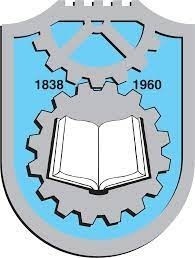
Универзитет у Крагујевцу Факултет инжењерских наука



### Тема:

**Детекција умора код возача**

Студенти: Предметни професор:

**Бојана Ивовић 637/2018 проф. др Владимир Миловановић Соња Арсић 629/2019**

**Садржај**

Задатак пројекта....................................................................................................................3 Значај и покретање апликације............................................................................................3 Делови изворног кода и опис програма и модела.............................................................6

Литература............................................................................................................................13

# Задатак пројекта

Циљ овог пројекта је направити систем који ће детектовати да ли су очи возача затворене неколико секунди. Систем ће, уколико је детектована поспаност, упозорити возача сигналом.

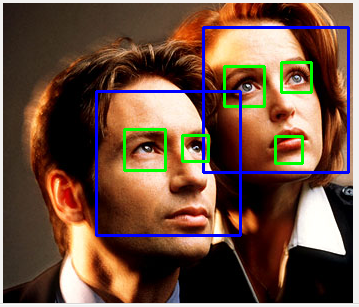
# Значај и покретање апликације

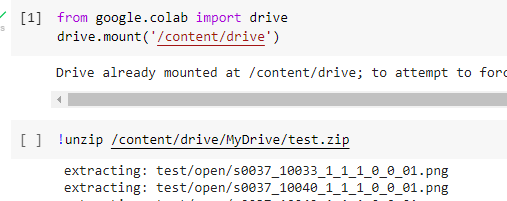
Детекција умора је технологија која може спречити настанак саобраћајних незгода проузрокованих од стране возача који заспе у току вожње.

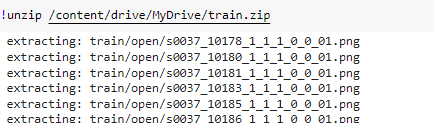
Према истраживању, више од 20% саобраћајних незгода су узроковане несавесним возачима.

За добијање података смо користиле **OpenCV** за сакупљање слика са веб камере (Он већ садржи унапред обучене класификаторе за лице, очи, осмех итд.). Детекција лица помоћу **Хаар каскада** је приступ заснован на машинском учењу где се каскадна функција обучава са скупом улазних података. Детекција лица је компјутерска технологија која одређује локације и величине људских лица на произвољним (дигиталним) сликама. Она детектује црте лица и игнорише било шта друго, као што су зграде, дрвеће и тела.

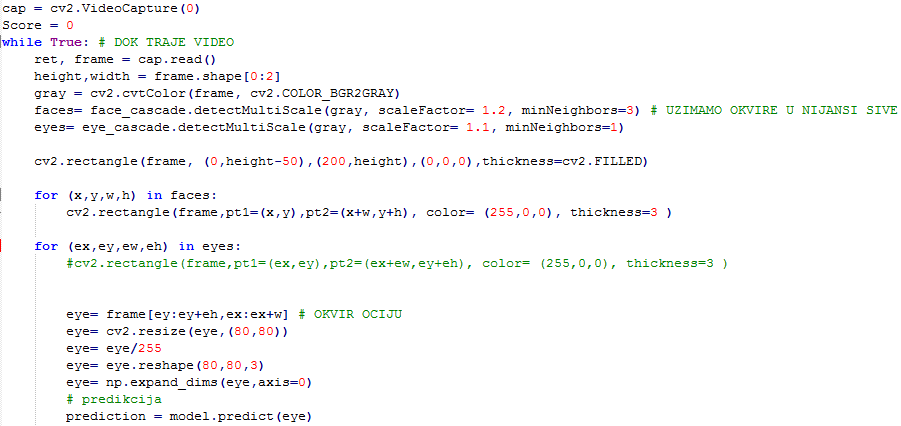








***Корак 1*** - Узимамо слике преко веб камере (хватамо сваки оквир).



***Корак 2*** - Детектујемо лице на слици. Ако се пронађу лица, враћа позиције откривених лица као **(x,y,w,h).** Користимо ове координате да нацртамо правоугаонике на нашој слици. Када добијемо ове локације, можемо да креирамо регион дефинисаности за лице и применимо детекцију очију на овај регион. Детекција ради само на сликама у нијансама сиве.

