Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Факултет за информатички науки и компјутерско инженерство – Скопје

Семинарска работа по предметот „Дигитално процесирање на слика”

Тема:

Детекција на маска на лице

Ментор:
Проф. Ивица Димитровски

Изработиле:
Димитар Балоски –221068
Софија Речкоска – 221064

Содржина

1. Вовед

Со појавата на глобалната пандемија на COVID-19, заштитните маски станаа клучен дел од мерките за јавно здравје, играјќи значајна улога во намалувањето на ширењето на вирусот. Во контекст на јавната безбедност, особено во затворени простори и јавни собири, детекцијата на маски на лицето се покажа како витално средство за надгледување и спроведување на здравствените мерки.

Системите за автоматска детекција на маска користат техники од областа на компјутерскиот вид и машинското учење за да препознаат дали лицата на сликите или видеата носат маски. Овие технологии овозможуваат не само брза и ефикасна идентификација на оние што ги почитуваат правилата, туку и на оние што не ги следат, што помага во превенција и контрола на инфекции.

YOLO е еден од најпопуларните и најефикасни алгоритми за детекција на објекти во реално време. Со неговата еволуција од првата до последната верзија, YOLO се подобруваше во однос на точноста и брзината, што го направи погоден за широка употреба во различни апликации. YOLOv8, како најнова верзија на овој алгоритам, претставува значителен напредок во поглед на прецизноста и ефикасноста во споредба со претходните верзии, што го прави идеален избор за задачи како што е детекцијата на маски.

Целта на овој проект е да се имплементира систем за автоматска детекција на заштитни маски, кој ќе се користи за анализа на слики и видеа. Преку овој систем, ќе се испита ефикасноста и точноста на детекцијата во различни услови и сценарија, како и неговата способност за работа во реално време. Проектот ќе овозможи практична примена на технологијата, со цел да се утврди нејзината употребливост и можности за подобрување во реални ситуации.

2. Теоретска основа за YOLOv8

YOLOv8 е најновата итерација на моделите за детекција на објекти од семејството You Only Look Once (YOLO), познати по нивната брзина и прецизност. Развиена од тимот на Ultralytics, YOLOv8 го гради успехот на своите претходници, воведувајќи неколку клучни иновации кои ја унапредуваат детекцијата на објекти во реално време.

Детекцијата на објекти се состои од идентификација и лоцирање на објекти од интерес во слика или видео. Традиционалните методи често се потпираа на пристапи со лизгачки прозорци, кои беа пресметковно сложени и бавни. YOLO ја револуционизира оваа област со третирање на детекцијата на објекти како единствен проблем на регресија.

Наместо користење на лизгачки прозорци, YOLO директно предвидува bounding boxes и веројатности за класификација на објектите од влезната слика со едно поминување напред, што значително ја зголемува брзината на процесирање.

2.1. Архитектура на YOLOv8

Алгоритмот на YOLOv8 се состои од три основни блока, каде што се одвиваат сите клучни процеси: **`Рбет`**, **`Врат`,** и **`Глава`**. Секој од овие блокови игра важна улога во детекцијата на објекти, и нивната функција е опишана подолу:

- ***'Рбет (Backbone):***
  Функција:
  `Рбетот`, или извлекувачот на карактеристики, е одговорен за идентификување и извлекување на значајни карактеристики од влезната слика. Тоа е основата на мрежата, која ги обработува сировите податоци и создава почетни слоеви на информации.

Активности:
- Во почетните слоеви, `Рбетот` снима основни обрасци, како што се рабови и текстури, кои се критични за понатамошната анализа.
- Овој дел од мрежата доловува карактеристики на различни нивоа на апстракција, што значи дека може да препознае сложени обрасци и структури како што се движите низ мрежата.
- Обезбедува богата, хиерархиска претстава на влезната слика, подготвувајќи ги карактеристиките за понатамошна обработка во следните фази.

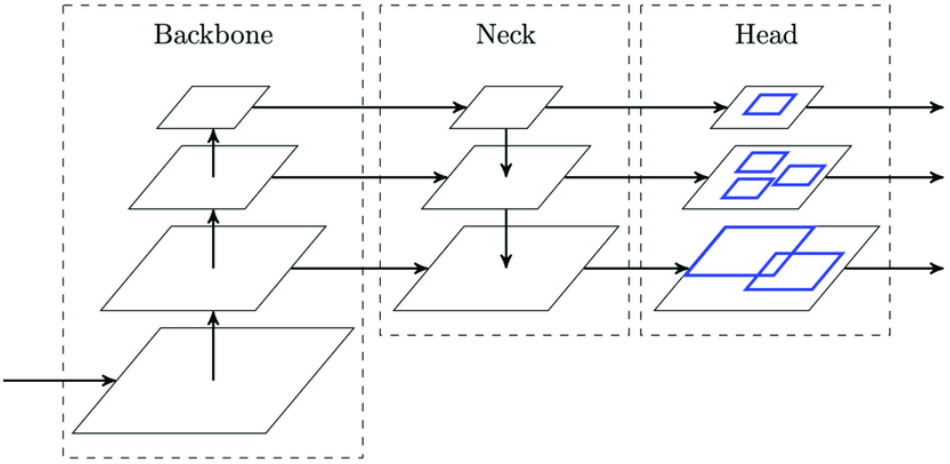
- ***`Врат` (Neck):***
  Функција:
  `Вратот` функционира како мост помеѓу `Рбетот` и `Главата`, овозможувајќи интеграција и соединување на карактеристиките. Тој собира и комбинира информации од различни нивоа на `Рбетот`, создавајќи пирамиди од карактеристики што се користат за откривање на објекти со различни големини.

