicolor-full	86	19	20	88	19	22	3	1
bottle-milk-full	90	20	21	91	19	21	3	1
bottle-oil	211	43	46	228	48	48	3	1
bottle-oil-full	15	8	8	20	4	8	3	1
bottle-blue5l	289	56	60	339	74	75	3	1
bottle-blue5l-full	65	22	23	73	16	24	3	1
glass-transp	133	32	34	145	32	37	3	1
glass-dark	109	23	24	112	24	25	3	1
glass-green	109	22	23	111	24	25	3	1
juice-cardboard	192	59	63	213	47	71	3	1
milk-cardboard	278	80	85	320	70	96	3	1
detergent-white	261	39	42	262	57	44	3	1
detergent-color	237	40	43	243	53	44	3	1
detergent-transparent	208	37	39	215	47	42	3	1
detergent-box	50	16	17	55	11	17	3	1
canister	118	26	28	123	26	30	3	1
cans	412	83	88	461	101	10	3	1
Укупно	6372	1284	1366	7276	1547	1493	84	28

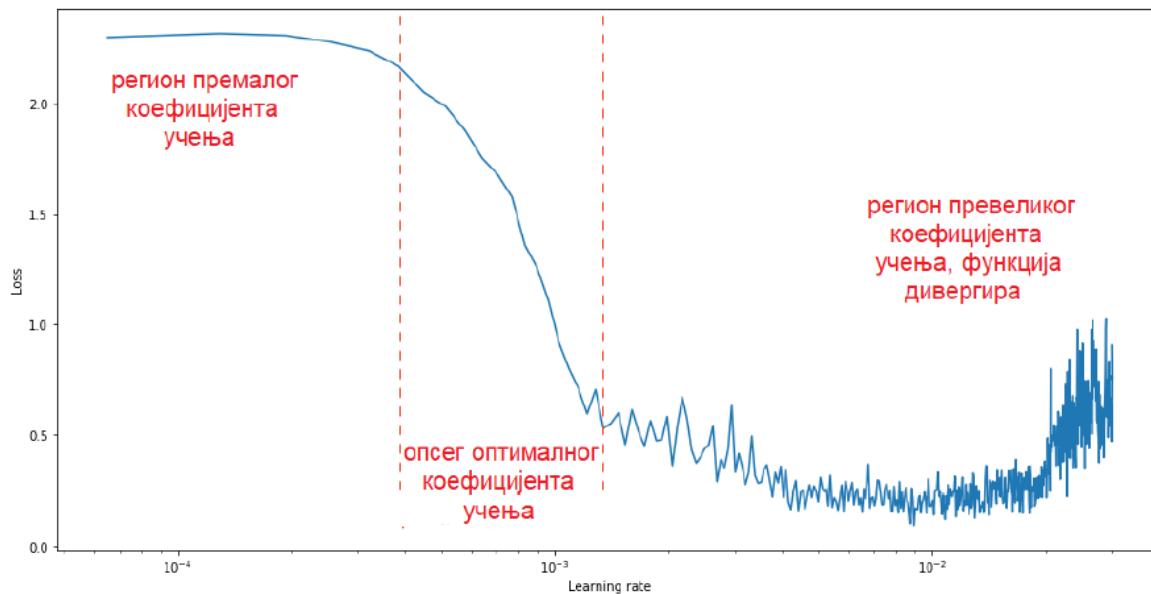
Табела 1. Детаљан приказ броја примера у скуповима WaRP-D, WaRP-C, и WaRP-S [2]

На обучавајућем скупу података мрежа прилагођава вредности тежина између неурона сходно вредностима улаза. Улога валидационог скупа је заштита од преобучавања. Наиме, приликом учења може се догодити да мрежа показује врло добре перформансе (велику тачност и малу средњеквадратну грешку) на подацима из обучавајућег скупа, док на неком другом скупу податка има лоше перформансе. У том случају каже се да је дошло до појаве преобучавања, односно да је мрежа почела да учи на особеностима улазних података. Заштита од преобучавања може се остварити ранијим заустављањем обучавања мреже уколико у неколико узастопних итерација долази до константног пораста грешке на валидационом скупу.

Раним заустављањем онемогућава се достизање максималне тачности на обучавајућем скупу, али се обезбеђује боља генерализација на још “невиђеним” подацима. Насупрот томе, превелике вредности коефицијента учења доводе до великих промена тежина (енг. *overshooting*) што узрокује немогућност да се дође до минимума, и потенцијалне дивергенције функције грешке, што је илустровано на сликама 7 и 8 [6].



Слика 7. Илустрација обучавања неуралне мреже са различитим вредностима коефицијента учења



Слика 8. Графички приказ функције грешке током обучавања употребом различитих вредности коефицијента учења

2.1 Скуп података за детекцију

Скуп података за детекцију WaRP-D састоји се од 1,962 фотографије мешовитог отпада у тренирајућем, 490 у валидационом и 522 у тестирајућем скупу. Највећи део слика је у *full HD* резолуцији од 1920×1080 пиксела, док је мањи део у нешто нижој резолуцији са истим односом висине и ширине. Свака фотографија у просеку садржи 3 објекта за рециклажу који су описаны својим ограничавајућим регионима (енг. *bounding boxes*) и класом припадности у текстуалном формату. Један такав пример визуализован је на слици 9, тако да су различите класе наговештene другачијом бојом.

Сваки регион описан је са по 5 параметара: својом класом (0-27), координатама центра (x , y) и дужинама страница кутије (w , h) у релативним димензијама. За потребе неких модела детекције, прво је неопходно пресликati ове координате у облик (x_1, y_1, x_2, y_2, S) , где су (x_1, y_1) и (x_2, y_2) апсолутне координате доњег левог и горњег десног темена правоугаоника, а S представља површину правоугаоника. Такође, слике су скалиране на димензије 480x270 и 256x144 и подељене на мини шарже величине 4 како би свака шаржа и параметри модела могли да се уклопе у расположиву меморију графичке картице.



Слика 9. Обучавајући пример из скupa података за детекцију рециклажног отпада

Важно је напоменути да се овај скуп података никако не сматра недовољним, штавише, расположив је велики број слика за које је просечном моделу потребно неколико сати да их обради у једном тренирајућем циклусу. Самим тим аугментације података типа ротације и окретање у огледалу по x и у осама зарад повећања броја примера нису неопходне.

Неке од занимљивих трансформација слика које могу да повећају метрике јесу мозаичка и насумична афина трансформација, приказане на слици 10. Мозаичка трансформација обухвата пресликање најчешће четири слике у једну и на тај начин помаже моделу да боље разликује објекте различитих облика и величина на истом примеру. Насумична афина трансформација укључује насумичне транслације, ротације, скалирање и смицање слика. У комбинацији, ове две трансформације смањују вероватноћу преобучавања повећавају свеукупну тачност модела [7].



Слика 10. Примери мозаичке (горе) и комбинације мозаичке трансформације са насумичним скалирањем слика (доле)

2.2 Скуп података за класификацију

WaRP-C скуп за класификацију се састоји од преко 8000 исечака слика издвојених из скупа за детекцију који варирају између 35 и 703 пиксела ширине и дужине. Слике су различитих димензија а број примера по класама није балансиран како би се осликало реално стање у индустријским условима. Канистери су, као класа, најмање заступљени са само 123 примера у тренирајућем скупу, док је најмање заступљена под-класа „пуна флаша за уље“ са 20 примера. Насупрот томе, најзаступљенија под-класа је провидна флаша са преко 1000 различитих примера. Скуп је подељен на тренирајући (70%), валидациони (15%) и тестирајући (15%) тако да је однос заступљености сваке класе очуван.

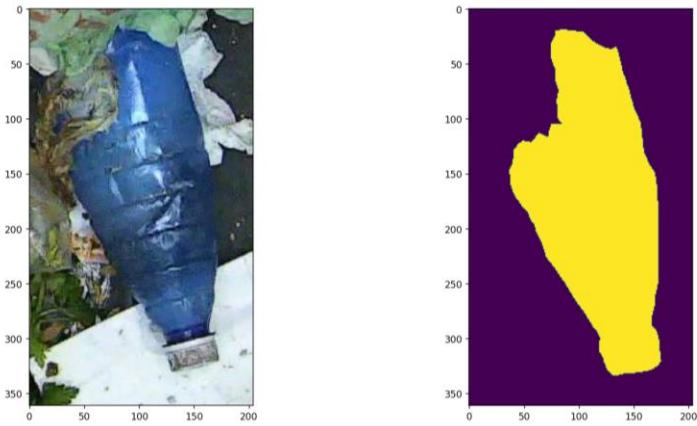
У сврху обучавања модела са надгледањем првобитно је било потребно унификовати широк распон димензија слика скалирањем на величину 224x224, а затим попунити остатак слике нулама како би се одржале њихове пропорције. Фотографије су такође окренуте хоризонтално и вертикално, као у огледалу, са вероватноћом од 0.5 како би се повећала моћ генерализације модела. Пример слике од учитавања до крајњег резултата приказан је на слици 11.



Слика 11. Класификациони примери пластичне флаше и канистера са трансформацијама

2.3 Скуп података за сегментацију

Скуп података за сегментацију WaRP-S је најмањи и саджи само по 4 слике рециклажног отпада из сваке под-класе, од којих је по једна искоришћена за тестирање а остале три за тренинг и валидацију модела семантичке сегментације. Слике и лабеле су високог квалитета са пажљиво одабраним примерима великог деформитета, али недостатак броја примера за потребе обучавања модела надгледаног учења је тешко надоместити. Димензије слика су исте као у класификационом скупу тако да је претпроцесирање било слично. Слике су скалиране на квадратне димензије 224x224 и попуњене нулама, само овог пута је лабела у облику слике такође морала бити скалирана и претворена у бинарну слику. Један пример слике и одговарајуће лабеле дат је на слици 12.

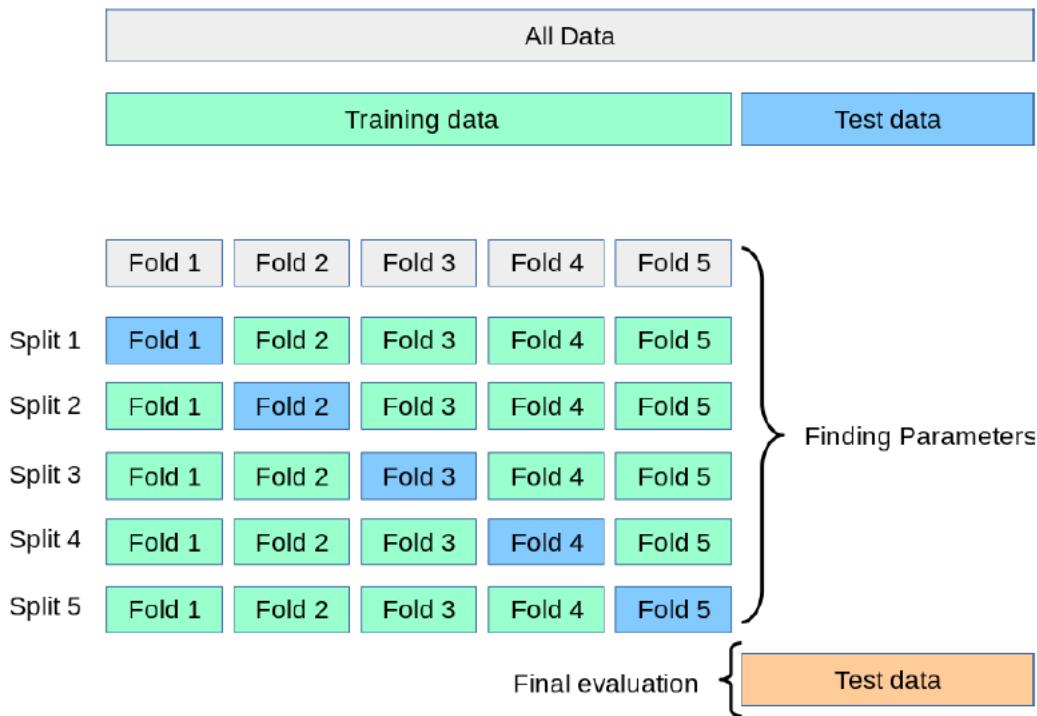


Слика 12. Пример слике и лабеле пластичне флаше у проблему сегментације

Цена означавања једног оваквог скupa је знатно већа него лабелирања скупова за детекцију и класификацију. Како би се обезбедио скуп високог квалитета, потребно је уложити доста времена у обележавање сваког пиксела слике. Такође, неконзистентности између лабелара могу бити честе, те има смисла што је сегментациони скуп података најмањи између претходна три. Скуп фотографија и лабела је проширен тако што су слике пресликане осносиметрично са вероватноћом 50% по x и y оси, као и у претходном поглављу. Упркос томе, скуп података је мали за проблеме овог типа, па је стандардни приступ поделе на тренирајући и валидациони скуп, тј. валидација изостављањем, неефикасан начин евалуације модела машинског учења. Уместо тога модел је током тренинга евалуиран методом унакрсне валидације.

Циљ унакрсне валидације је процена перформанси и генерализације модела машинског учења, користећи доступне тренинг податке на ефикаснији начин. Основна идеја је поделити тренирајући скуп података на више подскупова, тренирати модел на некима од њих, а затим га евалуирати на преосталим. Овај процес се понавља неколико пута, а просечна перформанса се користи као коначна естимација метрике на непознатим подацима. Предност ове методе је што уједно добијамо и варијансу ове естимације. Са друге стране, мана унакрсне валидације је то

што се модел тренира неколико пута, али како је скуп података мали, ни цена тренирања није велика. На слици 13 је илустрована једна подела скупа, при чему је у нашем случају одабран број подскупова $k = 3$. Модел се тренира на два подскупа док се на трећем евалуира. Овај процес се понавља три пута, по једном за сваки подскуп, а затим се рачуна просечна метрика у сва три случаја и користи се за даље подешавање хиперпараметара модела.



Слика 13. Илустрација поделе скупа приликом унакрсне валидације [8]

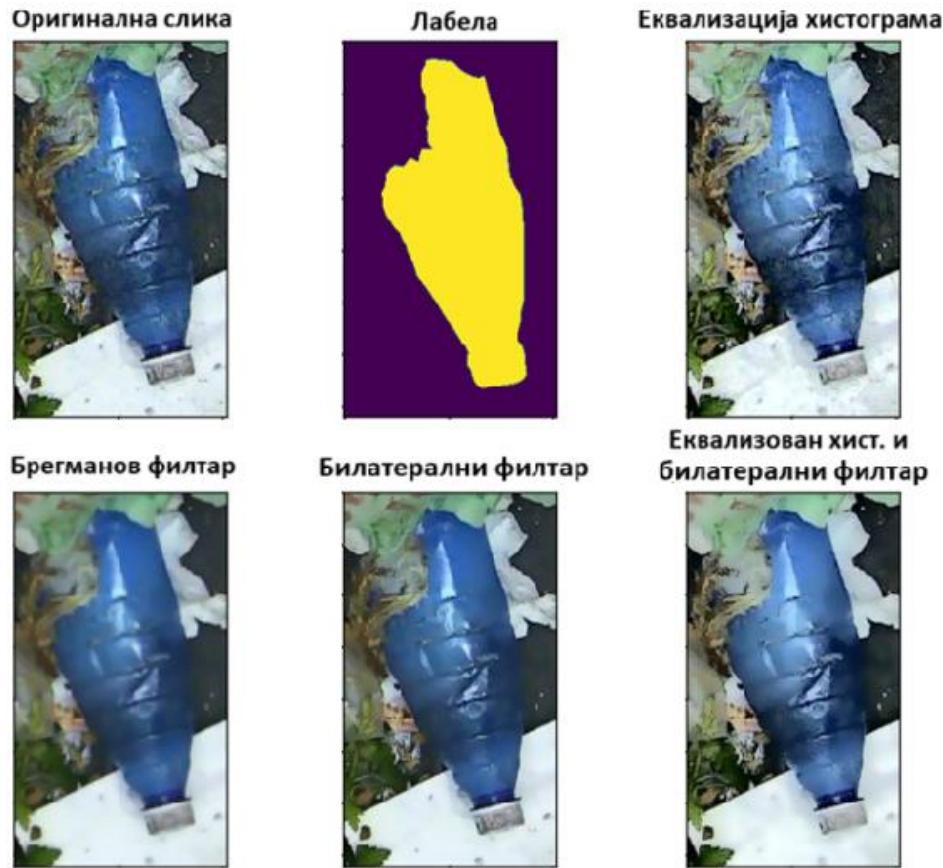
Испитане су следеће аугментације података са циљем унапређења метрика модела: еквализација хистограма, филтрирање Брегмановим и билатералним филтером и њихове комбинације. Циљ еквализације хистограма је побољшање контраста слике равномерном расподелом интензитета осветљености. На тај начин различити делови слике могу бити видљиви и могу се открити детаљи који су можда били заклоњени због лошег контраста. Иако може побољшати контраст слике, еквализација хистограма понекад може довести до губитка детаља у областима где је контраст већ добар или до појаве визуелног шума.

За уклањање шума коришћена су два филтера: Брегманов и билатерални. Брегманов филтар решава Рудин-Ошер-Фетами проблем оптимизације за налажење слике u са мањом варијансом од оригиналне слике f под условом да слике остану сличне.

$$\min \sum_{i=0}^{N-1} \left(|\Delta u_i + \frac{\lambda}{2} (f_i - u_i)^2| \right),$$

где је λ позитиван параметар који диктира ниво сличности слика који се захтева. Први члан описује тоталну варијансу, те како λ тежи 0, овај члан доминира и филтрирање је интензивније.

Билатерални филтар је одабран као нелинеарни филтар који очувава ивице и уклања шум. Овај филтер ради по принципу два Гаусовска филтра по простору и интензитету пиксела, чиме се одржава драстична промена у боји код ивица. Одабрани параметри филтра били су 15 за варијансу боја и 75 за просторну варијансу, а тако добијени резултати приказани су на слици 14.