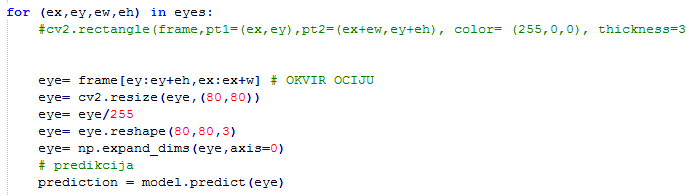
**DetectMultiScale** функција се користи за откривање лица. Потребна су 3 аргумента - **улазна слика, scaleFactor and minNeighbors.**

**ScaleFactor** - овај фактор размере се користи за креирање пирамиде размере као што је приказано на слици. Претпоставимо да је фактор скале 1,03, то значи да користимо мали корак за промену величине, односно смањимо величину за 3 %, повећавамо шансу да се пронађе одговарајућа величина са моделом за детекцију.

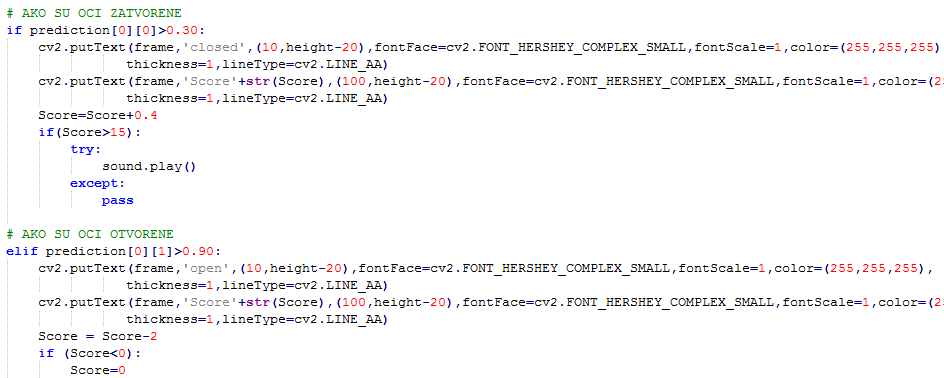


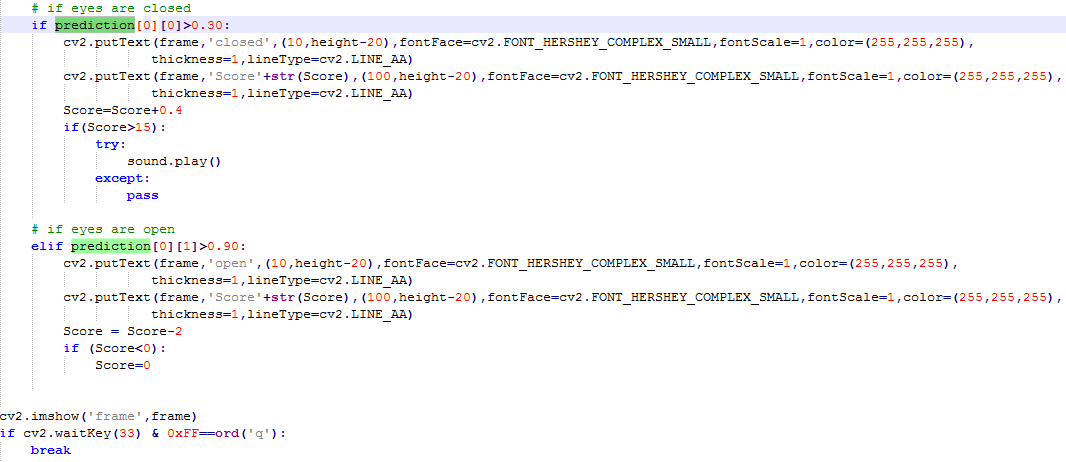
**MinNeighbors**: Параметар који специфицира колико суседа сваки правоугаоник кандидата треба да има да га задржи. Овај параметар ће утицати на квалитет откривених лица: већа вредност доводи до мање детекције, али са већим квалитетом. Користимо 5 у коду.