  Активности:
  - Спојува и комбинира карактеристики од различни скали за да осигури дека мрежата може да идентификува објекти од различни големини и во различни делови на сликата.
  - Интегрира контекстуални информации, земајќи ги предвид пошироките делови од сцената, што помага да се подобри точноста на детекцијата.
  - Ја намалува просторната резолуција и димензионалноста на карактеристиките за да се зголеми брзината на процесирање, иако тоа може да влијае на квалитетот на детекцијата.

- ***`Глава` (Head):***
  Функција:
  `Главата` е завршниот дел од мрежата и е задолжена за генерирање на конечните излези, како што се bounding boxes (ограничувачки кутии) и оценките за доверба што се користат за финалната детекција на објекти.

  Активности:
  - Генерира bounding boxes кои го опкружуваат секој можен објект на сликата, што помага во нивната локација и идентификација.

- Доделува оценки за доверба на секоја кутија, кои укажуваат на веројатноста дека навистина постои објект на тоа место.
- Ги класифицира објектите според нивните категории, што овозможува финална идентификација на видот на објектите што се детектирани.



Слика 1. (Архитектура на YOLOv8)

Овие три блока работат заедно за да овозможат YOLOv8 да постигне брза и точна детекција на објекти, дури и во комплексни и динамични сценарија.[1]

2.2. Принцип на функционирање на YOLOv8 за детекција на објекти

YOLOv8 го применува својот моќен модел за детекција на објекти преку серија од чекори што овозможуваат прецизна и брза анализа на сликите. Процесот започнува со внесување на сликата во мрежата, каде што таа се обработува за да се идентификуваат различни објекти. Еве како овој процес функционира детално:

I. Поделба на сликата во мрежа:

- YOLOv8 ја дели влезната слика на мрежа од ќелии. Секоја ќелија има задача да предвиди дали во нејзиниот домен постои објект и ако постои, да ги предвиди неговите координати и класификација. Оваа поделба овозможува паралелна анализа на различни делови од сликата, што значително го зголемува времето на процесирање.

II. Екстракција на карактеристики:

- Во оваа фаза, `Рбетот` на мрежата ја обработува сликата за да извлече значајни карактеристики, како што се форми, рабови, текстури и други визуелни елементи. Овие карактеристики се користат за идентификација на потенцијалните објекти.

III. Спојување на карактеристики:

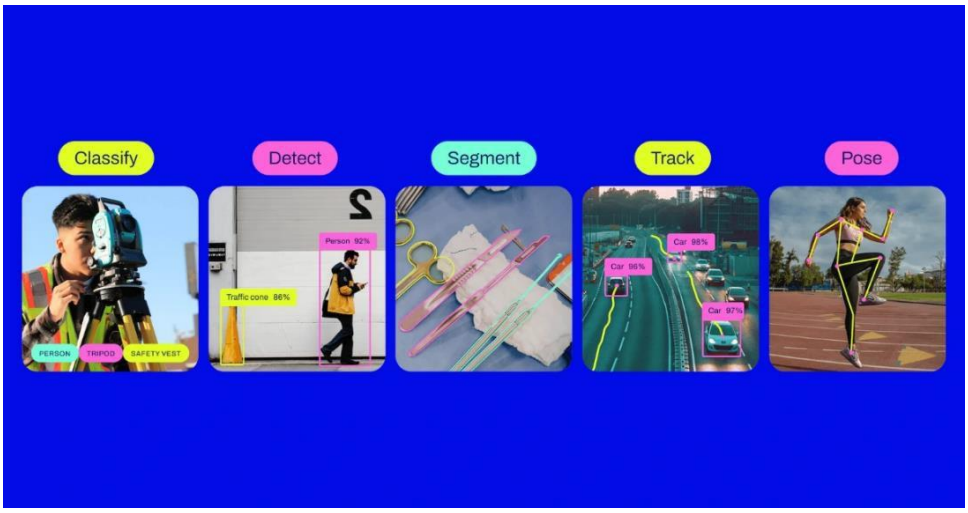
- Потоа, `Вратот` ги комбинира карактеристиките извлечени од различни делови на `Рбетот`, креирајќи пирамиди на карактеристики кои ги содржат информации на различни скали. Ова овозможува YOLOv8 да препознае објекти со различни големини и во различни контексти.