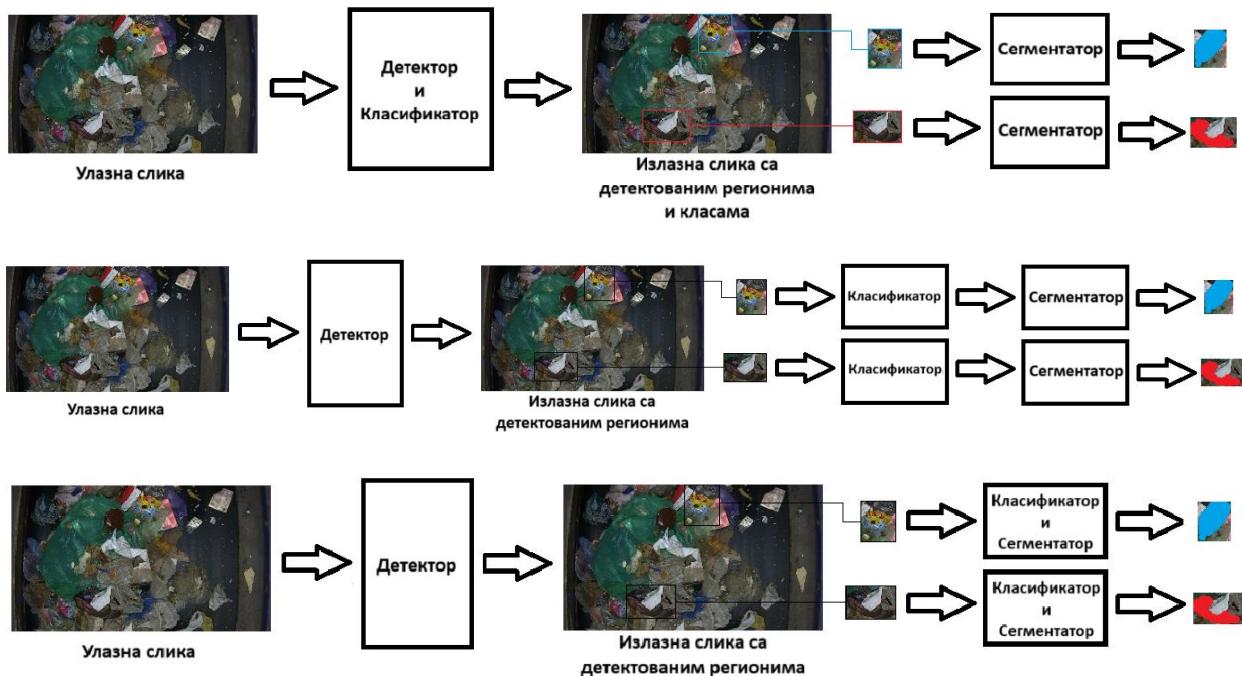


Слика 14. Различите врсте претпроцесирања података скупа за сегментацију

3 МЕТОДОЛОГИЈА РАДА

Одабран програмски језик у овом мастер раду био је *Python*. Ово је интерпретирани програмски језик високог нивоа чије бројне библиотеке омогућавају решавање великог распона проблема. *Python* омогућава једноставан и ефикасан рад са матрицама, примену алгоритама, трансформација и симулација помоћу библиотеке *pintpy*, исцртавање функција и графички приказ података и сигнала уз помоћ библиотеке *matplotlib* и обраду и квантитативну анализу слике користећи функције из библиотека *opencv* и *scipy*. Помоћу ових пакета могуће је остварити побољшање квалитета слике, смањење шума, побољшање резолуције слике ради боље разумљивости детаља, сегментацију детаља, анализу облика и текстура. Поред тога, овај програмски језик подржава брз и ефикасан рад са неуронским мрежама у *PyTorchu*-у, па се наметнуло као логичан избор за израду овог мастер рада. Већина функција садржи и добру документацију што омогућава кориснику да прегледа алгоритме, разуме изворни код и направи своје функције.

Проблем сегментације инстанци могуће је поделити на неколико једноставнијих проблема, а затим решити сваки појединачно. У овом раду предложено је неколико структура везивања модела на ред и испитана је њихова ефикасност. Први у низу од проблема који се решава јесте детекција. Сваки детектор инхерентно решава и проблем класификације, па је прва предложена архитектура сачињена из два модела, детектора и сематичког сегментатора обученог са надгледањем (слика 15а).



Слика 15. Предлози структуре сегментације инстанци рециклажног отпада

Међутим, класификација која је уграђена у детектор није нужно оптимизована када модел није сигуран, те остале структуре користе посебно трениран класификатор на WaRP-C скупу. Разлика између предлога са слике 15б и 15в лежи у сегментацији која може да буде тренирана са надгледањем или као нуспојава класификатора у зависности од расположивости ознака скупа за сегментацију. Предност сегментације без надгледања лежи у цени обележавања региона отпада која је велика, па се по цену нешто мање тачности може уштедети на времену и ресурсима.

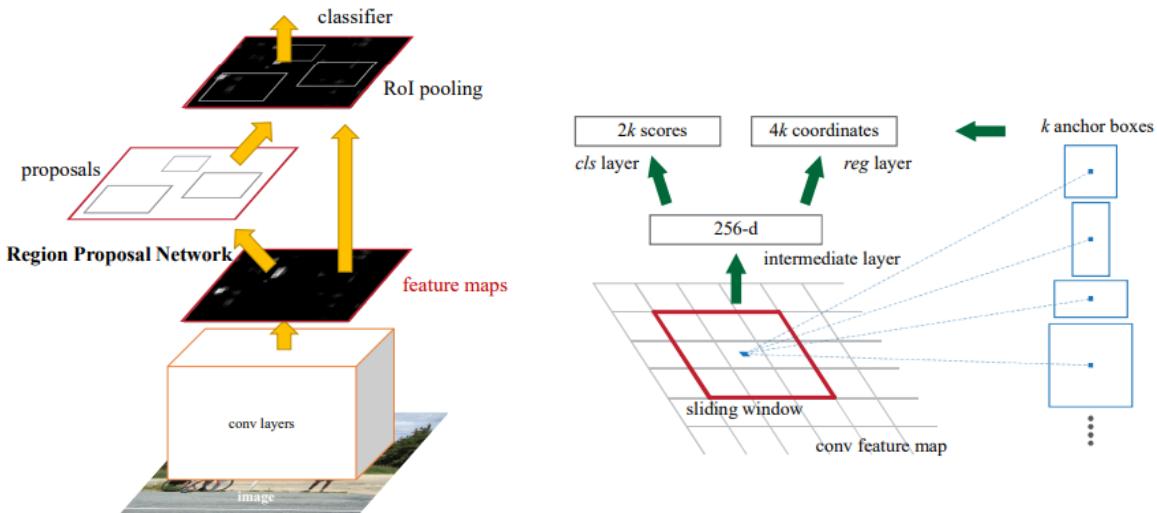
3.1 Детекција

Проблем детекције решен је помоћу две најзаступљеније архитектуре у овој области, Брзе R-CNN неуралне мреже и Yolov5 модела. Првобитно, скуп података припремљен као што је описано у другом поглављу, доведен је у шаржама на улаз Брзе R-CNN (енг. *Fast R-CNN*) мреже која је претходно тренирана на COCO скупу података. *Faster R-CNN* (*Faster Region-based Convolutional Neural Network*) је модел за детекцију објекта представљен 2015. године од стране Роса Гришика, који је унапредио постојеће R-CNN (*Region-based Convolutional Neural Network*) и SPP Net (*Spatial Pyramid Pooling Network*), пружајући значајна побољшања у брзини и тачности детекције.

Кључна карактеристика архитектуре Faster R-CNN-а лежи у томе што користи концепт предлога региона (*region proposal*) који се генеришу помоћу алгоритма попут селективне претраге. Ови региони представљају потенцијалне локације објекта на слици, који се затим обрађују путем конволуционе неуралне мреже како би се одредиле релевантне карактеристике.

Кораци по којим се може разложити рад Faster R-CNN-а су следећи:

- Одређивање одличја:** Слика се обрађују кроз претходно обучену конволуциону неуралну мрежу попут VGG16 или ResNet чиме се добијају њена одличја.
- Генерисање предлога региона:** Алгоритам за генерисање предлога региона идентификује потенцијалне области на слици које би могле да садрже тражене објекте, на основу одличја добијених у претходном кораку. Тиме говори даљим слојевима „где да гледају“.
- Region of Interest (ROI) Pooling:** Овај стадијум је кључан за убрзање процеса и омогућава да се обраде региони различитих димензија тако што се за сваки генеришу вектори одличја фиксне дужине.
- Класификација и регресија:** Класификациони слој одређује вероватноћу присуства објекта и његову класу, а регресиони прецизно локализује границу региона.



Слика 16. Архитектуре Faster R-CNN-а и Region Proposal Network-а (RPN) [9]

Тренинирање мреже подразумева истовремену оптимизацију компоненти R-CNN-а и RPN-а (енг. *Region Proposal Network*) (слика 16). У функцији губитка факторишу губици класификације и регресије истовремено [9]. Приликом инференце, често се добија велики број преклапајућих региона који се морају накнадно елиминисати неким од алгоритама.

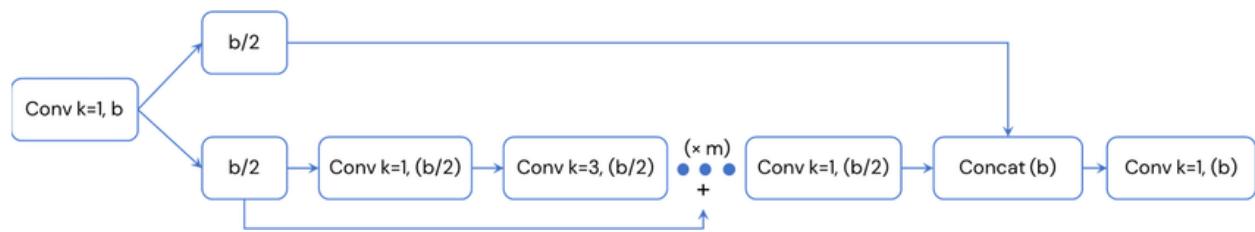
Метода уклањања преклапајућих региона примењена у овом мастер раду била је не-максимално сузбијање (енг. *non-maximum suppression, NMS*). Циљ ове врсте постпроцесирања резултата модела детекције јесте одабир једног региона од много преклапајућих који има највећу вероватноћу да садржи објекат. NMS се описује са два параметра или прага: праг сигурности коју предвиђа модел, као и праг односа пресека и уније преклапајућих региона. Идеја јесте одбацивање оних региона који имају мању сигурност и довољно велики пресек са неким од бољих региона.

Са друге стране, Yolov5 је моћни алгоритам за детекцију објеката развијен од стране *Ultralytics* тима. Одликују га велика брзина и ефикасност, различите стратегије за модификовање тренирајућег скупа, тренирање и обрачунавање губитка. Састоји се из три главна дела:

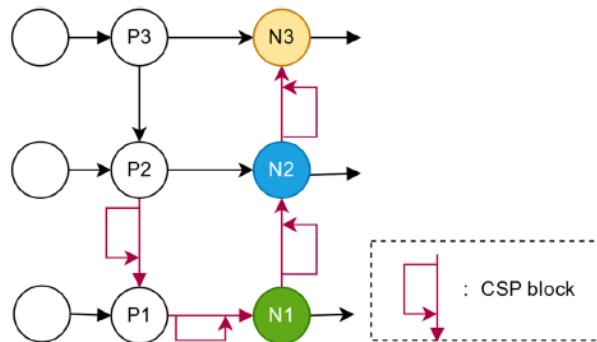
1. **Основа** (енг. *backbone*) Yolov5 модела је модификација *Darknet* архитектуре - *CSP-Darknet53* чији је задатак извлачење обележја. Упркос бољим резултатима које је имао *CSPResNext50* приликом класификације, *CSP-Darknet53* је изабран као основа зато што има више улазних и конволуционих слојева, па се показао боље приликом детекције малих објеката и вишебројних објеката различитих величина и типова на истој слици. *Darknet* се састоји из великог броја конволуционих слојева који обрађују слику на различитим нивоима компресије. Парцијални међуслој префикс (енг. *CSP-, Cross stage partial*) потиче из *CSPNet* архитектуре неуралне мреже и означава технику која смањује сложеност обучавања тако што подели улазна обележја на два дела, а затим их споји након примене одређених трансформација (слика 17).

На овај начин се повећава тачност и уједно смањује време инференције смањењем броја градијената које је потребно израчунати [11].

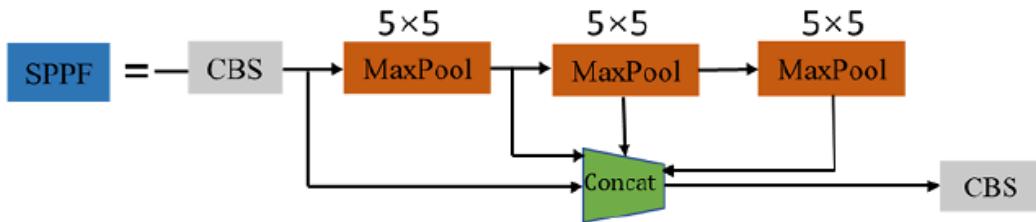
2. **Врат** (енг. *neck*), који повезује основу и главу. Имплементиран је као комбинација *SPPF* и *CSP-PAN* (*Cross stage partial PAN*) структура. *CSP-PAN* је архитектура која генерише и детаљна обележја предмета, и апстрактна семантичка обележја на основу улаза, и тако недвосмислено доприноси повећању тачности и успешности детекције малих објеката (слика 18). Улога *SPPF* архитектуре у *Yolov5* мрежи је имплементација вектора обележја константне дужине на основу улаза променљивих дужина за потребе потпуно повезаног слоја. Састоји се из три редно везана слоја сажимања величине 5×5 и два CBS (*Convolution + Batch Normalization + SiLU*) слоја чији је задатак извлачење и спајање обележја из улазних величина (слика 19) [12].
3. **Глава** (енг. *head*), чији је задатак да генерише излазне вредности. *Yolov5* се састоји из 3 главе чији су излази региони, класе и сигурности модела добијени на различитим нивоима компресије слике [13].



Слика 17. Парцијални међуслој (CSP) у *Yolov4* моделу [11]



Слика 18. Cross stage partial PAN структура у *Yolov5* моделу [12]



Слика 19. Архитектура SPPF блока унутар Yolov5 модела [13]

Једна од главних предности Yolov5 модела неуралних мрежа за детекцију је и начин на који се рачуна губитак. Функција губитка се састоји из следећих компоненти:

1. Губитак координата – рачуна се на основу разлике координата тачних и предвиђених региона,
2. Губитак класификације – бинарни губитак унакрсне ентропије за нетачно класификовани објекти унутар предвиђених региона,
3. Губитак објектности – бинарни губитак унакрсне ентропије који рачуна вероватноћу да се предмет налази унутар региона, добија се као однос пресека и уније,
4. Губитак контракције – фокусира мотивацију између истраживања и искоришћавања.

Само најбољи региони добијају координатни губитак, чији је задатак да фино подеси ивице региона, и класификациони губитак, чији је задатак да успешно одреди којој категорији детектовани објекат припада. Поред ових губитака, најбољи региони такође добијају и губитак објектности, док сви остали региони добијају само губитак објектности. Његов задатак је да научи мрежу да предвиди вероватноћу да се предмет садржи у региону, која није ништа друго до однос пресека и уније тачног и предвиђеног региона. На овај начин модел се тренира да предвиди своју сигурност, док најбољи региони такође фино подешавају своје границе и класу, што унатраг повећава пресек, односно сигурност, ка јединици. Губитак објектности је користан приликом одређивања најбољег од више сличних региона који су нуспојава архитектуре мреже (слика 20).

Губитак контракције се рачуна на основу промене позиције и величине региона између два корака. Овај губитак постоји у почетним епохама како би мотивисао истраживање, а затим се смањује током каснијих епоха. Иако мали, овај губитак никада није једнак нули и његов задатак је да полако или константно гура регион ка својој почетној позицији и облику. Идеја иза овог губитка је да се региони који нису нашли ниједан предмет врате на почетну сугестију која је дизајнирана да покрије цео простор подједнако, те су шансе да се објекат пронађе у будућности веће.



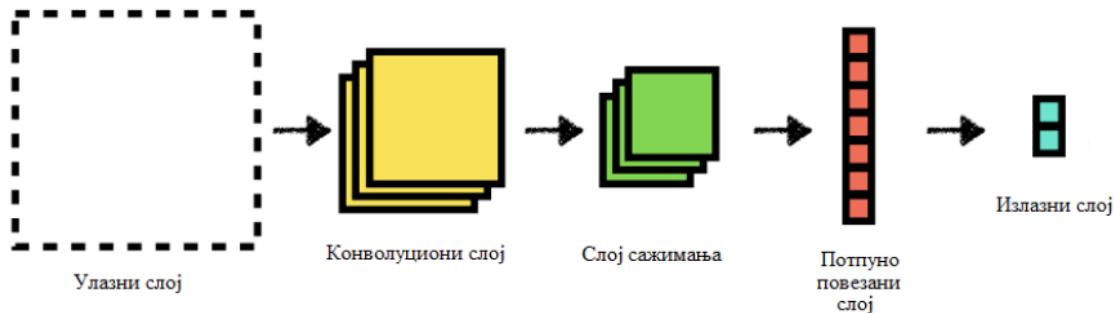
Слика 20. Преклапајући региони од којих се бира најбољи, а који настају као последица архитектуре модела

3.2 Класификација

Модел обучен за решавање класификационог проблема била је конволуциона неурална мрежа DenseNet121 (енг. *Densely Connected Convolutional Network*). Конволуционе мреже су посебан тип неуронских мрежа за процесирање неструктурираних података, а широку примену су нашле у области класификације, сегментације и детекције дигиталне слике. Погодније су за коришћење од потпуно повезаних неуронских мрежа које могу бити веома временски захтевне при решавању оваквих проблема. За разлику од класичних неуралних мрежа, конволуционе неуралне мреже уводе две нове градивне јединице: слој конволуције (енг. *convolutional layer*) и слој сажимања (енг. *pooling layer*). У случају класификације, на излазу конволуционе неуралне мреже налазе се потпуно повезани неурони између слојева и сигмоидна функција активације, док се у проблемима сегментације на излаз надовезују слојеви деконволуције и инверзног сажимања.

Конволуциону неуронску мрежу можемо приказати као низ слојева, при чему сваки слој трансформише резултат претходног слоја по одређеним правилима и производи излаз који се назива активациона мапа. Основне компоненте конволуционе неуралне мреже су: улазни слој, конволуциони слој, слој сажимања, активациона функција и потпуно повезани слој (слика 21) [14].

Моделирање конволуционе неуралне мреже најчешће се одвија у три основне фазе. У првој фази се примењују конволуције, што за резултат има активацијоне мапе. У другој фази, свака активациона мапа пролази кроз трансформацију неке од активационих функција. У трећој фази користе се функције сажимања како би се модификовала активациони слој.



