

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

ЗАВРШНИ РАД

**Компаративна анализа алгоритама за
детекцију објеката на проблему
препознавања заштитних маски**

Ментор

Др Павле Милошевић
Доцент

Студент

Жељана Грујић 2018/0045

Београд, 2022. године

АПСТРАКТ

Предмет овог рада је упоредна анализа перформанси алгоритама за детекцију објеката на проблему препознавања заштитних маски. Проблем којим се овај рад бави јесте класификације особа на фотографијама на оне које носе или не носе маску. Потребно је пронаћи вредности за аргументе прослеђене неуронској мрежи и фино подесити хиперпараметре, како би мрежа поред свих околности и ограничења приликом обучавања успела да оствари најбоље резултате. Модели мрежа коришћених у анализи су унапред тренирани на великим скуповима података, преузети су са GitHub-а и искоришћени у експерименту. За потребе експеримента су коришћени YOLOv5 модел, и Detectron2 платформа са које је искоришћена имплементација Faster R-CNN модела.

Како би одговориле на проблем препознавања заштитних маски, претрениране мреже су додатно обучене на скупу података који садржи фотографије на којима се налазе особе са и/или без заштитних маски. Приликом додатног обучавања модела варирана су подешавања вредности аргумената који одређују начин њиховог учења и упоређиване су перформансе. Анализом перформанси одабрани су најефикасније обучени модели, чија способност детекције објеката је тестирана на непознатом скупу података. Најбоље резултате су остварили модел YOLOv5 трениран 100 епоха, који је на скупу података за тестирање достигао 73.30% тачности и Faster R-CNN модел са платформе Detectron2, трениран 2 000 итерација, који је на скупу података за тестирање достигао 70.97% тачности.

Кључне речи: *неуронске мреже, алгоритми за детекцију објеката, анализа перформанси, класификација објеката, заштитна маска*

ABSTRACT

The subject of this paper is a comparative analysis of the performance of object detection algorithms on the problem of recognizing protective masks. The problem that this paper deals with is the classification of persons in photographs into those who wear or do not wear a mask. It is essential to find the values for the arguments passed to the neural network and fine-tune the hyperparameters, so that the network can achieve the best results despite all circumstances and limitations during training. The network models used in the analysis were pre-trained on large datasets, downloaded from GitHub, and used in the experiment. For the purposes of the experiment, the YOLOv5 model and the Detectron2 platform, from which the implementation of the Faster R-CNN model was used.

In order to answer the problem of recognizing protective masks, pretrained networks were additionally trained on a dataset containing photos of people with and/or without protective masks. During the additional training of the models, values of the arguments that determine the way of their learning were varied and the performances were compared. Through performance analysis, the most efficiently trained models were selected and their ability to detect objects was tested on an unknown dataset. The best results were achieved by the YOLOv5 model trained for 100 epochs, that has reached 73.30% accuracy on the testing data set, and the Faster R-CNN model from the Detectron2 platform, trained for 2 000 iterations, that has reached 70.97% accuracy on the testing data set.

Keywords: *neural networks, object detection algorithms, performance analysis, object classification, protective mask*

САДРЖАЈ

1.	УВОД.....	1
2.	ТЕОРИЈСКЕ ОСНОВЕ	3
2.1	ПРЕПОЗНАВАЊЕ ОБЈЕКТА	3
2.2	АРХИТЕКТУРА ДЕТЕКТОРА ЗАСНОВАНИХ НА ДУБОКИМ НЕУРОНСКИМ МРЕЖАМА.....	5
2.3	<i>YOLO</i> модел за детекцију објекта	6
2.3.1	Архитектура <i>YOLO</i> модела.....	8
2.3.1.1	Улазни слој мреже	9
2.3.1.2	Окосница мреже.....	10
2.3.1.3	Врат мреже	10
2.3.1.4	Глава мреже.....	11
2.3.2	Хиперпараметри <i>YOLO</i> модела	13
2.4	<i>R-CNN</i> модел за детекцију објекта	15
2.4.1	<i>Detectron2</i> платформа.....	15
2.4.2	<i>Faster R-CNN</i>	18
2.4.2.1	Окосница мреже.....	19
2.4.2.2	<i>Region Proposal Network</i>	19
2.4.2.3	ROI глава мреже	20
2.4.3	Хиперпараметри <i>Faster R-CNN</i> модела, <i>Detectron2</i> платформе	21
2.5	МЕТРИКЕ ЗА ЕВАЛУАЦИЈУ ПЕРФОРМАНСИ	22
3.	ПРЕГЛЕД ЛИТЕРАТУРЕ.....	26
4.	ЕКСПЕРИМЕНТ.....	29
4.1	ОПИС СКУПА ПОДАТАКА.....	29
4.1.1	Сређивање скупа података	35
4.2	ЕВАЛУАЦИЈА ПЕРФОРМАНСИ МОДЕЛА.....	36
4.2.1	Евалуација перформанси на подскуповима за тренинг и валидацију	37
4.2.1.1	Перформансе: <i>YOLOv5</i> 50/100/400 епоха.....	38
4.2.1.2	Перформансе: <i>Detectron2-Faster R-CNN</i> 1 000/2 000/4 000 итерација	47
4.2.2	Евалуација перформанси на подскупу за тест	54
5.	ЗАКЉУЧАК	58
6.	ЛИТЕРАТУРА	60

1. УВОД

Детекција објеката (енг. *object detection*) је само једна од техника која спада међу области које се баве компјутерским видом (енг. *computer vision*) (fritz.ai, 2021). Компјутерски вид користи технике вештачке интелигенције (енг. *artificial inteligence*) како би омогућио машинама, односно рачунарима да стекну висок ниво раумевања и препознавања битних података са фотографија и видео записа (ibm.com). Конкретно, под појмом детекција објеката се подразумева техника којом рачунар врши лоцирање објекта и одређивање којим класама припадају лоцирани објекти, на фотографијама или видео снимку (mathworks.com). Иако делује да је детекција објеката иновативна технологија, заправо она је присутна још од краја XX века, али услед недоступних и неадекватних фотографија и видео материјала, као и недовољних рачунарских ресурса, тада није стекла велику популарност као што је то случај данас. Почетком XXI века, развојем конволуцијских неуронских мрежа (енг. *convolutional neural network*) и концепта дубоког учења (енг. *deep learning*) детекција објеката добија све већи значај (Chinmoy, 2020).

Данас се технике детекције објеката користе у различитим научним дисциплинама, од маркетинга, преко саобраћаја па чак и до медицине. Своју примену налазе у системима аутономних аутомобила, аутономних робота, пребројавања масе људи, видео надзора, препознавања емоција, детекције аномалија и многим другим системима (fritz.ai, 2021).

Појавом новонастале епидемиолошке ситуације, ради превенције ширења заразних болести и јачања здравственог система, у читавом свету је прописана мера ношења заштитних маски. Како би увођење ове мере имало позитиван утицај на поправљање епидемиолошке ситуације, било је потребно да све особе, било да се налазе у затвореном или отвореном простору, носе неку врсту заштитних маски. Поред тога што је полицији било поверено да проверава да ли се прописана мера спроводи на јавним местима, радници у затвореном простору су такође били у обавези да обезбеде њено спровођење. Нови прописи обавезног ношења заштитних маски доносе са собом проблем како исконтролисати и проверити да ли свака особа носи маску, па тако детекција објеката добија значајну примену и на поменутом проблему.

Овај рад се бави управо анализом алгорита за детекцију објеката, који ће бити обучени да препознају заштитне маске, тачније да ли особа носи или не носи заштитну

маску. Експеримент је спроведен над скупом података који чине фотографије, забележене широм света током тадашње епидемиолошке ситуације.

У другом поглављу овог рада су дате теоријске основе детекције објеката. Укратко је описана историја развоја техника детекције објеката какве су познате данас, наведени су проблеми са којима се суочава детекција објеката и начин њиховог решавања, представљена су теоријска објашњења *YOLOv5* и *Detectron2* алгоритама који су коришћених у експерименту. Такође детаљно је анализирана архитектура мрежа коришћених у експерименту, објашњени су хиперпараметри везани за конкретну мрежу, као и метрике на основу којих се врши евалуација модела.

Треће поглавље садржи сумарни преглед литература која се баве проблемом препознавања заштитних маски.

Четврто поглавље садржи резултате свих симулација које су извршене при обучавању мрежа и покушају решавања проблема. У овом поглављу се налази и детаљан опис скупа података, поступак сређивања скупа података, приказ резултата, и упоредна анализа перформанси најбољих решења.

У петом поглављу је дат закључак које је решење најбоље и који су разлози за то, као и потенцијалне смернице за употребу у будућности.

Шесто поглавље се односи на преглед коришћене литературе.

2. ТЕОРИЈСКЕ ОСНОВЕ

Детекција објеката, као једна од техника компјутерског вида, се бави откривањем инстанци објеката одређене класе на фотографији или видео снимку (Zou et al., 2019). Задатак детекције објеката јесте да обезбеди информације о томе који се објекат где налази, и колико објеката има на фотографији.

2.1 ПРЕПОЗНАВАЊЕ ОБЈЕКТА

Детекцији објеката се може приступити на више начина: један приступ је заснован на машинском учењу (енг. *machine learning-based approach*), а други приступ је заснован на дубоком учењу (енг. *deep learning based approach*).

Приступ заснован на машинском учењу је заправо традиционалнији приступ детекцији објеката који је доминантно био у употреби све до средине друге деценије двадесет првог века, а методе које су се користиле за детекцију објеката заснивале су се на анализи дистрибуције боја на фотографији (енг. *color histogram*) и проналаску оштрих ивица како би се покушала одредити група пиксела која указује на објекат (fritz.ai, 2021). На основу овако добијених информација су вршена предвиђања локације објекта и припадности класи. Овој групи раних детектора објекта припадају *Viola-Jones* детектор, *Hog* детектор и *Deformable Part-based* модел.

Viola-Jones детектор је био заснован на покретном прозору (енг. *sliding window*) који је пролазио кроз све сегменте фотографије и покушавао да пронађе онај део фотографије који садржи објекат (Zou et al., 2019). Иако на изглед веома једноставан процес, иза њега се крио велики број калкулација за које тадашњи рачунари нису били погодни.

Hog детектор је покушавао да издвоји контраст на различитим деловима фотографије, помоћу вектора градијента који се рачунао за сваки угао и пиксел (Kundrotaitė, 2021). Своју примену је пронашао, на пример, у системима аутономних аутомобила и генерално апликацијама за детекцију људи.

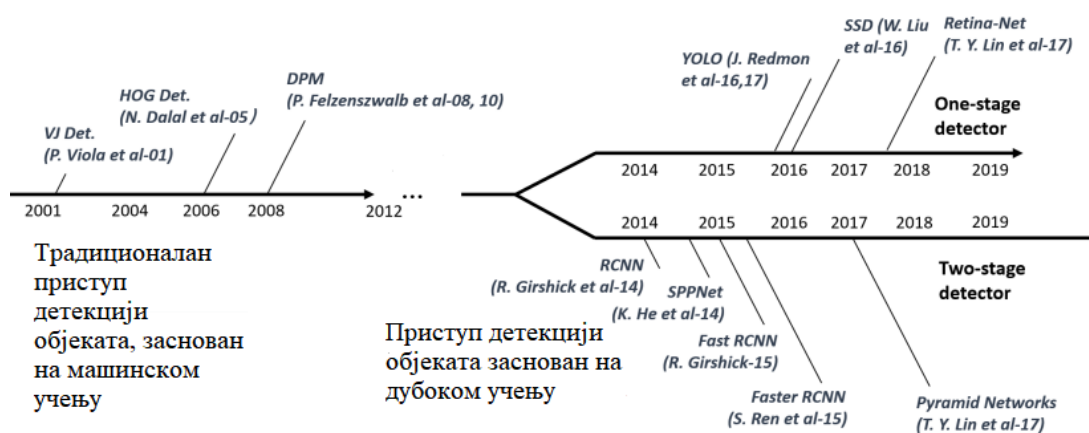
Deformable Part-based модел проблему детекције објеката приступа према принципу „завади па владај“ (енг. *divide and conquer*), па да би се детектовао жељени објекат прво је потребно детектовати све његове битне делове (Zou et al., 2019). Овај детектор је представљао врхунац примене традиционалног приступа детекције објеката.

Појава *AlexNet* неуронске мреже 2012. године, означила је почетак новог приступа детекцији објеката заснованог на дубоком учењу (Chinmoy, 2020). Убрзо након тога је 2014. године предложен низ софистициранијих и прецизнијих мрежа од којих се посебно издваја *Regresions with CNN (RCNN)* као нови детектор. Иновативни *RCNN* модел је био заснован на алгоритму селективне претраге који је предлагао око 2 000 региона од интереса (енг. *region of interest*), за које је постојала вероватноћа да садрже објекат, они су затим прослеђивани до неуронске мреже која је вршила даље анализе и коначну предикцију (Chinmoy, 2020). Иако је побољшана тачност предвиђања, овај модел је био поприлично спор. Као одговор на проблем брзине, настају *Fast R-CNN* и *Faster R-CNN* детектори. *Faster R-CNN* је алгоритам селективне претраге потпуно заменио неуронском мрежом чији је задатак да предвиди регионе од интереса (Chinmoy, 2020).

Нове промене у приступу детекцији објеката на основу дубоког учења доноси појава *You Only Look Once (YOLO)* детектора (Zou et al., 2019). За разлику од *RCNN* детектора и његових наследника, *YOLO* детектор напушта концепт претраге, а затим верификације региона од интереса, и уводи нову филозофију примене неуронске мреже на читаву фотографију. Иако је овим значајно убрзан процес детекције објеката, *YOLO* детектор се ипак суочио са проблемом смањене тачности. Као покушај решавања недостатка *YOLO* детектора, појављује се *Single Shot Detector (SSD)* који успева да обједини и брзину и тачност детекције (Zou et al., 2019). Додатна побољшања доноси појава *RetinaNet* детектора, који у своју архитектуру интегрише *Focal loss* функцију која ригидније кажњава направљене грешке класификације (Zou et al., 2019). На овај начин је додатно повећана тачност детекције, уз задржавање задовољавајуће брзине детекције. Детекција објеката из године у годину постаје све популарнија тема, па се тако развијају још савременија решења, од којих једно обухвата читаво *Facebook-ово* истраживање и креацију *Detectron2* библиотеке.

Детектори објекта се могу поделити на две групе. Прву групу чине једностепени детектори објеката (енг. *single stage object detectors*) који изостављају процес претраге и валидацију региона од интереса, већ директно врши лоцирање и предвиђање објеката. Овој групи детектора припада фамилија *YOLO* детектора и *SSD*. Другу групу чине двостепени детектори, који детекцији објеката приступају кроз две фазе, у првој извлачење региона од интереса затим у другој валидација и класификација објеката. Овој групи припадају *RCNN*, *Fast R-CNN*, *Faster R-CNN*, *Mask R-CNN* детектори.

На Слици 1. је приказан хронолошки развој области детекције објеката.



Слика 1. Еволуција области детекције објеката (Zou et al., 2019)

Предности коришћења детекције објеката се могу искористити у ситуацијама када је потребно детектовати објекте који заузимају већи део фотографије, објекте који су јасно разграничени, приликом детекције групације објеката на једном месту и при брзом одређивању локације оваквих објеката на великом скупу података. Мане са којима се ова област суочава огледају се у ситуацијама када је потребно детектовати издужене објекте као што је оловка, не може се извршити детекција појава које немају конкретан облик и границу, као што су сунчева светлост, магла, небо (Rizzoli, 2022). Упркос свему детекција објеката налази широку примену у препознавању карактера, индентификацији на основу личних докумената, детекцији и препознавању објеката, анализи медицинских феномена, праћењу лопти у спорту, роботизици и многим другим областима (Vahab et al., 2020).

2.2 АРХИТЕКТУРА ДЕТЕКТОРА ЗАСНОВАНИХ НА ДУБОКИМ НЕУРОНСКИМ МРЕЖАМА

Структура алгоритама за детекцију објеката се састоји из два основна дела (Gutta, 2021): први део се назива окосница (енг. *backbone*), а други је познат под називом глава (енг. *head*). У савременијим алгоритмима за детекцију објеката појављује се још један део који се налази између поменутих два, врат (енг. *neck*) (Gutta, 2021).

Окосница мреже представља основни класификациони модел на коме се заснива алгоритам, који игра важну улогу приликом извлачења и агрегирања свих битних карактеристика са улазних фотографија (енг. *feature maps*), које носе значајне информација и које се прослеђују наредним слојевима мреже (Solawetz, 2020).

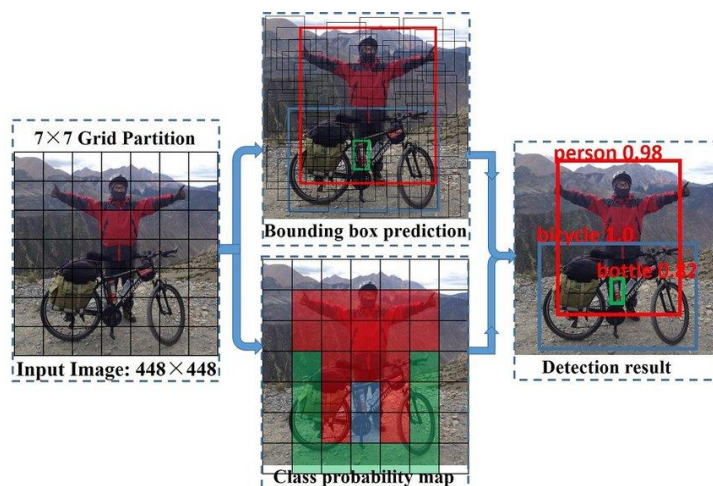
Након што окосница мреже извуче све битне карактеристике са фотографије, оне се прослеђују даље до дела врат мреже. Врат мреже од добијених карактеристика генерише пирамиде карактеристика (енг. *feature pyramids*) помоћу којих модел може боље да генерализује објекте на фотографији, и да боље идентификује исте објекте, различитих величина и размера (Patel, 2020).

Овако обрађене информације се прослеђују до последњег дела, главе мреже, чија је улога да предвиди класу којој објекат треба да припадне и да исцрта правоугаоник око идентификованог објекта (енг. *bounding box*) (Solawetz, 2020).

2.3 YOLO модел за детекцију објеката

You Only Look Once представља породицу јако ефикасних и веома популарних алгоритама за детекцију објеката, који се у великом броју случајева примењују на проблеме детекције објеката у реалном времену. Појава YOLO алгорита 2015. године, променила је начин на који се тада приступало детекцији објеката (Jiang et al., 2021). Фамилија YOLO алгоритама у дотадашњи приступ уводи значајне промене у погледу брзине обављања рачунарских операција током учења модела, по чему они постају препознатљиви. За разлику од *R-CNN* фамилије алгоритама, YOLO алгоритми имају једноставнију архитектуру која им гарантује већу брзину. Процес детекције објеката је упрошћен и своди се на проблем регресије, како да се на основу улазне фотографије директно одреди локација и класа којој објекат припада. Улазна фотографија се дели на $S \times S$ ћелија, где се за сваку ћелију рачуна вектор који носи информације о томе да ли она садржи објекат или не, као и информације о класи којој тај објекат припада. Свака од ћелија је одговорна да детектује објекат и да за њега предложи један или више оквира око објекта од интереса (енг. *bounding box*) који указују на његову локацију, као и вредност поверења (енг. *confidence score*) да тај оквир садржи објекат. Након тога се применом *Non-Maximum Suppression* методе изостављају сви оквири који не садрже објекат, као и они који се међусобно преклапају, остављајући тако најбоље решење од свих понуђених (Redmon et al., 2016).

На Слици 2. се може видети основна идеја функционисања *YOLO* алгоритама.