IV. Предвидување на bounding boxes и класификација:

- `Главата` на мрежата е задолжена за генерирање на ограничувачки кутии (bounding boxes) за секој детектиран објект. Покрај тоа, таа доделува оценки за доверба на секоја кутија, што укажува колку е веројатно дека навистина постои објект на таа локација. Мрежата исто така врши класификација на објектите, доделувајќи им категории (во случајов, "со маска", "без маска" или "маска носена неправилно").

V. Извлекување на резултатите:

- На крајот, YOLOv8 ги филтрира резултатите за да ги задржи само оние предвидувања кои имаат највисоки оценки за доверба. Овие резултати се претставуваат како финални излези, каде што секој објект е јасно означен на сликата со неговата ограничувачка кутија и соодветна класификација.



Слика 2. (Чекори за детекција на објекти)

Овој интегриран пристап овозможува YOLOv8 да изведува детекција на објекти во реално време со висока прецизност, што го прави овој модел еден од најдобрите избори за задачи како препознавање на лица со или без маска.[2]

3. Прибирање и подготвување на податоци

Процесот на прибирање и подготвување на податоците е критичен за успешната имплементација на YOLOv8 за задачи како детекција на маски. Во ваков проект, датасетот вклучува слики на лица со три категории: лица со правилно носена маска, без маска, и лица кои носат маска неправилно. Овој процес вклучува неколку клучни чекори:

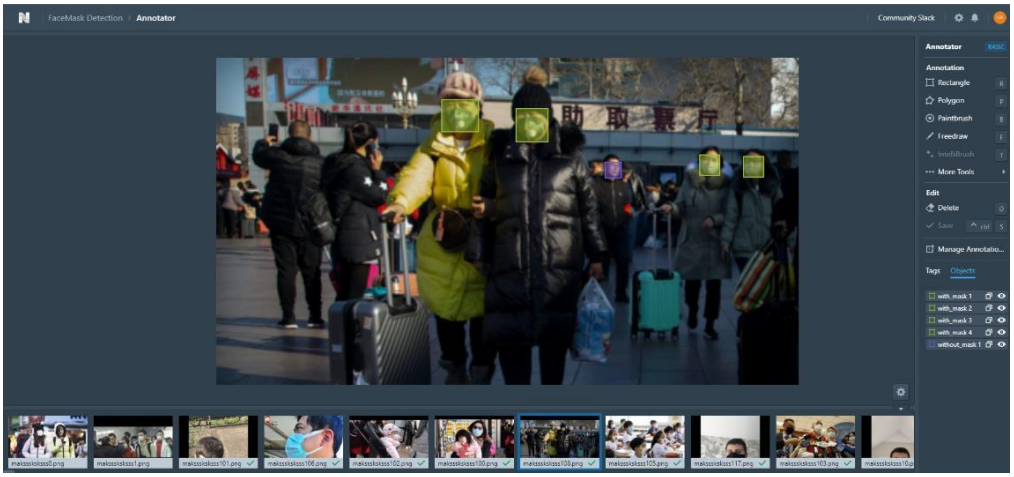
3.1. Избор на база на податоци за тренинг

Првиот чекор е изборот на соодветна база на податоци што ќе содржи слики кои ги опфаќаат трите категории на лица: со маска, без маска и со неправилно носена маска. Потребна

е база која ги отсликува различните услови на носење маска, за да се осигура дека моделот ќе може да се справи со различни ситуации во реалниот свет.

3.2. Аннотирање на податоците

Откако е избран датасетот, следниот чекор е аннотирање на сликите. Во случајот со детекција на маски, тоа значи да се обележат областите каде што се наоѓаат лицата, заедно со класификација во една од трите категории: „со маска“, „без маска“ и „маска носена неправилно“. Аннотирањето се врши со алатки кои овозможуваат поставување bounding boxes околу лицата, доделувајќи ја соодветната класа на секој објект. Прецизноста на аннотирањето е клучна за постигнување на добри резултати, бидејќи неправилно означени податоци можат да го намалат квалитетот на моделот.



Слика 3. (Аннотирање на слика)

3.3. Поделба на податоците на тренинг, валидација и тест сетови

Со цел моделот правилно да се обучи и евалуира, податоците треба да се поделат на тренинг, валидација и тест сетови.

- ***Тренинг сет***

  Збир на податоци што се користат за учење на моделот. За време на тренингот, моделот анализира слики од овој сет за да ги препознае објектите и да научи како да ги класифицира, на пример, лица со маска, без маска и со неправилно носена маска. Квалитетот и разновидноста на тренинг сетот се клучни за добрите перформанси на моделот при реална употреба.