Слика 21. Шематски приказ слојева конволуционе неуралне мреже [14]

Најважнији градивни елемент конволуционе неуралне мреже представља конволуциони слој који обавља већи део трансформације података. Он се састоји се из два дела. У првом делу долази до конволуције улазне активационе мапе са одређеним бројем филтера који могу препознати различите карактеристике. На улаз првог конволуционог слоја мреже се доводи вишедимензиони вектор, односно матрица пиксела у случају дигиталне обраде слике. Након тога, у другом делу конволуционог слоја, на добијени резултат делује се неком од активационих функција. Хиперпараметри који се дефинишу у сваком конволуционом слоју су број филтера d , величина филтера f , допуњавање (енг. padding) p и корак s . Активационе функције конволуционих слојева су најчешће функције рампе (енг. *ReLU*), које су се најбоље показале у пракси.

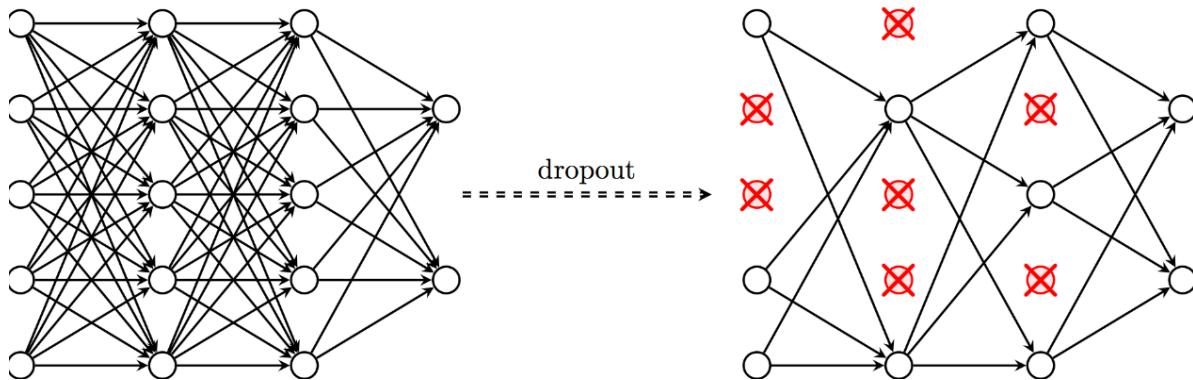
Неурони у првом конволуционом слоју нису повезани са сваким појединачним пикселом улазне слике, већ само са пикセルима који припадају његовом рецептивном пољу. У складу са претходно описаним, сваки неурон из другог конволуционог слоја је повезан само са неуронима који припадају малом региону у првом конволуционом слоју (односно неуронима који припадају рецептивним пољима неурона из другог конволуционог слоја). Мрежа са оваквом архитектуром у првом скривеном слоју обрађује једноставна обележја на слици, а касније их комбинује у сложенија обележја у наредним скривеним слојевима. Описана хијерархијска структура одговара сликама из реалног живота што је разлог зашто су се конволуционе неуралне мреже показале као веома успешан метод приликом обраде дигиталне слике.

За разлику од класичних неуралних мрежа које поседују једнодимензионе слојеве неурона, улаз неурона у првом слоју конволуционе мреже представља изравнату матрицу пиксела слике. Код конволуционих неуралних мрежа, сваки слој представља матрицу неурона који су „стимулисани“ пикセルима из одговарајућег рецептивног региона неурона у претходном слоју, односно пикселима улазне слике у случају првог конволуционог слоја. Сваки конволуциони слој се састоји од више мапа обележја истих димензија, те се они могу посматрати као тродимензионална структура.

Сви неурони једне мапе обележја имају исте вредности параметара тежина и помераја (енг. *bias*). Рецептивно поље неурона се сада проширује дуж свих мапа обележја, односно у

конволуционом слоју долази до примене више филтара на улазну слику, чиме се обезбеђује детекција одређених обележја било где на улазној слици. Чињеница да сви неурони који су везани за једну мапу обележја поседују исте параметре значајно редукује број параметара модела. За разлику од класичне неуралне мреже која научи да препознаје одређену структуру на једном делу слике, надаље може структуру препознати само на тој позицији, конволуционе неуралне мреже поседују могућност препознавања научених облика било где на улазној слици [15].

Слојеви за одбацивање постављају се најчешће иза слојева за сажимање са циљем заштите од преобучавања и могу се комбиновати заједно са другим техникама регуларизације (енг. *weight decay, batch normalization*). Механизам рада заснива се на деактивацији одређеног броја неурона током обучавања мреже чиме се постиже побољшање генерализације мреже јер се подстиче да различити неурони “науче” да препознају иста обележја (слика 22).



Слика 22. Механизам одбацивања у неуралној мрежи

Нормализована експоненцијална функција (енг. *softmax*) је функција која улаз од K реалних бројева нормализује у дистрибуцију вероватноћа од K вредности. Пре процеса нормализације неке од вредности на излазу неурона могу бити негативне или веће од 1, али након примене *softmax* функције вредности свих излаза биће у интервалу (0,1) и у збиру једнаки 1, те се они могу интерпретирати као вероватноће. Ова функција користи се у неуралним мрежама у циљу мапирања ненормализованих излаза мрежа у вероватноће за предикцију сваке класе.

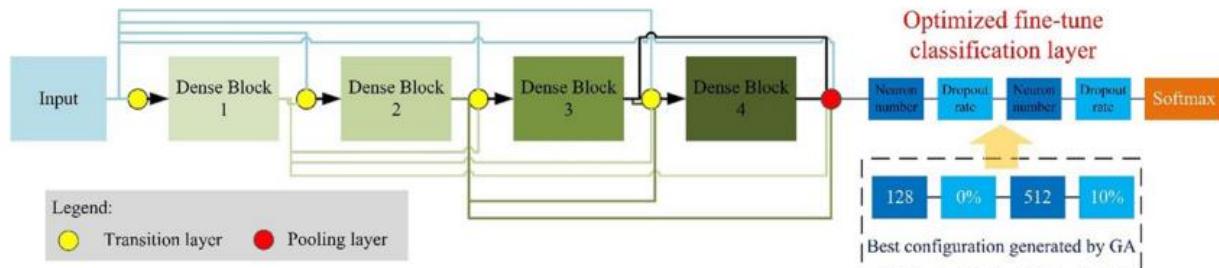
DenseNet је архитектура неуралне мреже која користи концепт густих конектора: сваки слој на улаз добија информацију из свих претходних слојева. Број 121 се односи на варијанту која има укупно 121 слојева, укључујући конволуционе, нормализационе и остale слојеве. Оваква мрежа је ефикасна и смањује утицај нестајања градијената (енг. *vanishing gradient problem*) јер се параметри размењују између почетних и каснијих слојева (слика 23).

Основни градивни блокови DenseNet неуралне мреже су:

- Густи блок (Dense Block):** Састоје се од неколико густо повезаних слојева који на улаз добијају информацију из свих претходних блокова, што омогућава поново коришћење

информација и градијената. Ова иновација се показала да за исти број параметара даје боље резултате него мрежа без ње

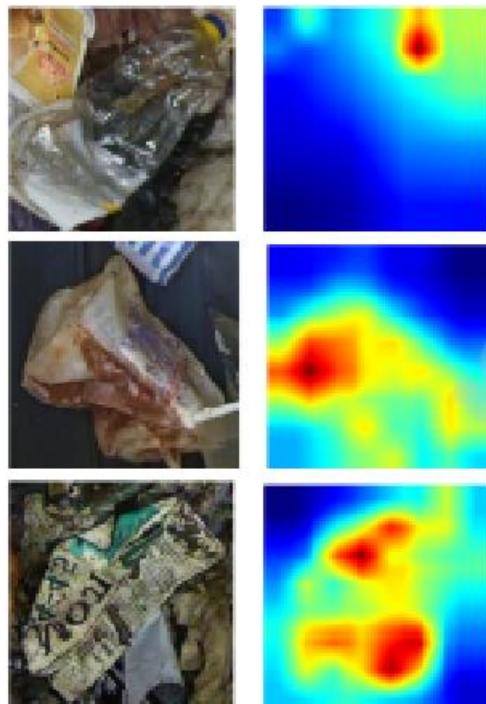
2. **Bottleneck слој:** Унутар сваког густог блока постоје *bottleneck* слојеви који имају 1x1, а затим и 3x3 конволуциони слој. Слој 1x1 служи да смањи дубину (број канала), а 3x3 да ухвати сложене правилности.
3. **Прелазни слој:** Налази се између густих слојева и служи да смањи просторну димензију (висину и ширину) слика и мапа особина. То значи да постоји 1x1 конволуциони и 2x2 *average pooling* слој који компресују обележја тако да се њима може лакше манипулисати.
4. **Global Average Pooling (GAP) слој:** Операција у којој се рачуна просек сваке мапе обележја и тиме добија један број за сваки канал. Ефективно, просторна димензија се смањује на 1x1 и на тај начин се добијају коначна „поравната“ одличја која се доводе на улаз класификатора.
5. **Потпуно повезани слој:** Потпуно повезани слој са softmax активацијом на самом крају има улогу класификатора над издвојеним карактеристикама слике.



Слика 23. Архитектура DenseNet неуралне мреже [16]

Осим верзије од 121 слојева, у раду је разматрана и 169 варијанта DenseNet архитектуре. Она је описана са (6, 12, 32, 32) блоком уместо (6, 12, 24, 16), али принцип рада остаје исти. Број параметара који се тренирају је кључни фактор са аспекта хардверских ресурса али и могућности преобучавања и брзине закључивања, како би модел могао да се извршава у реалном времену. Имплементација DenseNet неуронске мреже дата је у прилогу Б [17].

Поред класификације, могућност коју сваки модел конволуционих неуралних мрежа пружа јесте техника звана Grad-CAM (енг. *Gradient-weighted Class Activation Mapping*). Овај приступ користи градијенте у последњем слоју неуралне мреже да утврди који делови слике су највише допринели при доношењу класификационе одлуке. Већи коефицијенти одговарају тачно оним местима на слици која су највише допринела при одлуци, те се они могу искористити за добијање семантичке сегментације слике без претходног обучавања на обележеном скупу података у ту сврху [18]. Једана такав пример дат је на слици 24.



Слика 24. Приказ градијената класификатора (Grad CAM) као алтернатива сегментацији са надгледањем [2]

3.3 Семантичка сегментација

Не постоје универзална решења за проблем сегментације, а методе које се користе углавном зависе од конкретне примене. У оквиру дигиталне обраде слика развијено је неколико сегментационих метода, од којих су најпознатије следеће: 1) амплитудни праг (енг. *thresholding*), 2) сегментација на основу текстуре, 3) подударање са шаблоном и 4) сегментација на основу ширења региона (енг. *region growing*). Ови типови алгоритама се базирају на сепарацији карактеристичних делова користећи текстуру, поређење са прагом (енг. *thresholding*), или поређење са шаблоном. Код алгоритама који се заснивају на обради слике на бази пиксела, сваки пиксел у слици сврстава се у једну од n категорија на основу интензитета, локалног градијента и боје. У најпростијем случају, када имамо само 2 класе, дефинишемо праг интензитета T_g тако да важи:

$$(x,y) \in c_0, \text{ ако је } g(x,y) < T_g$$

$$(x,y) \in c_1, \text{ ако је } g(x,y) \geq T_g$$

при чему се исти праг примењује на целој слици, тј. $\forall(x,y)$. Основни проблем код овог приступа јесте присуство шума и неједнозначност мерења тј. могућност да се исти интензитет g појављује у различитим сегментима. Дакле, циљ је налажење оптималног прага класификације.

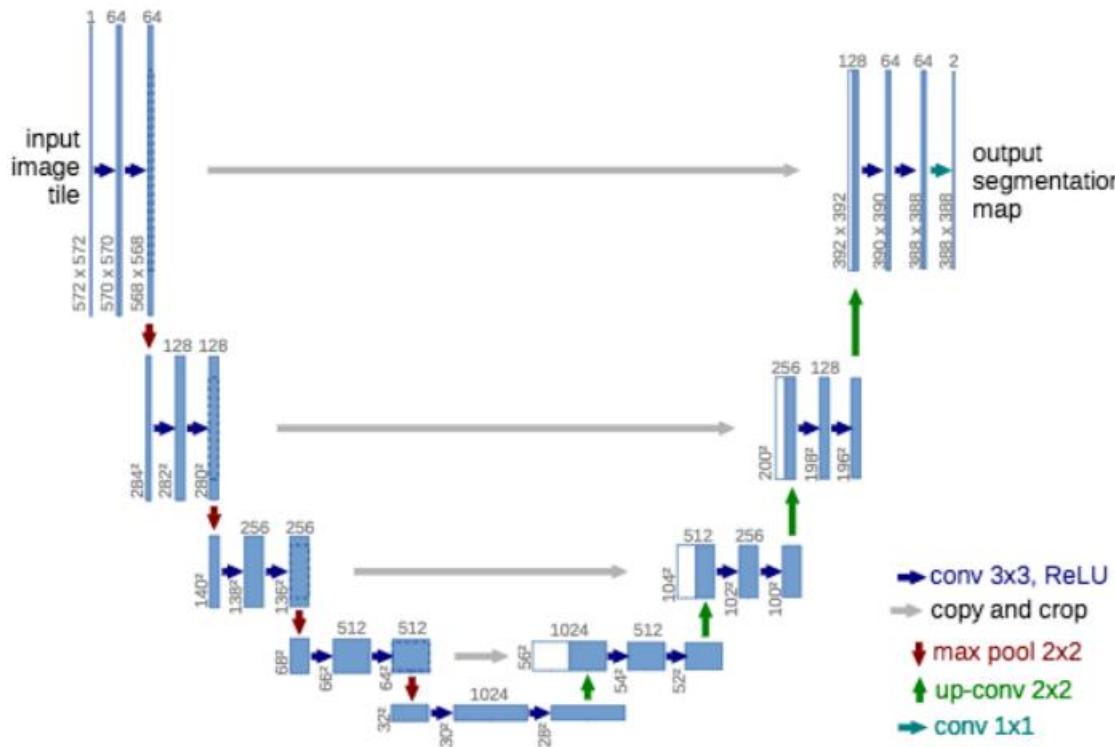
То је могуће учинити помоћу неке од аутоматских метода: анализом облика хистограма, одређивањем параметрисаних расподела, анализом ивичних пиксела, ентропијским методама (поређењем локалне сличности) и просторним методама.

Први корак је углавном побољшање контраста, јер већина слика има ниску резолуцију. Они укључују модификације хистограма, филтерима средњих вредности или Гаусовим филтером. Следећи корак је разлагање слике на сумњиве, односно потенцијалне зоне региона од интереса, и коначно, сегментација дела слике. Приступи за сегментацију се према начину рада могу поделити на статистичке и структурне технике. Статистичке методе означавају пикселе према вероватноћи припадања одређеној категорији зависно од интензитета пиксела, док су структурне технике засноване на просторним особинама слике.

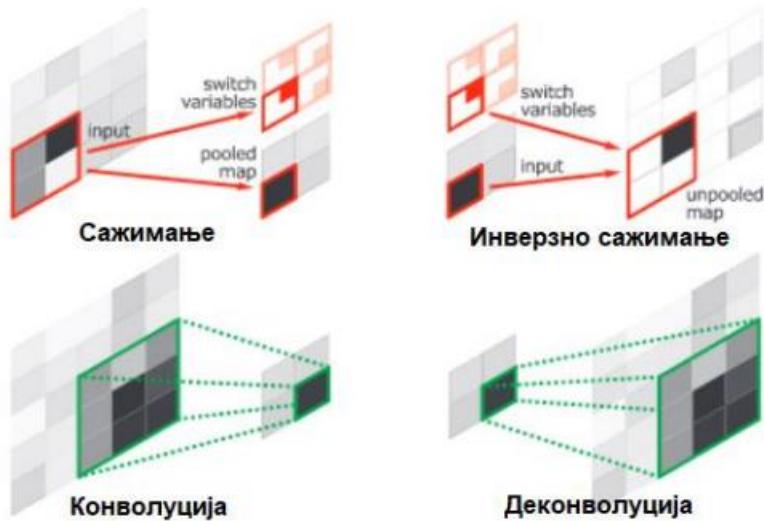
Иако често даје задовољавајуће резултате, сегментација само на основу прага није довољна за постизање високе тачности која се тражи када су у питању индустријски захтеви, па се често примењује неки од алгоритама машинског учења. Дубоко машинско учење је у претходној деценији успешно решило неколико проблема компјутерске визије различитих нивоа сложености, од класификације слика, детекције објекта, па све до семантичке сегментације и сегментације инстанци.

Дубока неурална мрежа коришћена у овом раду за сематичку сегментацију рециклажног отпада била је U-Net конволуциона неурална мрежа. Принцип рада ове неуралне мреже заснива се на две основне целине: прва целина обухвата сажимање и компресију карактеристичних облика применом слојева конволуције и сажимања, док се у дугом делу мреже поново шири применом инверзних операција. Слојеви обе гране на истој дубини су међусобно повезани како би се избегло губљење примарних одличја добијених конволуцијом. Улазни сигнал ове мреже је сама слика, док се на излазу налази матрица истих димензија која садржи вероватноће припадности пиксела региону од интереса, што ову неуронску мрежу чини *end-to-end* решењем аутоматске сегментације облика [19]. Шематски приказ архитектуре неуронске мреже дат је на слици 25, док је њена имплементација приложена у прилогу В [20].

Приликом пројектовања конволуционах неуралних мрежа постоји тежња да се димензије филтара смањују са уласком у дубље нивое. Ово се постиже техникама сажимања које подразумевају да се репрезентација филтра издели на делове (најчешће димензија 2×2) и изабере медијана или максимална вредност унутар једног дела која ће представљати вредност пиксела у редукованом филтру. На тај начин задржавамо информацију о томе да ли је неко обележје препознато у посматраном региону уз губитак информације о тачном подрегиону у ком је обележје препознато. Слојеви деконволуције и инверзног сажимања представљају инверзне операције конволуцији и сажимању и кључне су компоненте неуралних мрежа за сегментацију региона. Деконволуција се може посматрати као транспонована конволуција, или конволуција слике са допуњеним вредностима. Инверзно сажимање подразумева проширивање слике попуњавањем непознатих вредности нулама. Принцип рада конволуције, сажимања и њихових инверзних операција приказан је на слици 26 [21].