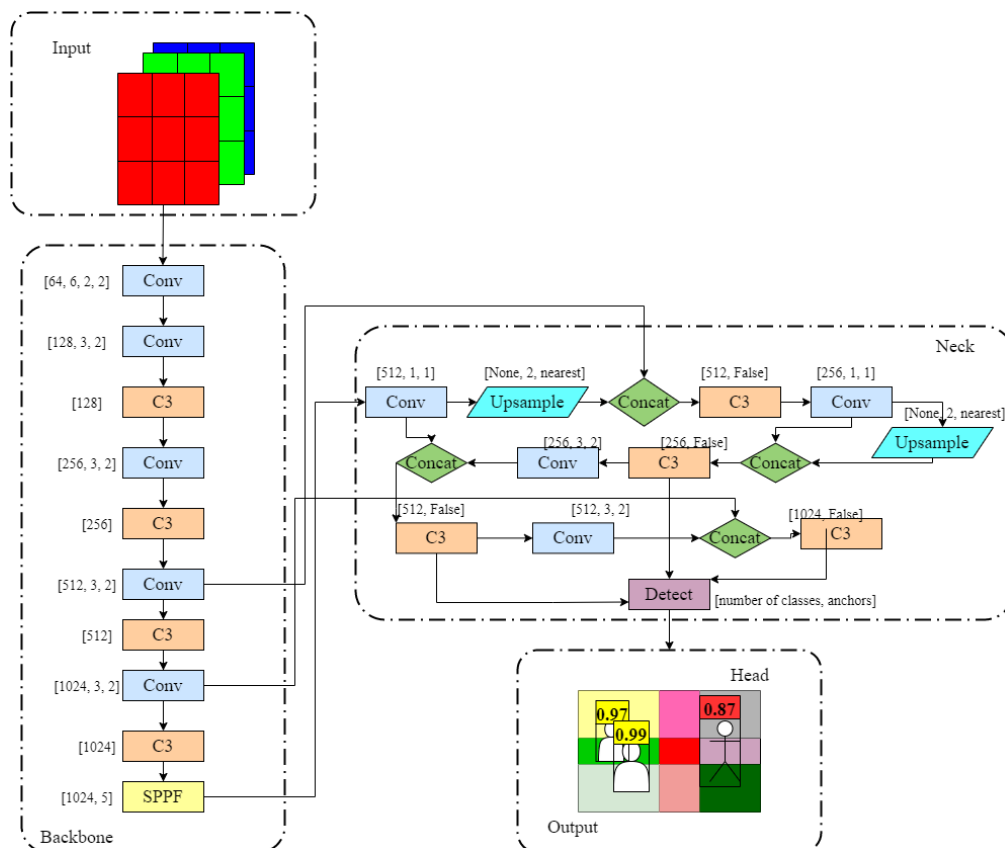
Слика 2. Основна идеја функционисања *YOLO* алгоритама (Wu & Zhou, 2019)

Верзија *YOLOv1* имплементирала је основну идеју овог приступа (Thuan, 2021). Верзија *YOLOv2* доноси нова побољшања прецизности модела применом методе усидрених региона (енг. *anchor box*) за предвиђање оквира. Усидрени региони представљају листу предефинисаних оквира који се највише подударају са циљаним објектом. За груписање сличних оквира коришћен је алгоритам к средњих вредности (енг. *k-means*), као основни класификатор коришћена је *Darknet-19* неуронска мрежа, примењена је и нормализација на нивоу целине (енг. *batch normalization*) метода која додатно повећава прецизност. Верзија *YOLOv3* у своју архитектуру интегрише *Darknet-53* и *ResNet* мрежу, чиме остварује бољу локализацију и детекцију мањих и збијенијих објеката. Верзија *YOLOv4* уводи највише иновација, интеграцијом додатног *Spatial Pyramid Pooling* блока, чији задатак јесте да поспешу детекцију објеката различитих димензија, користи *Mosaic* метод за побољшање улазних података, генерализовани пресек над унијом (енг. *generalized intersection over union, GIoU*) као параметар за *loss* функцију, која указује на настале грешке приликом детекције, и примењује *MISH* активациону функцију (енг. *a self regularized non-monotonic activation function*). Верзија *YOLOv5* се појавила 2020. године и веома је слична претходној верзији, али уводи мање промене и побољшања прецизности и брзине (Thuan, 2021). Имплементирана је помоћу *PyTorch framework-a* чиме је омогућена флексибилност контролисања конфигурације модела, омогућава повећање разноликости улазних података кроз примену подешавања боја, *Mosaic* методе, ротације, и промене величине фотографије. Последње верзије *YOLOv6* и *YOLOv7* су се појавиле 2022. године и са собом доносе изузетну брзину и тачност у поређењу са претходним верзијама. (*models.roboflow.com*, 2019)

2.3.1 Архитектура YOLO модела

YOLOv5 модел је један од новијих верзија детектора, породице *YOLO*. Карактерише га велика брзина и добра прецизност детекције објеката. У оквиру ове верзије постоје четири модела, који се разликују по броју филтера за извлачење и агрегирање карактеристика са фотографије (енг. *feature extraction modules*), по брзини, броју параметара и величини модела (Song et al., 2021). Називи поменутих модела су: *YOLOv5x* (енг. *YOLOv5 extended*), *YOLOv5l* (енг. *YOLOv5 large*), *YOLOv5m* (енг. *YOLOv5 medium*), *YOLOv5s* (енг. *YOLOv5 small*). За потребе експеримента је коришћен модел *YOLOv5s* због мале величине модела, способности веома брзог учења и задовољавајуће прецизности приликом детекције објеката.

Архитектура модела *YOLOv5* фамилије се састоји из четири дела: улазни слој (енг. *input layer*), окосница мреже, врат мреже и глава мреже. На Слици 3. је приказана архитектура *YOLOv5s* мреже нацртана према структури модела дефинисаној у *yolov5s.yaml* фајлу.



Слика 3. Архитектура YOLOv5s мреже

У Табели 1. је приказана структура *YOLOv5s* модела, а у наставку следи опис сваког дела.

Табела 1. Детаљан приказ структуре *YOLOv5s* модела

<i>YOLOv5s</i>	Модули
Улазни слој	<i>Mosaic data augmentation, adaptive anchor, adaptive image scaling</i>
Окосница мреже (CSPDarknet)	<i>CSP1_X, CBL Conv-BN-Leaky_relu, Res unit, Convolutional layer, Concatenate function, SPP</i>
Врат мреже (PANet)	<i>FPNet, CSP2_X</i>
Глава мреже	<i>Loss function, Non-Maximus Suppression, GIoU</i>

2.3.1.1 Улазни слој мреже

Улазни слој *YOLOv5s* модела примењује метод Мозаичке аугментације података (енг. *Mosaic data augmentation*), метод адаптивних усидрења (енг. *adaptive anchor*) и адаптивно скалирање слика (енг. *adaptive image scaling*) (Guo et al., 2022).

Метод Мозаичке аугментације података је један од начина да се повећа робусност модела кроз улазне податке. Мозаичка метода спаја четири насумично изабране фотографије у једну, како би обогатила улазне податке новим случајевима и на тај начин повећала способност мреже да детектује и мање објекте на фотографији (Guo et al., 2022).

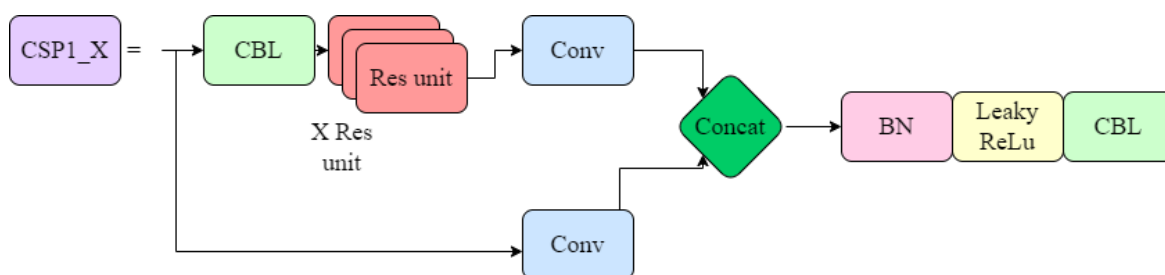
Метод адаптивних усидрења омогућава генерисање великог броја помоћних усидрених региона приликом тренирања мреже, а затим ће од њих, при предвиђању, бити изабран онај који је најближи стварном оквиру (енг. *ground truth*) који је већ исцртан око објекта (Guo et al., 2022). Оквир који је ручно исцртан око објекта, приликом аотирања фотографија, назива се стварни оквир (енг. *ground truth box*), док се оквир који мрежа предвиди и исцрта назива очекивани оквир (енг. *bounding box*).

Применом методе адаптивног скалирања слика, фотографије неуједначених величина се скалирају на одговарајућу стандардну величину пре него што се пошаљу као улаз, ка мрежи (Guo et al., 2022).

2.3.1.2 Окосница мреже

YOLOv5 детектори обједињују *CSPNet* (енг. *cross stage partial network*) и *Darknet*, стварајући тако *CSPDarknet* која представља окосницу алгоритма. *CSPNet* смањује количину прорачуна и информација о промени градијента, трансформишући их у мапу карактеристика одговарајућих димензија (Qu, 2021). На овај начин се смањује број параметара и број операција, са покретним зарезом, у секунди (енг. *FLOPS*), и тако се постиже већа брзина и тачност модела, као и смањење величине модела.

Детаљније, *YOLOv5s* модел се састоји из два типа, веома сличних, *CSP* структура, *CSP1_X* која је у саставу окоснице мреже, и *CSP2_X* у саставу врата мреже. *CSP1_X* модул се састоји из једног *CBL* (енг. *case-based learning*) блока, затим садржи *Conv-BN-Leaky_relu* активациону функцију (енг. *convolutional layer-batch normalization-leaky rectified linear unit*), *Res* јединицу која представља модул који служи за изградњу *ResNet* дубоке мреже и конволуциони слоји (енг. *convolutional*) (Qu, 2021). На Слици 4. је приказан *CSP1_X* модул.



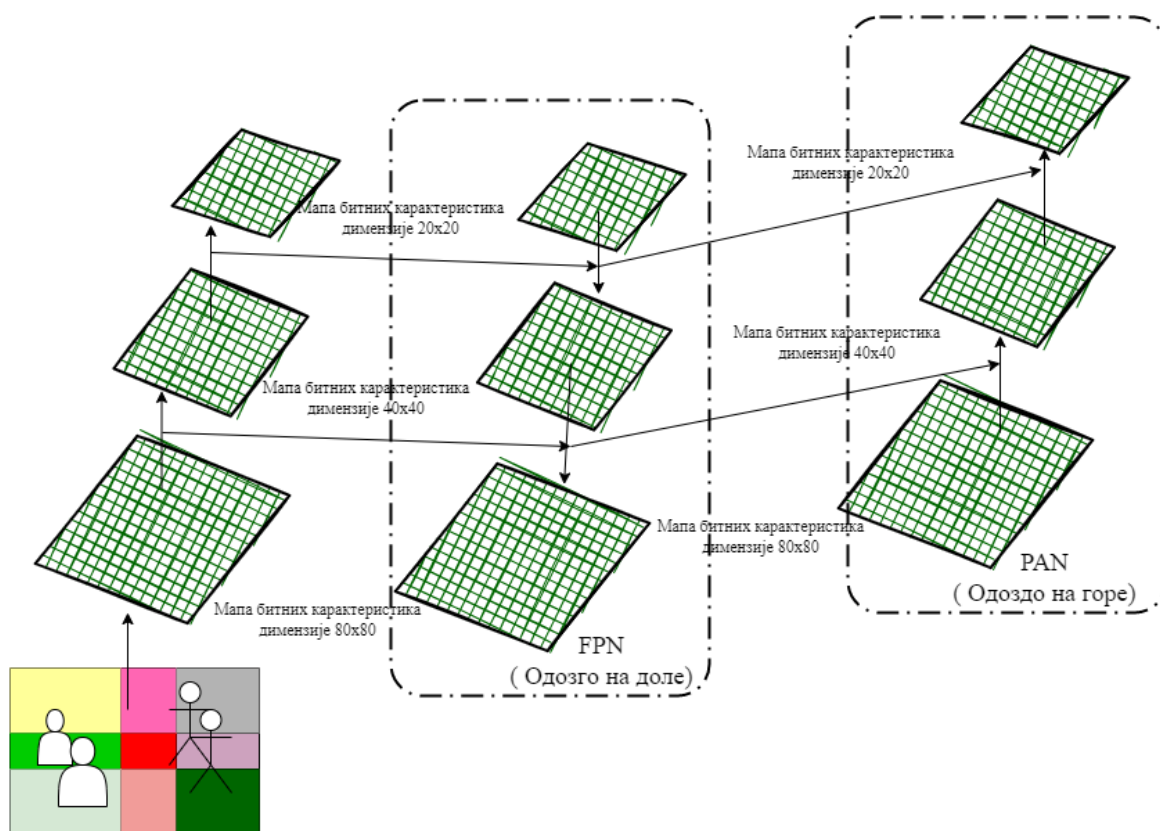
Слика 4. *CSP1_X* модул

Поред тога, окосница мреже *YOLOv5s* модела користи и *SPP* (енг. *spatial pyramid pooling*) структуру која претвара мапе карактеристика (енг. *feature maps*) произвољне величине, у векторе карактеристика фиксне величине, без губљења информација и нарушавања тачности модела (Guo et al., 2022).

2.3.1.3 Врат мреже

Врат мреже за генерисање пирамида карактеристика користи различите технике, а у *YOLOv5s* моделу се користи *FPNet* (енг. *feature pyramid network*) која, кроз слојеве, преноси информације о карактеристикама одозго на доле, и *PANet* (енг. *path aggregation network*) која информације о циљаним објектима на фотографији преноси одоздо на горе (Guo et al., 2022).

Пирамиде карактеристика су јако корисне јер помажу моделу да ефикасно ради, и да оствари добре перформансе на подацима које никада није видео. На Слици 5. су приказане технике *FPNet* и *PANet*.



Слика 5. Пирамиде карактеристика генерисане коришћењем *FPNet* и *PANet*

CSP2_X структура врата мреже умеће два *CBL* блока која омогућавају мрежи да смањи број прорачуна, а да при томе задржи што више информација о битним карактеристикама извученим са фотографије (Guo et al., 2022).

2.3.1.4 Глава мреже

Овај део модела, на основу свих добијених информација, обавља завршну класификацију, исцртава оквире око објеката и исписује вероватноћу припадности предвиђеној класи (Guo et al., 2022).

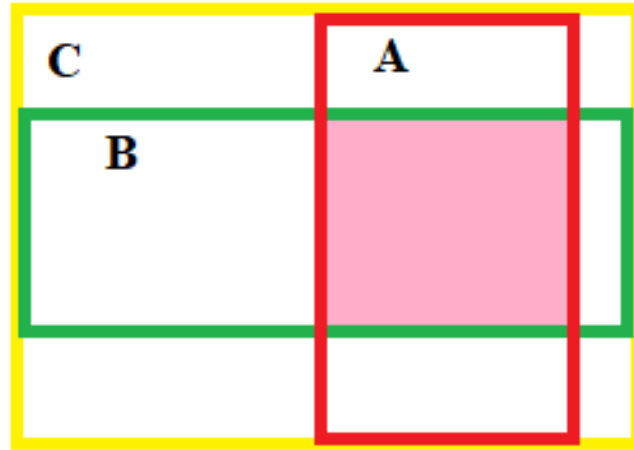
У делу глава мреже се рачуна *loss* функција и примењује *Non-Maximum Suppression* техника. Функција губитка (енг. *loss function*) се састоји из три дела: *bounding box regression loss*, *confidence loss* и *classification loss*. *Bounding box regression loss* који указује на грешку између предефинисаног *stvarnog* оквира у анотацији, и очекиваног оквира којег модел предвиди и исцрта око објекта, *confidence loss (the objectness of the box)* указује на грешку где модел предвиди и исцрта оквир који не садржи ни један

објекат, *classification loss* указује на грешку предвиђене вероватноће припадности класи (Guo et al., 2022).

$$LOSS = L_{GIoU} + L_{conf} + L_{class} \quad (1)$$

где је $LOSS$ укупна вредност функције, L_{GIoU} се односи на *bounding box regression loss*, L_{conf} се односи на *confidence loss*, L_{class} се односи на *classification loss*.

Bounding box regression loss се израчунава на основу $GIoU$ параметра, који поред тога што приликом прорачуна, узима у обзир величину подручја преклапања између предвиђеног оквира и анотацијом предефинисаног стварног оквира, већ узима у обзир и величину подручја која се не преклапају (Song et al., 2021). На Слици 6. је приказан пример преко кога се може израчунати $GIoU$.



Слика 6. Пример за израчунавање вредности $GIoU$ (Song et al., 2021)

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \setminus (A \cup B)|}{|C|} = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \quad (2)$$

где IoU представља стандардну метрику за прецизност детекције објеката, која показује у којој мери се преклапају анотацијом предефинисан оквир и моделом предвиђен оквир. Вредности које може узети IoU налазе се у интервалу $[0,1]$. IoU не узима у обзир случај када се оквири не преклапају, или када су орјентисани у различитим смеровима, стога се, као свеобухватнији показатељ за евалуацију прецизности модела, користи $GIoU$. Он може узети вредности из интервала $[-1,1]$ (<https://patrickllgc.github.io/>, 2019).

Non-Maximum Suppression је техника која има за циљ одабир најбољег оквира, од свих генерисаних *anchor* оквира који служе као помоћ за предикцију коначног оквира (Subramanyam, 2021).

2.3.2 Хиперпараметри *YOLO* модела

Хиперпараметри представљају врсту параметара чије вредности диктирају цео процес дубоког учења (енг. *deep learning*) модела (Munawar, 2022). Такође, предефинисане вредности хиперпараметара ће утицати на то које ће тежине бити подешене параметрима мреже, приликом процеса учења. *YOLOv5s* модел има 29 хиперпараметара, који се могу подешавати и тако утицати на сам процес тренирања мреже. Боља оптимизација хиперпараметара утиче на то да ће тежинске вредности мреже бити боље подешене, а самим тим ће модел бити ефикаснији. Код *YOLOv5* модела, као оптимизациона функција се може изабрати или алгоритам стохастичког градијентног спуста (енг. *stochastic gradient descent, SGD*) или *Adam* оптимизатор. Разлика између њих је то што *SGD* оптимизатор одржава једну стопу учења која се не мења током тренинга, док *Adam* оптимизатор прилагођава стопу учења током тренинга. Код *YOLOv5s* модела је подразумевано постављен *SGD* оптимизатор, али се по жељи може променити. У *hyp.scratch-low.yaml* фајлу се налазе сви хиперпараметри, са предефинисаним вредностима, дефинисаних приликом учења мреже на јавном *COCO* скупу података. У Табели 2. су објашњени сви хиперпараметри чија оптимизација може довести до побољшања резултата модела (Munawar, 2022).

Табела 2. Хиперпараметри модела

Назив хиперпараметра	Опис
<i>Lr0 (Learning-rate start)</i>	Односи се на почетну брзину учења. Постављена вредност износи 0.01 што значи да приликом сваке итерације током тренирања, мрежа ће направити прогрес већи за 0.01.
<i>Lrf (Learning-rate end)</i>	Кључан параметар за проверу следећег услова: <i>If(current_learning_rate >= learning_rate_end): stop_training else continue_training</i>
<i>Momentum</i>	Веома популарна техника која се користи заједно са <i>SGD</i> оптимизатором. Уместо да користи градијент тренутне итерације, <i>momentum</i> техником се агрегирају сви градијенти прошлих итерација, како би се одредио правац којим треба наставити.
<i>Weight-decay</i>	Регуларизациона техника која спречава модел да преучи или да не научи уопште. Функционише тако што се <i>loss</i> функцији додаје казнени члан (енг. <i>penalty term</i>) који утиче на смањење тежина током процеса пропагације уназад (енг. <i>backpropagation</i>). Постављена вредност износи 0.0005
<i>Box (box loss gain)</i>	Пондер који се додаје <i>bounding box regression loss-у</i> , постављен је на вредност 0.05
<i>Cls (classification loss gain)</i>	Пондер који се додаје <i>classification loss -у</i> , постављен је на вредност 0.5
<i>Obj (the objectness of the box gain)</i>	Пондер који се додаје <i>classification loss -у</i> , постављен је на вредност 1, и скалира се према пикселима
<i>IoU_t</i>	Граница постављена за <i>IoU</i> параметар. Предефинисана вредност је 0.20
<i>Degree</i>	Користи се за ротацију насумично изабраних фотографија. Фотографије се могу ротирати чак и за 360 степени. Иницирана вредност је 0.0
<i>Scale</i>	Користи се за промену величине улазних фотографија, како би постале одговарајуће за улаз у мрежу, и како би се постигли бољи резултати
<i>Flipud</i>	Техника којом се, насумично изабране фотографије из скупа података, ротирају на горе или на доле, како би се побољшала способност предвиђања. Хиперпараметар је изражен кроз вероватноћу, а иницирана вредност је 0.0
<i>Fliplr</i>	Техника којом се, насумично изабране фотографије из скупа података, ротирају на лево или на десно, како би се побољшала способност предвиђања. Хиперпараметар је изражен кроз вероватноћу, а иницирана вредност је 0.5
<i>Mosaic</i>	Користи се како би повећао тачност предвиђања модела, тако што од постојећих фотографија, прави колаж, и на тако добијеним новим фотографијама тренира модел. Хиперпараметар је изражен кроз вероватноћу, а иницирана вредност је 1.0

2.4 *R-CNN* модел за детекцију објеката

Кључни концепт на основу кога функционише фамилија *Region-Based Convolutional Neural Network (R C-NN)* алгоритама заснива се на предлагању региона од интереса који потенцијално садрже жељени објекат (Chingis, 2021). Први представник ове фамилије јесте сам *R-CNN* алгоритам по коме фамилија носи име. *R-CNN* користи алгоритам селективне претраге који пролази кроз сваки сегмент фотографије и покушава да на основу боје, ивица, величине и облика пиксела утврди да ли може тај сегмент да сврста у групу сегмената који чине један конкретан објекат. Користећи овај принцип, алгоритам селективне претраге генерише 2 000 предложених региона од интереса по фотографији, затим се свим регионима своди величина на исте димензије, тако да они попримају облик квадрата, а затим се прослеђују до конволутивне неуронске мреже. Задатак неуронске мреже јесте да на основу прослеђених информација предвиди и исцрта оквире, а затим се применом *SVM* (енг. *support vector machine*) алгоритма врши класификација региона у зависности да ли је ту присутан објекат одређене класе или не (Chingis, 2021). Недостаци *R-CNN* алгоритма као што су веома дуг период тренирања мреже, велики број прорачуна и чињеница да алгоритам селективне претраге није способан да учи, стога ни да се прилагођава различитим варијацијама, решени су развојем нове верзије алгоритма, назване *Fast R-CNN*. Уместо да се 2 000 предложених региона, по фотографији, прослеђује до мреже, у новој верзији се читава слика прослеђује мрежи, а затим се са ње извлачи мапа битних карактеристика, на основу које се генеришу региони од интереса (Ananth, 2019). Приликом генерисања региона од интереса и *R-CNN* и *Fast R-CNN* користе *SVM* алгоритам. Нова верзија ове фамилије, *Faster R-CNN* алгоритам, уједно и најпопуларнији представник ове фамилије, потпуно елиминира употребу *SVM* алгоритма, а уместо њега унеће засебну неуронску мрежу звану *Region Proposal Network (RPN)*, која ће бити способна да научи да ли је објекат присутан на одређеној локацији и његову величину (Ananth, 2019). Ово се реализује генерисањем усидрених региона различите величине и ротације који указују на потенцијални објекат на тој локацији.