- ***Сет за валидација***

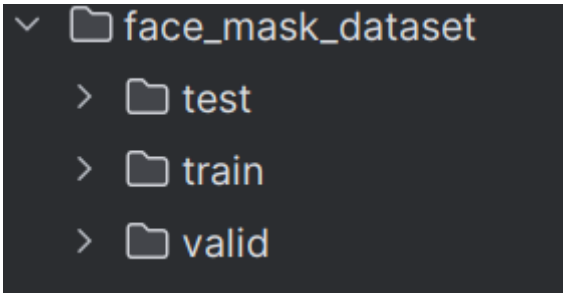
  Сетот за валидација се користи за проценка на перформансите на моделот за време на тренингот. Овој сет не учествува директно во учењето, туку служи за проверка на точноста на моделот и за прилагодување на хиперпараметрите, како што се стапката на учење или бројот на епохи. Целта е да се спречи пренавикнување на моделот на тренинг податоците и да се осигура дека ќе се однесува добро на нови, невидени податоци.

- ***Тест сет***

  Тест сетот се користи за непристрасна евалуација на целосно обучениот модел. Се применува откако моделот е истриниран со тренинг и валидација сетови, за да се

оцени неговата способност за работа на нови, невидени податоци. Овој сет претставува "златен стандард" за оценување и е клучен при споредба на различни модели.

Тест сетот обично е внимателно куриран, вклучувајќи примери кои ги претставуваат сите класи на проблемот, за да се осигура дека моделот ќе биде ефикасен во различни ситуации. За разлика од валидација сетот, тест сетот се користи само за финална евалуација и е од суштинско значење за објективно оценување на перформансите на моделот.[3]



Слика 4 .(Структура на dataset)

4. Тренинг на моделот

Тренингот на моделот за детекција на маски со YOLOv8 е комплексен процес кој вклучува конфигурирање на моделот, оптимизација на хиперпараметрите и следење на перформансите. Секој од овие чекори е клучен за постигнување на оптимални резултати. Подолу е подетален опис на секој чекор:

4.1. Конфигурирање на YOLOv8

4.1.1. Архитектура на моделот

За задачата на детекција на маски, треба да се осигура дека архитектурата на моделот е прилагодена на бројот на класи и спецификата на задачата. Во случајов, моделот треба да биде конфигуриран да детектира три класи: "со маска", "без маска" и "маска носена неправилно". Ова се постигнува со модификација на излезниот слој на моделот за да се одрази бројот на класи.

4.1.2. Конфигурација на податоци

Податоците треба да бидат подготвени во формат кој YOLOv8 го поддржува. Овој формат обично вклучува слики и соодветни текстуални датотеки со ознаки. Секој запис во текстуалните датотеки треба да вклучува координати на bounding box и класификација на објектите (1 за СО МАСКА, 0 за БЕЗ МАСКА). Правилниот формат овозможува моделот да ги прочита и разбере податоците во текот на тренингот.

```
1 0.181640625 0.33469945355191255 0.05859375 0.10109289617486339
0 0.3994140625 0.33060109289617484 0.080078125 0.12021857923497267
1 0.6669921875 0.3128415300546448 0.068359375 0.13934426229508196
```

Слика 5. (Текстуална датотека со ознаки за одредена слика)

4.2. Хиперпараметри и нивна оптимизација

Хиперпараметрите играат важна улога во тренингот на YOLOv8 и нивната оптимизација е клучна за постигнување на добри резултати. Основните хиперпараметри и нивната оптимизација вклучуваат:

- *Број на епохи*: Бројот на епохи е бројот на целосни преминувања низ тренинг податоците. Премногу мал број на епохи може да резултира со недоволно обучен модел, додека премногу голем број може да доведе до пренасочување (overfitting). Често се користи рано стопирање (early stopping) како техника за да се спречи пренасочување.

- *Batch Size*: Ова е бројот на примери што се обработуваат одеднаш. Поголем број за batch size може да ја забрза обуката, но може да бара поголема меморија. Помала големина на партијата може да доведе до помалку стабилно учење, но понекогаш може да биде поефикасна за мали машини со помалку ресурси.

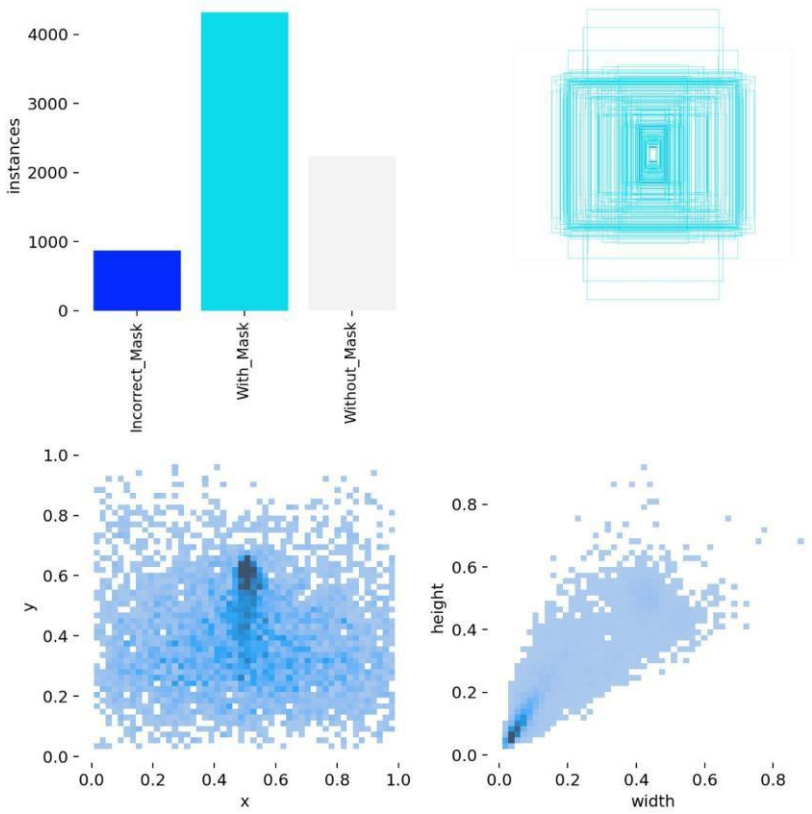
- *Image size*: За време на тренингот, сите слики се рескалираат на големината дефинирана со imgsz. Ова обезбедува конзистентност на податочниот сет, што е важно за стабилен тренинг. Во случајов вредноста на imgsz е поставена на 800, а тоа значи дека сите слики ќе се рескалираат на 800x800px.