Слика 25. Шематски приказ архитектуре U-Net конволуционе неуралне мреже [19]



Слика 26. Принцип рада слојева у конволуционој неуралној мрежи која врши сегментацију [21]

Вредности пиксела улазне слике скалирани су на опсег реалних бројева од -0.5 до 0.5. Коришћена је активациона функција ReLu између конволуционих слојева, док су димензије и број конволуционих слојева у левој грани мреже исти као приликом пројектовања класификатора. Излаз овакве неуралне мреже је матрица димензија $224 \times 224 \times 2$ која се надовезује на softmax слој чији резултат даје вероватноће припадности сваког пиксела сегментованом региону.

За рачунање функције грешке коришћена је унакрсна ентропија свих пиксела, а за оптимизацију коришћен је RMSProp оптимизатор. Као мера оцене прецизности сегментације у односу на обележене слике од стране стручњака коришћен је Дајсов коефицијент (енг. *Dice coefficient*), који се рачуна на следећи начин:

$$A = \frac{2|x \cap y|}{|x| + |y|}$$

где x представља скуп сегментованих пиксела нашег модела, а y су селектовани пикели од стране стручњака, при чему је ова функција додефинисана јединицом када су и x и y празни скупови. Може се приметити да се овако дефинисана прецизност разликује од функције грешке коју смо оптимизовали приликом тренирања, што је оправдано из разлога што је Дајсов коефицијент величина која није осетљива на мале промене излаза, поготово у екстремима, чинећи је лошим параметром оптимизације током тренирања једне неуралне мреже. Такође, постоји могућност аугментовања функције грешке додавањем сабирка 1-Dice, али то се показало као неефикасно у нашем случају, јер доводи до конвергенције модела који обележава готово читаву слику као објекат од интереса, што може резултовати великим вредношћу коефицијента, али тада модел губи своју поенту.

4 Резултати

У овом раду испитане су две архитектуре неуралних мрежа за детекцију и по један модел за класификацију и семантичку сегментацију рециклажног отпада са покретних трака. Више хиљада слика су процесиране и подељене у скупове за тренирање, валидацију и тестирање модела. Резултати су представљени кроз следећа четири поглавља и обухватали су подешавање параметара, аугментацију података, заштиту од преобучавања и финално, евалуацију на невиђеном тестирајућем скупу као коначном естимацијом ефикасности модела. Разматрана је брзина извршавања инференције модела и дискутован је услов за њихову употребу у продукцији.

На крају, одабрани модели су повезани на ред у виду једне скрипте која на улазу прима фотографију мешовитог отпада из птичије перспективе, а на излазу садржи исту фотографију са сегментованим и класификованим регионима од интереса, који представљају рециклажни отпад који је потребно издвојити. Ова скрипта је покривена тестовима као вид заштите од грешака у будућим изменама, измерене су њене перформансе и дискутована је примена у реалном времену.

Сваки корак тренирања и евалуације модела су извршени на рачунару са следећом конфигурацијом, притом се водило рачуна да нису покренути други програми који би утицали на резултате:

1. Процесор – 12th Gen Intel Core i7-12700H, 2300MHz
2. Графичка карта - NVIDIA GeForce RTX 3050Ti Laptop 8G
3. RAM меморија – Samsung 16GB 3200MHz DDR4
4. Матична плоча - HP 8A4F-37.55

За имплементацију модела коришћен је програмски језик *Python* и библиотека за машинско учење *PyTorch*. За визуелизација графика зависности губитака и тачности током тренирања претежно су коришћени алати *Tensorboard* и *Excel*, док је визуелизација слика постигнута помоћу библиотеке *Matplotlib* у *Python*-у. Тест случајеви написани су помоћу библиотеке *pytest*.

4.1 Резултати детекције

Први одабрани модел детекције рециклажног отпада био је Faster R-CNN описан у претходном поглављу, који је првобитно трениран на COCO V1 јавном скупу података за детекцију. Затим је тако оптимизован модел дотрениран кроз шест епоха на скупу података из овог рада, који је обухватао 1962 примера подељених у мини шарже величине 4. Величина мини шарже била је ограничена RAM меморијом GPU уређаја који је поред слика морао да остави простор за 40 милиона параметара модела за обучавање. Испитане слике су скалиране на димензије 256x144 као и 480x270 тако да је одржан однос висине и дужине оригиналних слика.

Коришћен је стохастички градијентни спуст са моментом као оптимизатор, чија је почетна стопа учења била 10^{-4} а моменат 0.9. Стопа учења је адаптирана током тренирања по принципу косинусног каљења:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right) \right)$$

Функције губитка током неколико епоха тренирања дате су на слици 27. Посматране су две врсте губитка – губитак објектности који кажњава процену вероватноће да одабрани регион садржи објекат модела на основу количника пресека и уније детектованих и истинитих објеката, и губитак регресије предлога региона који мери квалитет ивица детектованих региона у поређењу са истинитим.

График зависности губитка на тренирајућем скупу од броја обрађених шаржи

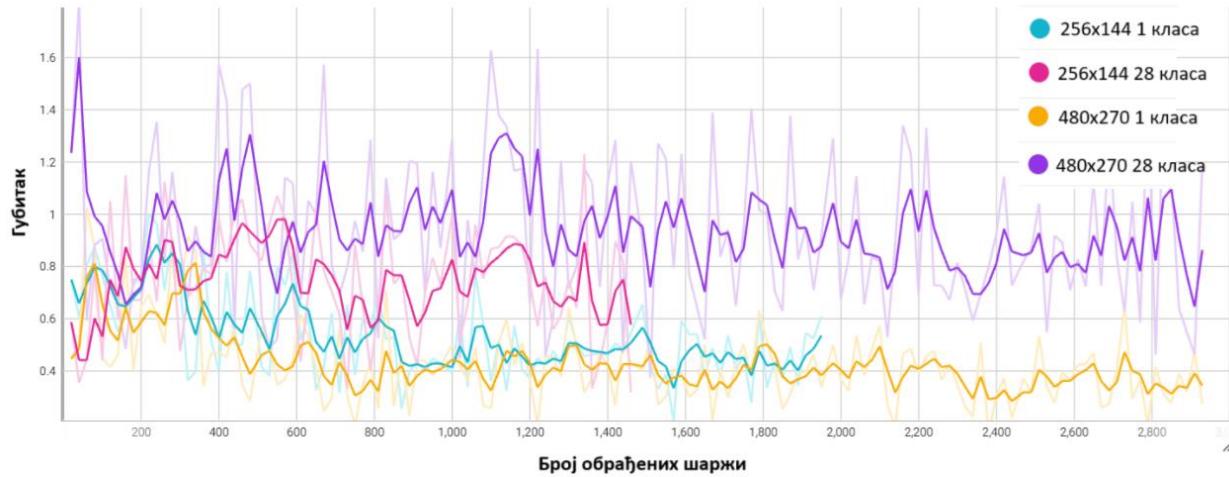
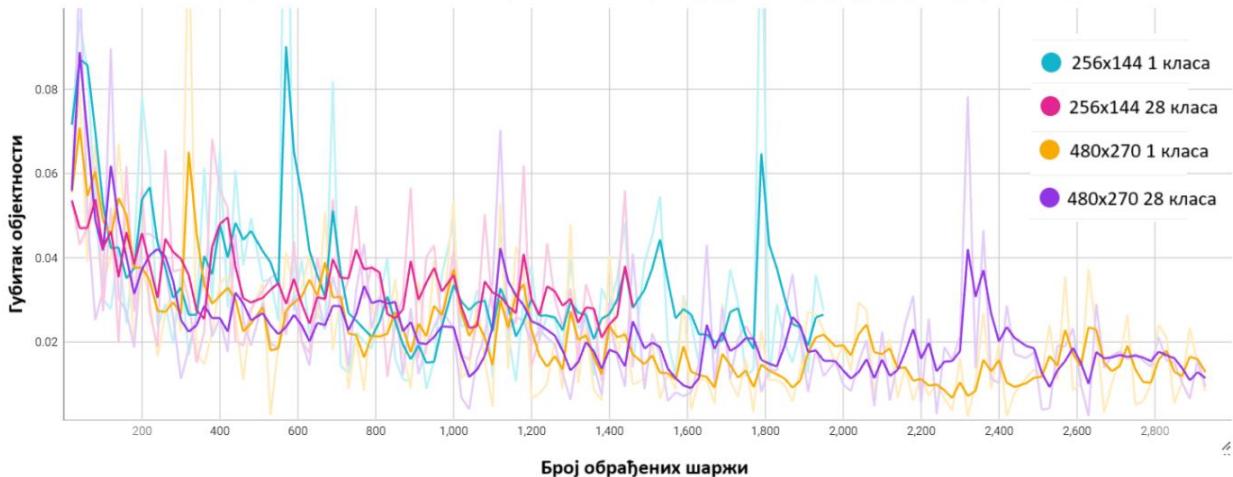


График зависности губитка објектности на тренирајућем скупу од броја обрађених шаржи





Слика 27. Графици зависности губитка од броја процесираних примера на тренирајућем скупу

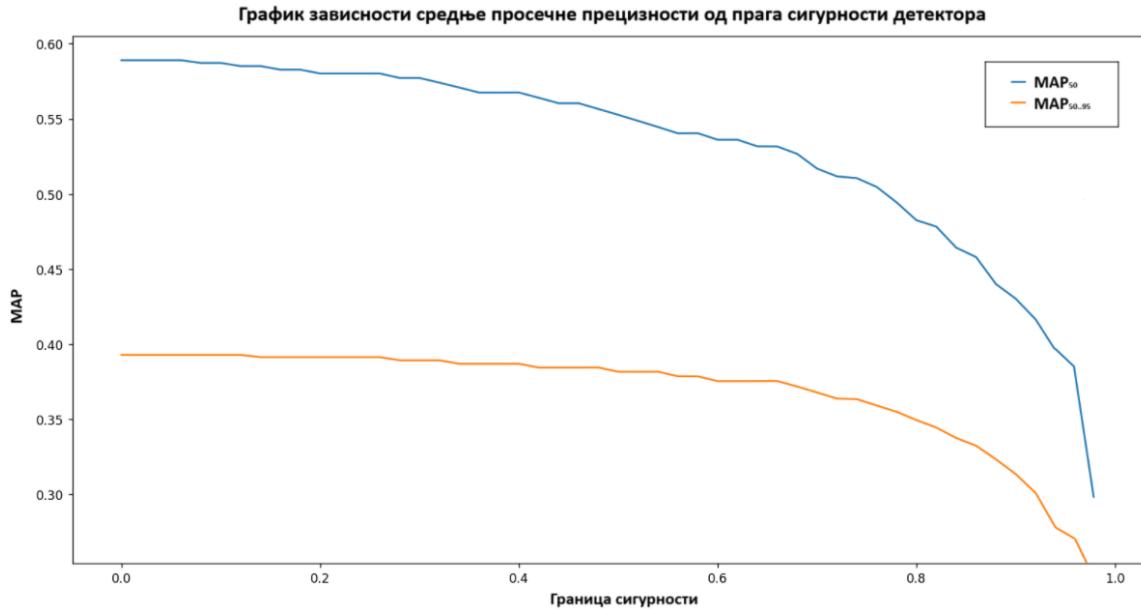
Испитана су по два модела за обе величине улазне фотографије, један који врши и класификацију и један који се бави само постојањем објекта било које категорије. На валидационом скупу модели над slikama веће резолуције су се знатно боље показали, а између модела са и без класификације, модели који су имали само задатак детектовања објекта су очекивано дали већу тачност. Највећа постигнута средња тачност износила је 61% за модел детектора над slikama димензија 480x270 на валидационом скупу. Детаљнији преглед метрика и поређења модела дат је у табели 3 испод.

Метрика коришћена за евалуацију модела била је средња просечна прецизност на бази преклапања региона од 50% (mAP_{50}) и средња просечна прецизност за преклапања између 50% и 95%, са кораком од 5% ($mAP_{50..95}$). Преклапање се рачуна по принципу пресека и уније као $\frac{2|xy|}{|x|+|y|}$, где апсолутне заграде представљају површину, а x и y истинити и детектовани регион, редом. Добијени резултати на валидационом скупу за модел над slikama димензија 480x270 и једном излазном класом били су $mAP_{50}=61.0\%$, $mAP_{50..95}=40.0\%$, док су најбољи резултати за више класа износили $mAP_{50}=37.9\%$, $mAP_{50..95}=16.7\%$. Време извршавања модела приликом евалуације било је у просеку 120ms за сваки модел независно од његове величине.

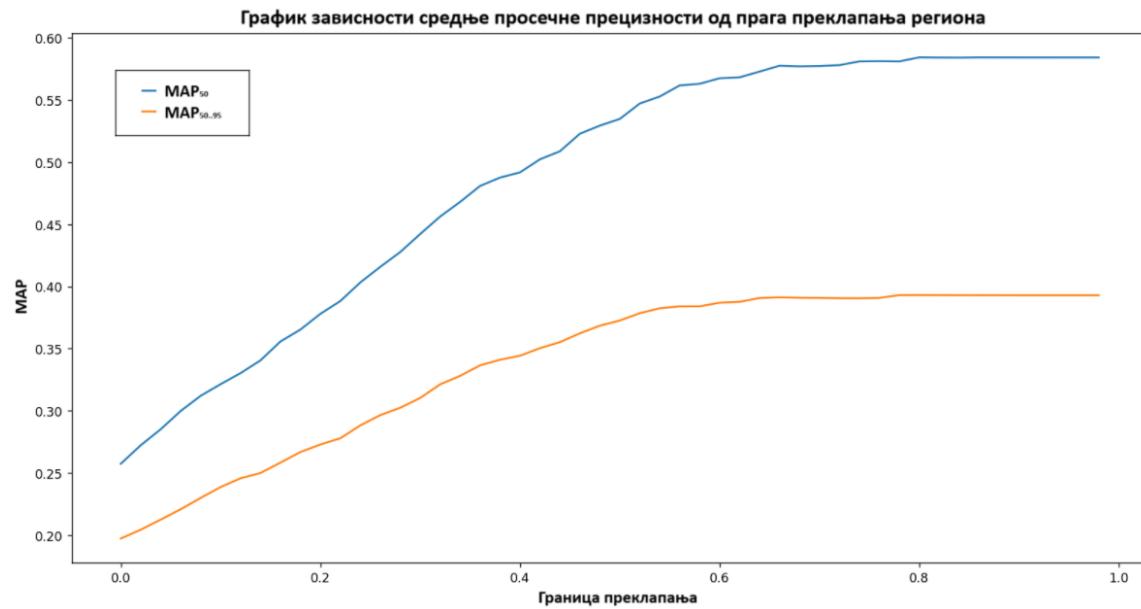
Модел	Број параметара	MAP_{50}	$MAP_{50..95}$
Fast-RCNN 256x144 1 класа	41.2M	34.7%	17.4%
Fast-RCNN 256x144 28 класа	41.3M	28.1%	11.8%
Fast-RCNN 480x270 1 класа	41.3M	61.0%	40.0%
Fast-RCNN 480x270 28 класа	41.4M	37.9%	16.7%

Табела 3. Резултати Faster R-CNN модела детекције на валидационом скупу

Резултат детекције садржао је велики број нежељених објеката са високим нивоом преклапања, те је било потребно уклонити дупликате. Приступ у овом раду било је немаксимално сажимање по нивоу сигурности модела и преклапања региона већим од неког прага. Посматране су метрике mAP_{50} и $mAP_{50.95}$ за различите вредности ових прагова. Како нам ове метрике не дају увид у лажне аларме, она има особине одзива, те је на основу графика са слика 28 и 29 потребно утврдити границе тако да се одстрани што већи број преклапајућих региона, а да се одзив не смањи знатно. Са графика 28 видимо да је потребно максимизовати праг сигурности, а са слике 29, минимизовати границу преклапања.



Слика 28. Зависност средње просечне прецизности од прага сигурности детектора

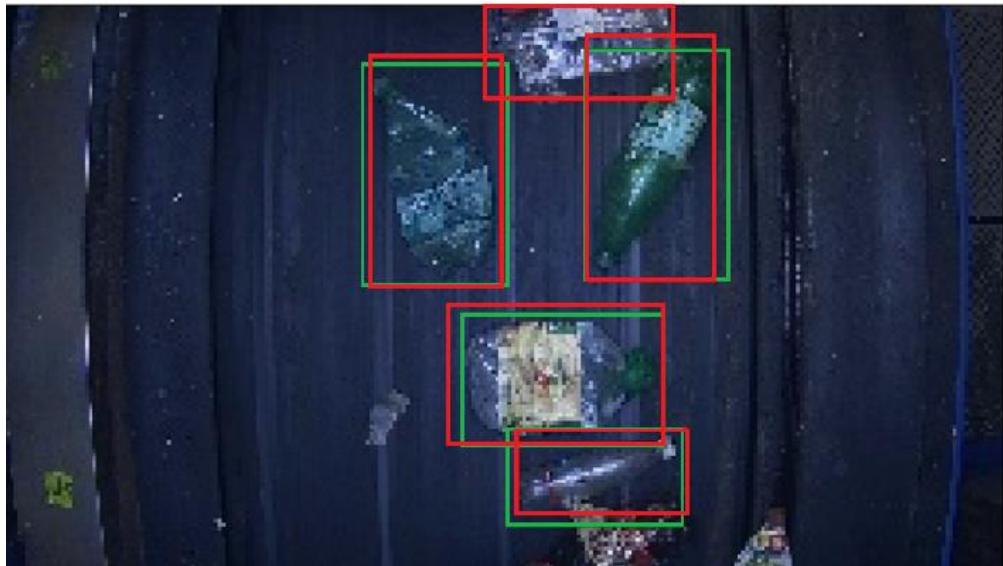


Слика 29. Зависност средње просечне прецизности од прага преклапања региона

Резултати у табели 3 и визуелни примери на слици 30. испод, дати су на основу резултата модела са прагом сигурности од 0.38 и прагом преклапања 0.6, где су истинити објекти обележени зеленом, а предикције црвеном бојом.



