**Система за следене спазването на противоепидемиологични мерки в реално време**

**Проект : SaferTogether**

**Отбор SI**

**Силвия Войнова и Искра Тенева**

*ППМГ „Академик Никола Обрешков“*

GITHUB: https://github.com/SVoynova/FaceMaskDetector.git

**Резюме:**

Животът ни се промени драстично през последната година. Една от основните промени обаче е носенето на маска на обществени места. Откриването на хората, които не използват своите предпазни средства е решаващо, защото подобно поведение води до по-бързото разпространяване на COVID-19 и до по-големи човешки и финансови загуби. Целта на този проект е да подпомогне овладяването на развилата се пандемия, като следва принципа "Превенцията е по-добра и ефикасна от лечението". Тази технология може да се приложи в градски транспорт, училища, банки, магазини, ресторанти и други. Разделянето на хората в две групи, такива с маска и такива без, се случва благодарение на обработка на снимки (image processing) и дълбоко обучение (deep learning). Използвани са различни библиотеки на python като Tensorflow и PyQt, конволюционна невронна мрежа за обучаване на модела и OpenCV за следенето в реално време.

**Въведение:**

През последната година COVID- 19 завладя света и вече има повече от 125 милиона регистрирани случаи по целия свят . С бързото му разпространение носенето на медицински маска се превърна от личен избор в необходимост, спазването на да 2 метра дистанция в ежедневие, а постоянното използване да дезинфектант в навик. Но както всичко останало, и мерките за безопасност срещат недоволството на част от хората, които отказват да ги спазват и смятат, че са безполезни и създадени, за да ограничат свободата им. Това, което не осъзнават, е, че с отказа си да спазват правилата, те застрашават не само своя живот, но и живота на всички около тях. Идеята на нашето приложение е да премахне тази заплаха и да помогне за ограничаването на разпространението на непознатия вирус. Използвайки камерите, разположени на обществени места, нашият алгоритъм събира информация в реално време и чрез конволюционна невронна мрежа определя дистанцията между хората, следи за носене на маски и спазване на препоръчителните мерки срещу коронавирус. Една от функциите на този модел ще бъде да запазва информация за броя на хората, носещи маски, без това да касае идентичността им. Чрез нея да създава протокол и база данни, отнасящи се до броя на хората, спазващи мерките, и броя на тези, които не ги спазват. По този начин властите ще могат по-лесно да се ориентират в градската среда, да разберат къде не се спазват мерките и да разпределят ресурсите си по-ефективно като се съсредоточат в проблемните райони.

За гражданина ще се осигури уебстраница, в която той ще може да намери информация за степента на безопасност на мястото, към което се е запътил. Това ще бъде подкрепено с процентно представяне на спазването или съответно неспазването на противоепидемичните мерки.

**Разработка**

**Избор на вид машинно обучение**

* За посочения по-горе проблем е нужен вид машинно обучение, което да класифицира подадените му входни данни, базирано на примерните двойки вход-изход.
* Избраният обучителен метод е **supervised learning**

**Събиране на база данни и етикирането им**

* ***Изисквания към базата данни***, която да бъде използванa в обучението на модела:
  + Трябва да е достатъчно разнообразна (т.е. да включва примери от различни полове, раси, възрасти, националности и т.н.) (за да се избяга underfitting)
  + Не трябва да предизвиква overfitting на модела (прекаленото съсредоточаване в детайлите би провалило обучавания модел)
* Избраната база данни, която отговаря на нашите изисквания, именно разнообразие от снимки на хора с маски, без маски и с неправилно поставени маски, е предоставена от [Kaggle](https://www.kaggle.com/andrewmvd/face-mask-detection?select=images).

Картина, която съдържа текст, екранна снимка, различен

Описанието е генерирано автоматично

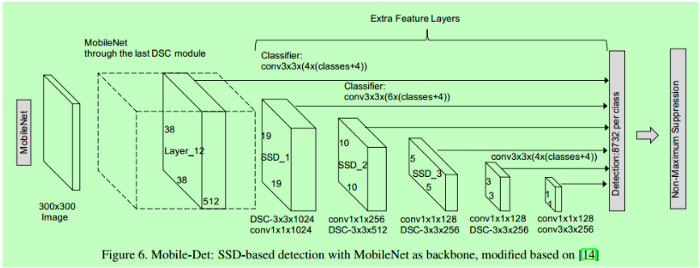
* Разделяме dataset-а на 25% за training и 75% за testing, следвайки препоръчителните съотношения за оптимално обучение на модела.