- *Resume*: Овозможува продолжување на тренингот од претходно зачуван checkpoint, што е корисно ако тренингот е прекинат. Со ова, моделот ги вчитува претходно зачуваните тежини и продолжува со учењето без потреба да се почне од почеток. Овој пристап заштедува време и ресурси, овозможувајќи ефикасен и стабилен тренинг, при што моделот ги задржува сите претходно научени карактеристики.

```python
import numpy as np
import cv2


from ultralytics import YOLO


model = YOLO("yolov8n.yaml")  # build a new model from YAML


# # Training the model
results = model.train(data="data.yaml", epochs=13, resume=True, batch=8, imgsz=800)

```

Слика 6. (Хиперпараметри на моделот)

4.3. Графици на перформанси



Слика 7. (Анализа на податоците добиени после тренинг на моделот)

- *Бар графиконот* ја илустрира распределбата на категориите на носење маски, откривајќи дека најголем број од примерите се од луѓе кои правилно носат маски, што би можело да влијае на балансот при тренинг на моделот. Дисбалансот меѓу категориите (помалку примери за „Без маска“ и „Погрешно носење маска“) може да доведе до потреба за дополнителни техники како oversampling или подесување на тежините за порамнување на класификацијата.

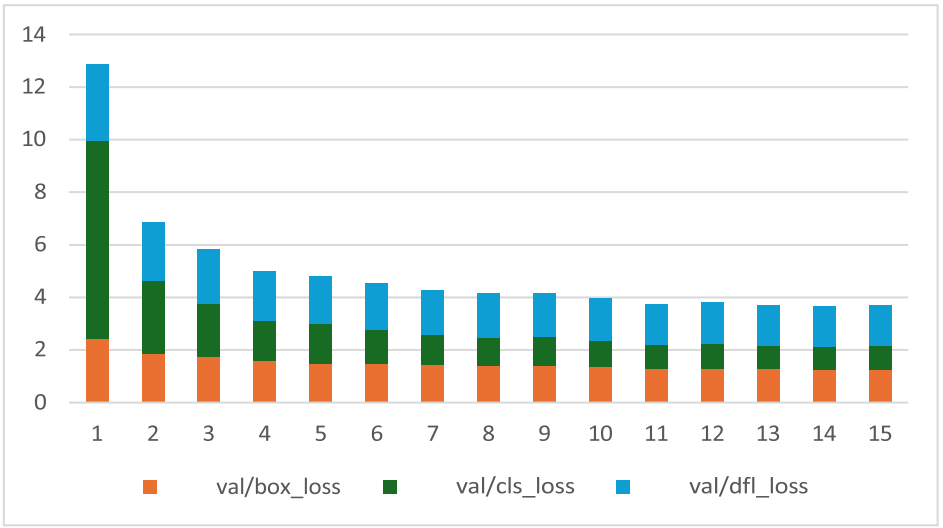
- *Bounding box визуелизацијата* покажува дека повеќето рамки се преклопуваат, укажувајќи на тоа дека детектираните објекти (лица) се во слични позиции на сликите. Ова е корисно за утврдување на точноста на позиционирањето на објектите.

- *Scatter графиконот за координатите (x, y)* укажува на тоа дека повеќето објекти се концентрирани во централниот дел од сликата. Ова може да биде предност за детекцијата на лица, бидејќи тие обично се наоѓаат во центарот на сликите, но исто така треба да се внимава на детекциите во аглите и периферните делови на сликите.

- *Scatter графиконот за ширина и висина* покажува дека повеќето објекти се со мали димензии, што може да укаже на релативна униформност на големината на лицата во податочниот сет.

Слика 8. (Перофрманси по завршување на тренингот)

Како што се зголемува бројот на epoch, train/box_loss, train/cls_loss, и train/dfl_loss генерално опаѓаат, што укажува на подобрување во точноста на предвидувањето на bounding box-овите, класификацијата и Discriminative Feature Learning.