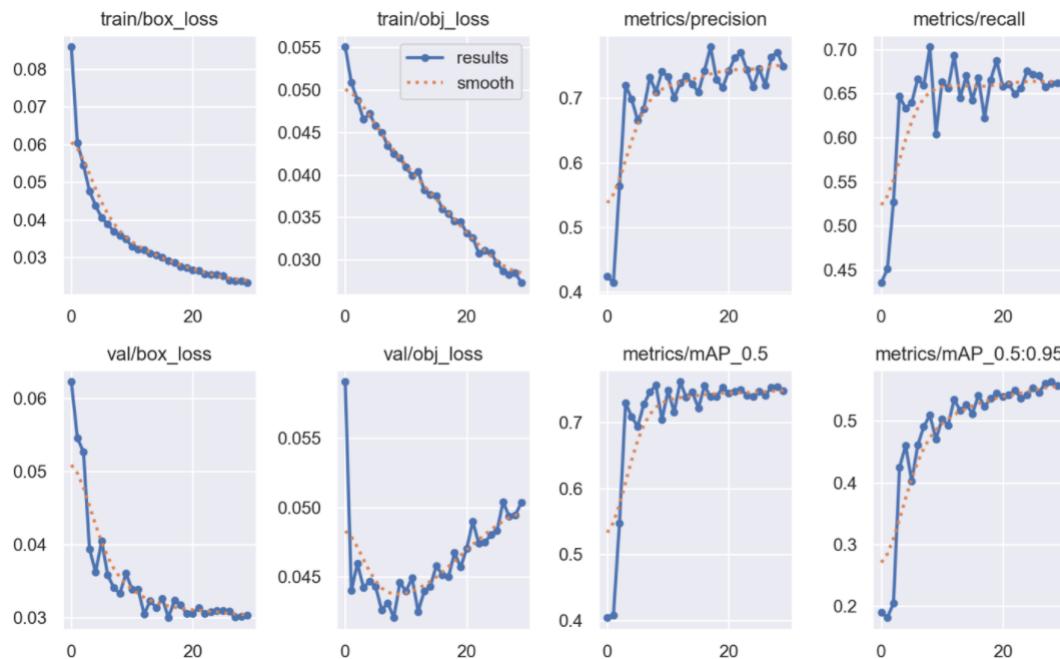
Слика 30. Примери резултата детекције Faster R-CNN мреже различитих нивоа успешности



Са слика 30. се може видети да, упркос неколико очигледних пропуста модела, постоји одређени квалитет у детекцији ивица субјективно и боље од детекције стручњака. Лажни аларми често су подразумевали пластичне кесе које одскачу од позадине или личе на пластичне флаше – проблем који се може решити пројектовањем класификатора који је научен да их препозна и сврста у посебну класу.

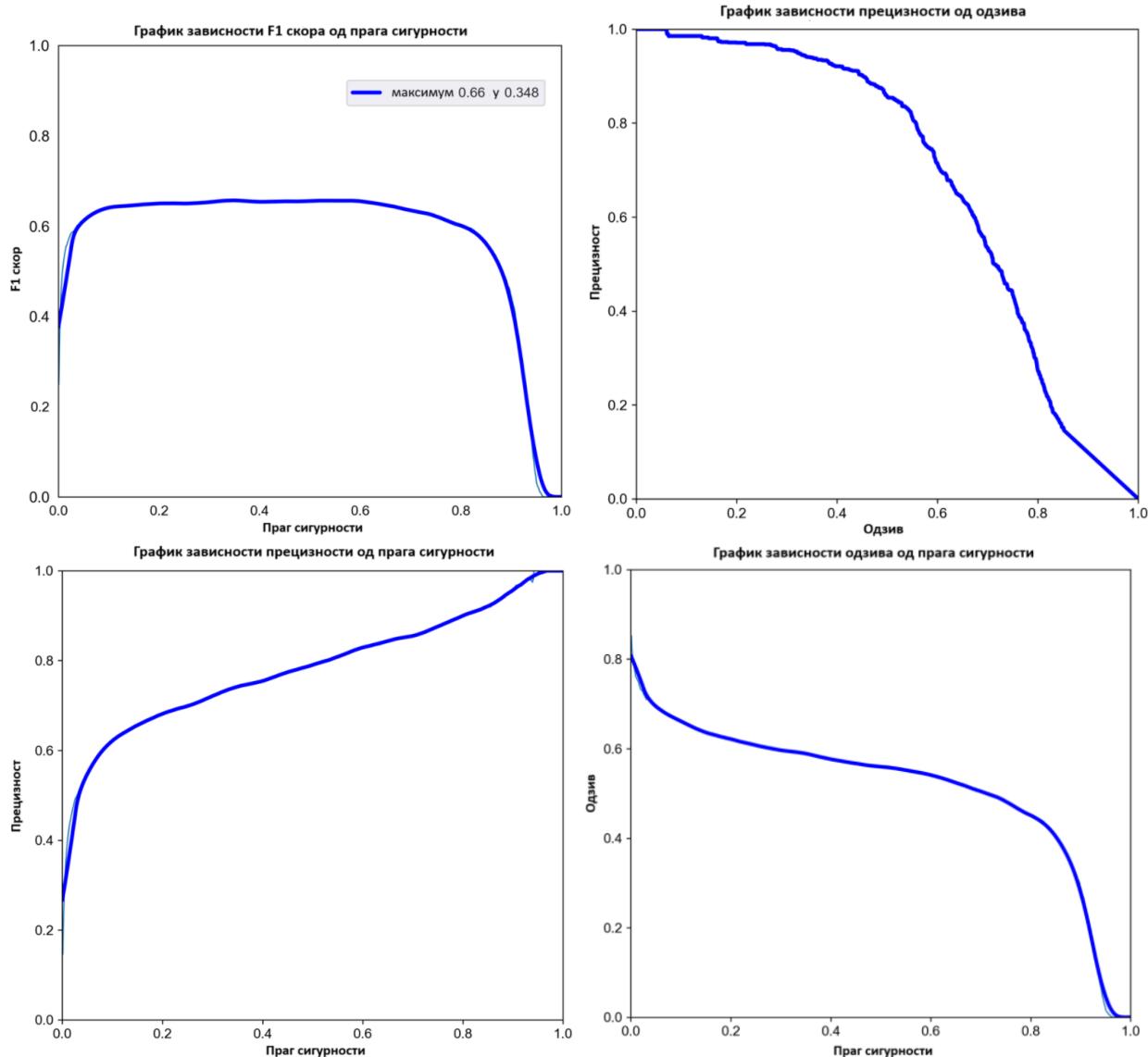
Главни недостатак овог модела јесте његова обимност, 41 милиона параметара које је потребно оптимизовати кроз три стадијума детекције. Тренирање сваки епохе трајало је неколико сати па је тиме и ограничен број експеримената попут различитих аугментација података, предобраде обраде слике и хиперпараметара. Еквализација хистограма и уклањање шума нису позитивно утицале на метрике на основу неколико почетних епоха, па су ти експерименти изостављени из резултата овог рада.

Оно што се показало као најзначајнији допринос при решавању проблема преобучавања била је аугментација слика у виду мозаика, који Yolov5 модел омогућава на једноставан начин. Овај модел, претходно трениран на COCO скупу даје могућност одабира величине модела у виду S, M, L и X величина у растућем поретку. Како детекцију изводи у једном пролазу, Yolov5 је знатно оптимизованији него његов претходник. Овакав приступом дао је већу могућност при експериментисању са хиперпараметрима и границама одлуке. Осим тога, модел је трениран на већем броју епоха за делић времена које је било потребно Faster R-CNN моделу. У овом раду разматрани су модели мале и средње величине који су имали 157 односно 212 слојева. Графици зависности губитака објектности и инстанци, као и прецизност, одзив и mAP метрике током тренирања дати су на слици 31. Слике су имале величину 480x270, а број излазних класа је вариран између једне и свих 28 класа.



Слика 31. Графици губитка Yolov5 модела у зависности од броја епоха током тренирања

Слично као и код Faster R-CNN модела, потребно је утврдити праг сигурности модела испод којег се објекти одбацују. На слици 32 можемо видети графике зависности прецизности, одзива и F1 скора у зависности од одабраног прага на валидационом скупу. Максимални F1 скор за модел среће величине се добија када је изабрани праг 0.348. У наставку, резултати су приказани баш за ову вредност прага, како у табели 4, тако и при визуелној репрезентацији.



Слика 32. Графици зависности прецизности, одзива и F1 скора од прага сигурности

Најбољи резултат постигнут је са моделом средње величине и једном излазном класом са преко 75% средње просечне прецизности за 50% преклапања на валидационом скупу и 68.4% на тестирајућем скупу. Резултати поређења свих модела на валидационом скупу дати су у табели 4, а на тестирајућем у табели 5.

Модел	Број параметара	MAP50	MAP50..95
Yolov5 medium 480x270 1 класа	20.8M	75.4%	56.3%
Yolov5 medium 480x270 28 класа	20.9M	73.4%	53.3%
Yolov5 small 480x270 1 класа	7.01M	75.0%	53.6%
Yolov5 small 480x270 28 класа	7.08M	70.1%	47.3%

Табела 4. Резултати Yolov5 модела на валидационом скупу

Резултате свих модела детекције на тестирајућем скупу могу се видети у табели 5. Постаје јасно да је Yolov5 неурална мрежа надмашила свог конкурента у сваком могућем погледу. Средња просечна прецизност MAP50 најбољег Faster R-CNN модела износила је 46.2%, док је најбољи Yolov5 имао 68.4% прецизности за свега 8 милисекунди извршавања по одбируку. Јасно је да у овом случају нема дилеме и да је Yolov5 модел коришћен као први елемент имплементације аутоматске сегментације инстанци рециклажног отпада од почетка до краја. Неколико примера детекције победничког Yolov5 модела са обележеним регионима од стране стручњака приказани су на слици 33.

Модел	MAP50	MAP50..95	Време извршавања по слици
Fast-RCNN 256x144 1 класа	28.9%	10.9%	126ms
Fast-RCNN 256x144 28 класа	22.7%	5.5%	117ms
Fast-RCNN 480x270 1 класа	46.2%	30.0%	121ms
Fast-RCNN 480x270 28 класа	31.8%	9.9%	126ms
Yolov5 medium 480x270 1 класа	68.4%	50.3%	8ms
Yolov5 medium 480x270 28 класа	43.5%	32.3%	9ms
Yolov5 small 480x270 1 класа	67.8%	47.0%	5ms
Yolov5 small 480x270 28 класа	63.9%	42.3%	6ms

Табела 5. Резултати модела детекције на тестирајућем скупу

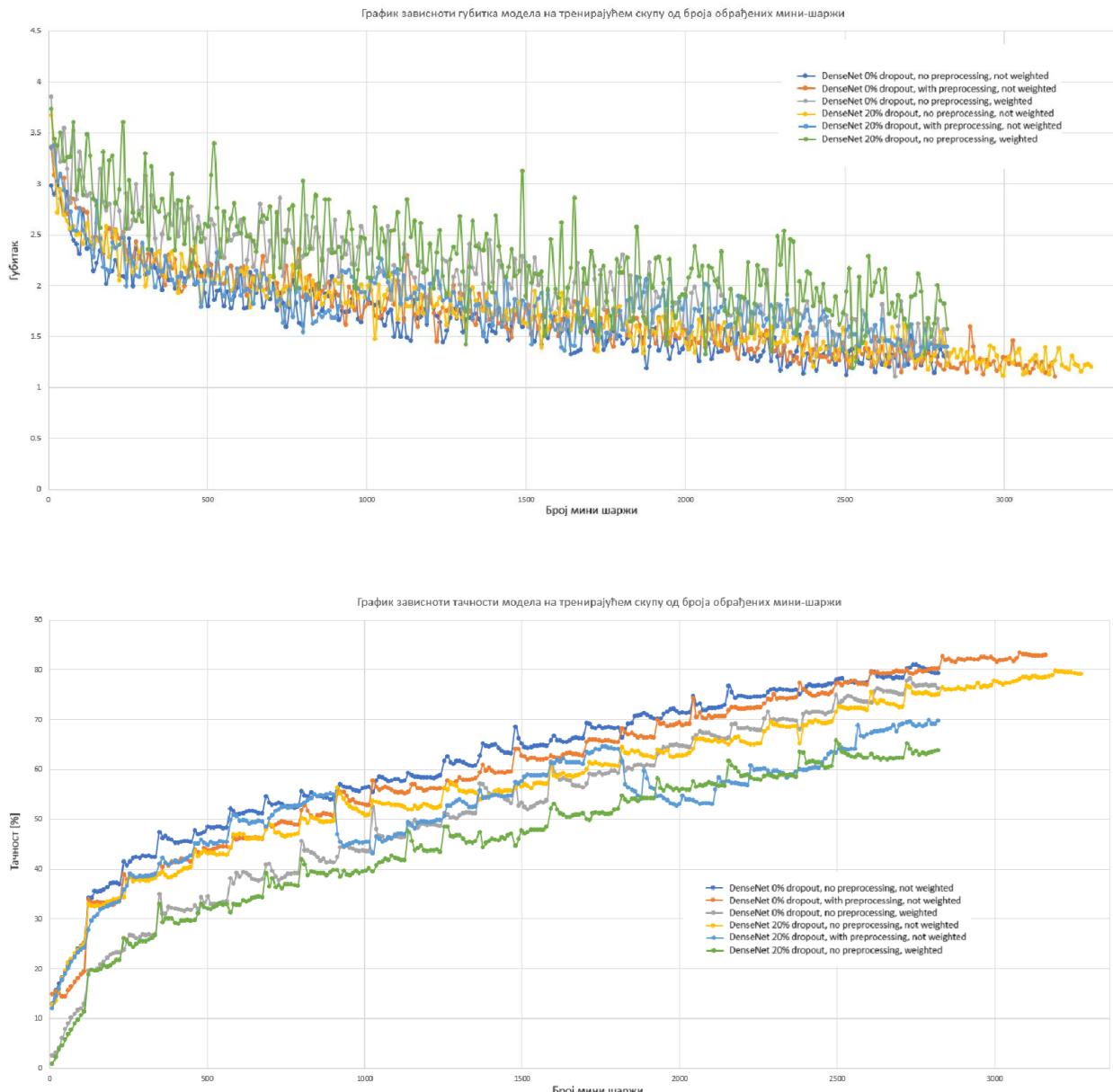




Слика 33. Резултати детекције средњег Yolov5 модела (десно) у поређењу са ознакама лабелара (лево)

4.2 Резултати класификације

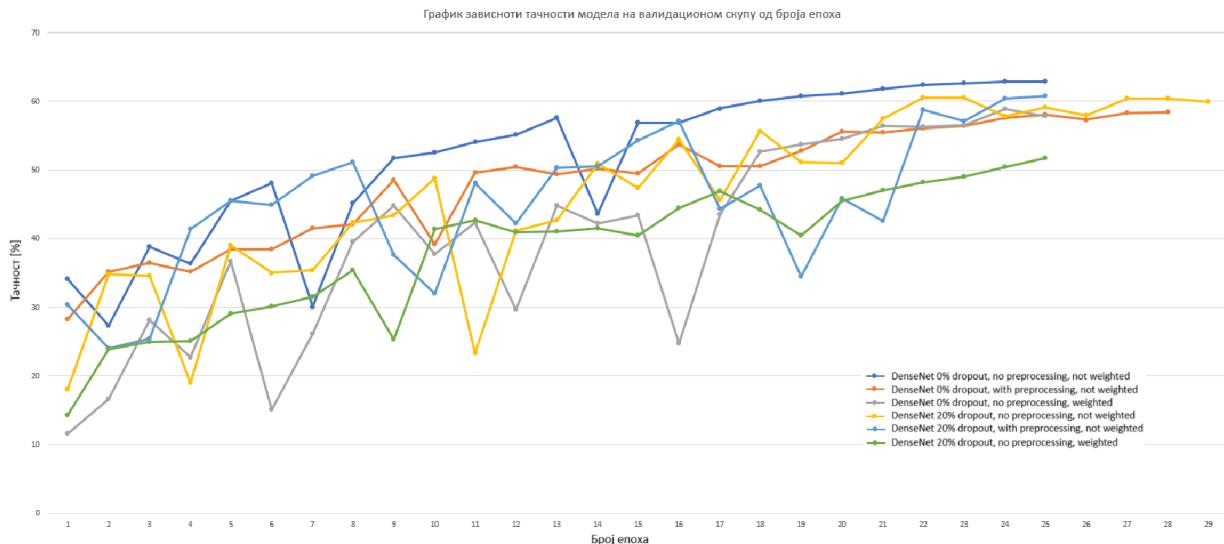
Предложена архитектура класификатора рециклажног отпада била је DenseNet. Модел је трениран кроз 25-29 епоха на 7240 слика отпадног материјала, 1583 примера су коришћена за валидацију, док је финална апроксимација тачности модела добијена на основу 1551 до тада невиђених примера. Слике су аугментоване насумичним хоризонталним и вертикалним пресликавањем, скалиране на 224x224 пиксела и нормализоване на опсег од -0.5 до 0.5. Величина мини шарже била је 64, а адаптивна стопа учења имала је почетну вредност 0.01, прилагођена методом косинусног каљења кроз епохе. Методом стохастичког градијентног спуста са моментом, оптимизована је функција губитка за различите хиперпараметре модела која је приказана на слици 34, заједно са тачношћу модела на тренирајућем скупу.



Слика 34. Губитак модела на тренирајућем скупу кроз епохе (горе) и тачност модела на тренирајућем скупу кроз епохе (доле)

Испитана је следећа комбинација хиперпараметара:

1. слој одбаџивања (без одбаџивања и са 20% одбаџивања)
2. претпроцесирање слика методом еквализације хисограма
3. отежавање класа пропорционално борју примера $w_i = \frac{N}{N_t * N_c}$, где је N укупан број примера, N_i број примера класе i , а N_c укупан број класа



Слика 35. Тачност модела на валидационом скупу кроз епохе

На слици 35, дата је тачност модела на валидационом скупу кроз епохе, при чему је испитано шест комбинација хиперпараметара. DenseNet169 се показао као модел са превеликим бројем параметара те је његово тренирање кроз две епохе трајало неколико сати и превремено је заустављено. Између осталих шест модела, модел без одбацивања, претпроцесирања података и отежавања класа дао је нешто боље резултате, те је он одабран као победник.

На сликама 36. и 37. дате су матрице конфузије за моделе са и без предобраде података на тестирајућем скупу. Ова два модела дала су најбољу тачност на валидационом скупу од преко 60% те су одабрана за евалуацију на тестирајућем. Класа која се најмање појављивала – пуне пластична флаша за уље, неуспешно је класификована од стране оба модела. Једине архитектуре модела за класификацију које су успешно сврстале мањинске класе биле су отежане пропорционално броју примера, али збој значајно мање тачности код већинских класа, ови модели су знатно подбацили на пољу свеукупне тачности. У зависности од индустријских захтева, на овај начин можемо подешавати жељене тачности одређених категорија отпада, али увек по цену преосталих поткатегорија.