2.4.1 *Detectron2* платформа

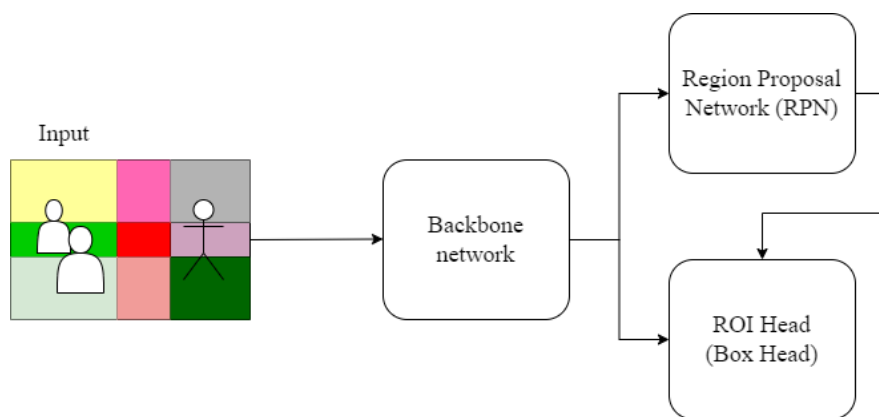
Detectron2 платформа представља најновију технологију развијену од стране *Facebook AI Research* групе (*ai.facebook.com*, 2019). Настао је као побољшање *Detectron* модела, креираног 2018. године са циљем испуњавања нових корисничких и индустријских захтева са којима се сусрео *Facebook*. Почетне идеје истраживачког одељења су веома

брзо реализоване у производ, па тако *Detectron2* налази своју примену у системима уређаја за видео позиве *Facebook* портала, под називом *Smart Camera*. *Detectron2* верзија је за разлику од *Detectron-a* који је написан помоћу *Caffe2*, имплементиран помоћу *PyTorch-a* што му омогућава додатну флексибилност, проширивост и модуларан дизајн који оставља простора за кастомизацију модела према потребама пројекта (ai.facebook.com, 2019). *Detectron2* се може сматрати библиотеком која обједињује све моделе који су били доступни у оригиналном *Detectron-u* као што су *Faster R-CNN*, *Mask R-CNN*, *RetinaNet*, *DensePose*, али такође садржи и имплементације најсавременијих алгоритама за детекцију објеката укључујући *Cascade R-CNN*, *Panoptic FPN*, *TensorMask*. Поред детекције објеката, *Detectron2* подржава низ задатака као што су семантичка сегментација (енг. *semantic segmentation*), детекција кључних тачака (енг. *key point detection*) и паноптична сегментација (енг. *panoptic segmentation*) (ai.facebook.com, 2019).

За обучавања мреже на претходно описаном скупу података, биће коришћен већ научени модел мреже. Модел је претходно обучен на јавном *COCO* (енг. *common objects in context*) скупу података, и биће додатно научен и модификован тако да детектује објекте конкретног скупа података (енг. *transfer learning*). Веома битан корак јесте избор основног модела са *Detectron2* платформе, који треба да укаже који претходно обучени модел треба преузети и користити. За основу модела *Detectron2* се може поставити било који од наведених детектора: *RetinaNet*, *R-CNN*, *Faster R-CNN*, *Mask R-CNN* и многи други од понуђених опција¹. У овом експерименту је за основни модел са платформе *Detectron2* изабрана *faster_rcnn_X_101_32x8d_FPN_3x* мрежа са претренираним тежинама.

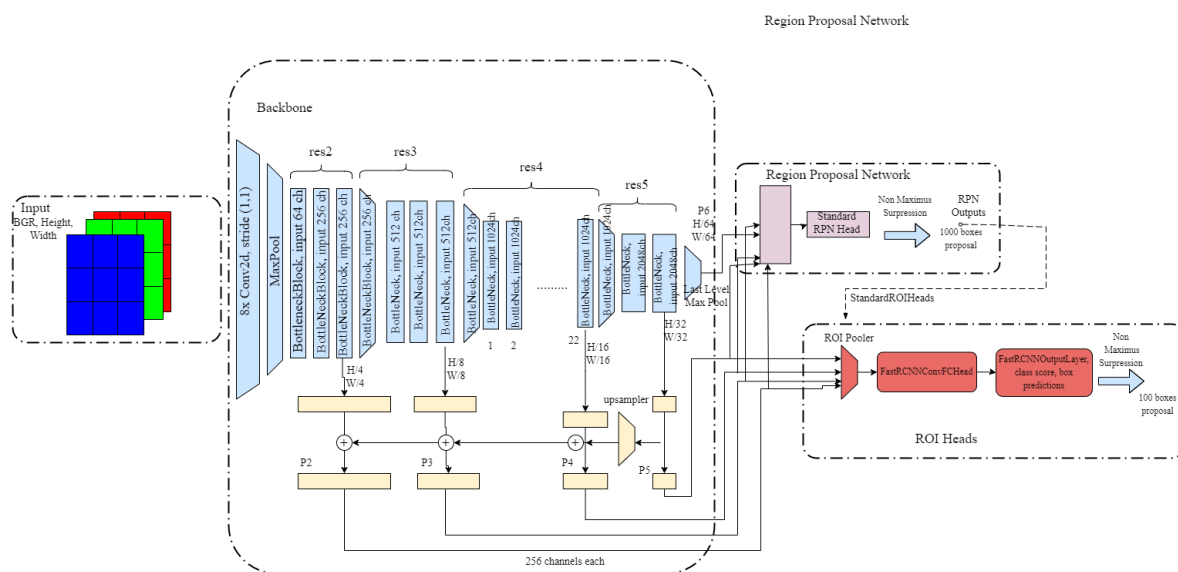
Архитектура *Faster R-CNN* мреже која је изабрана за основу *Detectron2* библиотеке, се састоји из три дела: окоснице мреже, *RPN* (енг. *region proposal network*), и *ROI* глава (енг. *box head*) дела. Апстраховани приказ сржи *Detectron2* библиотеке након поставке *Faster R-CNN* детектора за основни модел је приказан на Слици 7.

¹ Детектори који се могу одабрати са *Detectron2* платформе, налазе се на страници github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#coco-object-detection-baselines



Слика 7. Упрошћени приказ сржи *Detectron2* са *Faster R-CNN* основом

На Слици 8. се може видети детаљни приказ архитектуре мреже изабране са *Detectron2* платформе, нацртан према конфигурацији коришћеног модела. За основу је са *Detectron2* платформе изабран *Faster R-CNN* детектор са *FPN* окосницом мреже, јер остварује високу прецизност приликом детекције како ситнијих тако и већих објеката на фотографији.



Слика 8. Детаљни приказ *Faster R-CNN* детектора

У Табели 3. је такође дат приказ архитектуре *Faster R-CNN* детектора, са *Detectron2* платформе, а у наставку следи опис сваког дела.

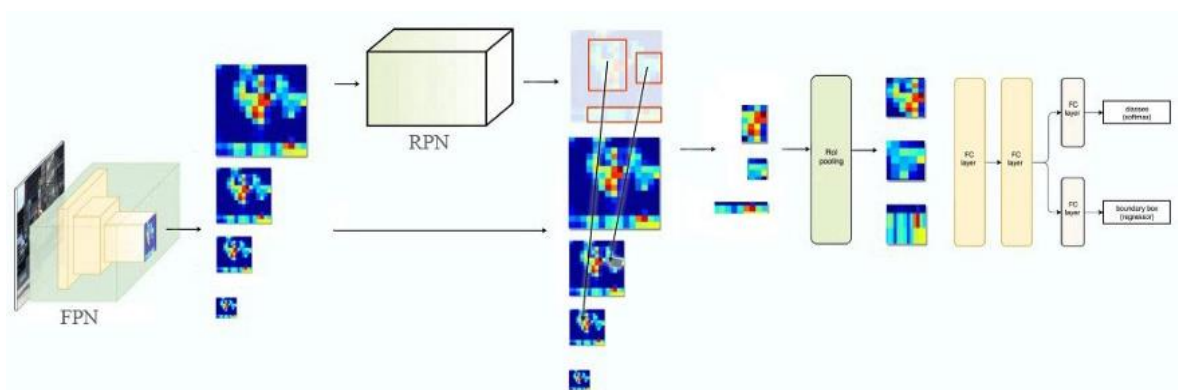
Табела 3. Архитектура *Faster R-CNN* детектора, *Detectron2* платформе

<i>Detectron2</i> платформа	Модули
Окосница мреже (<i>Faster R-CNN</i> детектор са <i>FPN</i> окосницом)	<i>FPN</i> -> <i>ResNet</i> -> - <i>BasicSteam</i> - <i>BottleneckBlock</i> - <i>LastLevelMaxPool</i>
<i>RegionProposalNetwork</i>	<i>RPN</i> -> - <i>StandardRPNHead</i> - <i>RPNOutput</i>
<i>ROI</i> глава	<i>StandardROIHeads</i> -> - <i>ROI Pooler</i> - <i>FastRCNNConvFCHead</i> - <i>FastRCNNOutputLayers</i> - <i>FastRCNNOutputs</i>

2.4.2 *Faster R-CNN*

Faster R-CNN алгоритам припада фамилији *R-CNN* алгоритама настао са циљем побољшања перформанси свог претходника *Fast R-CNN* алгоритама (Nguyen et al., 2018). Процес детекције објеката *Faster R-CNN* алгоритама своди се на два најважнија корака, предвиђање региона од интереса на фотографији, а затим одабир најрелевантнијих региона и класификација објеката. Након што се улазна фотографија проследи мрежи, помоћу *RPN* мреже се предлаже око 1 000 и више усидрених региона и за сваки од њих се одређује да ли садржи објекат или не. Затим се примењује *RoI Pooling* метода која предложене усидрене регионе који садрже објекат трансформише тако да буду исте величине, затим се на основу њих врши филтрирање предложених кандидата и одређивање класе којој лоцирани објекат припада (Nguyen et al., 2018).

На Слици 9. се може видети основна идеја функционисања *Faster R-CNN* алгоритама.



Слика 9. Основна идеја функционисања *Faster R-CNN* алгоритама (Chingis, 2021)

2.4.2.1 Окосница мреже

Окосница модела, изабраног са *Detecreon2* платформе, тачније *Base-RCNN-FPN* мрежа (Honda, 2022) са улазне фотографије извлачи мапу битних карактеристика и скалира је на различите величине. Улаз у саму мрежу јесте фотографија, преференцијално у *BGR* формату, у супротном ако фотографија остане у изворном *RGB* формату, прецизност модела може опати. Приликом процеса извлачења битних карактеристика, на фотографију се примењују различити филтери. Филтери се померају за одређени број корака (енг. *stride*) и тако клизе дуж целе фотографије. Излаз из окоснице мреже јесу четири тензора $P2$, $P3$, $P4$ и $P5$, а они се управо разликују у величини корака за које се филтер помера дуж слике. Величине корака код излаза $P2$ износи четири, што би значило да се филтер у сваком кораку помера за по четири пиксела. Величине корака који су примењени на излазе су $P3$, $P4$, и $P5$ су 8, 16 и 32. *BasicSteam* блок служи како би компресовао битне информације мапа карактеристика, а *Bottleneck* блок како би смањио број сложених калкулација и број параметара модела. Излаз $P5$ се прослеђује до *LastLevelMaxPool* слоја, који мења димензију тензора, која претходно износи $1/32$, на димензију $1/64$. У финалној листи резултата, добијених из окоснице, се налазе тензори $P2$, $P3$, $P4$, $P5$ и $P6$, који носе информације о битним карактеристикама фотографије, који се прослеђују наредним деловима мреже. Разлог зашто се креирају мапе карактеристика различитих димензија, тачније тензори $P2$ - $P6$ јесте како би се мрежа оспособила да добро детектује, како веће објекте тако и оне мање (Honda, 2022).

2.4.2.2 Region Proposal Network

Region Proposal Network (RPN) представља главну компоненту *Faster R-CNN* детектора (Honda, 2022). Ова мрежа повезује мапе карактеристика, прослеђених од стране окоснице мреже, у виду тензора различитих димензија: $P2$, $P3$, $P4$, $P5$, $P6$ са информацијама о ширини, дужини и локацији стварних оквира који означавају одређени објекат, добијених из анотација. *RPN* се састоји из *RPN* главе мреже која процесира мапе карактеристика добијене из *FPN* мреже. Сваки од излаза од $P2$ до $P6$ се прослеђује до *RPN* главе мреже, која за сваку улазну фотографију креира мапу вероватноћа постојања објекта (енг. *probability map of object existence*) на конкретном делу фотографије и претпоставља релативни облик *anchor* оквира који ће се исцртати око тог објекта (Honda, 2022).

Генератор усидрења (енг. *anchor generator*) *RPN* мреже генерише велики број помоћних усидрених региона око циљаног објекта, од којих ће коначно бити изабран онај који је најприближнији стварном оквиру. За сваки предложени усидрени регион и стварни оквир се рачуна IoU (енг. *intersection over union*) и бирају се они усидрени региони који имају највећи степен преклапања са постојећим стварним оквиром (Honda, 2022). Како би се квантитативно изразила разлика између предвиђеног и стварног оквира рачуна се функција губитка. *RPN* мрежа рачуна две функције губитка: *localization loss* (*L1 loss function*) и *objectness loss*. *Localization loss* се може израчунати помоћу следеће формуле:

$$L1_{loss_{function}} = \sum_1^n |y_{true} - y_{predicted}| \quad (3)$$

Применом *Non-Maximum Suppression* технике, од свих предложених усидрених региона бира се 1 000 најбољих, које се прослеђују као улаз у наредни део мреже (Honda, 2022).

2.4.2.3 ROI глава мреже

Мапе карактеристика *P2-P5*, које су излаз из *FPN* мреже, најбољи предложени усидрени региони и стварни оквири из анотација чине улазне податке у део мреже *ROI* глава. На основу предложених усидрених региона и мапа карактеристика одређују се региони од интереса на које мрежа треба да се фокусира. Након тога се примењује *ROI pooling* процес како би се одређени региони од интереса свели на исту величину, опсецањем. У циљу прецизног опсецања *ROI* области са фотографије користи се метода *ROIAlignV2*. Након *ROI pooling* процеса, добијени исечци се прослеђују *Box Head* делу мреже. *Base R-CNN FPN* детектор који је изабран за основу модела, има *FastRCNNConvFCHead box head*, који класификује објекте у оквиру датог региона од интереса и фино подешава позицију и облик оквира. За крајњи излаз се рачуна функција губитка, тачније рачуна се *localization loss* (*L1 loss function*) и *classification loss* (*softmax loss*). Укупна вредност функције губитка се добија када се саберу вредности за *objectness loss* и *localization loss* *RPN* дела, као и *localization loss* и *classification loss* *ROI Head* дела. Како би се уклонили оквири који се преклапају и који су сувишни, и овде се примењује *Non-Maximum Suppression* техника (Honda, 2022).

2.4.3 Хиперпараметри *Faster R-CNN* модела, *Detectron2* платформе

Оптимизација хиперпараметара мреже је од велике важности како би модел био ефикаснији. Као оптимизациона функција *Faster R-CNN* модела је подразумеван *SGD* оптимизатор. У Табели 4 су објашњени хиперпараметри чија оптимизација може довести до побољшања резултата модела.

Табела 4. Хиперпараметри модела

Назив хиперпараметра	Опис
<i>Cfg.MODEL.WEIGHTS</i>	COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml, основни модел претходно научен на <i>COCO</i> скупу података
<i>cfg.SOLVER.IMS_PER_BATCH</i>	Број фотографија у једној целини, постављена вредност је четири фотографије у једној целини
<i>Cfg.SOLVER.BASE_LR</i>	Основна стопа учења, иницијална вредност износи 0.001
<i>Cfg.SOLVER.WARMUP_ITERS</i>	Стопа учења се полако повећава кроз специфицираних <i>N</i> итерација, пре него што достигне константну вредност <i>BASE_LR</i> . Постављена вредност за <i>N</i> износи 1000 итерација.
<i>Cfg.SOLVER.MAX_ITER</i>	Максимални број итерација за тренинг, почетна вредност се одређује по жељи
<i>Cfg.SOLVER.GAMMA</i>	Вредност за коју се <i>Learning Rate</i> параметар смањује приликом процеса учења модела. Износи 0.05
<i>Cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE</i>	Број региона од интереса (<i>ROI</i>) по фотографији. Постављена вредност је 64
<i>Cfg.MODEL.ROI_HEADS.NUM_CLASSES</i>	Број класа конкретног скупа података +1, почетна вредност 2+1=3
<i>Cfg.TEST.EVAL_PERIOD</i>	Број итерација након којих се врши евалуација модела, постављена вредност износи 500 итерација
<i>Cfg.MODEL.ANCHOR_GENERATOR.SIZE</i>	Димензије усидрених региона који се генеришу Вредности: 32, 64, 128, 256, 512
<i>Cfg.MODEL.ANCHOR_GENERATOR.ASPECTS_RATIOS</i>	Рацио према коме ће се исцртати различити облици усидрених региона, скалирано према димензијама <i>P2-P6</i> излаза. Вредности: 0.5, 1.0, 2.0

2.5 МЕТРИКЕ ЗА ЕВАЛУАЦИЈУ ПЕРФОРМАНСИ

Упоредна анализа и евалуација перформанси оба модела биће извршена на основу следећих метрика: тачност (енг. *accuracy*), прецизност (енг. *precision*), средња просечна прецизност (енг. *mean average precision, mAP*) и одзив (енг. *recall*). Поред наведених метрика за обе мреже ће бити забележено време утрошено за тренинг модела.

Метрике *YOLOv5* модела које су коришћене у раду (Dubey, 2020):

- *mAP* (енг. *mean average precision*) - Најпопуларнија метрика за мерење тачности модела. Рачуна се тако што се за сваки оквир који се исцрта на фотографији исписује вероватноћа којом он предвиђа циљани објекат. Затим се према граници *IoU_t* која износи 0.5, одређује да ли тај оквир садржи објекат посматране класе или не, разврставајући тако *TP* (енг. *true positive*) оквир, који садржи објекат и *FP* (енг. *false positive*) оквир који не садржи објекат. На Слици 10. су приказани постојећи стварни оквири, предвиђени усидрени региони и начин на који се рачуна *IoU*.