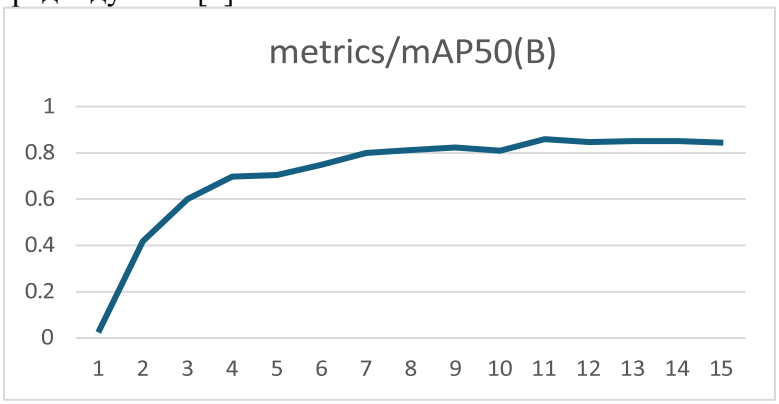
Слика 9. (Перофрманси за валид сетот)

Загубите на валидирачкиот сет (val/box_loss, val/cls_loss, val/dfl_loss) се пониски во подоцнежните epoch, што значи дека моделот генерално постигнува подобри резултати и на валидирачкиот сет.

4.4. Оценување на перформансите на моделот

Следењето на перформансите е важно за евалуација на успехот на тренингот и за правење на потребни прилагодувања. Овој процес вклучува:

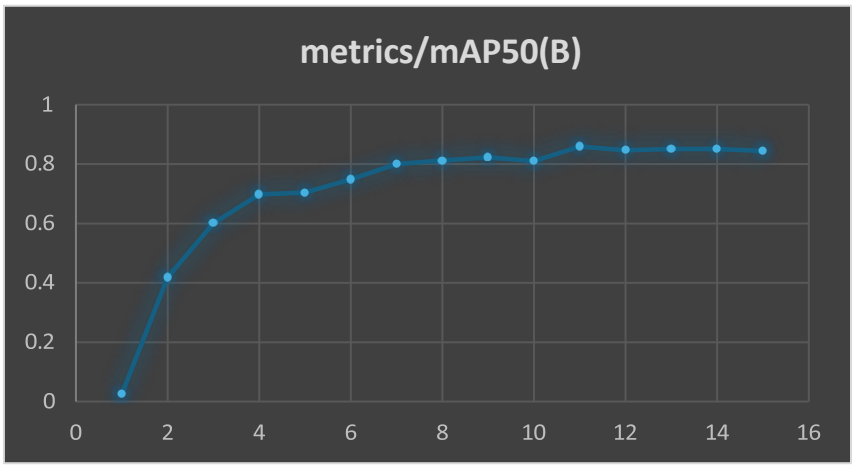
- ***Метрики:*** Клучните метрики за оценување на перформансите на моделот вклучуваат:
  - *Precision (Прецизност)*: Мерење на точноста на позитивните предвидувања, пресметано како односот на точните позитивни предвидувања кон вкупниот број на примери што моделот ги предвидел како позитивни (и вистинити и погрешни позитивни). Висока прецизност значи дека моделот ретко прави погрешни позитивни предвидувања.[4]



Слика 11. (Резултати за прецизност)

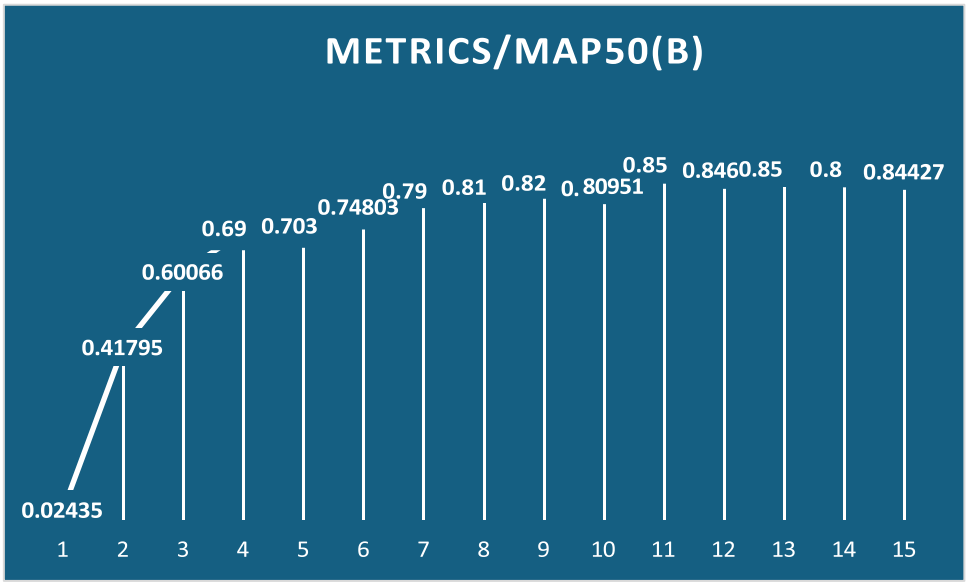
Како што може да се забележи на графикот, прецизноста се зголемува со зголемување на епохите, што дава знак дека моделот прави минимални погрешни предвидувања