Модели са 20% одбацивања имали су нешто мању укупну тачност и на валидационом и на тестирајућем скупу што значи да није дошло до преобучавања, те са повећањем броја параметра мреже и додатним тренирањем постоји простор за додатно унапређење метрика модела.

		Матрица конфузије																										
		0	5	10	15	20	25																					
Тачна класа	Предвиђена класа	0	5	10	15	20	25																					
0	65 2 9 4 0 0 5 0 0 0 0 1 1 0 13 0 0 0 0 0 2 1 0 0 0 0 0 1 0	9 26 1 3 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0	8 0 56 0 1 3 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 8 14 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	0 0 0 0 86 6 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0	0 0 0 0 0 2 30 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 1 0 0 0 61 5 0 0 0 0 0 0 0 0 2 0 0 0 0 1 0 0 3 0 1 0	1 0 1 0 0 0 4 24 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 38 1 0 0 0 0 0 2 3 0 0 0 0 6 0 0 0 3 4	0 0 0 0 0 0 0 4 17 0	0 0 0 0 1 0 1 0 2 2 8 2 0 1 0 0 2 3 0 1 0 0 1 0 0 0 4 0	0 0 0 0 0 0 1 0 1 0 15 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0	2 0 6 0 0 0 0 0 1 0 0 0 16 0 18 1 0 0 2 0 1 0 0 0 0 0 0 1	0 1 0 0 0 0 0 0 0 0 0 2 0 2 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1 2 13 0 1 0 3 0 0 0 2 0 14 0 165 9 4 1 3 1 1 4 2 0 0 4 3 1	1 5 0 0 0 1 1 0 1 0 1 1 0 0 22 55 0 0 1 0 0 0 0 0 0 2 1 0	0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 23 1 0 1 0 0 0 0 1 5 7	0 1 1 0 0 0 0 0 2 0 0 1 0 0 0 0 0 19 1 0 0 1 1 0 0 1 1 1	2 0 0 0 1 0 8 0 1 0 1 0 3 0 0 2 0 0 66 0 4 0 0 0 1 1 5 3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 14 0 0 0 0 0 0 1 0	0 0 0 0 0 1 2 1 0 0 1 1 0 0 0 2 2 2 1 1 25 1 0 2 0 0 1 0	1 1 1 0 0 1 2 0 5 0 0 0 0 0 11 1 1 3 2 1 1 5 0 0 0 0 4 1	0 0 0 0 0 0 0 11 1 0 0 0 0 1 0 4 2 0 1 1 0 18 0 0 0 0 4	0 24 1 0 0	0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 18 0 2 0	1 2 0 0 1 0 0 0 3 0 0 0 2 0 0 0 0 0 1 0 0 4 0 0 1 21 0 0	0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 3 0 0 1 2 0 0 0 0 0 46 12	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 4 0 2 0 2 0 2 0 0 0 10 73

Слика 36. Матрица конфузије модела без предобраде података

		Confusion Matrix																										
		0	5	10	15	20	25																					
Actuals	Predictions	0	5	10	15	20	25																					
0	69 4 6 1 0 0 3 0 0 0 1 0 1 0 16 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	5 28 1 2 0 0 0 0 0 0 0 0 0 0 4 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	4 1 43 0 0 1 0 0 0 0 0 1 0 22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 9 11 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 87 4 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0	1 0 0 0 5 26 0 1 1 0 0 0	0 0 0 0 0 0 53 15 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 2 0 1 1	0 0 2 0 0 0 6 22 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 2 0 0 0	0 0 0 0 0 0 0 39 5 0 0 0 0 0 2 1 2 0 0 0 0 5 0 0 0 0 3	0 0 0 0 0 0 0 5 14 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0	1 0 0 0 5 0 0 1 1 1 7 2 0 0 2 0 0 1 0 3 0 2 0 0 0 2 0	0 0 0 0 0 0 1 0 0 0 2 12 0 0 0 3 0 0 0 0 1 0 0 1 0 0 0 1	3 1 4 0 0 0 0 0 1 0 0 1 17 0 18 1 0 0 0 0 0 2 0 0 0 0 0 0	0 0 0 1 0 0 0 0 0 0 0 1 0 3 2 0 0 0 0 1 0 0 0 0 0 0 0 0	9 2 7 0 1 0 0 0 0 2 0 8 0 17 710 2 0 1 1 1 2 1 0 0 6 3 1	2 6 0 0 0 0 0 1 0 0 0 0 0 0 16 62 1 0 0 0 3 0 3 0 0 0 5 5	0 0 0 0 0 0 3 1 3 0 0 0 0 0 1 1 17 0 0 0 3 0 3 0 0 0 0	3 0 0 0 0 0 2 0 2 0 0 1 1 0 6 0 0 11 0 0 3 0 1 0 0 0 0	1 0 1 0 3 0 7 1 0 0 1 1 1 0 3 1 4 1 61 0 3 0 0 2 5 0 1 1	0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1 10 3 0 0 0 0 0 1 0	0 0 0 0 0 2 0 1 1 1 0 0 2 1 0 1 1 3 3 0 3 2 1 1 0 0 0 1 0	2 0 0 0 0 0 2 1 5 1 0 1 4 0 8 3 1 2 1 0 1 8 1 0 0 0 0	0 0 0 0 0 0 0 9 0 0 0 0 0 1 0 5 2 0 0 1 0 18 0 0 0 0 7	0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 21 0 0 0	0 0 0 0 0 0 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 20 0 1 0	0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 7 2 0 0 4 0 0 2 0 1 0 18 0 0	0 0 0 0 0 0 6 0 2 0 0 0 0 0 1 0 4 0 1 0 4 0 0 0 2 0 36 12	0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 5 0 6 0 2 0 2 0 2 0 0 0 6 64

Слика 37. Матрица конфузије модела са предобрадом података

У табели 6 дати су F1 скорови (хармонијска средина прецизности и одзива) надкласа за најбољи модел, на основу којих се може видети утицај небалансираног скупа на предрасуде класификатора. Одвојено, у табели 7. су дати и F1 скорови за појединачне класе при чему је прављена разлика између поткатегорија. Из ових табела може се видети да је најмање заступљена класа, детерценти, уједно и најлошије класификована, што је очекивано. Овај проблем се делимично решава увођењем тежина класа, али приликом евалуације тих модела цена смањења свеукупне тачности и тачности осталих класа била је превелика. Увођење мање драстичних тежина класа може бити један од приступа за надоградњу овог рада у будућности.

	Пл. флаше	Канисарери	Конзерве	Дешерченди	Стакло	Карбон
F1 скор	67.3%	60.3%	72.9%	48.8%	74.5%	66.4%
Бр. шачних	699	19	66	62	63	119
Број примера	1031	30	98	144	86	162

Табела 6. F1 скорови по класама најбољег модела класификације

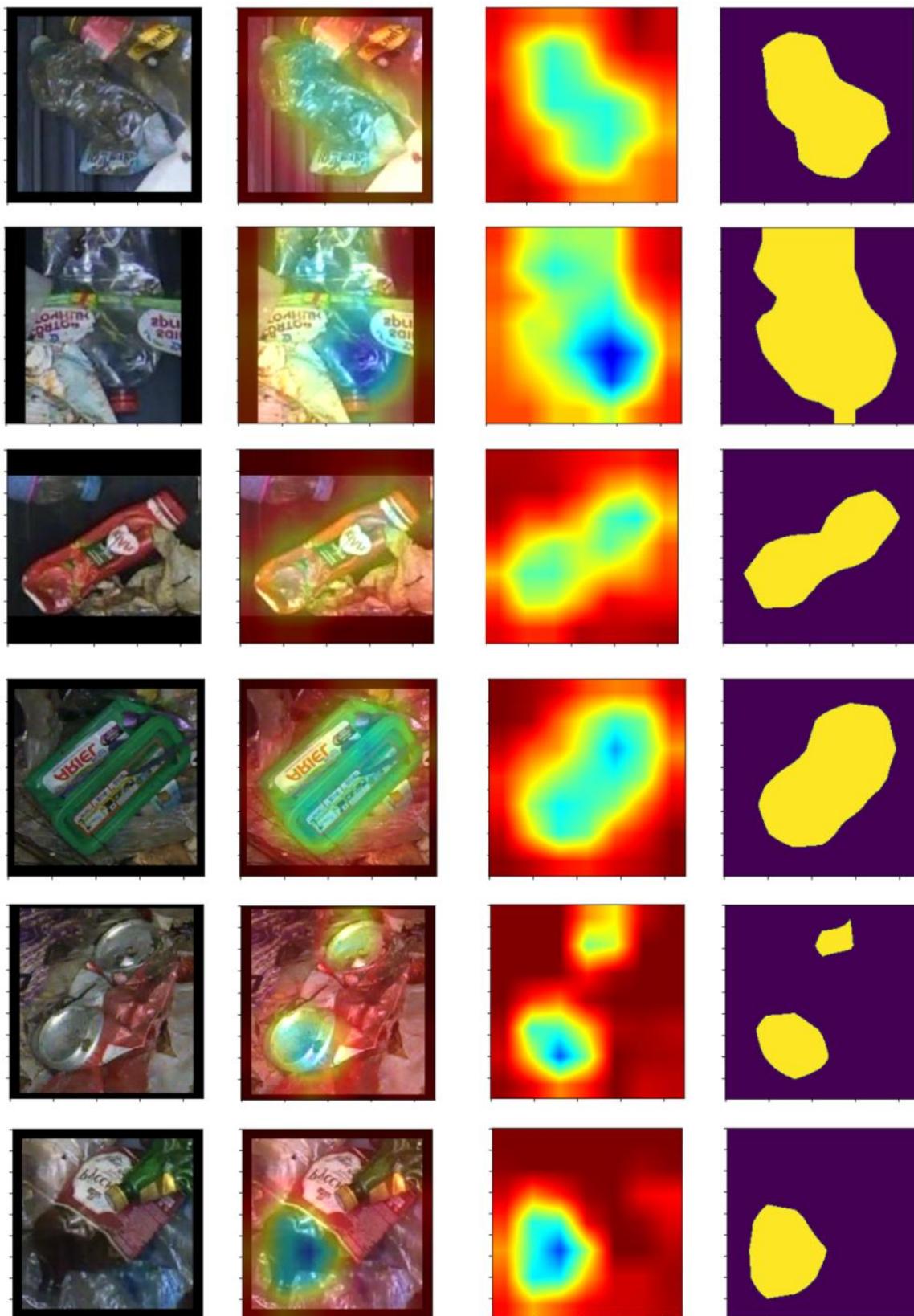
	Пл. флаше	Канисарери	Конзерве	Дешерченди	Стакло	Карбон
F1 скор	91.1%	60.3%	72.9%	53.5%	78.1%	78.8%
Бр. шачних	946	19	66	68	66	141
Број примера	1031	30	98	144	86	162

Табела 7. F1 скорови за надкласе модела без слоја одбацивања

Укупна тачност модела била је 66.3% (односно 84.2% за шест надкласа), док је његово извршавање приликом евалуације трајало реда величине 80 милисекунди по примеру, што је доволјно добар перформанс за употребу у продукцији.

Након класификације, примењена је CAM (енг. *Class Activation Map*) метода на излазни слој DenseNet неуралне мреже, која нам омогућава да видимо мапу пиксела који су највише допринели при доношењу класификационе одлуке. Како бисмо проценили ефикасност ове методе, стандардна евалуација по принципу количника пресека и уније је примењена над тестирајућим скупом података за сегментацију. Добијен Дајсов коефицијент износио је 44.3% који је упоређенес са резултатима семантичке сегментације са надгледањем.

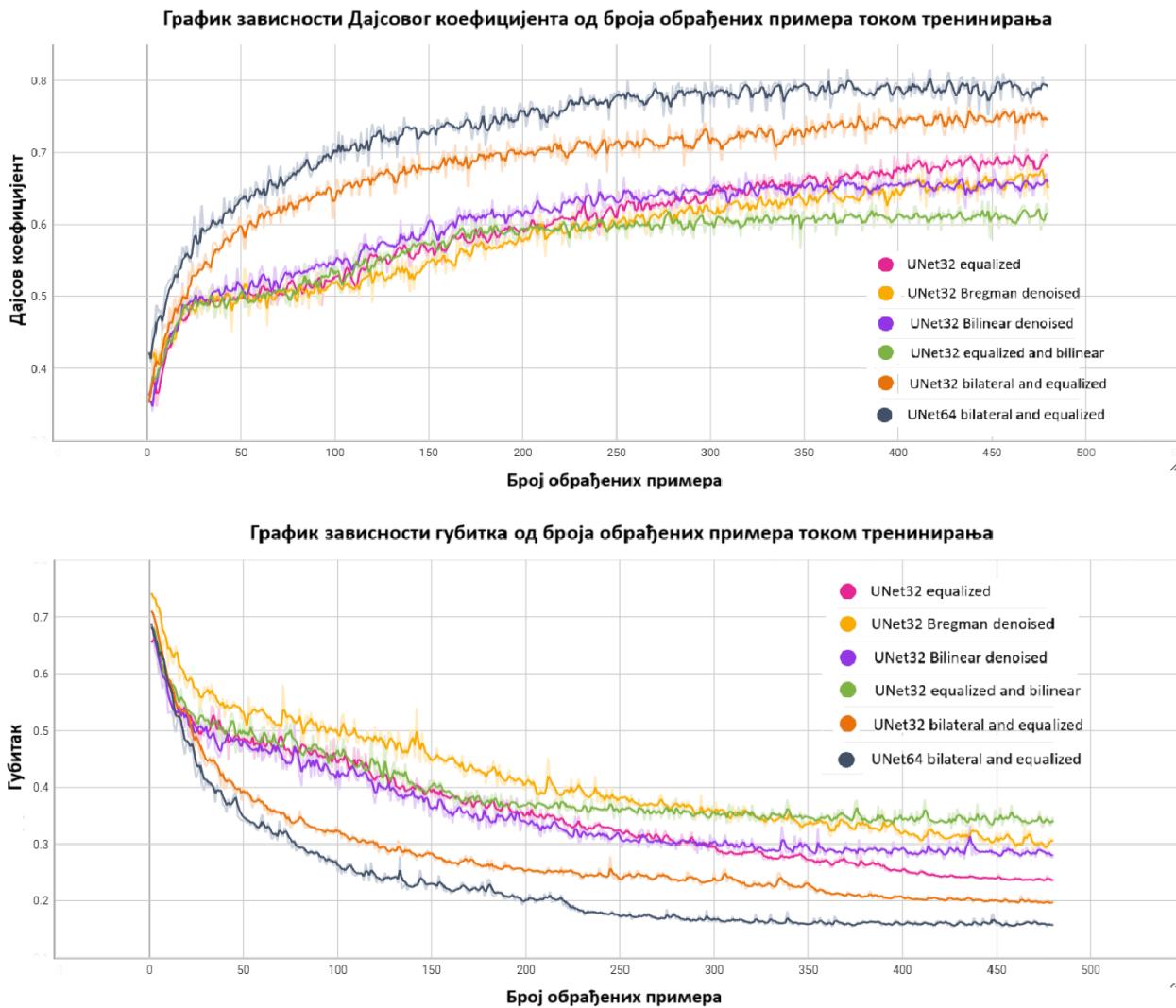
Визуелна презентација резултата Grad CAM сегментације дата је на слици 38. Иако овај приступ може дати завидне резултате, те корелација између пиксела на којима се налази објект за рециклажу и значајних региона за класifikатор постоји, еквиваленција не важи увек, те су укључени и примери где је разлика између ова два појма очигледна.



Слика 38. Резултати Grad CAM сегментације

4.3 Резултати семантичке сегментације

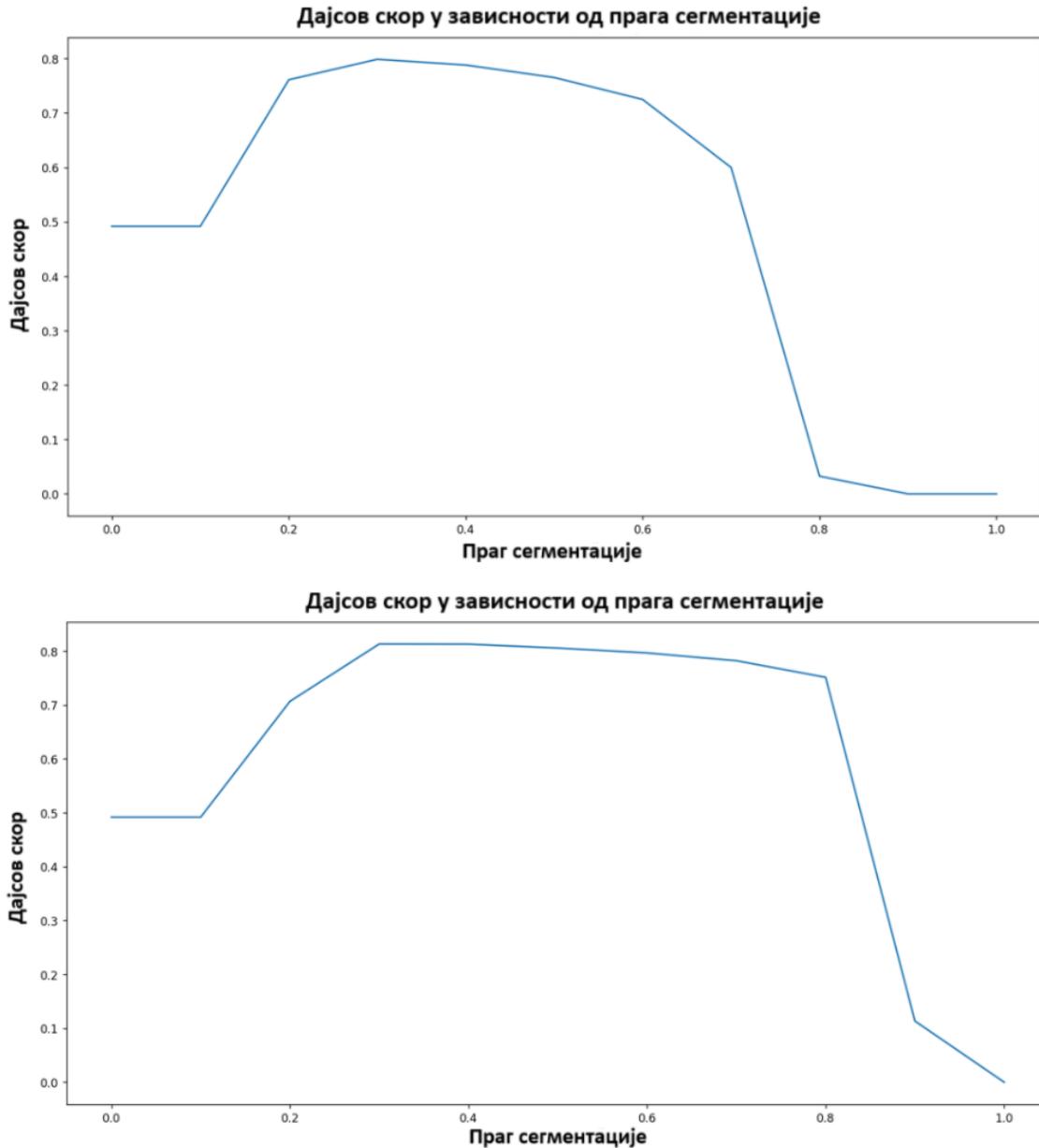
U-Net конволуциона неурална мрежа је модел за семантичку сегментацију са надгледањем, што значи да је неопходно поседовати обележене примере за тренирање и тестирање. Методом унакрсне валидације, испитано је неколико варијанти овог модела за сегментацију рециклажног отпада. Први од хиперпараметара био је број канала у првом и последњем слоју конволуције, док су дубљи слојеви имали пропорционално више канала. Испитане су варијанте са 32 и 64 канала, затим је испитан утицај аугментације улазних слика на моћ генерализације, као и билинеарно скалирање броја канала приликом деконволуције модела. Величина мини шарже била је 28, а број епоха тренирања износио је 40. Оптимизовали смо бинарну унакрсну ентропију са стопом учења од 10^{-5} , а коришћен је RMS Prop оптимизатор са моментом приликом градијентног спуста.