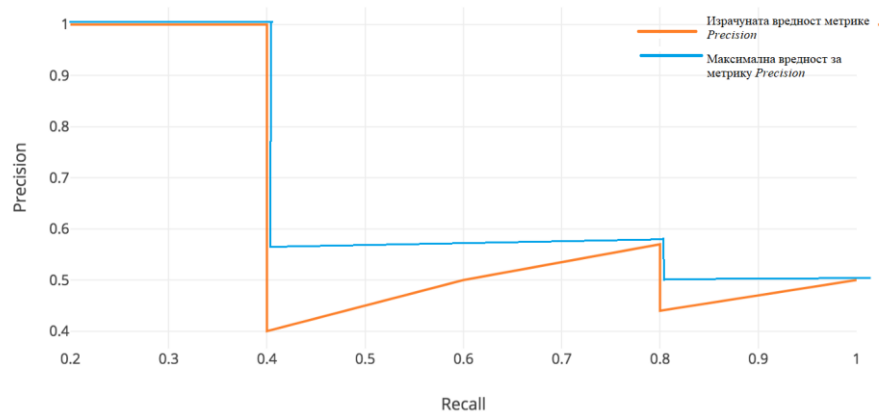
Слика 10. Рачунање *IoU*, означени стварни оквири и предвиђени усидрени региони

Након тога се, у зависности од броја *TP* и *FP* инстанци рачунају метрике прецизност и одзив. Прецизност пропорционално зависи од броја предвиђених *TP* оквира, док одзив пропорционално зависи од броја предвиђених *TP* у односу на укупан број стварних оквира. У Табели 5. је показан начин рачунања метрика прецизност и одзив, према примеру са Слике 10.

Табела 5. Израчунавање метрика прецизност и одзив

Ранг	Тачно предвиђен оквир?	Прецизност	Одзив
1	Тачно	1.00	$\frac{1}{5} = 0.20$
2	Тачно	1.00	$\frac{2}{5} = 0.50$
3	Нетачно	$\frac{2}{3} = 0.67$	$\frac{2}{5} = 0.40$
4	Нетачно	$\frac{2}{4} = 0.50$	$\frac{2}{5} = 0.40$
5	Нетачно	$\frac{2}{5} = 0.40$	$\frac{2}{5} = 0.40$
6	Нетачно	$\frac{3}{6} = 0.50$	$\frac{3}{5} = 0.60$
7	Тачно	$\frac{4}{7} = 0.67$	$\frac{4}{5} = 0.80$
8	Нетачно	$\frac{4}{8} = 0.50$	$\frac{4}{5} = 0.80$
9	Нетачно	$\frac{4}{9} = 0.44$	$\frac{4}{5} = 0.80$
10	Тачно	$\frac{5}{10} = 0.50$	$\frac{5}{5} = 1.00$

- Вредности за прецизност и одзив се затим нацртају на графику, где се на x оси наносе вредности за одзив, а на y оси вредности за прецизност. Након тога се рачуна површина испод прецизност-одзив (енг. *Precision-Recall*) криве, а вредност која се добије предстаља просечну прецизност (енг. *average precision*, *AP*) дате класе. На Слици 11. је нацртана Прецизност-Одзив крива за добијене вредности из Табеле 5.



Слика 11. Прецизност-Одзив крива

Овај поступак је потребно поновити за сваку фотографију и за сваку класу, након чега се добијају укупне AP вредности за сваку класу. Како би се добила вредности mAP потребно је сабрати вредности добијене за AP за све класе и поделити их бројем класа (у нашем случају постоје две класе).

- $mAP@.5:.95$ - Вредност ове метрике се добија када се израчунају вредности за mAP где је $IoU_t=0.5$, затим за mAP где је $IoU_t=0.55$, и тако редом све вредности за mAP са кораком 0.05 све док се не достигне $IoU_t=0.95$, а затим се добијене вредности упросече.
- Прецизност- Показује процентуално колико је модел имао тачних погодака TP , од свих оних које је предвидео као тачне, $TP+FP$. Вредност за прецизност се рачуна према формули:

$$P = \frac{TP}{TP+FP} = \frac{TP}{\text{укупан број детекција}} \quad (4)$$

где је TP број оквира који су тачно исцртани око објекта, чија је вероватноћа предвиђања већа од IoU_t параметра, а FP број оквира који су исцртани али њихова вероватноћа предвиђања не превазилази IoU_t параметар.

- Одзив- Показује процентуално колико је модел имао тачних погодака TP , од свих оних који су у скупу података стварно тачни, $TP+FN$.

Вредност за одзив се рачуна према формули:

$$R = \frac{TP}{TP+FN} = \frac{TP}{\text{укупан број стварних оквира}} \quad (5)$$

где се FN односи на оквира које модел није предвидео и који се нису исцртали око објекта, а требало је да буду исцртани и предвиђени са одређеном вероватноћом.

Метрике *Faster R-CNN* модела, *Detectron2* платформе које су коришћене у раду (Dubey, 2020):

- AP (енг. *average precision*) – Просечна прецизност када је граница $IoU_t=0.5$, па све до вредности $IoU_t=0.95$ са кораком 0.05, са максималним бројем детекција по фотографији 100.
- $AP_{IoU_t=0.50}$ - Просечна прецизност када је граница IoU_t параметра постављена на 0.5, са максималним бројем детекција по фотографији 100.
- $AP_{IoU_t=0.75}$ - Просечна прецизност када је граница IoU_t параметра постављена на 0.75, са максималним бројем детекција по фотографији 100.
- AR (енг. *average recall*)- Просечан одзив када је граница $IoU_t=0.5$, па све до вредности $IoU_t=0.95$ са кораком 0.05, са максималним бројем детекција по фотографији 100
- $AR_{IoU_t=0.50}$ - Просечан одзив када је граница IoU_t параметра постављена на 0.5, са максималним бројем детекција по фотографији 100
- $AR_{IoU_t=0.75}$ - Просечан одзив када је граница IoU_t параметра постављена на 0.75, са максималним бројем детекција по фотографији 100.
- AR_{small} - Просечан одзив приликом детекције малих објеката
- AR_{medium} - Просечан одзив приликом детекције објеката средње величине
- AR_{large} - Просечан одзив приликом детекције великих објеката

3. ПРЕГЛЕД ЛИТЕРАТУРЕ

Појавом епидемиолошке ситуације настаје обавеза ношења заштитних маски, па се тако развија и све већа потреба препознавања заштитних маски и идентификација особа које носе и које не носе маску. Детекција објеката добија на значају, а многе институције и научни центри се баве овим питањем, покушавајући да примене и тестирају успешност различитих алгоритама за детекцију објеката на поменутом проблему.

У раду (Yadav, 2020) аутор представља систем за детекцију објеката у реалном времену. Систем треба да изврши аутоматизовано надгледање особа које се крећу по улици, и на тај начин помогне органима власти да спрече ширење епидемије услед непоштовања превентивних мера ношења маски и одржавања социјалне дистанце. Задатак система јесте да у реалном времену изврши детекцију особа, да утврди да ли лоцирана особа носи маску или не, као и да израчуна да ли особа одржава социјалну дистанцу од најмање 2 метара удаљености од друге особе. За детекцију особа у реалном времену и за детекцију маски је коришћен *SSD-MobileNetV2* модел. Окосницу модела чини *MobileNetV2* мрежа која поједностављује процес рачунања, прецизна је и компјутерски ефикасна. *MobileNetV2* мрежа је у овом случају коришћена како би извукла квалитетне мапе карактеристика. На основу ових мапа карактеристика *SSD* детектор врши детекцију и класификацију објеката. Модел је претходно трениран на скупу података од 3 165 фотографија, током 20 епоха. Током тестирања систем је за утврђивање социјалне дистанце и препознавање заштитних маски, остварио прецизност од 91.70%.

У раду (Hussain et al., 2021) аутори представљају иновативно решење „Паметна капија за преглед и дезинфекцију заснована на ИоТ-у (енг. *IoT-based Smart Screening and Disinfection Walkthrough Gate- SSDWG*)“ која би се налазила на јавним местима, а чија је улога превенција ширења заразе. Задатак *SSDWG-a* јесте брзинско мерење температуре помоћу бесконтактнoг сензора, чување података потенцијално оболелих особа ради даље контроле, препознавање заштитних маски и класификација особа на оне које правилно, неправилно или уопште не носе маску. Извршена је упоредна анализа перформанси следећих детектора *VGG-16* (енг. *visual geometry group from oxford*), *MobileNetV2*, *Inception v3*, *ResNet-50*, и *CNN* на проблему препознавања заштитних маски. Модели су тренирани на следећа три скупа података, *MaskedFace-Net* скуп података из ког је издвојено 67 562 фотографије, *FMLD* скуп података који броји 51 433

фотографија, и *MAFA* скуп података који броји 30 811 фотографија. Највећу прецизност је постигао *VGG-16* детектор, са 99.80% тачности, затим *MobileNetV2* са 99.60%, *Inception v3* са 99.40%, *ResNet-50* са 99.20% и *CNN* са 99.00% тачности препознавања заштитних маски.

У раду (Ieamsaard et al., 2021) аутори се баве компаративном анализом *YOLOv5* аглоритма тренираног на 20, 50, 100, 300 и 500 епоха, како би се утврдило за који број епоха ће модел бити најефикаснији на проблему препознавања заштитних маски. Задатак детектора је да класификује особе са фотографије у три групе, особе које исправно носе маску, неисправно носе маску или је не носе уопште. За експеримент је коришћен јавно доступан скуп података који укупно броји 853 фотографије, од чега је 682 искоришћено за тренинг, 85 за валидацију и преосталих 86 за тест. Модел трениран 20 епоха је остварио 79 тачних детекција и седам погрешних, модел трениран 50 епоха је остварио 81 тачних и пет нетачних детекција, на 100 епоха је остварио такође 81 тачних и пет нетачних, на 300 епоха су оставрене најбоље перформансе са 83 тачних детекција и три нетачне, чиме модел постиже чак 96.50% тачности, док је на 500 епоха модел остварио 82 тачне и 4 нетачне детекције.

У раду (Snyder & Husari, 2021) аутори су представили иновативно решење „*Thor*“ који заправо представља мобилног робота, чији је задатак да помоћу технике дубоког учења са слика и снимака у реалном времену препозна особе које не носе маску. Најважнији задатак пројекта јесте научити модел да ефикасно детектује особе, чак и на снимцима и фотографијама слабијег квалитета и под утицајем различитих спољашњих фактора и осветљења. Детекција особа које не носе маску је реализована кроз три корака, прво се врши детекција особе на видеу, затим се на особи детектује и извлачи предео лица, а на самом крају се врши класификација према томе да ли особа са видео снимка носи или не носи маску. За детекцију објеката је коришћена *RetinaNet* мрежа која у својој архитектури интегрише *ResNet-50* и *FPN* помоћу којих извлачи мапе битних карактеристика, на основу којих врши детекцију особа на фотографији или снимку. *RetinaNet* мрежа је тренирана на *COCO* скуп података који броји 330 000 фотографија и 91 тип објеката. За детекцију предела лица је коришћена *Multi-Task cascaded Convolutional Neural Network (MTCNN)*. *MTCNN* детекцију врши кроз три корака, прво *Proposal Network (P-Net)* генерише предложене регионе од интереса, затим се предложени региони од интереса филтрирају помоћу *Refinement Network (R-Net)*, а затим се на самом крају одређује коначна локација лица на фотографији помоћу *Output Netwrok (O-Net)*. *MTCNN* мрежа је тренирана на *CelebA* скуп података који броји 202

599 фотографија и *Wider Face* скупу података који броји 393 703 фотографија. За класификацију особа на оне које носе или не носе маску је коришћен *MobileNetV2* модел, који је трениран на *CMCD* (енг. *Custom Mask Community Dataset*) скупу података који броји 1 376 фотографија. Коначно тестирање ефикасности модела је извршено на фотографијама извучених са видео снимака које је прикупио мобилни робот „*Thor*“ пролазећи кроз студентски дом. Из 150 видео снимака је извучено 229 корисних фотографија. *RetinaNet* је успела да детектује свих 229 особа са фотографија постижући 100.00% прецизности, *MTCNN* мрежа је постигла прецизност од 94.40% приликом детекције лица на фотографијама, док је *MobileNetV2* постигао 81.30% прецизности приликом препознавања заштитних маски.

У раду (Loey et al., 2021) аутори представљају хибридни модел за детекцију заштитних маски, који интегрише дубоко учење и машинско учење. Циљ експеримента јесте извршити упоредну анализу коришћених алгоритама за детекцију објеката, како би се пронашло најбоље решење за препознавање заштитних маски, које се може уградити у надзорне камере на јавним местима, како би се спречило ширење епидемије од стране особа које не носе маску. За извлачење мапа карактеристика са фотографија је коришћен *ResNet-50* детектор који је представник дубоког учења, док су за класификацију коришћени представници машинског учења, *SVM* (енг. *support vector machine*) алгоритам, дрво одлучивања и ансамбл алгоритам (интеграција *KNN* алгоритма, линеарне и логистичке регресије). Моделу су тренирани на *Real-World Masked Face Dataset (RMFD)* скупу података који укупно броји 95 000 фотографија од чега је издвојено 5 000 фотографија са особама које носе маску и 5 000 фотографија са особама без маске, и на *Simulated Masked Face Dataset (SMFD)* скупу података који броји 1 570 фотографија. Тестирање је извршено на фотографијама из претходна два поменута скупа података, али и на *Labeled Faces in the Wild (LFW)* скупу података који је намењен искључиво само за тестирање модела. Дрво одлучивања је на *LFW* тест скупу података постигао тачност од 99.89%, *SVM* алгоритам је на *LFW* тест скупу података постигао тачност од 100.00%, ансамбл алгоритам је на *LFW* тест скупу података постигао тачност од 100.00%.

4. ЕКСПЕРИМЕНТ

Ово поглавље детаљно описује скуп података на ком је извршен експеримент. Такође објашњава архитектуру и параметре коришћених неуронских мрежа, као и упоредну анализу добијених резултата.

4.1 ОПИС СКУПА ПОДАТАКА

За потребе учења алгоритама за детекцију објеката, коришћен је *Моха3К* скуп података (shitty-bots-inc.github.io), који броји укупно 3 000 фотографија, које су сачуване под називом од *МОХА_0.jpg* до *МОХА_2999.jpg*. Део фотографија је преузет са *Kaggle* сајта, тачније 678 фотографија које су направљене на подручју Кине, Русије и Италије. Већина ових фотографија приказује јавна места, као што су супер-маркети, тржни центри, шеталишта, аеродруми, аутобуске станице, јавни превоз, на којима се појављује велики број људи који носе или не носе заштитне маске (Roy et al., 2020), као што је приказано на Слици 12.



Слика 12. Пример фотографија које се налазе у скупу података

У другом делу од 757 фотографија су приказани крупни кадрови профила и предње стране лица људи, што се може видети на Слици 13., а осталих 1 565 фотографија је преузето са интернета, и на њима се такође може уочити велики број људи, фотографисан како обавља свакодневне активности током текуће пандемије, углавном са подручја Индије (Roy et al., 2020).



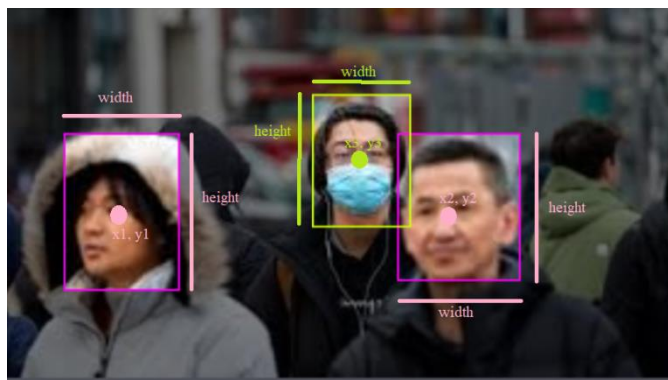
Слика 13. Пример крупних кадрова фотографија из скупа података

На фотографијама је извршена индентификација и означавање свих појединаца које носе или не носе маску. Стога се на читавом скупу података разликују две класе, особе које носе маску и особе које не носе маску. Означавање фотографија је извршено цртањем оквира у пределу главе, без обзира да ли особа носи или не носи заштитну маску, што се може видети на Слици 14. где је зеленим оквиром означена једна особа која носи маску, а љубичастим две особе које не носе маску.



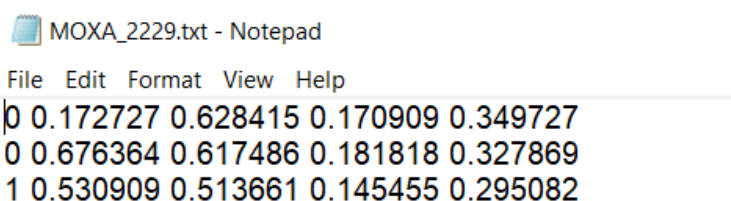
Слика 14. Пример означене фотографије, и класе којима означени објекти припадају

За сваку фотографију посебно се чува анотација у форми текстуалног документа, где су записане координате центра оквира који се означава око главе, ширина и висина поменутог оквира, и класа којој означена особа припада. На Слици 15. су приказане информације које се чувају у сваком тектуалном документу.



Слика 15. Информације које се чувају за сваку ознаку на фотографији

Уколико особа носи маску, онда ће она у анотацији бити означена класом 1, у супротном ће бити означена класом 0. На Слици 16. је приказана текстуална анотација која се чува за горе поменуту фотографију. Прва два реда се односе на особе које нису носиле маске а последњи за особу која носи маску. Прво је наведена припадност класи 0 или 1, затим вредност за x координату центра, вредност за y координату центра, затим вредност за ширину означеног оквира, и на крају за висину оквира. Такође, анотација фотографија може бити сачувана у било ком формату.



Слика 16. Текстуална анотација фотографије

У текстуалној анотацији, димензије x и y координата, као и ширине и дужине оквира су скалиране према димензијама фотографија. Да би се успешно извукле статистике над целим скупом података, као што су најмања, највећа, просечна вредност, стандардна девијација и квантили за x и y координате, као и за ширину (енг. *width*) и висину (енг. *height*) фотографија, потребно је пронаћи првобитне вредности зависне од пропорција сваке фотографије. У Табели 6. су наведене вредности текстуалне анотације за фотографију *Моха_1.jpg*.

Табела 6. Текстуална анотација фотографије *Моха_1.jpg*

Класа	x	y	Ширина оквира	Висина оквира
1	0.25	0.22	0.13	0.21
0	0.57	0.17	0.08	0.17
1	0.84	0.58	0.03	0.05

Након извлачења димензија фотографије, изражених у пикселима, потребно је поново израчунати вредности горе поменутих атрибута, према следећим формулама:

$$x = x * picture_width \quad (6)$$

$$y = y * picture_height \quad (7)$$

$$bounding_box_area = x * y \quad (8)$$

$$picture_area = picture_width * picture_height \quad (9)$$

где је *picture_width* ширина фотографије изражена у пикселима, *picture_height* висина фотографије изражена у пикселима, *bounding_box_area* површина на фотографији коју заузима оквир исцртан око објекта, који се рачуна за сваки објекат означен на фотографији, *picture_area* површина целе фотографије.

У Табели 7. су наведене вредности атрибута за фотографију *Моха_1.jpg*, након примене наведених формула.

Табела 7. Израчунате вредности атрибута за фотографију *Моха_1.jpg*

Класа	x	y	Ширина оквира	Висина оквира	Површина оквира	Ширина фотографије	Висина фотогр.	Површина фотогр.
1	489.50	280.50	241.00	267.00	64347.15	1920	1277	2451840
1	1101.00	216.50	148.00	217.001	32115.95	1920	1277	2451840
0	1617.50	746.50	61.00	69.00	4209.03	1920	1277	2451840

У Табели 8. су приказани основни статистички подаци израчунати на целокупном скупу података, где су вредности заокружљене на три децимале.