- *Recall* : Мерење на способноста на моделот да ги открие сите вистински позитивни примери, пресметано како односот на вистински позитивни предвидувања кон сите позитивни примери. Висок повик значи дека моделот не пропушта многу вистински позитивни примери.[5]



Слика 12. (Резултати за recall)

- *mAP (Mean Average Precision):* Средна вредност на точноста на различни класи. Оваа метрика обезбедува глобален преглед на перформансите на моделот на сите класи и помага да се утврди колку е успешен моделот во детекцијата на различни типови објекти.[6]



Слика 13. (Резултати за mAP)

Според графикот најдобра вредност за mAP е во 11-та епоха, што значи дека во тој момент моделот постигнува висок процент на точно предвидени објекти (висока прецизност), додека истовремено успева да детектира поголем број од вкупните објекти во сликата (висок recall).

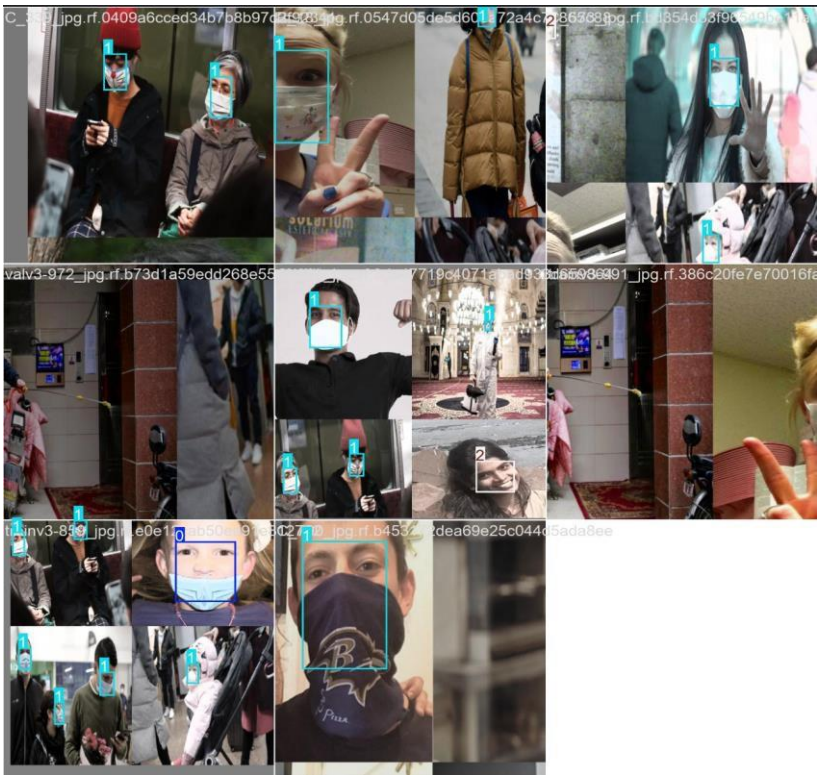
Следењето и евалуацијата на овие метрики помагаат во процесот на прилагодување на хиперпараметрите и унапредување на моделот. Тековното прилагодување и оптимизација на моделот на основа на овие метрики го подобруваат неговото успешно справување со нови и непознати податоци.

4.5. Резултати од тренинг фазата

По завршувањето на тренинг фазата на YOLOv8 моделот, резултатите покажуваат дека моделот постигнува значајни перформанси во детекција на маски, со особено внимание на

точноста на селекцијата на лицата и класификацијата на соодветните категории. Категориите се класифицирани како:

- 1 : Со маска
- 2 : Без маска
- 3 : Маска носена неправилно



Слика 10. (Резултати од тренинг фазата)

5. Тестирање на моделот

Тестирањето на моделот за детекција на маски е клучен чекор во процесот на машинско учење, бидејќи овозможува да се оцени неговата точност, ефикасност и способност за генерализација врз нови податоци. Овој процес не само што го потврдува квалитетот на моделот, туку и овозможува откривање на потенцијални слабости и можности за подобрување.

5.1. Подготовка на тест сетот

Тестирањето започнува со избор на соодветен тест сет — збирка на податоци што не биле користени за време на тренингот на моделот. Овие податоци треба да бидат репрезентативни за сценаријата во кои моделот ќе се користи. Во случајот на детекција на маски, тест сетот треба да вклучува:

- Лица со правилно носени маски
- Лица без маски
- Лица со неправилно носени маски

- Различни агли, осветлување, и позадини за да се процени како моделот се снаоѓа во различни услови.