Слика 39. Губитак и Дајсов скор на тренирајућем скупу у зависности од броја обрађених примера

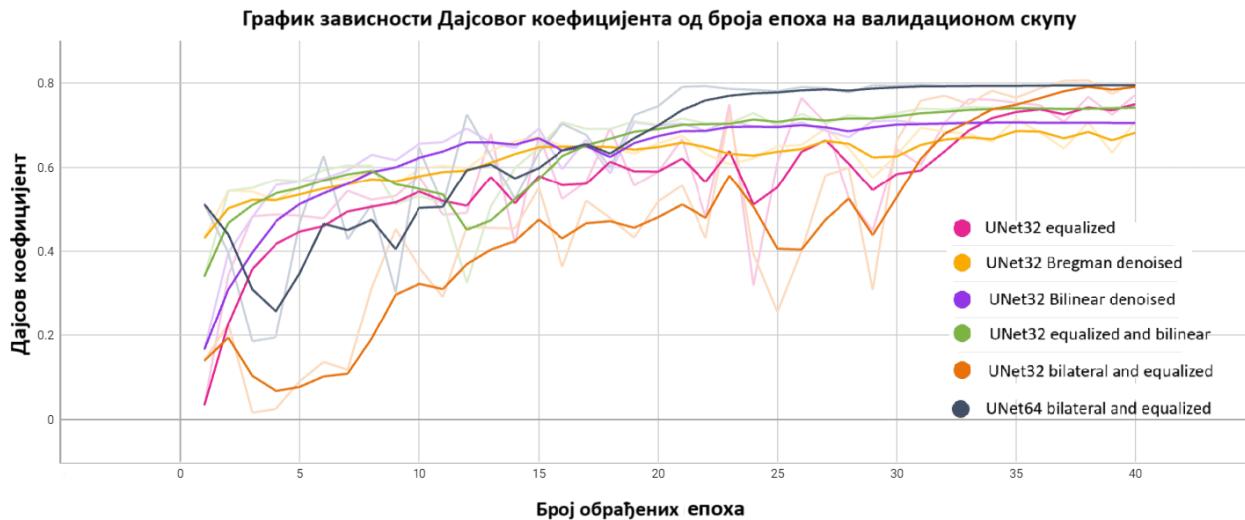
Дајсов скор описан у претходном поглављу коришћен је за евалуацију модела, што је приказано на слици 39, заједно са функцијом губитка на тренирајућем скупу кроз епохе. Имплементација модела у програмском језику *Python*, као и хиперпараметри дати су у прилогу В на крају овог рада.

Најбољи резултат на валидационом скупу дали су билинеарни модели са аугментацијом улазних слика еквализацијом хистограма са око 80% тачности. За ова два модела, испитани су прагови сигурности при одлуци да ли неки пиксел припада региону, а зависности Дајсовог коефицијента на валидационом скупу од прага приказане су на слици 40.



Слика 40. Дајсов скор у зависности од прага сегментације на валидационом скупу за моделе са 32 (горе) и 64 (доле) канала у првом и последњем слоју неуралне мреже

Са ових графика можемоочитати оптималан праг сегментације који је у оба случаја износио 0.35. График зависности Дајсовог коефицијента на валидационом скупу у зависности од броја епоха током тренирања за праг од 0.5 дат је на слици 41.



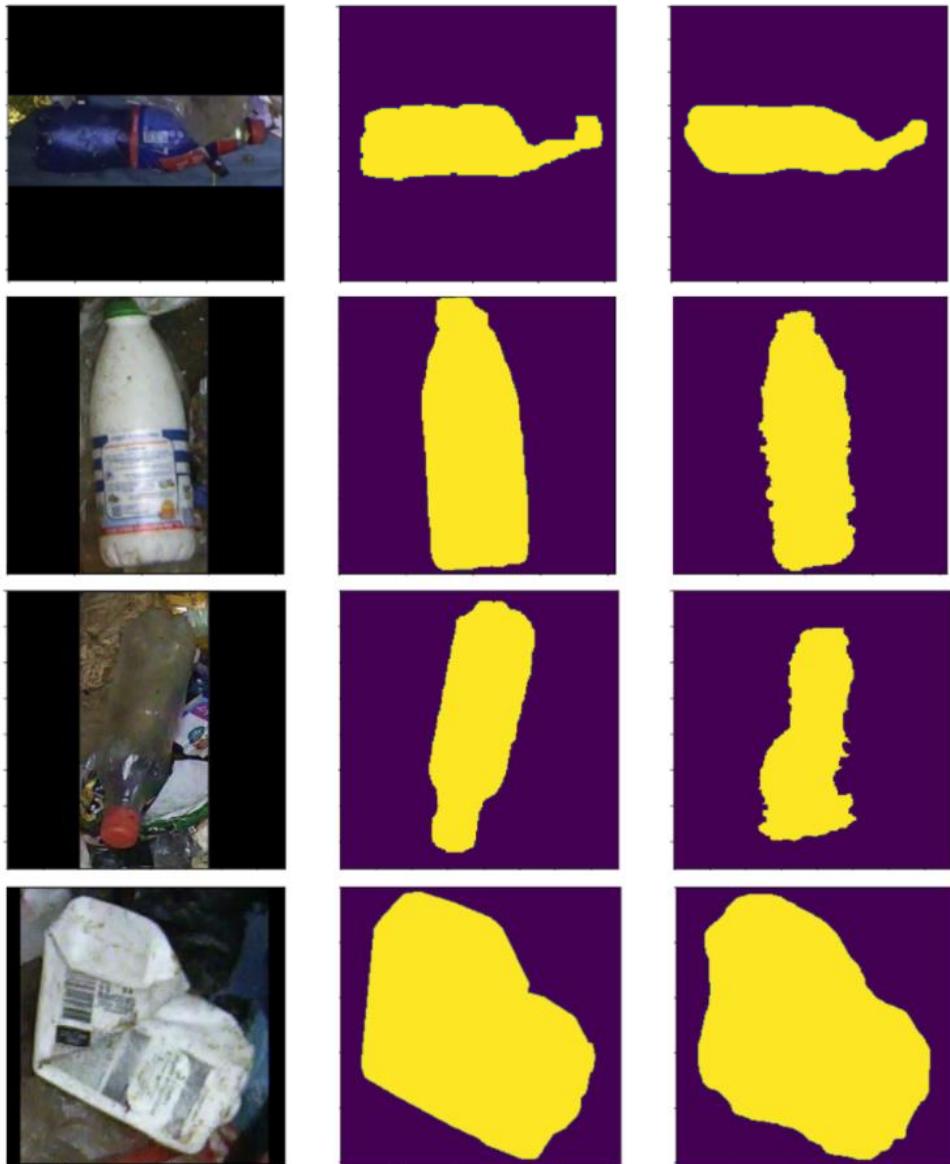
Слика 41. Дајсов скор на валидационом скупу у зависности од броја епоха

Различите форме U-Net неуралне мреже евалауиране су на валидационом скупу чији су резултати дати у табели 8. Између испитаних врста предобрада слика, еквализација хистограма је једина резултовала позитивним унапређењем метрика тачности у поређењу са моделом тренираним над оригиналним сликама, те је она изабрана као основа за тренирање већих модела. Најбоље показана архитектура била је она са 32 канала у првом слоју и билинеарним скалирањем броја канала, а за оптималан праг од 0.35 и еквализацијом хистограма као формом предобраде слике, која је резултовала Дајсовим коефицијентом од 80.7% на валидационом скупу. Недалеко од ње, билинеарна неурална мрежа са 64 канала у првом слоју дала је тачност од 79.6%, што значи да повећањем броја параметра не добијамо значајно другачији резултат.

Модел	Врста претпроцесирања	Дајсов скор
UNet 32	Без претпроцесирања	71.9%
UNet 32	Еквализација хистограма	77.2%
UNet 32	Брегманов филтер	71.5%
UNet 32	Билатерални филтер	71.2%
UNet 32	Екв. хист. и билатерални филтер	74.5%
UNet 32 bilinear	Еквализација хистограма	80.7%
UNet 64 bilinear	Еквализација хистограма	79.6%

Табела 8. Тачности сегментације модела за различите методе претпроцесирања слика на валидационом скупу

Слике су на крају скалиране на оригиналну величину, а затим су провучене кроз форму обраде маске поћу дилатације и ерозије бинарне слике за добијање глаткијих ивица, након чега су упоређене са тачним лабелама, што је приказано на слици 42. Брзина извршавања модела приликом евалуације износила је око 10ms по примеру, што као и у случају детекције, не представља критично место за оптимизацију ради продуктизације кода.





Слика 42. Резултати сегментације (десно) у поређењу са лабелом (средина) и оригиналном сликом (лево)

Коначни резултати семантичке сегментације дати су у табели 9. У складу са естимираним тачностима у процесу евалуације, билинеарни модел са 32 канала у првом слоју и евализацијом хистограма као видом обраде слике дао је резултат од 81.3% на тестирајућем скупу. У табели 9. упоређени су резултати свих модела на тестирајућем скупу, али су дати и резултати градијената класификационог модела, као и основни случај предвиђања целог детектованог региона. Узимајући у обзир мало време извршавања модела и јефтине тренирање, семантичка сегментација са надгледањем је вредна интеграције ако је прецизна граница одлуке од значаја у индустрији.

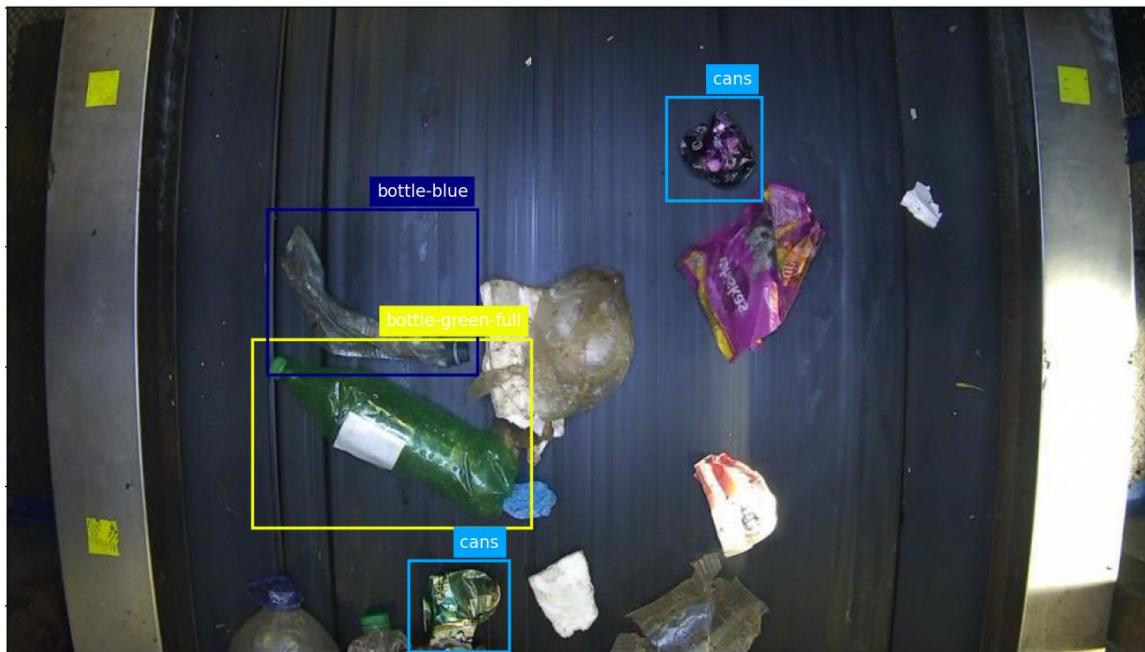
Grad CAM сегментација без надгледања дала је мањи Дајсов коефицијент него одабир целог региона детекције, па као такав не доноси значајно побољшање у односу на регион детекције који квалитетно лоцира објекат са фотографије. У референтном раду, D. Yudin et al. [2] показали су да употребом бољег класификатора се може добити тачност од 55% што би ово методи дало малу предност уколико обележени подаци не постоје или је цена аквизиције истих велика.

Модел	Врста претпроцесирања	Дајсов скор
Grad CAM	Без претпроцесирања	47.2%
YoloV5 детекција	Без претпроцесирања	49.1%
UNet 32	Без претпроцесирања	75.5%
UNet 32	Еквализација хистограма	79.9%
UNet 32	Брегманов филтер	75.9%
UNet 32	Билатерлани филтер	74.3%
UNet 32	Екв. хист. и билатерални филтер	77.5%
UNet 32 bilateral	Еквализација хистограма	81.3%
UNet 64 bilateral	Еквализација хистограма	80.5%

Табела 9. Тачности сегментације модела за различите методе претпроцесирања слика на тестирајућем скупу

4.4 Резултати сегментације инстанци

Проблем сегментације инстанци објекта рециклажног отпада смо поделили на три једноставнија проблема описана у претходним потпоглављима. Предложене су три структуре везивања ових модела у јединствени софтверски алат који за улаз има фотографију покретне траке из птичије перспективе, а на излазу је иста фотографија прекривена детектованим регионима у боји одговарајуће класе. На слици 43 дат је пример једне фотографије обележене од стране стручњака која је спроведена кроз процес сегментације инстанци на два предложена начина.



Слика 43. Обележен примерак рециклажног отпада на покретној траци

Време извршавања процеса од почетка до краја износило је између 500ms и 800ms. На ово време није утицао начин сегментације у виду посебног модела или резултата градијената класификације. Иако би сви модели у теорији требало да се изврше за око 100-200ms, појединачно процесирање слика и учитавање у меморију узима већи део овог временског интервала, те додатна оптимизација модела није била неопходна.

На слици 44. дат је излаз одабраног модела детекције за улазну фотографију који је успешно детектовао три од четири објекта на слици. Ово осликава реалну успешност модела која је естимирана на око 70% F1 скора на тестирајућем скупу.



Слика 44. Резултат детекције рециклажног отпада

Излаз модела детекције је затим доведен на улаз класификатора. У првом случају, класификатор обавља сортирање појединачних објеката у једну од 28 класа, а затим се његови градијенти користе за одабир оних пиксела који су највише допринели у тој одлуци и на тај начин добијамо сегментацију инстанци без супервизијског обучавања сегментатора. Резултат овог приступа дат је на слици 45. Време извршавања ове секвенце модела износило је између 600ms и 800ms, што је обухватало не само извршавање појединачних модела на једној фотографији, већ и учитавање слике у меморију и њену предобраду.



Слика 45. Резултат сегментације инстанци рециклажног отпада помоћу градијената класификатора

Други приступ обухватао је не само класификацију објекта посебним моделом, већ и обучавање U-Net конволуционе неуралне мреже за сегментацију отпада са супервизијом. Овакав приступ дао је значајно побољшање у виду метрика покривености региона од интереса, што се може и видети на слици 46.

Упркос коришћења додатног модела, перформанса целокупног процеса естимирана је на око 600ms, што значи да увођење додатног модела није повећало потребно време за извршавање које је првенствено потрошено на управљање меморијом. Као такав, овај модел се може применити у постројењу за сегментацију инстанци рециклажног отпада на покретним тракама у реалном времену. Свеукупна тачност модела естимирана је на нешто више од 50%, те се као такав може увести у асистирању мануелног сортирања у постојењу, а уз одређена унапређења потенцијално може постати нова парадигма у аутоматском сортирању рециклажног отпада.



Слика 46. Резултат сегментације инстанци рециклажног отпада са посебним моделом за класификацију и сегментацију

5 ДИСКУСИЈА И ЗАКЉУЧАК

Главни циљ развоја аутоматских поступака за препознавање рециклажног отпада је потпуна аутоматизација процеса сортирања помоћу роботских алата. Први корак у том процесу је прављење фотографије из птичије перспективе, скалирање те слике, а затим и сегментација инстанци отпада од интереса помоћу дубоких неуралних мрежа, чији резултат нам даје информацију о присуству отпада које је потребно издвојити и даљем приступу у рециклажи. Прецизно издвајање региона од интереса је кључно, како би роботска рука са великим прецизношћу могла да уклони означени отпад са покретне траке.

У овом раду испитана су два модела за детекцију и по један модел неуралних мрежа за класификацију и сегментацију рециклажног отпада са покретних трака. Више хиљада слика су процесирани и подељене у скупове за тренирање, валидацију и тестирање модела. Резултати су представљени кроз три одвојене целине и обухватали су подешавање параметара, заштиту од преобучавања и финално, евалуацију на невиђеном тестирајућем скупу као коначном естимацијом ефикасности модела. Разматрана је брзина извршавања инференције модела и дискутован је услов за њихову употребу у производњи.