Табела 8. Основни статистички подаци на скупу података

Класа	x	y	Ширина оквира	Висина оквира	Површина оквира	Ширина фотографије	Висина фотографије	Површина фотогр.
<i>count</i>	12169.0	12169.0	12169.0	12169.0	12169.00	12169.00	12169.00	12169.00
<i>mean</i>	341.88	169.48	61.99	80.15	13694.52	698.48	458.07	58947
<i>std</i>	397.20	209.87	85.64	105.24	6.2862.31	644.32	432.06	1456828
<i>min</i>	3.00	4.50	2.00	3.00	9.00	48.00	48.00	2.304
<i>25%</i>	113.00	59.00	15.00	19.00	288.00	275.00	181.00	50325
<i>50%</i>	196.50	94.50	32.00	42.00	1368.00	300.00	194.00	50512
<i>75%</i>	445.50	208.00	78.00	105.00	8085.02	1000.00	657.00	665000
<i>max</i>	4399.00	2943.50	1827.00	1783.00	3257542.00	4608.00	3306.00	14644580

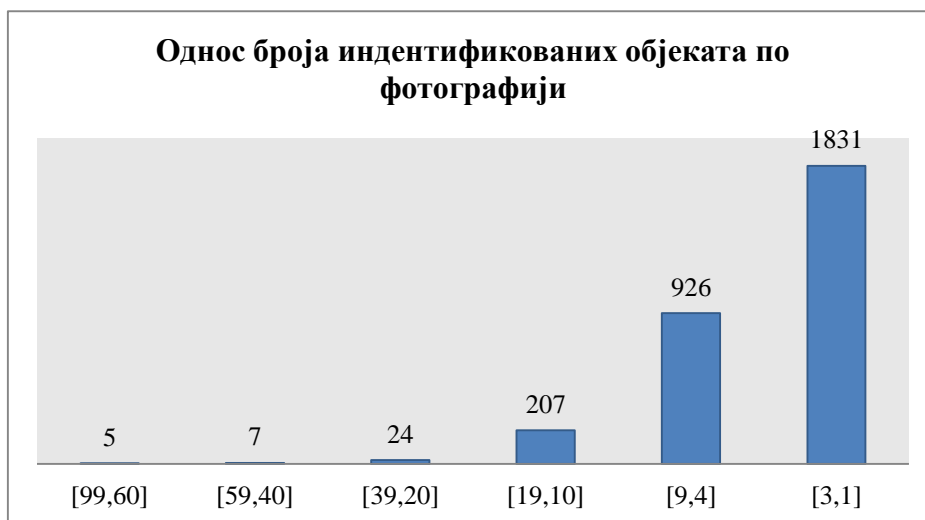
Укупан број ознака, тачније укупан број идентификованих објеката који припадају било класи 0, особе које не носе маску, било класи 1, особе које носе маску, износи 12 169. Посматрано на 3 000 слика, просечна вредност за центар оквира који се исцртава око објекта је $(x, y) = (196.500, 94.500)$. Према овоме можемо да претпоставимо где ће се наћи центар новог оквира. Просечна површина коју на фотографији заузимају исцртани оквири износи 1 367.997 пиксела. Према овим подацима од просечне величине фотографије, која износи 50 512 пиксела, 2.71% је прекривено оквирима који су исцртани око објеката, док остатак фотографије не садржи значајне објекте. На 75.00% фотографија је површина оквира који су се исцртали око објеката 8 085.020 пиксела, што значи да се на само 25.00% преосталих фотографија, тачније 750 фотографија, налазе већи оквири, који покривају већи део фотографије, него на осталих 2 250.

На Слици 17. се уочава да 75.00% особа на фотографијама које су забележене, носи маску, што од укупно 12 169 индентификованих објеката, чак 9 157 њих носи маску, док осталих 25.00%, односно 3 012 особа не носи маску. Скуп података је небалансиран, а доминирајућа класа јесте класа 1, која обухвата особе које носе маску.



Слика 17. Процентуални однос индентификованих класа

Од укупних 3 000 фотографија, на 61.00% њих, односно на 1 831 фотографији се налази између једног и три индентификованих, значајних објеката, тј. особа, док је на пет фотографија индентификовано између 99 и 60 значајних особа, које чине само 0.17% од укупног броја фотографија, што се може уочити на Слици 18.



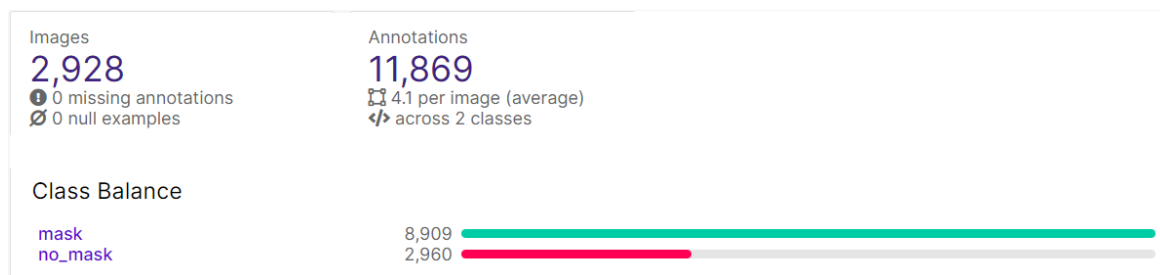
Слика 18. Број особа по фотографији

4.1.1 Сређивање скупа података

Како би подаци били адекватно припремљени за улаз у алгоритам, коришћена је платформа за компјутерски вид која носи назив *Roboflow* (*roboflow.com*, 2019). Ова платформа ставља на располагање велики број могућности сређивања података, као што су мењање величина фотографија, ротација, изоловање објеката, подешавање контраста, повећање робусности модела намерним убацивањем шума у виду замагљивања (енг. *blur*), потамњивања (енг. *grayscale*) и повећања осветљености (енг. *brightness*) фотографија.

Приликом сређивања података (енг. *preprocessing*) избачено је 72 фотографије и њихових текстуалних анотација јер су анотације грешком садржале означене области, чија је величина износила нула пиксела. До ових грешака најчешће долази када се приликом индентификације и означавања објеката, случајно кликне на неки део фотографије, где се генерише једна тачкица која се веома лако изгуби из вида, при чему она не носи никакву информацију, нема правоугаону површину која се очекује, и не указује на објекат од значаја. Овакве анотације се називају *zero area* анотације, нису погодне за улаз у модел јер врло лако могу збунити модел погрешним информацијама. Након овог корака у скупу података остаје 2 928 фотографија на којима је примењена ауто-ротација, чиме су све фотографије доведене у адекватан положај, као и промена величине, након чега све фотографије имају једнаке димензије од 416×416 пиксела, услед чега би требало да се повећа брзина детекције објеката на фотографији, а да се при томе не изгубе информације.

Након мањих корекција на почетном скупу података, коначни скуп података броји 2 928 фотографија, са укупним бројем од 11 869 индентификованих објеката од значаја, тј. особа које припадају класи 0, која указује на то да нема маске, или класи 1, указује на то да је присутна заштитна маска. Број особа које носе маску је и у овом случају знатно већи, него број оних који не носе маску, а просечан број индентификованих особа по фотографији износи четири. На Слици 19. се могу видети поменуте информације, које генерише платформа, у виду извештаја, након завршеног сређивања података.



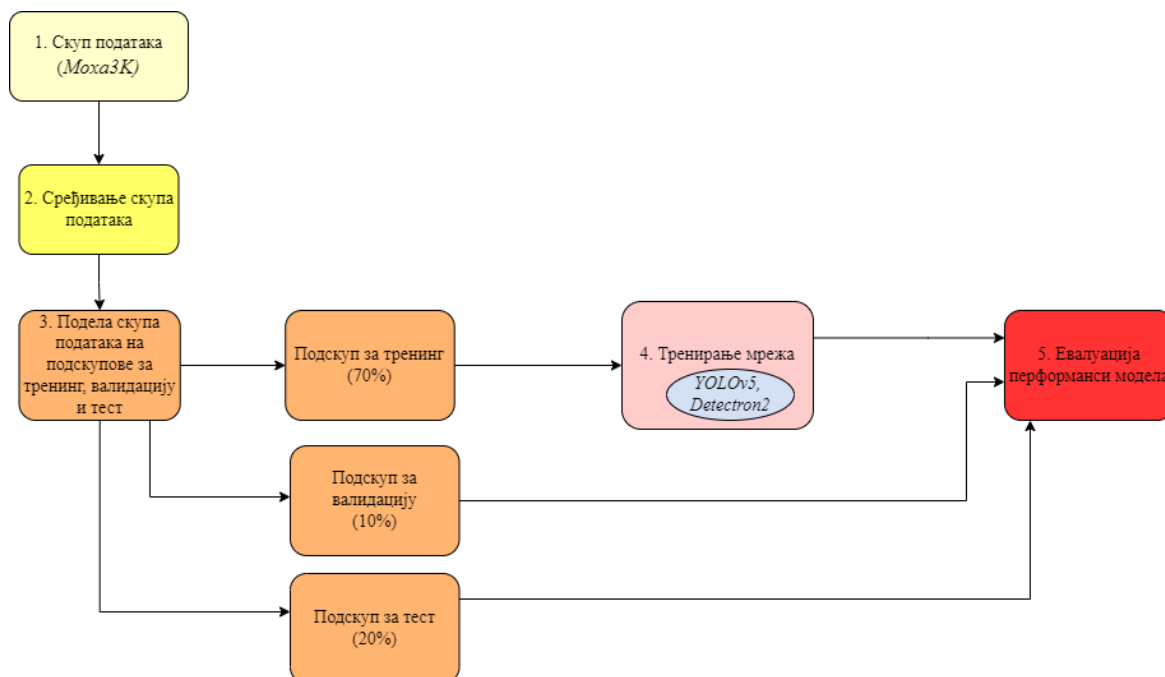
Слика 19. Извештај о броју фотографија, броју ознака и односу идентификованих класа

Овако припремљен скуп података је затим подељен на три подскупа, део који ће бити коришћен за тренинг, чини 70.00% од укупног броја фотографија и броји 2 051 фотографију, део за валидацију, чини 10.00% од укупног броја фотографија и броји 291 фотографије, део за тест који чини преосталих 20.00% и броји 586 фотографија. Модел ће учити на тренинг скупу података, затим ће тако научен модел бити примењен на скупу података за валидацију, са циљем финог подешавања и проналажења најбоље вредности за параметре модела, а на самом крају, модел са најбољим параметрима ће бити примењен на потпуно непознатом, тест скупу података.

4.2 ЕВАЛУАЦИЈА ПЕРФОРМАНСИ МОДЕЛА

У овом поглављу ће бити описан поступак тренирања *YOLOv5* и *Faster R-CNN* модела са *Detectron2* платформе. Припремљен скуп података је подељен на три подскупа у односу 70%-10%-20%. Оба детектора ће бити обучавања на подскупу података за тренинг, који броји 2 051 фотографију. Сврха подскупа за валидацију је да спречи модел да преучи (енг. *overfitting*), подскуп за тест чине фотографије које модели никада нису видели, а над фотографијама из овог подскупа ће научени модели покушати што прецизније да детектују објекте. Упоредном анализом и евалуацијом перформанси оба модела биће утврђено по којим карактеристикама се они разликују, и у којим случајевима је боље користити *YOLOv5* детектор, а у којим *Faster R-CNN* детектор, за постављени проблем класификације особа на фотографијама, на особе који носе маску и на особе које не носе маску. Циљ спровођења анализе је проналазак модела који ће на најбољи начин интегрисати способност и брзину учења са што већом прецизношћу и тачности детекције, а затим тај модел искористити као решење поменутог проблема.

На Слици 20. је дат шематски приказ корака које је потребно спровести за реализацију експеримента.



Слика 20. Шематски приказ корака за спровођење експеримента

4.2.1 Евалуација перформанси на подскуповима за тренинг и валидацију

Модел *YOLOv5* и *Faster R-CNN*, *Detectron2* платформе су учени на подскупу података за тренинг који броји 2 051 фотографију, након чега се на скупу података за валидацију који броји 294 фотографија, врши оцена успешности модела, као и фино подешавање хиперпараметара како би се додатно повећала способност мреже и прецизност детекције објеката. Пре почетка тренирања потребно је дефинисати и проследити детекторима вредности одређених аргумената.

Аргументи *YOLOv5* модела:

- *Img* - димензија фотографија
- *Batch* - односи се на број узорака фотографија које ће одједном бити прослеђене као улаз у мрежу. Цео подскуп података за тренирање је издељен на више целина (енг. *batch*), које се затим, једна по једна прослеђују као улаз у мрежу.
- *Epoch* - број којим се дефинише колико пута ће мрежа проћи кроз читав скуп података који јој се прослеђује за тренинг. Једна епоха значи да мрежа приликом тренинга једном прође кроз све фотографије које су јој прослеђене кроз

подељене целине, док би 50 епоха значило да ће мрежа 50 пута проћи кроз све фотографије и тако се обучавати истим подацима.

- *Data* - путања до локације на којој је смештен скуп података за тренирање
- *Weights* - почетне вредности за тежине подешене приликом тренирања на *COCO* скупу података

Аргументи *Faster R-CNN* модела:

- *Dataset.train* - путања до локације на којој је смештен скуп података за тренирање
- *Dataset.test* - путања до локације на којој је смештен скуп података за тестирање
- *Imgs_per_batch* - број фотографија које ће се наћи у једној од целина које ће бити прослеђене као улаз у мрежу
- *Max_iter* - укупан број итерација који се одређује алгоритму за пролазак кроз све прослеђене целине, где се приликом сваке итерације обрачунавају и подешавају нове тежине. Разлика између епохе и итерације се уочава на примеру где је дефинисано 1000 фотографија, подељених у две целине са по 500 фотографија, мрежа ће том приликом направити две итерације да би прошла кроз две целине, односно требаће јој једна епоха да прође кроз обе целине.
- *Model.weights* - почетне вредности за тежине, такође подешене приликом тренирања на *COCO* скупу података

4.2.1.1 Перформансе: *YOLOv5* 50/100/400 епоха

Вредности аргумената прослеђених *YOLOv5* детектору за *batch=16*, и *epoch=50/100/400*. Број целина указује на то да ће укупан број фотографија прослеђених за тренинг бити подељен на 128 целина од које ће свака садржати по 16 фотографија, а мрежа ће приликом процеса обучавања 50/100/400 пута проћи кроз сваку целину. Перформансе које је модел остварио на скупу података за валидацију, у зависности од броја епоха, приказане су у Табелама 9, 10. и 11.

Табела 9. Метрике *YOLOv5* модела тренираног 50 епоха

Класа	Фотографија	Означено	mAP@.5	mAP@.5:.95	Прецизност	Одзив
Укупно	291	1203	0.84	0.43	0.90	0.75
<i>Mask</i>		915	0.91	0.45	0.90	0.84
<i>No_mask</i>		288	0.78	0.41	0.87	0.66

Табела 10. Метрике *YOLOv5* модела тренираног 100 епоха

Класа	Фотографија	Означено	mAP@.5	mAP@.5:.95	Прецизност	Одзив
Укупно	291	1203	0.84	0.43	0.88	0.76
<i>Mask</i>		915	0.91	0.46	0.89	0.87
<i>No_mask</i>		288	0.77	0.40	0.888	0.66

Табела 11. Метрике *YOLOv5* модела тренираног 400 епоха

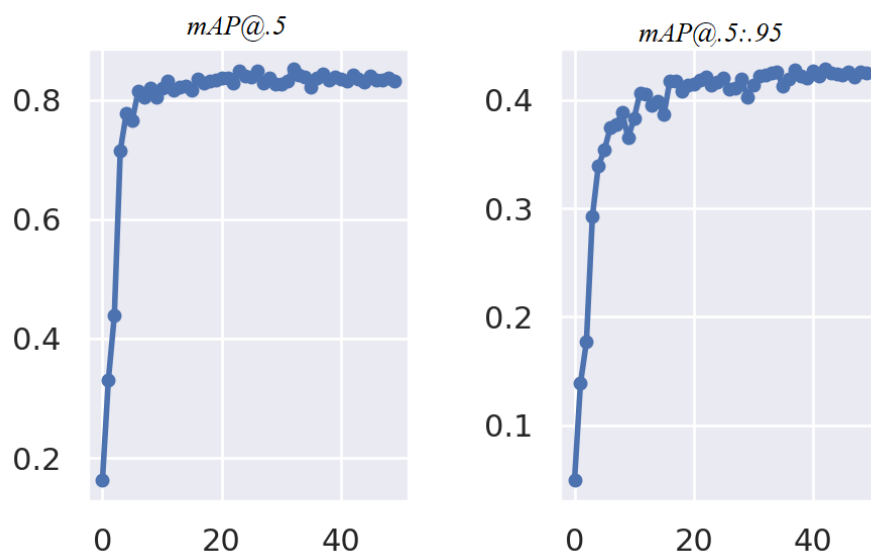
Класа	Фотографија	Означено	mAP@.5	mAP@.5:.95	Прецизност	Одзив
Укупно	291	1203	0.85	0.43	0.87	0.76
<i>Mask</i>		915	0.91	0.45	0.89	0.85
<i>No_mask</i>		288	0.78	0.40	0.84	0.67

Време које је било потребно за 50/100/400 епоха тренинга респективно, износи 19.98 минута, 40.44 минута и 166.74 минута.

За ово време мрежа је успела да постигне просечну прецизност од 84.00% односно 43.00% по оштријем критеријуму на 50 епоха, 84.00% односно 43.00% по оштријем критеријуму на 100 епоха и 85.00% односно 43.00% по оштријем критеријуму на 400 епоха.

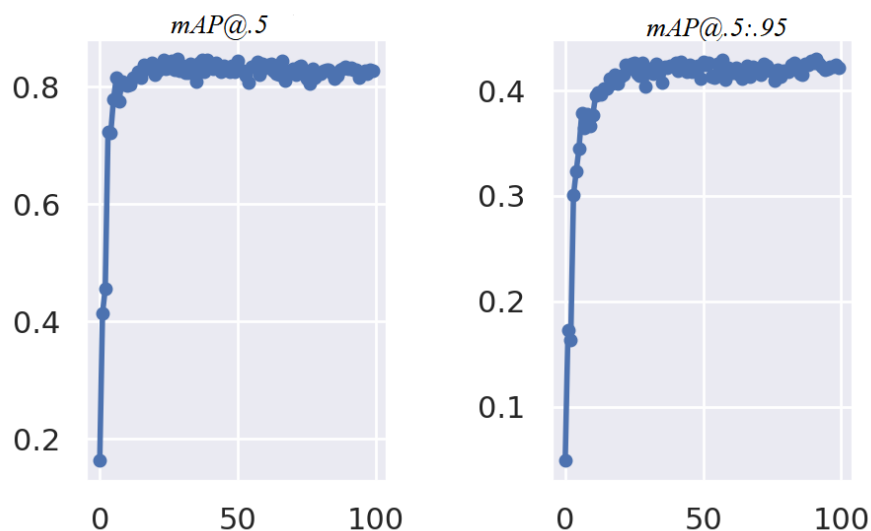
Посматрано на 291 фотографији из подскупа за валидацију, од укупно 1 203 означених класа, где се 915 односи на особе које носе маску, а преосталих 288 на оне које не носе маску, по оштријем критеријуму модел је успео да погоди да 45.00% особа носи маску, а 41.00% њих да сврста у групу која не носи маску на 50 епоха, на 100 епоха 46.00% је сврстано у групу која носи маску, а 40.00% њих у групу која не носи маску, док је на 400 епоха 45.00% сврстано у групу која носи маску, а 40.00% у групу која не носи маску.

На Слици 21. су приказани графици вредности за метрике $mAP@.5$ и $mAP@.5:.95$ које осцилирају током 50 епоха обучавања модела.



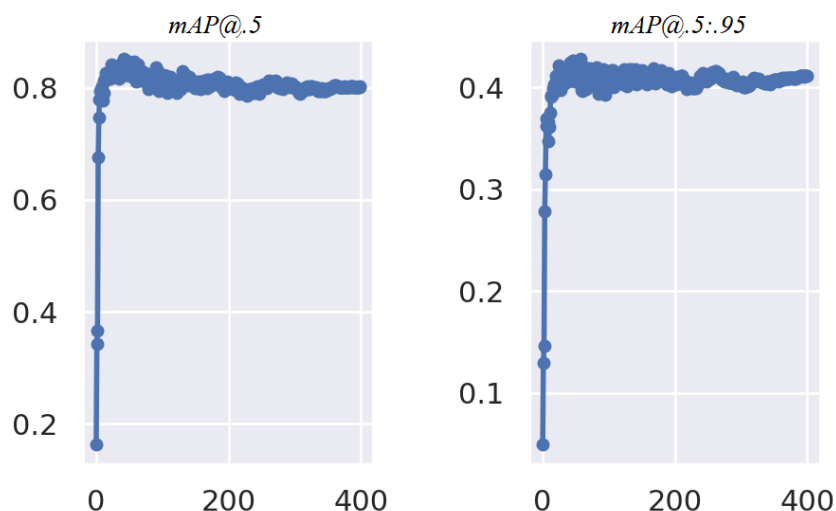
Слика 21. Графички приказ вредности метрика $mAP@.5$ и $mAP@.5:.95$, модел *YOLOv5*, 50 епоха

На Слици 22. су приказани графици вредности за метрике $mAP@.5$ и $mAP@.5:.95$ које осцилирају током 100 епоха обучавања модела.