Оваа разновидност е важна за да се осигура дека моделот може добро да се справи со реални услови, а не само со идеализирани слики од тренингот.

5.2. Примена на моделот врз тест податоците

Откако тест сетот е подготвен, моделот се применува врз сликите од тест сетот. Ова значи дека за секоја слика моделот прави едно поминување напред (forward pass) низ нејзините пиксели и генерира bounding boxes, како и соодветни ознаки за тие кутии. Процесот на детекција ги вклучува следните чекори:

1. ***Предобработка на сликите***: Моделот прво ја предобработува секоја слика со цел да ги прилагоди нејзините димензии на очекуваниот влез на YOLOv8. Ова вклучува промена на големината и нормализација на пиксел вредностите, со цел да се осигури конзистентност помеѓу тренинг и тест сликите.

2. ***Детекција на објекти***: Кога сликата е подготвена, YOLOv8 го применува своето 'рбет, врат и глава (backbone, neck, head) за да изврши извлекување на карактеристики, фузија на информации и предвидување на објекти. Моделот генерира bounding boxes за сите региони во сликата кои ги смета за потенцијални објекти (лица со маски, без маски или со неправилно носени маски).

3. ***Оценки за доверба***: Покрај генерирањето на ограничувачките кутии, YOLOv8 доделува оценка за доверба на секоја кутија. Оваа оценка покажува колку моделот е сигурен дека во таа кутија се наоѓа одреден објект (на пример, лице со маска). Оценките за доверба се клучни за филтрирање на потенцијални грешки. На пример, ако некоја кутија има ниска доверба, таа може да се отфрли.

4. ***Класификација на објекти***: Откако ќе ги предвиди ограничувачките кутии, моделот ги класифицира објектите што ги пронајде. Во нашиот случај, објектите ќе бидат класифицирани како со маска, без маска или маска носена неправилно.

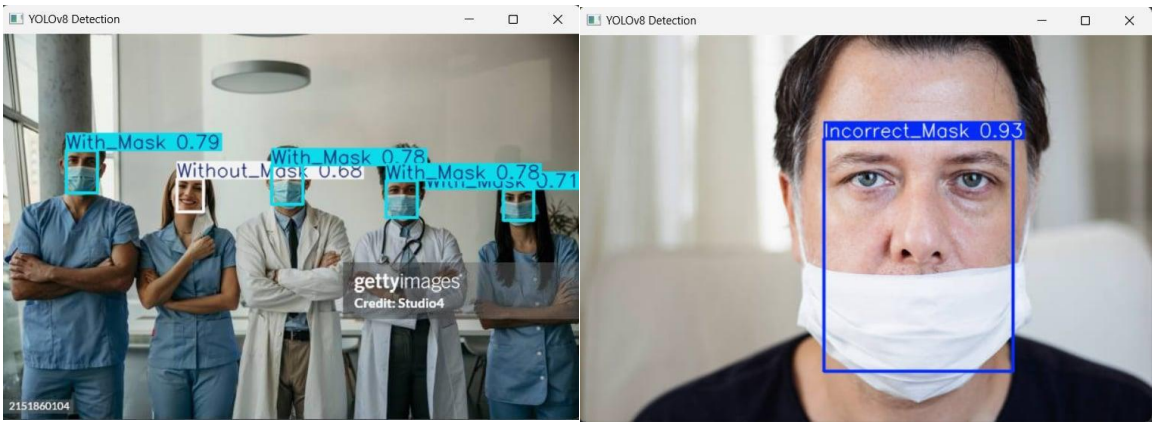
5.3. OpenCV и резултати од тестирањето на моделот

```python
def test_images():
    image_paths = [
        "test_sliki/test_image.jpg",
        "test_sliki/test_image2.jpg"
    ]

    for image_path in image_paths:
        image = cv2.imread(image_path)

        image = cv2.resize(image,  dsize: (860, 640))

        results = model.predict(image, iou=0.4)

        result_image = results[0].plot()
        cv2.imshow( winname: f"YOLOv8 Detection - {image_path}", result_image)
        cv2.waitKey(0)

    cv2.destroyAllWindows()
```

Слика 11. (Тестирање на слики со користење на OpenCV)

OpenCV (Open Source Computer Vision Library) е библиотека со отворен код која се користи за компјутерска визија и обработка на слики. Таа нуди широк спектар на алатки и функции за анализа и манипулација на слики и видео.

Предобработка на слики со OpenCV е клучен чекор во системите за детекција на објекти, вклучително и за детекција на маска. Овој чекор обезбедува правилна подготовка на сликите пред да се поднесат на YOLOv8 моделот, со што се обезбедува подобра прецизност и стабилност во детекцијата.



Слика 12. (Резултати од тестирањето на моделот)

6. Референци

[1] https://yolov8.org/yolov8-architecture/

[2] https://yolov8.org/how-does-yolov8-work/#The_Future_of_YOLOv8

[3] https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

[4] https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#what-is-precision

[5] https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall

[6] https://www.v7labs.com/blog/mean-average-precision