• Създаденият Label Map съдържа 3 класа:

* with\_mask
* mask\_weared\_incorrect
* without\_mask

**Избор на модел невронна мрежа**

* Най-добрият избор за поставения проблем е използването на ***конволюционна невронна мрежа (CNN)***, използваща се главно при класификация и най-вече image processing.
* Използване на ***SSD MobileNe***t /CNN архитектури, създадени да оптимизират броя операции/ - избрана поради високата й ефективност, приемайки леки спадове в точността



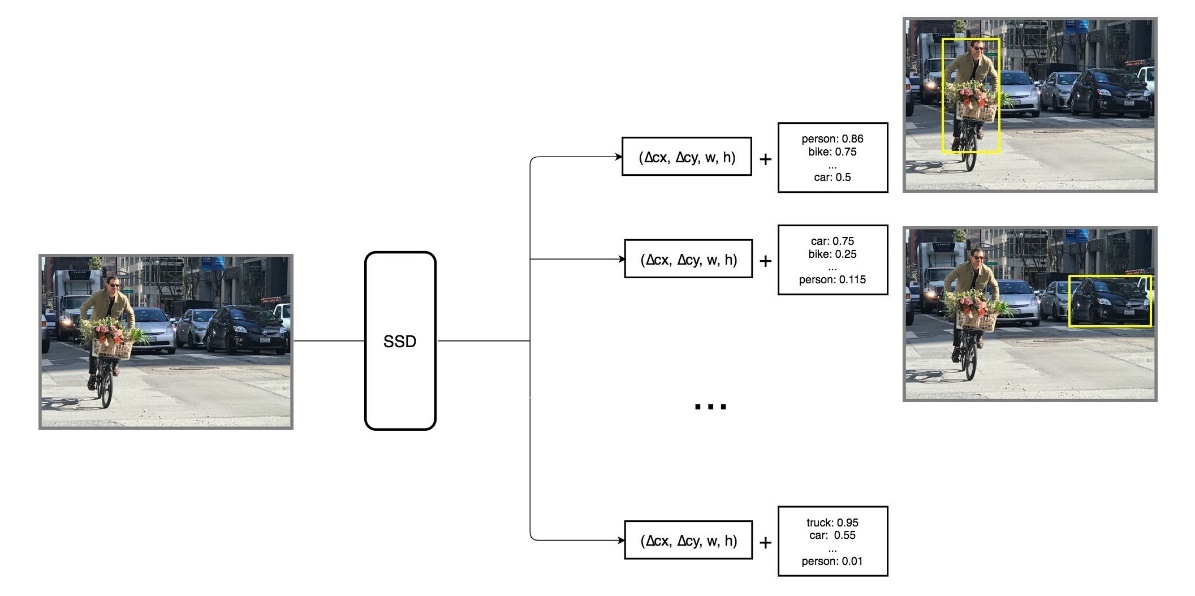
**Допълнителни използвани методи:**

* ***Transfer learning*** е изключително важна част от обученето на нашия модел в постигането на възможно най-висока ефективност за най-малко време. Използваме предварително тренираните модели, предоставени от Model Zoo, тренирани на базата данни COCO.

**Предпоставки:**

За осъществяването на нашата идея използвахме следното:

* ***TensorFlow* -** софтуерна библиотека с отворен код за машинно обучение.
* ***Object Detection API -*** framework за създаване на deep learning network, която решава проблема с object detection.
* ***Keras library -*** софтуерна библиотека с отворен код, която осигурява интерфейс на Python за изкуствени невронни мрежи.
* ***PyQt -*** обвързване на Python на кръстосания платформиран графичен интерфейс Qt, реализиран като плъгин за Python.
* ***Lxml -*** предоставя много прост и мощен API за анализиране на XML и HTML.
* ***OpenCV-*** библиотека от програмни функции, насочени главно към компютърно зрение в реално време.
* ***Protocol Buffers (protobuf) -*** механизми за сереализация на структурирана информация- XML, но по - малки, по -бързи и по - прости.
* ***Model Garden for TensorFlow -*** репоситори от различни имплементации на StateOfTheArt модели и моделирани решения.
* ***SSD-Mobilenet -*** главно създадена за object detection в реално време; учи се да предсказва локацията на ограничителната кутия, заобикаляща обекта. То класифицира обекта чрез "един поглед", за разлика от RCNN, която използва регионална мрежа, за да създаде ограничителната кутия, която по-късно се използва за класификация на обектите.
* ***IDE*** ***– Jupyter Notebook***



**Ресурси:**

* <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/_downloads/da4babe668a8afb093cc7776d7e630f3/generate_tfrecord.py>'
* <https://github.com/tensorflow/models>
* <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/_downloads/da4babe668a8afb093cc7776d7e630f3/generate_tfrecord.py>
* <https://opencv.org/>
* <https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md>
* <https://pypi.org/project/lxml/>
* <https://cocodataset.org/#home>