Слика 22. Графички приказ вредности метрика $mAP@.5$ и $mAP@.5:.95$, модел *YOLOv5*, 100 епоха

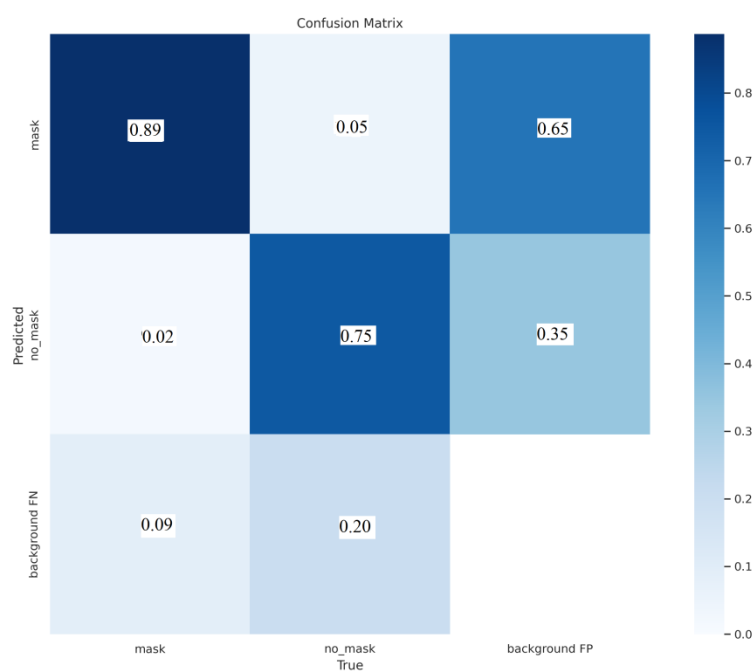
На Слици 23. су приказани графици вредности за метрике $mAP@.5$ и $mAP@.5:.95$ које осцилирају током 400 епоха обучавања модела.



Слика 23. Графички приказ вредности метрика $mAP@.5$ и $mAP@.5:.95$, модел *YOLOv5*, 400 епоха

Уколико се као значајнија класа одаберу особе које не носе маску, од модела се очекује да направи што мање погрешних класификација у којима ће предвидети да особа носи маску када је она заправо не носи, што се визуелно може приказати путем матрице конфузије.

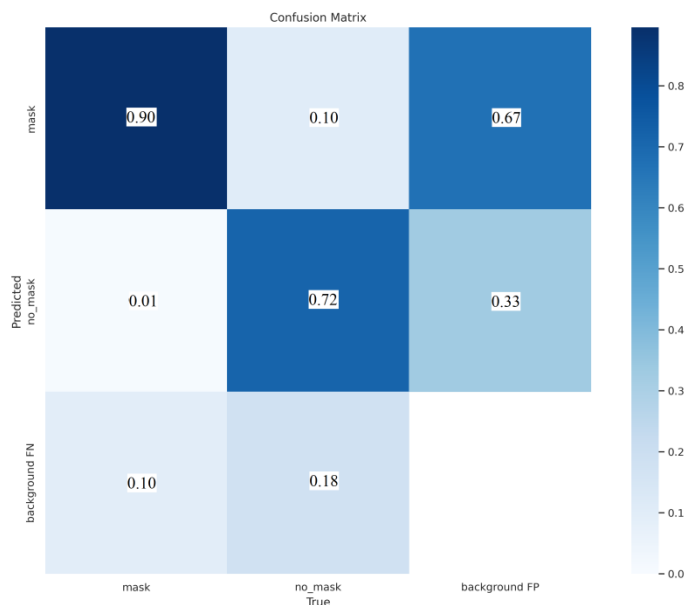
На Слици 24. је приказана матрица конфузије за тренинг од 50 епоха .



Слика 24. Матрица конфузије за модел *YOLOv5*, 50 епоха

Може се уочити да је модел на 50 епоха, у 75.00% случајева предвидео да особа не носи маску, при чему је особа заиста не носи, али да је у 5.00% случајева модел предвидео да особа носи маску, а особа заправо није носила маску.

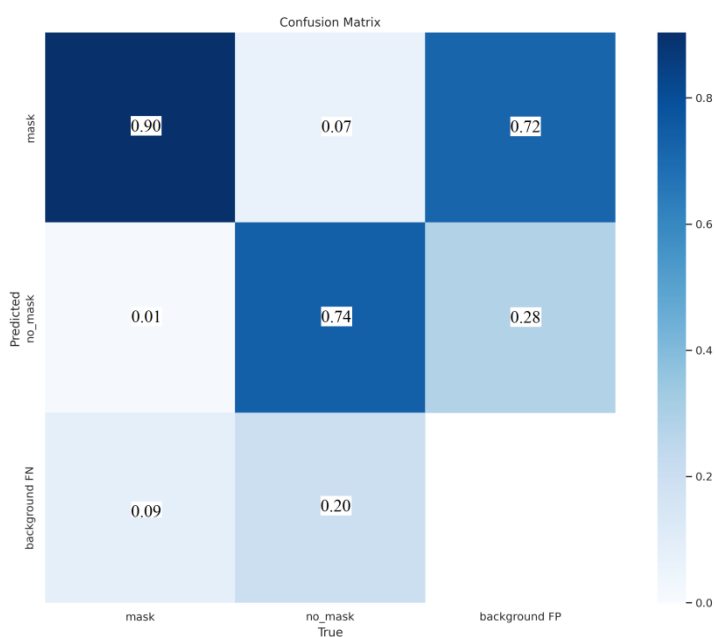
На Слици 25. је приказана матрица конфузије за тренинг од 100 епоха .



Слика 25. Матрица конфузије за модел *YOLOv5*, 100 епоха

На 100 епоха, модел у 72.00% случајева исправно предвиђа да особа не носи маску, а у 10.00% случајева погрешно предвђа да особа носи маску.

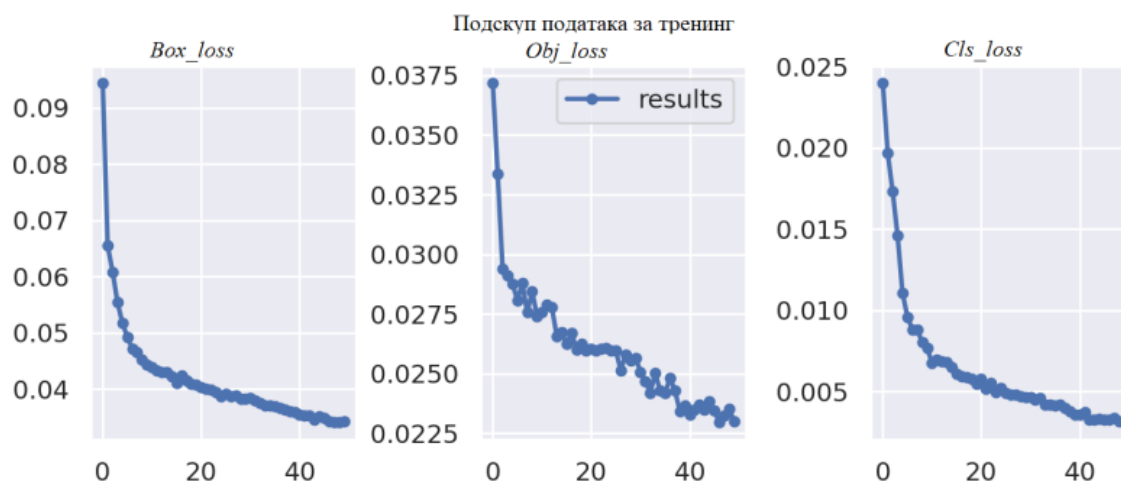
На Слици 26. је приказана матрица конфузије за тренинг од 400 епоха .



Слика 26. Матрица конфузије за модел *YOLOv5*, 400 епоха

На 400 епоха, модел у 74.00% случајева исправно предвиђа да особа не носи маску, а у 7.00% случајева погрешно предвиђа да особа носи маску.

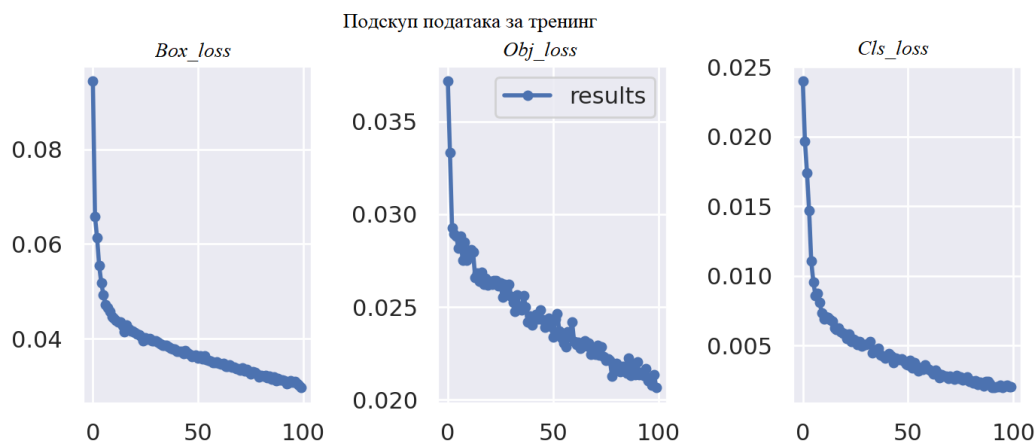
На Слици 27. је приказана промена вредности *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss* функција током 50 епоха тренинга.



Слика 27. *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss*, YOLOv5 50 епоха тренинг

Током тренинга 50 епоха, вредност грешке за *box_loss* је опала са 0.09 на 0.03, за *obj_loss* је опала са 0.04 на 0.024, а за *cls_loss* са 0.02 на 0.003.

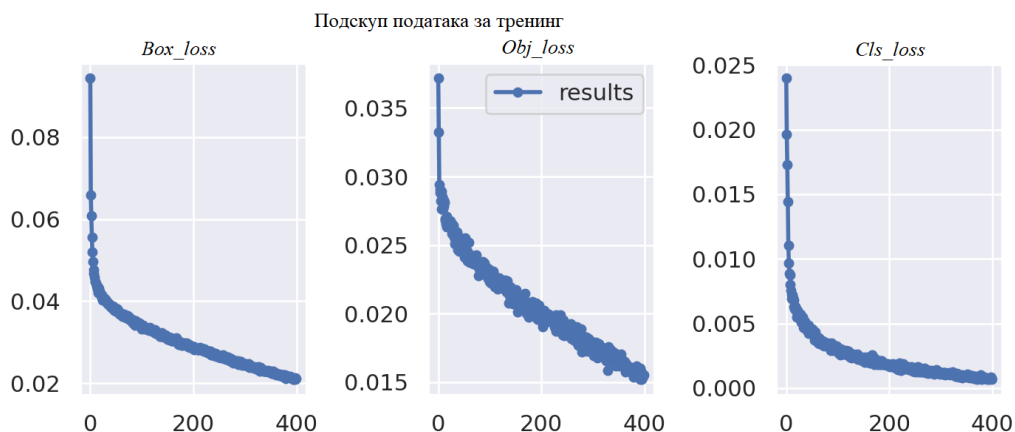
На Слици 28. је приказана промена вредности *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss* функција током 100 епоха тренинга.



Слика 28. *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss*, YOLOv5 100 епоха тренинг

Током тренинга 100 епоха, вредност грешке за *box_loss* је опала са 0.09 на 0.03, за *obj_loss* је опала са 0.04 на 0.02, а за *cls_loss* са 0.02 на 0.003.

На Сlici 29. је приказана промена вредности *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss* функција током 400 епоха тренинга.



Слика 29. *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss*, YOLOv5 400 епоха тренинг

Током тренинга 400 епоха, вредност грешке за *box_loss* је опала са 0.09 на 0.02, за *obj_loss* је опала са 0.035 на 0.015, а за *cls_loss* са 0.02 на 0.0007.

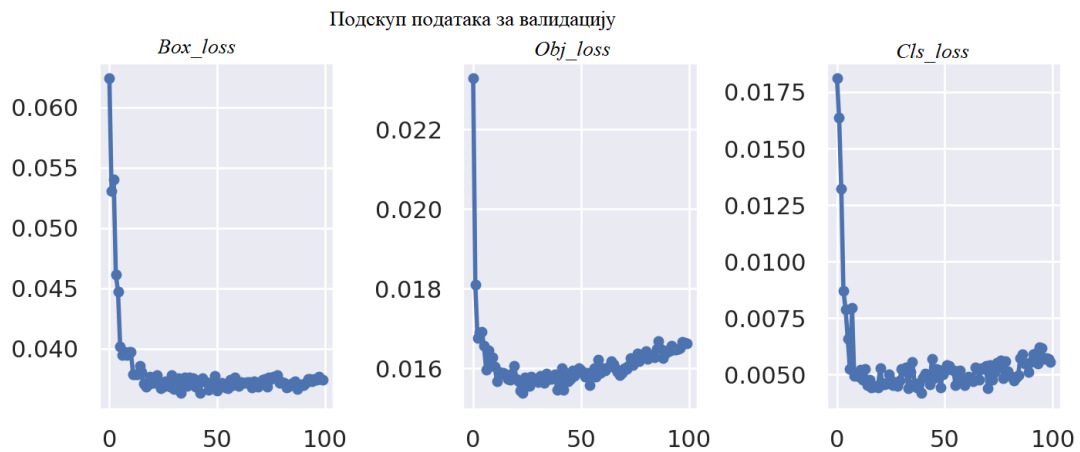
На Сликама 30, 31. и 32. је приказано како је вредност грешке за све три *loss* функције постепено опадала, у зависности од броја епоха, када се претходно истренирани модел примени на скуп података за валидацију. Јако слично примерима са Слика 27, 28. и 29. само што је приликом примене већ обучаваног модела на скуп података за валидацију, грешка коју модел прави приликом предикција знатно мања него она коју модел направи приликом тренинга.

Током валидације модела тренираног 50 епоха, вредност грешке за *box_loss* је опала са 0.06 на 0.04, за *obj_loss* је опала са 0.02 на 0.016, а за *cls_loss* са 0.02 на 0.005.



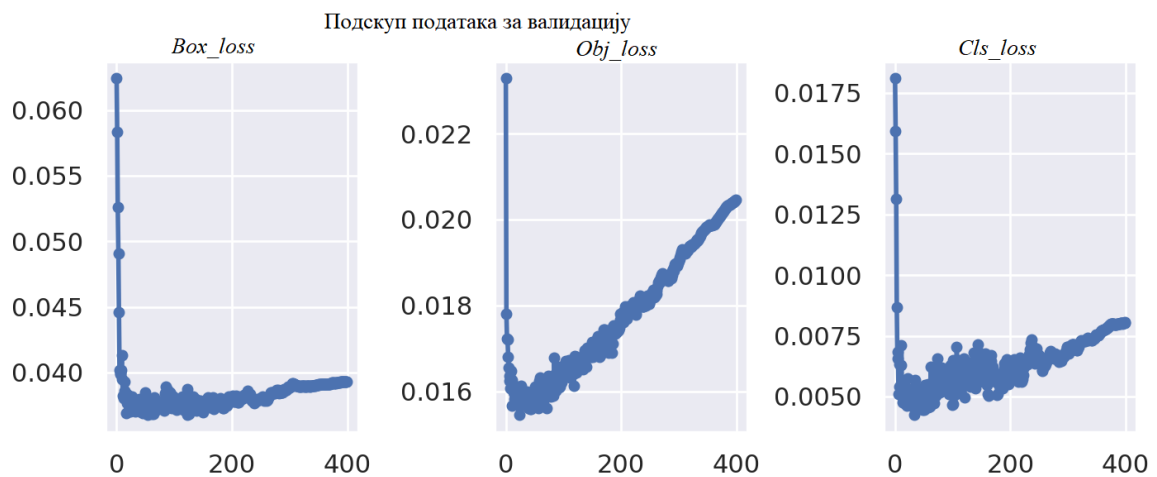
Слика 30. *Bounding box regression loss*, *Objectness of the box loss* и *Clasiffication loss*, YOLOv5 50 епоха валидација

Током валидације модела тренираног 100 епоха, вредност грешке за *box_loss* је опала са 0.06 на 0.04, за *obj_loss* је опала са 0.02 на 0.02, а за *cls_loss* са 0.02 на 0.006.



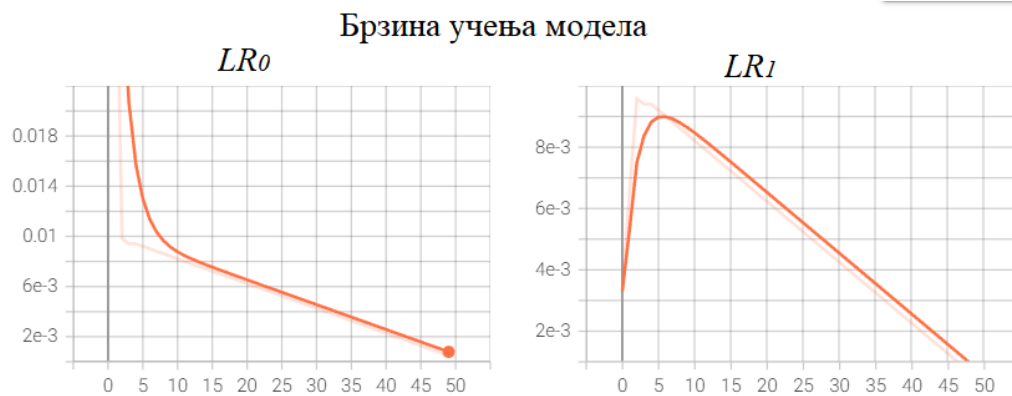
Слика 31. *Bounding box regression loss, Objectness of the box loss и Clasiffication loss, YOLOv5 100 епоха валидација*

Током валидације модела тренираног 400 епоха, вредност грешке за *box_loss* је опала са 0.06 на 0.04, за *obj_loss* је опала са 0.02 на 0.02, а за *cls_loss* са 0.02 на 0.008.



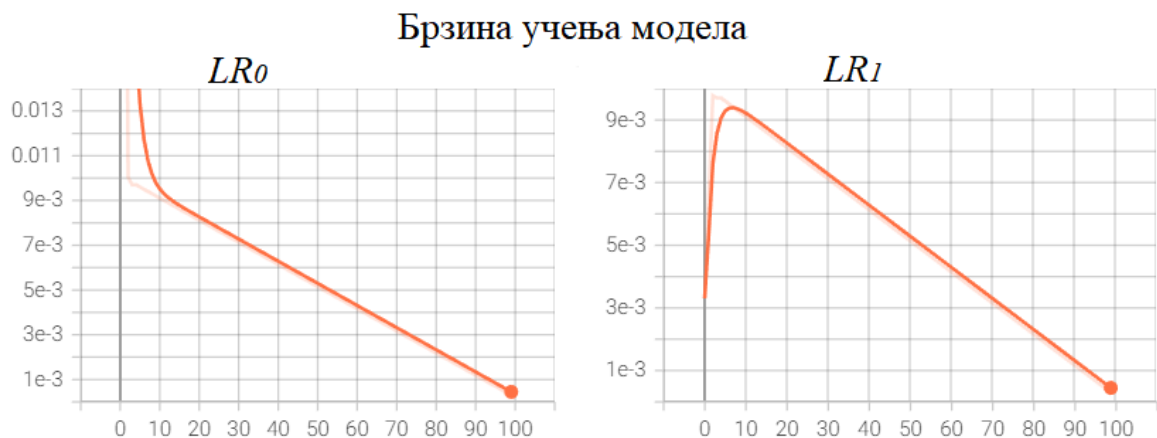
Слика 32. *Bounding box regression loss, Objectness of the box loss и Clasiffication loss, YOLOv5 400 епоха валидација*

На Слици 33. је приказана предефинисана LR_0 крива и пронађена минимална вредност за параметар LR , а затим се као почетна вредност узима пронађени минимум и на скупу података за валидацију успоставља нова LR_1 крива, за модел трениран 50 епоха.