Унапред тренирана неурална мрежа Faster R-CNN, која је затим дотренирана на примерима рециклажног отпада, искоришћена је као детектор и први корак у аутоматском процесу издвајања отпада у постројењу. Након скалирања 1962 примера на величину 256x144 пиксела, добијен је резултат од 29% средње просечне прецизности са границом од 50% преклапања. Употребом већих слика, димензија 480x270 пиксела, средња просечна прецизност повећана је на 46.2% на тестирајућем скупу. Примарни проблем при тренирању ове мреже била су хардверска ограничења која, због броја подесивих параметара мреже, нису дозвољавала веће мини шарже од 4 примера, а тренирање једне епохе трајало је више сати. Додатним тренирањем, бољим подешавањем хипер-параметара и коришћењем слика веће резолуције, може се добити mAP_{50} од око 50% [2]. Одстрањивање високо преклапајућих детекција у овом раду постигнуто је не-максималном супресијом, што није идеалан начин за мало вероватне класе. Паметнијим одсецањем може се значајно унапредити ова метрика, са незнатном ценом у виду времена тренирања или извршавања у продукцији. Увођењем YOLOv5 архитектуре који се у литератури показују као успешније решење овог проблема [2], а посебно мозаичком аугментацијом тренирајућих података постигнута је тачност од око 69% на тестирајућем скупу. Време инференције је уједно смањено са 120ms на око 10ms чиме детекција више није одузимала највећи део времена у низу аутоматске сегментације инстанци објекта. Проблем детекције је дефинитивно кључни корак у низу аутоматског препознавања, као почетни корак од кога зависи свеукупна тачност, те је увек исплативо уложити у његово унапређење.

Проблем класификације решен је помоћу неуралне мреже DenseNet121, а испитана је и DenseNet169 архитектура чија је главна ограничавајућа околност била хардверски лимит. Добијена тачност класификације била је 66.3% када је у питању 28 класа, односно 84.2% за шест надкласа, где је главни проблем представљала небалансираност података. Међутим, бољим аугментовањем примера и давањем већих тежина мањинским класама нису постигнути бољи

резултати. Употребом већих хардверских ресурса, подешавањем хиперпараметара, и дотренирањем DenseNet169 неуралне мреже може се постићи тачност преко 70% на овом скупу [2]. Поред детекције, проблем класификације се показао као кључни корак у низу аутоматског препознавања, од кога зависи свеукупна тачност и време извршавања, те има највише смисла улагати у његово унапређење у даљем раду.

Још један интересантан податак који се може добити из класификатора на бази конволуционих неуралних мрежа јесте приказ градијената у последњем слоју пропагацијом уназад (енг. *Grad CAM*), чиме се може видети који делови слике су највише допринели при доношењу одлуке. У поглављу 4.2 видели смо да се овај податак може искористи за добијање сегментованих региона без супервизираног обучавања модела семантичке сегментације. На тај начин може се избећи скуп и дуготрајан процес обележавања слика прецизним регионима посебно ако резолуција величине пиксела није неопходна при уклањању отпада са покретних трaka. Такође, превазилази се и проблем већине метода дубоког учења које, за разлику од стручњака, не могу да дају образложение иза своје одлуке, а које је од великог значаја при људском разумевању понашања модела.

Неколико варијанти U-Net конволуционе неуралне мреже пројектовано је као решење проблема сегментације рециклажног отпада и добијена је укупна тачност од 81.3% на тестирајућем скупу. Видно побољшање у виду метрика добијено је претпроцесирањем слика употребом филтра за уклањање шума, као и еквализацијом хистограма са циљем повећања контраста слике. Употреба модела са већим бројем параметара се није показала као неопходна и нема доказа да би довела до значајне промене у свеукупној тачности.

На крају, повезивањем сва три модела на ред и евалуацијом сегментације инстанци добијени су резултати које је тешко евалуирати као последице недостатка одговарајућих лабела. Естимацијом F1 скора на основу детекције и класификације дала је процену тачности сегментације инстанци од преко 50% на тестирајућем скупу. У зависности од захтева постројења, предложена су решења са сегментацијом, уколико је прецизно издвајање отпада неопходно, и без сегментације, ако су ограничавајући региони довољни за механичко издвајање. Модели машинског учења су заједно допринели $\sim 100\text{ms}$ приликом извршавања у продукцији што је чинило мање од 20% укупног времена извршавања процеса. Показано је да је проблем аутоматског издвајања рециклажног отпада решив помоћу дубоких неуралних мрежа, и као такав, може се применити у хибридном моделу асистирања мануелном сортирању. Резултат овог експеримента јесте софтверски код који се може применити у индустриском постројењу и уједно је покривен тестовима као вид заштите од грешака у будућим итерацијама.

Осим прослеђивања резултата једног модела на улаз другог, даљи рад би могао да обухвата и дотренирање модела класификације и сегментације на резултатима модела детекције како би обучавајући скуп података боље осликовао реални улаз модела у постројењу за рециклажу. Резултат овог експеримента била би свеукупна тачност и брзина извршавања у продукцији, а за извођење захтевала би ручно обележавање дела скупа података као референтне истинске вредности за евалуацију и тренирање модела.

6 ЛИТЕРАТУРА

- [1] На основу члана 38. став 1. Закона о планском систему Републике Србије, *Програм управљања оштадом у Републици Србији за период 2022 – 2031. године*, Службени гласник РС, број 30/18
- [2] D. Yudin, N. Zakharenkoc, A. Smetanind, R. Filonovb, M. Kichikb, V. Kuznetsov, D. Larichevb, E. Gudovd, S. Budennyya, A. Panov, *Hierarchical waste detection with weakly supervised segmentation in images from recycling plants*, Available at SSRN: <https://ssrn.com/abstract=4183424>, 2022
- [3] Kaggle, *WaRP - Waste Recycling Plant Dataset*, 2023. URL:
<https://www.kaggle.com/datasets/parohod/warp-waste-recycling-plant-dataset>
- [4] D. Sculley et. al. NIPS: Hidden Technical Debt in Machine Learning Systems, Advances in neural information processing systems, 2015
- [5] A. Ng, et al., Machine Learning in Production, Preuzeto sa: [DeepLearning.AI](#)
- [6] Kaggle 2018. Tuning your learning rate, Preuzeto sa:
<https://www.kaggle.com/residentmario/tuning-your-learning-rate>
- [7] Ultralytics Yolov5 Tutoriali, Preuzeto sa: <https://docs.ultralytics.com/yolov5/tutorials>
- [8] Website *scikit-learn.org*, Preuzeto sa: https://scikit-learn.org/stable/modules/cross_validation.html
- [9] S. Ren, K. He, R. Girshick, J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, (6), 2017. pp. 11371149, doi:10.1109/TPAMI.2016.2577031.
- [10] C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh, I. Yeh, *CSPNet: A New Backbone that can Enhance Learning Capability of CNN*. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
- [11] Parico, A.I.B.; Ahamed, T. Real Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT. Sensors 2021, 21, 4803. <https://doi.org/10.3390/s21144803>
- [12] Huo, B.; Li, C.; Zhang, J.; Xue, Y.; Lin, Z. SAFF-SSD: Self-Attention Combined Feature Fusion-Based SSD for Small Object Detection in Remote Sensing. Remote Sens. 2023, 15, 3027.
<https://doi.org/10.3390/rs15123027>
- [13] Qiu, M.; Huang, L.; Tang, B.-H. ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion. Remote Sens. 2022, 14, 3498.
<https://doi.org/10.3390/rs14143498>

-
- [14] Shashikant 2019. *Convolutional Neural Network: A Step By Step Guide*, Preuzeto sa: <https://towardsdatascience.com/convolutional-neural-network-a-step-by-step-guide-a8b4c88d6943>
- [15] J. Šušteršić, *Automatska segmentacija pneumotoraksa na rendgenskim snimcima pluća čoveka*, Diplomski rad, Elektrotehnički fakultet u Beogradu, 2019
- [16] W. Mao, W. Chen, C. Wang, Y. Lin, *Recycling waste classification using optimized convolutional neural network*, in Resources, Conservation and Recycling, 164, 2021 105132.
- [17] L. Tang Github, Preuzeto sa: <https://github.com/Lornatang/DenseNet-PyTorch/blob/main>
- [18] Y. Li, S. Xie, X. Chen, P. Dollar, K. He, R. Girshick, *Benchmarking Detection Transfer Learning with Vision Transformers*, Facebook AI Research (FAIR) 2021
- [19] Long J., Shelhamer E., Darrell T. 2015. Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 3431-3440. 2015 doi: 10.1109/CVPR.2015.7298965
- [20] Milesial Github, Preuzeto sa: <https://github.com/milesial/Pytorch-UNet/tree/master>
- [21] Turchenko V., Chalmers E., Luczak A. A Deep Convolutional Auto-Encoder with Pooling – Unpooling Layers in Caffe, 2017. abs/1701.04949
- [22] Q. Zhang, Q. Yang, X. Zhang, Q. Bao, J. Su, X. Liu, *Waste image classification based on transfer learning and convolutional neural network*, Waste Management 135, 2021 pp. 150–157
- [23] H. Kanyamahanga Github, Preuzeto sa: https://github.com/hubert10/fasterrcnn_resnet50_fpn_v2_new_dataset/tree/main

ПРИЛОГ А

У Прилогу А налази се део кода из скрипте *dataset.py* помоћу које је имплементиран општи интерфејс са скуповима података у виду итеративног типа у *Python-u*.

dataset.py

```
class ImageDataset(Dataset):
    def __init__(self, mode: str, mean: list, std: list) -> None:
        self.mode = mode
        self.image_file_paths = []
        self.post_transform = transforms.Compose([
            transforms.ConvertImageDtype(torch.float),
            transforms.Normalize(mean, std),
        ])

    def __len__(self):
        return len(self.image_file_paths)

class Prefetcher:
    def __init__(self, dataloader, device: torch.device):
        self.batch_data = None
        self.original_dataloader = dataloader
        self.device = device

        self.data = iter(dataloader)
        self.stream = torch.cuda.Stream()
        self.preload()

    def next(self):
        torch.cuda.current_stream().wait_stream(self.stream)
        batch_data = self.batch_data
        self.preload()
        return batch_data

    def reset(self):
        self.data = iter(self.original_dataloader)
        self.preload()

    def __len__(self) -> int:
        return len(self.original_dataloader)
```

ПРИЛОГ Б

У Прилогу Б налази се део кода из скрипте *DenseNet.py* која имплементира пројектовани модел за класификацију рециклирајућег отпада са надгледањем.

DenseNet.py

```
class DenseNet(nn.Module):
    def __init__(self, netType: int = 121, channels: int = 64, growth_rate: int = 32,
                 bottle_neck_size: int = 4, dropout_rate: float = 0.2, num_classes: int = 28) -> None:
        super(DenseNet, self).__init__()
        self.features = nn.Sequential(
            OrderedDict([
                ("conv0", nn.Conv2d(3, channels, (7, 7), (2, 2), (3, 3), bias=False)),
                ("norm0", nn.BatchNorm2d(channels)),
                ("relu0", nn.ReLU(True)),
                ("pool0", nn.MaxPool2d((3, 3), (2, 2), (1, 1))),
            ])
        )
        block_cfg = (6, 12, 24, 16) if netType == 121 else (6, 12, 32, 32)

    for i, repeat_times in enumerate(block_cfg):
        block = _DenseBlock(
            repeat_times=repeat_times,
            channels=channels,
            growth_rate=growth_rate,
            bottle_neck_size=bottle_neck_size,
            dropout_rate=dropout_rate,
        )
        self.features.add_module(f"denseblock{i + 1}", block)
        channels = channels + int(repeat_times * growth_rate)
        if i != len(block_cfg) - 1:
            trans = _Transition(channels, channels // 2)
            self.features.add_module(f"transition{i + 1}", trans)
            channels = channels // 2

    self.features.add_module("norm5", nn.BatchNorm2d(channels))
    self.features.add_module("relu5", nn.ReLU(True))
    self.avgpool = nn.AdaptiveAvgPool2d((1, 1))
    self.classifier = nn.Linear(channels, num_classes)
    self.gradients = None
    self._initialize_weights()
```

```

def forward(self, x: Tensor) -> Tensor:
    x = self.features(x)
    x.register_hook(self.activations_hook)

    out = self.avgpool(x)
    out = torch.flatten(out, 1)
    out = self.classifier(out)
    return out

def activations_hook(self, grad):
    self.gradients = grad

def _initialize_weights(self) -> None:
    for module in self.modules():
        if isinstance(module, nn.Conv2d):
            nn.init.kaiming_normal_(module.weight)
        elif isinstance(module, nn.BatchNorm2d):
            nn.init.constant_(module.weight, 1)
            nn.init.constant_(module.bias, 0)
        elif isinstance(module, nn.Linear):
            nn.init.constant_(module.bias, 0)

def get_activations_gradient(self):
    return self.gradients

def get_activations(self, x):
    return self.features(x)

class _DenseBlock(nn.ModuleDict):
    def __init__(self, repeat_times: int, channels: int, growth_rate: int,
                 bottle_neck_size: int, dropout_rate: float) -> None:
        super(_DenseBlock, self).__init__()
        for i in range(repeat_times):
            layer = _DenseLayer(
                channels=channels + i * growth_rate,
                growth_rate=growth_rate,
                bottle_neck_size=bottle_neck_size,
                dropout_rate=dropout_rate,
            )
            self.add_module(f"denselayer{i + 1}", layer)

    def forward(self, x: List[Tensor] | Tensor) -> Tensor:
        out = [x]

```

```

for _, layer in self.items():
    denselayer_out = layer(out)
    out.append(denselayer_out)
out = torch.cat(out, 1)
return out

class _Transition(nn.Sequential):
    def __init__(self, in_channels: int, out_channels: int) -> None:
        super(_Transition, self).__init__()
        self.norm = nn.BatchNorm2d(in_channels)
        self.relu = nn.ReLU(True)
        self.conv = nn.Conv2d(in_channels, out_channels, (1, 1), (1, 1), (0, 0), bias=False)
        self.pool = nn.AvgPool2d((2, 2), (2, 2))

    class _DenseLayer(nn.Module):
        def __init__(self, channels: int, growth_rate: int, bottle_neck_size: int, dropout_rate: float) -> None:
            super(_DenseLayer, self).__init__()
            growth_channels = int(bottle_neck_size * growth_rate)
            self.dropout_rate = float(dropout_rate)

            self.norm1 = nn.BatchNorm2d(channels)
            self.relu1 = nn.ReLU(True)
            self.conv1 = nn.Conv2d(channels, growth_channels, (1, 1), (1, 1), (0, 0), bias=False)

            self.norm2 = nn.BatchNorm2d(growth_channels)
            self.relu2 = nn.ReLU(True)
            self.conv2 = nn.Conv2d(growth_channels, growth_rate, (3, 3), (1, 1), (1, 1), bias=False)
            self.dropout = nn.Dropout(dropout_rate, True)

        def forward(self, x: List[Tensor] | Tensor) -> Tensor:
            if isinstance(x, Tensor):
                x = [x]
            out = torch.cat(x, 1)
            out = self.norm1(out)
            out = self.relu1(out)
            out = self.conv1(out)
            out = self.norm2(out)
            out = self.relu2(out)
            out = self.conv2(out)
            if self.dropout_rate > 0:
                out = self.dropout(out)
            return out

```

ПРИЛОГ В

У Прилогу В налази се део кода из скрипте *UNet.py* која имплементира пројектовани модел за семантичку сегментацију рециклијажног отпада са надгледањем.

UNet.py

```

class UNet(nn.Module):
    def __init__(self, base:int = 32, bilinear:bool = False):
        super(UNet, self).__init__()
        self.inc = (DoubleConv(3, base))
        self.down1 = (Down(base, base*2))
        self.down2 = (Down(base*2, base*4))
        self.down3 = (Down(base*4, base*8))
        factor = 2 if bilinear else 1
        self.down4 = (Down(base*8, base*16 // factor))
        self.up1 = (Up(base*16, base*8 // factor, bilinear))
        self.up2 = (Up(base*8, base*4 // factor, bilinear))
        self.up3 = (Up(base*4, base*2 // factor, bilinear))
        self.up4 = (Up(base*2, base, bilinear))
        self.outc = (OutConv(base, 1))

    def forward(self, x):
        x1 = self.inc(x)
        x2 = self.down1(x1)
        x3 = self.down2(x2)
        x4 = self.down3(x3)
        x5 = self.down4(x4)
        x = self.up1(x5, x4)
        x = self.up2(x, x3)
        x = self.up3(x, x2)
        x = self.up4(x, x1)
        logits = self.outc(x)
        return logits

class DoubleConv(nn.Module):
    def __init__(self, in_channels, out_channels, mid_channels=None):
        super().__init__()
        if not mid_channels:
            mid_channels = out_channels
        self.double_conv = nn.Sequential(
            nn.Conv2d(in_channels, mid_channels, kernel_size=3, padding=1, bias=False),

```

```

        nn.BatchNorm2d(mid_channels), nn.ReLU(inplace=True),
        nn.Conv2d(mid_channels, out_channels, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(out_channels), nn.ReLU(inplace=True)
    )

def forward(self, x):
    return self.double_conv(x)

class Down(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.maxpool_conv = nn.Sequential(nn.MaxPool2d(2),
            DoubleConv(in_channels, out_channels)
    )

    def forward(self, x):
        return self.maxpool_conv(x)

class Up(nn.Module):
    def __init__(self, in_channels, out_channels, bilinear=True):
        super().__init__()
        if bilinear:
            self.up = nn.Upsample(scale_factor=2, mode='bilinear', align_corners=True)
            self.conv = DoubleConv(in_channels, out_channels, in_channels // 2)
        else:
            self.up = nn.ConvTranspose2d(in_channels, in_channels // 2, kernel_size=2, stride=2)
            self.conv = DoubleConv(in_channels, out_channels)

    def forward(self, x1, x2):
        x1 = self.up(x1)
        diffY = x2.size()[2] - x1.size()[2]
        diffX = x2.size()[3] - x1.size()[3]

        x1 = F.pad(x1, [diffX // 2, diffX - diffX // 2, diffY // 2, diffY - diffY // 2])
        x = torch.cat([x2, x1], dim=1)
        return self.conv(x)

class OutConv(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(OutConv, self).__init__()
        self.conv = nn.Conv2d(in_channels, out_channels, kernel_size=1)

    def forward(self, x):
        return self.conv(x)

```