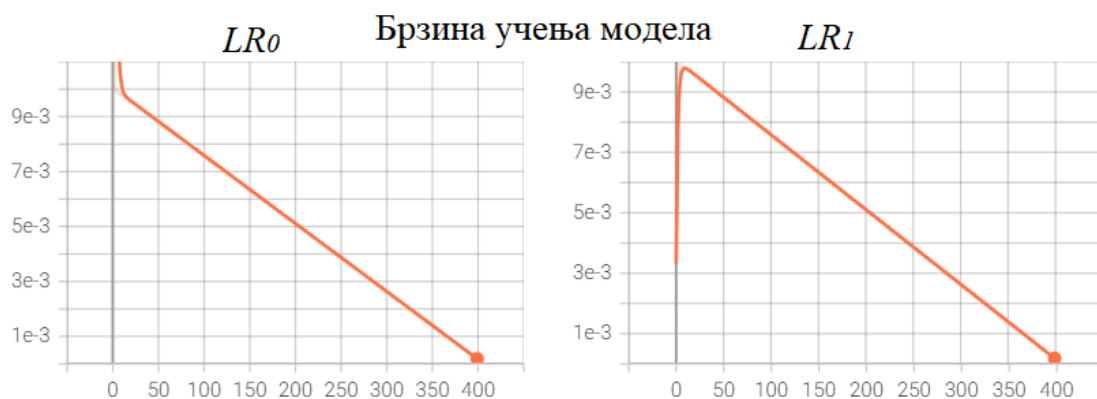
Слика 33. Брзина учења модела, *YOLOv5* 50 епоха

На Слици 34. су приказане LR_0 и LR_1 криве за модел трениран 100 епоха.



Слика 34. Брзина учења модела, *YOLOv5* 100 епоха

На Слици 35. су приказане LR_0 и LR_1 криве за модел трениран 400 епоха.



Слика 35. Брзина учења модела, *YOLOv5* 100 епоха

4.2.1.2 Перформансе: *Detectron2-Faster R-CNN* 1 000/2 000/4 000 итерација

Вредности аргумената прослеђених *Faster R-CNN* детектора на *Detectron2* платформи су за $batch=4$, и $max_iter=1\ 000/2\ 000/4\ 000$. Везу између броја итерација *Faster R-CNN* детектора и епоха *YOLOv5* детектора, је могуће успоставити путем следеће формуле:

$$max_iter = \frac{\text{величина подскупа за тренинг}}{\text{величина } batch-a} \times \text{број пролазака} \quad (9)$$

Перформансе које је модел остварио на скупу података за валидацију, у зависности од броја итерација, приказане су у Табелама 12, 13. и 14.

Табела 12. Метрике *Detectron2, Faster R-CNN* детектор трениран 1 000 итерација

Класа	Фотогр.	Означено	AP@.5	AP@.5:.95	AP@.75	AP _{small}	AP _{medium}	AP _{large}
Укупно	291	1203	0.80	0.39	0.33	0.26	0.43	0.59
<i>Mask</i>		915		0.43	/	/	/	
<i>No_mask</i>		288		0.35	/	/	/	
Класа	Фотогр.	Означено	AR@.5:.95		AR _{small}	AR _{medium}	AR _{large}	
Укупно	291	1203	0.50		0.40	0.54	0.66	

Табела 13. Метрике *Detectron2, Faster R-CNN* детектор трениран 2 000 итерација

Класа	Фотогр.	Означено	AP@.5	AP@.5:.95	AP@.75	AP _{small}	AP _{medium}	AP _{large}
Укупно	291	1203	0.79	0.40	0.34	0.26	0.44	0.62
<i>Mask</i>		915		0.43	/	/	/	
<i>No_mask</i>		288		0.36	/	/	/	
Класа	Фотогр.	Означено	AR@.5:.95		AR _{small}	AR _{medium}	AR _{large}	
Укупно	291	1203	0.50		0.40	0.54	0.69	

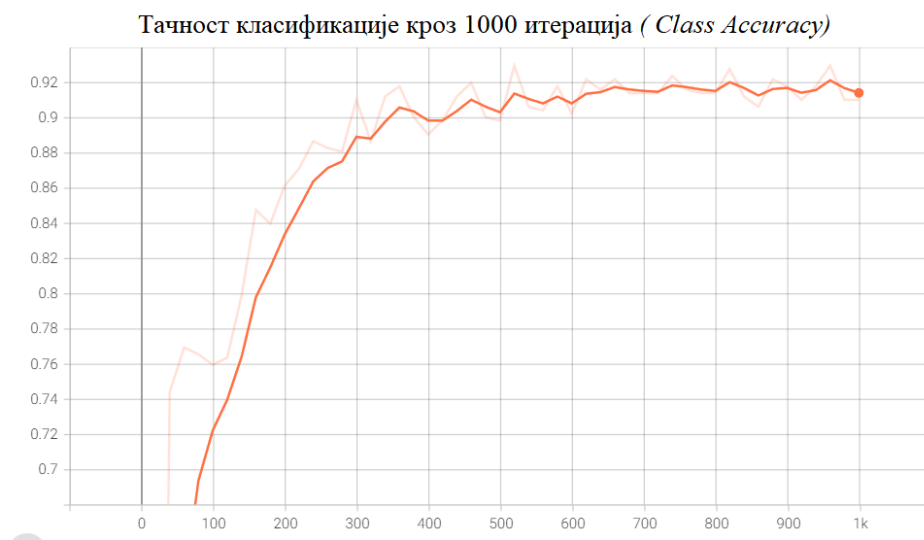
Табела 14. Метрике *Detectron2, Faster R-CNN* детектор трениран 4 000 итерација

Класа	Фотог	Означено	AP@.5	AP@.5:.95	AP@.75	AP _{small}	AP _{medium}	AP _{large}
Укупно	291	1203	0.81	0.41	0.36	0.27	0.44	0.63
<i>Mask</i>		915		0.45	/	/	/	
<i>No_mask</i>		288		0.38	/	/	/	
Класа	Фотогр.	Означено	AR@.5:.95		AR _{small}	AR _{medium}	AR _{large}	
Укупно	291	1203	0.50		0.40	0.54	0.69	

Време које је било потребно за 1 000 итерација тренинга износи 36.02 минута, за 2 000 итерација 71.58 минута, а за 4 000 итерација 143.04 минута. За ово време мрежа је успела да постигне просечну прецизност од 80.00% односно 39.00% по оштријем критеријуму током 1 000 итерација, током 2 000 итерација прецизност од 79.00% односно 40.00% по оштријем критеријуму, а током 4 000 итерација прецизност од 81.00% односно 41.00% по оштријем критеријуму.

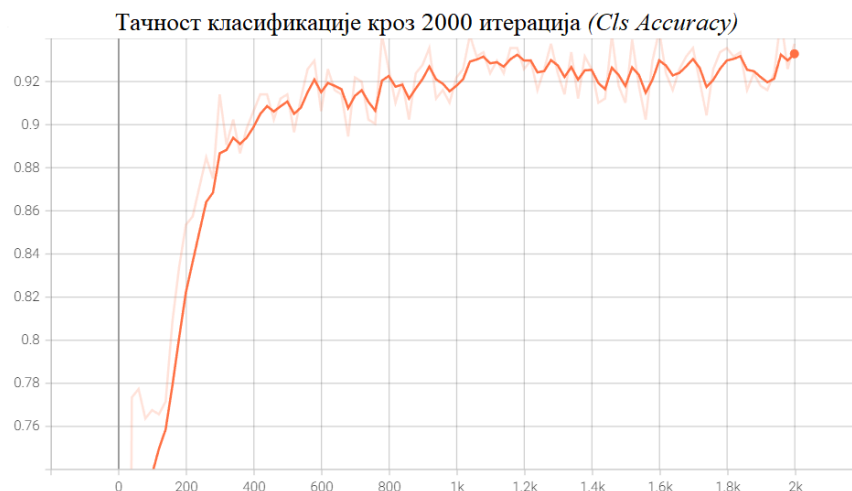
Посматрано на 291 фотографији из подскупа за валидацију, од укупно 1 203 означених класа, где се 915 односи на особе које носе маску, а преосталих 288 на оне које не носе маску, по оштријем критеријуму модел је, за 1 000 итерација тренинга, успео да погоди да 43,00% особа носи маску, а 35.00% њих да сврста у групу која не носи маску, за 2 000 итерација по оштријем критеријуму 43.00% њих је разврстано у класу која носи маску, а 36.00% у класу која не носи маску, док је за 4 000 итерација по оштријем критеријуму 45.00% њих је разврстано у класу која носи маску, а 38.00% у класу која не носи маску.

На Слици 36. је приказан график вредности за метрику тачност, која указује на тачност класификације модела, како осцилира кроз 1 000 итерација обучавања модела.



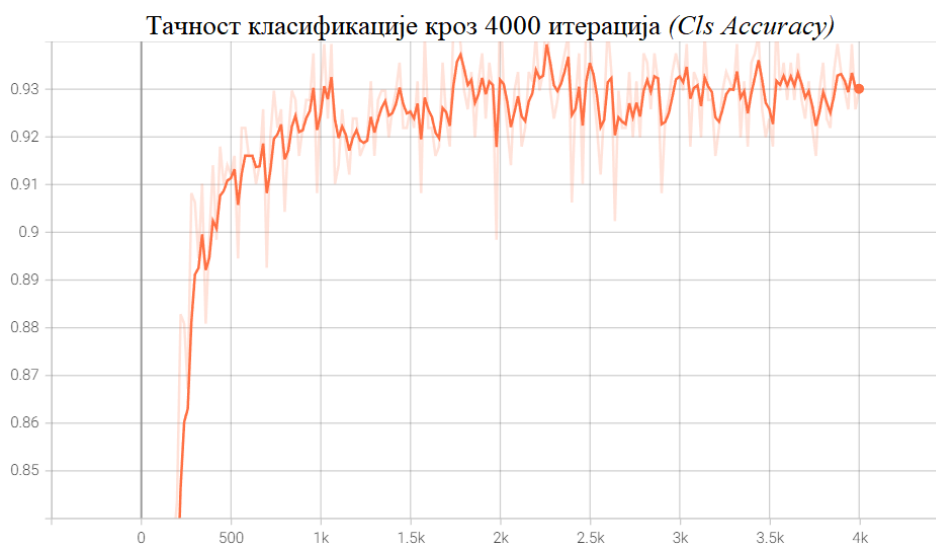
Слика 36. Тачност класификације, *Detectron2* 1 000 итерација

На Слици 37. је приказан график вредности за метрику прецизност кроз 2 000 итерација обучавања модела.



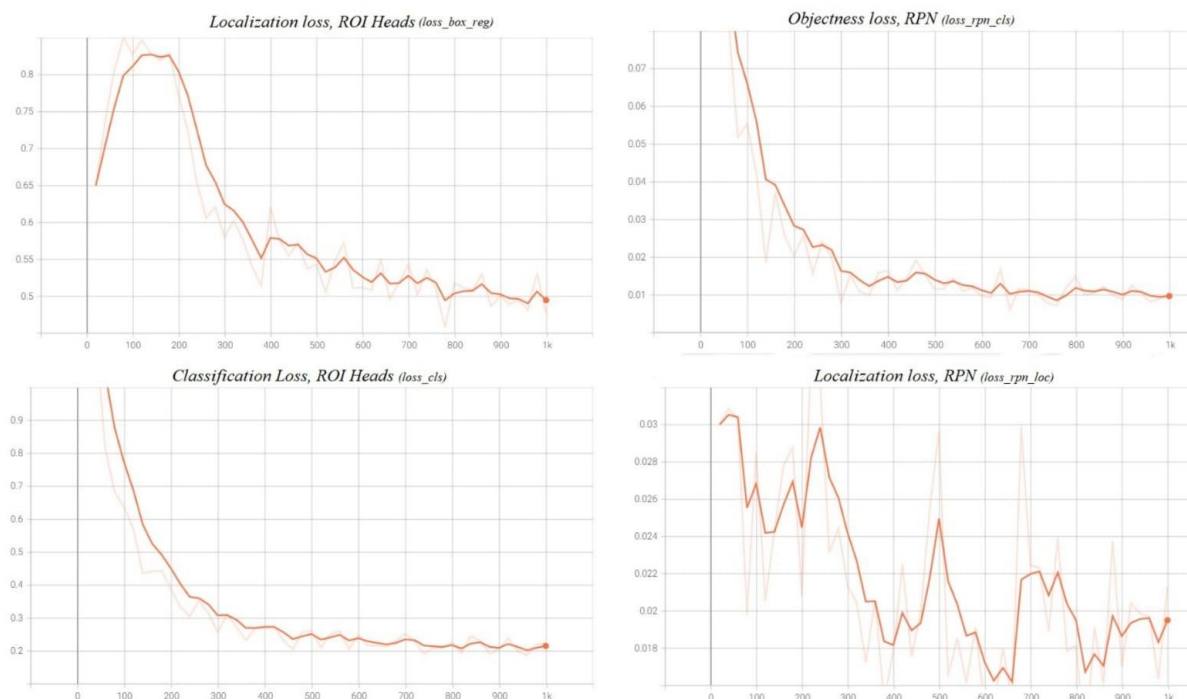
Слика 37. Тачност класификације, *Detectron2* 2 000 итерација

На Слици 38. је приказан график вредности за метрику прецизност кроз 4 000 итерација обучавања модела.



Слика 38. Тачност класификације, *Detectron2* 4 000 итерација

На Слици 39. је приказана промена вредности *Localization loss* и *Clasiffication loss* функција које се рачунају у делу *ROI* глава, и промена вредности *Localization loss* и *Objectness loss* функција које се рачунају у *Region Proposal Network* делу *Faster R-CNN* детектора изабраног са *Detectron2* платформе током 1 000 итерација тренинга.

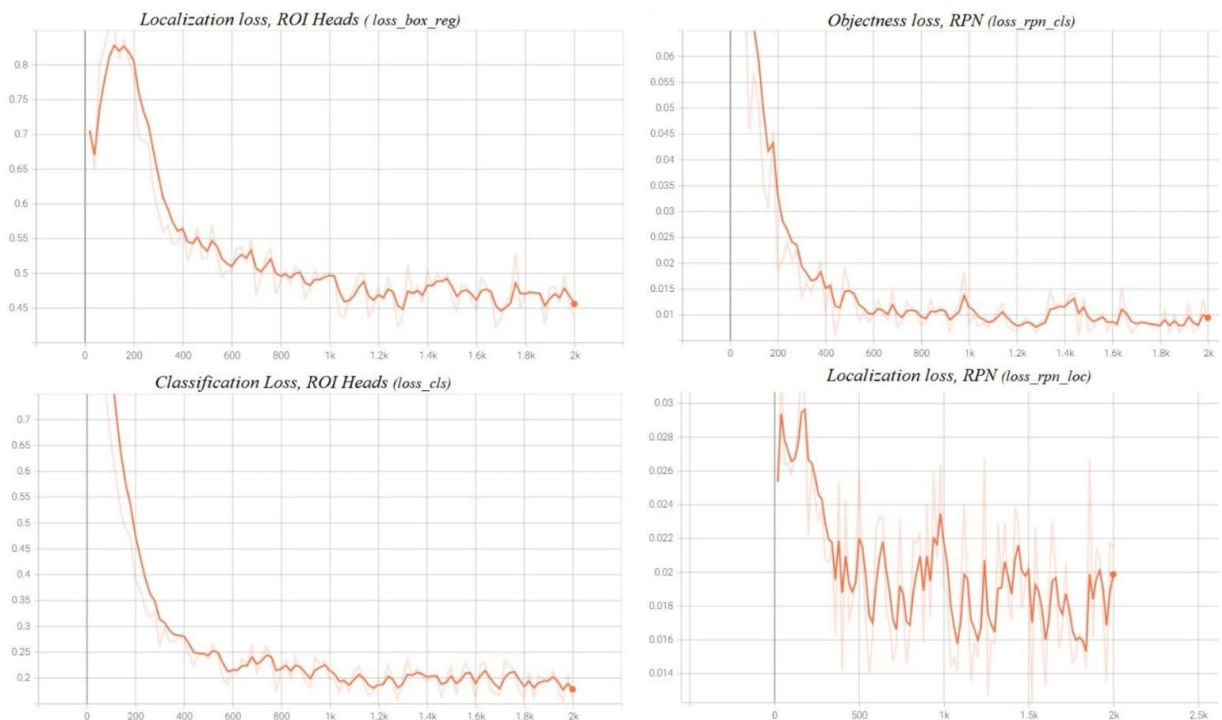


Слика 39. *Localization loss ROI, Clasiffication loss ROI, Objectness loss RPN, Localization loss RPN, Detectron2 1 000 итерација*

Током тренинга вредност грешке за *loss_box_reg* је опала са 0.72 на 0.47, а за *loss_cls* са 1.29 на 0.21.

Вредност за *loss_rpn_cls* опада са 0.02 на 0.01. Грешка погрешне класификације све време осцилира током хиљаду итерација, а вредности за *loss_rpn_loc* се крећу између 0.03 на самом почетку, 0.01 као најмања достигнута вредост, а затим поново расте на 0.02.

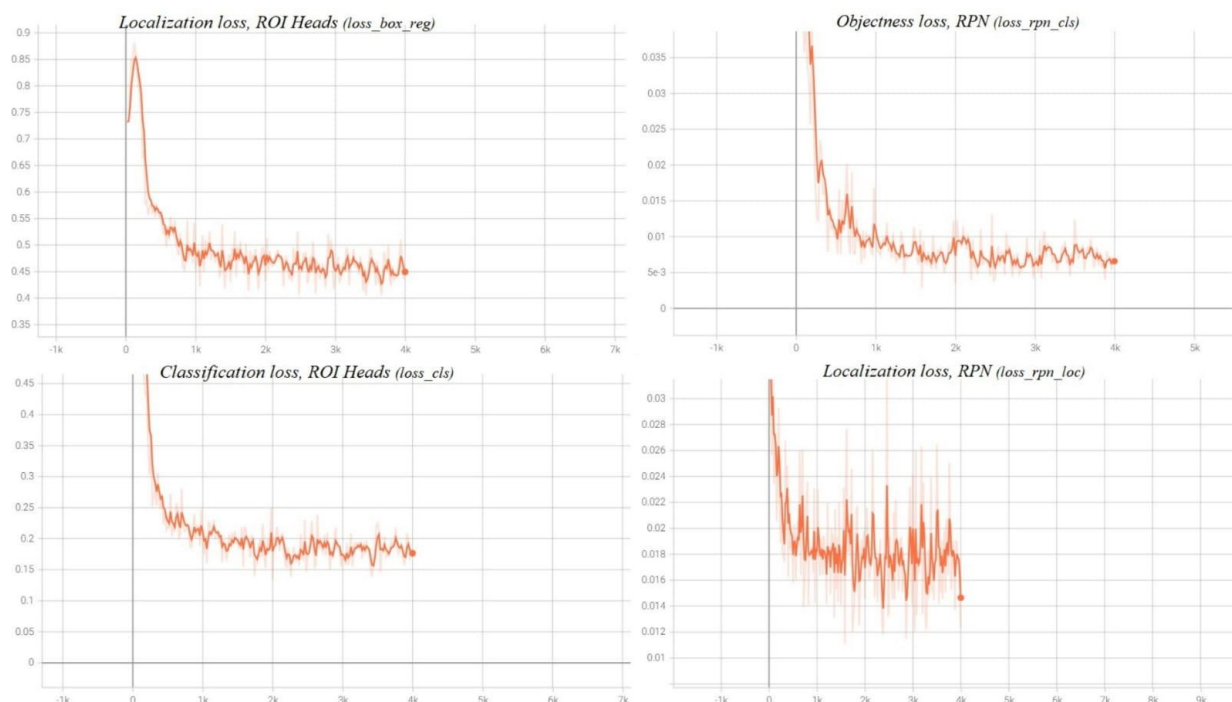
На Слици 40. је приказана промена вредности *Localization loss ROI*, *Classification loss ROI*, *Objectness loss RPN*, *Localization loss RPN* функција *Faster R-CNN* детектора изабраног са *Detectron2* платформе, током 2 000 итерација тренинга.



Слика 40. *Localization loss ROI*, *Classification loss ROI*, *Objectness loss RPN*, *Localization loss RPN*, *Detectron2* 2 000 итерација

Током тренинга вредност грешке за *loss_box_reg* је опала са 0.71 на 0.44, за *loss_cls* са 1.56 на 0.16, за *loss_rpn_cls* опада са 0.12 на 0.009а за *loss_rpn_loc* вредности се крећу између 0.03 на самом почетку, 0.01 као најмања достигнута вредност, а затим поново расте на 0.02 .

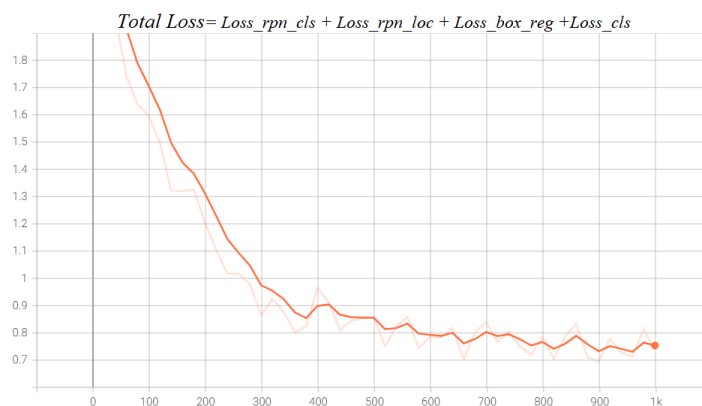
На Слици 41 је приказана промена вредности *Localization loss ROI*, *Classification loss ROI*, *Objectness loss RPN*, *Localization loss RPN* функција *Faster R-CNN* детектора изабраног са *Detectron2* платформе, током 4 000 итерација тренинга.



Слика 41. *Localization loss ROI*, *Classification loss ROI*, *Objectness loss RPN*, *Localization loss RPN*, *Detectron2* 4 000 итерација

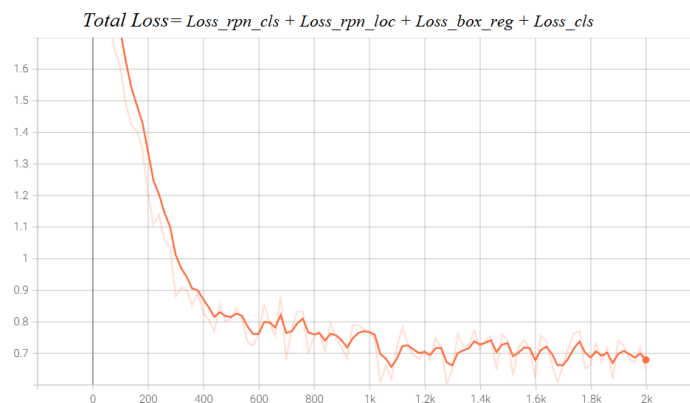
Током тренинга вредност грешке за *loss_box_reg* је опала са 0.71 на 0.44, за *loss_cls* са 1.56 на 0.16, за *loss_rpn_cls* опада са 0.11 на 0.009, а за *loss_rpn_loc* вредности се крећу између 0.03 на самом почетку, 0.01 као најмања достигнута вредност, а затим поново расте на 0.02 .

На Слици 42. је приказана промена вредности *Total Loss* функције током 1 000 итерација тренирања мреже, која има опадајући карактер.



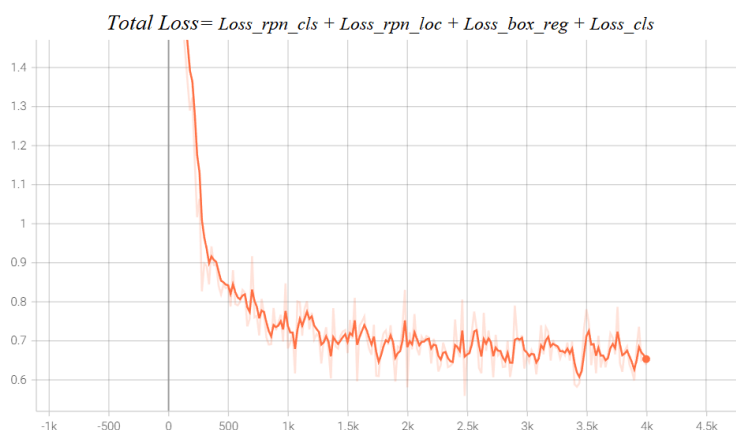
Слика 42. *Total Loss*, *Detectron2* 1 000 итерација

На Слици 43. је приказана промена вредности *Total Loss* функције током 2 000 итерација тренирања мреже.



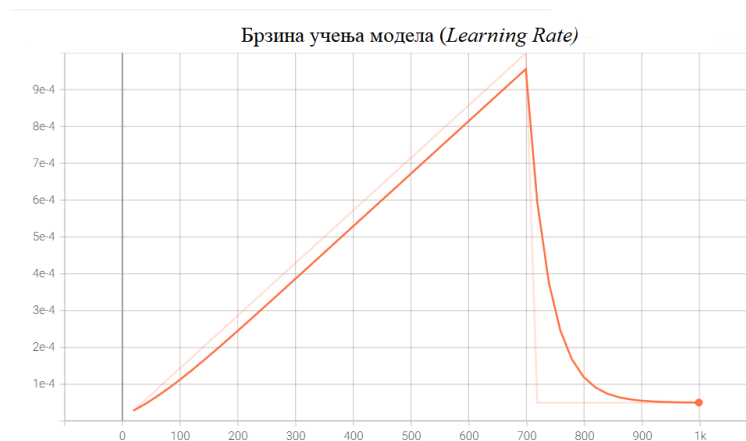
Слика 43. *Total Loss*, *Detectron2* 2 000 итерација

На Слици 44. је приказана промена вредности *Total Loss* функције током 4 000 итерација тренирања мреже.



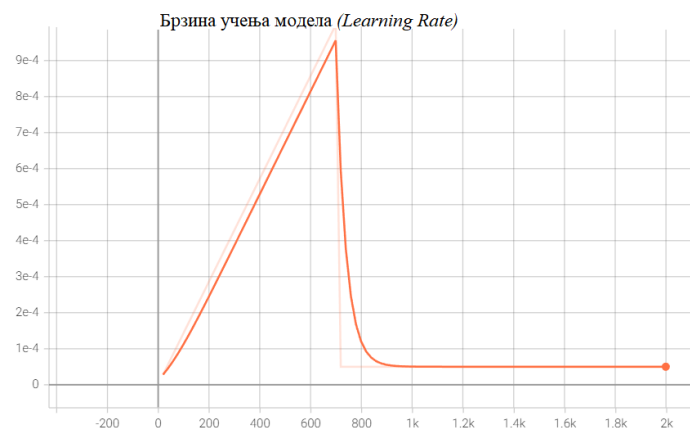
Слика 44. *Total Loss*, *Detectron2* 4 000 итерација

Са Слике 45. се може видети брзина учења модела кроз 1 000 итерација.



Слика 45. Брзина учења модела, *Detectron2* 1 000 итерација

На Слици 46. је приказана брзина учења модела кроз 2 000 итерација.



Слика 46. Брзина учења модела, *Detectron2* 2 000 итерација

На Слици 47 је приказана брзина учења модела кроз 4 000 итерација.



Слика 47. Брзина учења модела, *Detectron2* 4 000 итерација

4.2.2 Евалуација перформанси на подскупу за тест

Последња етапа подразумева да се модели са најбољим перформансама примене на издвојеном скупу података за тест који броји 583 фотографија са којима модел није упознат, а затим да се изврши евалуација перформанси на добијеним резултатима, упоредна анализа и избор модела који има највећу способност да што релевантније предвиди да ли особа носи маску или не.

Према резултатима експеримента као оптимална решења су изабрана *YOLOv5* трениран 100 епоха и *Faster R-CNN* са *Detectron2* платформе, трениран 2 000 итерација. Перформансе модела ће бити оцењене на подскупу података за тестирање. Ова подешавања су одабрана као најбоља решења ради избегавања потенцијалног проблема када модел није довољно научио (енг. *underfitting*) на 50 епоха односно 1 000 терација,

као и потенцијалног проблема да модел превише научи до најситнијег детаља (енг. *overfitting*) кроз 400 епоха односно 4 000 итерација. У Табели 15. су упоредно приказани резултати остварени на подскупу података за тест.

Табела 15. Резултати које остварују алгоритми на подскупу података за тест

Класа	Фотографија	Означено	YOLOv5 AP@.5	Detectron2 AP@.5	YOLOv5 AP@.5:.95	Detectron2 AP@.5:.95
Укупно	586	2318	73.30%	70.97%	33.00%	34.13%
<i>Mask</i>		1756			39.30%	39.24%
<i>No_mask</i>		562			26.80%	29.03%

Према приказаним резултатима *YOLOv5* и *Faster R-CNN* са *Detectron2* платформе постижу јако сличне резултате са 73.30% односно 33.00% прецизности по оштријем критеријуму за *YOLOv5*, и 70.97% односно 34.00% прецизности по оштријем критеријуму за *Detectron2*. Приликом тумачења резултата потребно је имати на уму време потребно за тренинг мреже, које је код *YOLOv5* алгоритама знатно краће, са друге стране када се у обзир узме број епоха и итерација, и када се број итерација преведе на број епоха, *Faster R-CNN* пролази кроз фотографије мањи број пута, при чему достиже сличну прецизност.

Наредни примери представљају излазне фотографије на којима су алгоритми извршили детекцију објеката.

На Слици 48. је приказана способност детекције *YOLOv5* алгоритма, када су објекти у крупном плану.



Слика 48. Излаз *YOLOv5* алгоритма, објекти су у крупном плану

На Слици 49. је приказана способност детекције *Faster R-CNN* алгоритма *Detectron2* платформе, када су објекти у крупном плану.



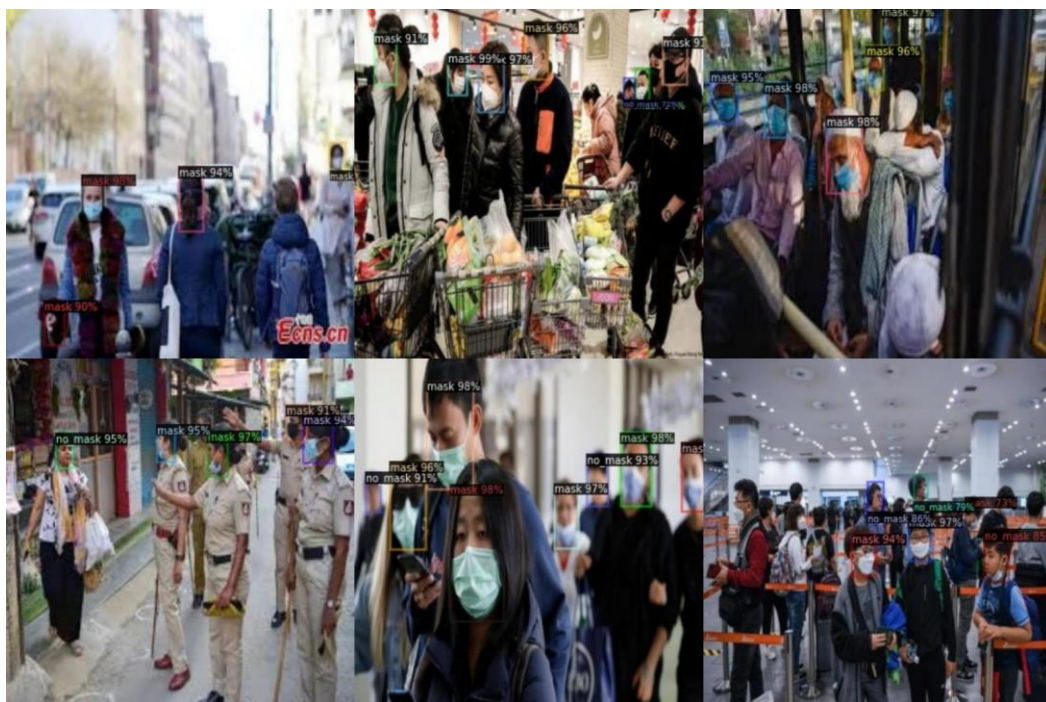
Слика 49. Излаз *Detectron2*, објекти су у крупном плану

На Слици 50. је приказана способност детекције *YOLOv5* алгоритма, када су објекти збијени и у гужви.



Слика 50. Излаз *YOLOv5* алгоритма, објекти су збијени

На Слици 51. је приказана способност детекције *Faster R-CNN* алгоритма *Detectron2* платформе, када су објекти збијени и у гужви.



Слика 51. Излаз *Detectron2*, објекти су збијени

На Слици 52. је дат упоредни приказ способности детекције *YOLOv5* алгоритма на левој страни и *Faster R-CNN* алгоритма *Detectron2* платформе на десној страни.



Слика 52. Упоредни приказ 1, *YOLOv5* VS *Detectron2*

5. ЗАКЉУЧАК

Постоји велики број иновативних алгоритама за детекцију објеката, такође и велики број варијација истог модела које се могу применити на конкретни проблем. Узимајући ово у обзир, проналазак најбољег модела за решавање проблема и постизање жељених резултата може бити веома изазовно. Задатак одабира модела може се олакшати ако се узму у обзир следећи фактори: величина скупа података, величина модела, потребни ресурси које модел црпи приликом тренинга, брзина и прецизност коју модел може да оствари.

Имајући на уму наведене факторе може се извести следећи закључак. Уколико је скуп података који се користи за обучавање мреже мањи, са на пример 300-500 фотографија бољи одабир ће бити *Faster R-CNN* детектор одабран са *Detectron2* платформе, који ће постићи већу прецизност него *YOLOv5*, са друге стране уколико је скуп података велики, као што јесте у овом случају, са више од 2 000 фотографија за обучавање, онда неће постојати велика разлика у прецизности детекције коју остварује *YOLOv5* и коју остварује *Faster R-CNN*, али би постојала разлика у брзини, где би *YOLOv5* био знатно бржи. Уколико се посматра величина модела и потребни ресурси *YOLOv5* има једноставнију архитектуру од *Faster R-CNN* детектора, стога и од читаве библиотеке *Detectron2* па самим тим захтева мање ресурса, па у случају да нема много велике разлике између постигнутих прецизности или једноставно није толико велики акценат на прецизности, бољи одабир би био *YOLOv5* алгоритам. Уколико је акценат само на прецизности, без обзира на утрошено време и ресурсе, прави избор јесте *Detectron2* платформа с обзиром да поред *Faster R-CNN* детектора нуди више различитих опција за одабир основног модела, укључујући технике детекције објеката, семантичке сегментације, индентификације кључних тачака. Иако избор *Detectron2* платформе и *Faster R-CNN* детектора пружа велику тачност детекције објеката, при томе се компензује утрошено време. Ипак ако је акценат на брзини детекције, као што би то био случај за детекцију објеката у реалном времену, онда је прави избор *YOLOv5* алгоритам који при великој брзини детекције објеката остварује и задовољавајућу прецизност.

Када су у питању случајеви употребе детекције објеката у будућности, постоји безброј могућности. Иако је последњих година остварен велики напредак на пољу детекције објеката, захваљујући константној еволуцији вештачке интелигенције, тек се очекују нова открића на пољу детекције објеката. До сада је детекција објеката интегрисана у

системе аутономних возила, системе детекције лица на мобилним телефонима, али постоје многи системи у којима она може наћи примену као што су дронови, роботи помоћници који би обављали досадне и опасне послове уместо људи, роботи и нано-роботи који ће истраживати подручја на која људи не могу да ступе и која нису видели као што су велике дубине мора или друге планете, као и многе друге примене (Verschae & Ruiz-del-Solar, 2015).

6. ЛИТЕРАТУРА

Ananth S. (2019) *Fast R-CNN for object detection* [<https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022>, датум приступа 2.9.2022.]

Chingis (2021) *Object Detection Explained: Faster R-CNN* [<https://medium.com/mllearning-ai/object-detection-explained-faster-r-cnn-23e7ab57991d>, датум приступа 6.9.2022.]

Chingis (2021) *Object Detection Explained: R-CNN* [<https://towardsdatascience.com/object-detection-explained-r-cnn-a6c813937a76>, датум приступа 2.9.2022.]

Chinmoy B. (2020). *Evolution of Object Detection* [<https://medium.com/analytics-vidhya/evolution-of-object-detection-582259d2aa9b> , датим приступа 18.8.2022.]

Dubey V. (2020) *Evaluation Metrics for Object Detection algorithms* [<https://medium.com/@vijayshankerdubey550/evaluation-metrics-for-object-detection-algorithms-b0d6489879f3>, 30.7.2022.]

Fritz.ai (2021) [<https://www.fritz.ai/object-detection/> датум приступа 18.8.2022]

Guo, S., Li, L., Guo, T., Cao, Y., & Li, Y. (2022). Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5. *Sensors*, 22(13), 4933, 3-6.

Gutta S. (2021) *Object Detection Algorithm- YOLOv5 Arhitecture* [<https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef>, датум приступа 20.7.2022.]

Honda H. (2022) *Digging into Detectron2-part 1:5* [<https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd>, датум приступа 28.7.2022.]

Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J. A., & Hou, W. (2021). IoT and deep learning based approach for rapid screening and face mask detection for infection spread control of COVID-19. *Applied Sciences*, 11(8), 3495.

Ieamsaard, J., Charoensook, S. N., & Yammen, S. (2021, March). Deep learning-based face mask detection using yolov5. In *2021 9th International Electrical Engineering Congress (iEECON)* (pp. 428-431). IEEE.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066-1073.

Kundrotaite R. (2021) *The Evolution of Object Detection* [<https://www.sentisight.ai/the-evolution-of-object-detection/>, датум приступа 18.8.2022.]

Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167, 108288.

Munawar R. M. (2022) *How do Hyperparameters of YOLOv5 Work?* [<https://chr043416.medium.com/how-hyperparameters-of-yolov5-works-ec4d25f311a2>, датум приступа 27.7.2022.]

Nguyen, C. C., Tran, G. S., Nghiem, T. P., Doan, N. Q., Gratadour, D., Burie, J. C., & Luong, C. M. (2018, April). Towards real-time smile detection based on faster region convolutional neural network. In *2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)* (pp. 1-6). IEEE, 3-5.

Patel M. (2020) *YOLOv5- Explained And Demystified* [<https://towardsai.net/p/computer-vision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-and-demystified>, датим приступа 26.7.2022.]

Qu, R., Yang, Y., & Wang, Y. (2021, December). COVID-19 Detection Using CT Image Based On YOLOv5 Network. In *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)* (pp. 622-625). IEEE.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Rizzoli A. (2022) *The Ultimate Guide to Object Detection* [<https://www.v7labs.com/blog/object-detection-guide>, датум приступа 18.8.2022]

Roy B., Nandy S., Ghosh D., Dutta D., Biswas P., Das T. (2020). *MOXA: A Deep Learning Based Unmanned Approach For Real-Time Monitoring of People Wearing Medical Masks*. [<https://doi.org/10.1007/s41403-020-00157-z>, датум приступа 18.7.2022.]

Snyder, S. E., & Husari, G. (2021, March). Thor: A deep learning approach for face mask detection to prevent the COVID-19 pandemic. In *SoutheastCon 2021* (pp. 1-8). IEEE.

Solawetz J. (2020) *YOLOv5 New Version- Improvements And Evaluation* [<https://blog.roboflow.com/yolov5-improvements-and-evaluation/>, датум приступа 20.7.2022.]

Song, Q., Li, S., Bai, Q., Yang, J., Zhang, X., Li, Z., & Duan, Z. (2021). Object Detection Method for Grasping Robot Based on Improved YOLOv5. *Micromachines*, 12(11), 1273, 7-12.

Subramanyam S. V. (2021) *Non Max Suppression (NMS)* [<https://medium.com/analytics-vidhya/non-max-suppression-nms-6623e6572536>, датум приступа 26.7.2022.]

Thuan, D. (2021). Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm.

Vahab, A., Naik, M. S., Raikar, P. G., & Prasad, S. R. (2019). Applications of object detection system. *International Research Journal of Engineering and Technology (IRJET)*, 6(4), 4186-4192.

Verschae, R., & Ruiz-del-Solar, J. (2015). Object detection: current and future directions. *Frontiers in Robotics and AI*, 2, 29, 4-5.

Yadav, S. (2020). Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence. *Int. J. Res. Appl. Sci. Eng. Technol*, 8(7), 1368-1375.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 1-5.

[https://patrick-lilgc.github.io/Learning-Deep-Learning/paper_notes/giou.html, датум приступа 26.7.2022.]

AI.Facebook.com (2019) *Detectron2: A PyTorch-based modular object detection library* [<https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/>, датум приступа 22.8.2022]

Ibm.com [<https://www.ibm.com/topics/computer-vision>, датум приступа 18.8.2022.]

Mathworks.com [https://www.mathworks.com/discovery/object-detection.html, датум приступа 18.8.2022.]

Models.roboflow.com (2019) [https://models.roboflow.com/, датум приступа 18.8.2022]

Roboflow.com (2019) [https://roboflow.com/, датум приступа 19.7.2022.]

Shitty-bots-inc.github.io [https://shitty-bots-inc.github.io/MOXA/index.html, датум приступа 18.7.2022.]