***Корак 3 -*** Детектујемо очи и те оквире дајемо неуронској мрежи за класификовање



***Корак 4*** - Рачунамо вредности да проверимо да ли је особа поспана.





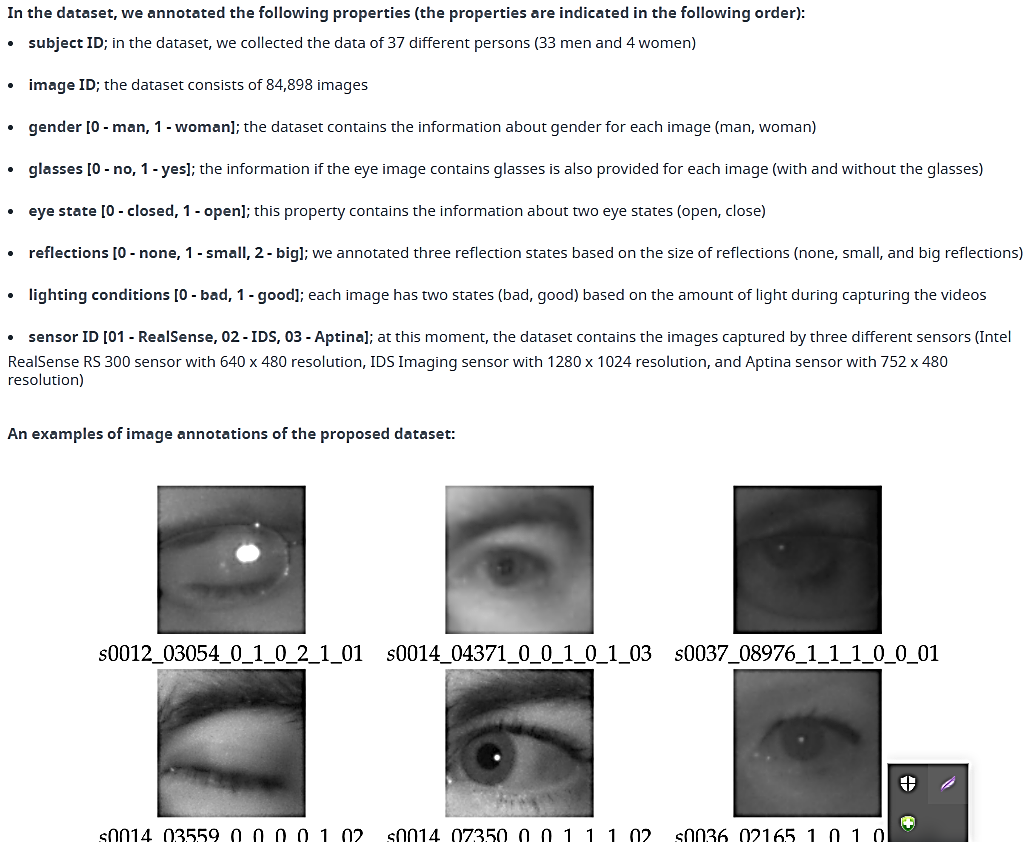
Вредност коју добијамо одређује колико дуго је особа држала очи затвореним. Уколико су очи затворене, повећаваћемо вредност, када се очи отворе, смањиваћемо је.

Дефинисаћемо праг, на пример, уколико је праг већи од 15, очи возача су затворене дужи период и огласиће се сигнал.

# Делови изворног кода и опис програма и модела

- Делимо слике на тренинг и тест скуп (**Open** и **Closed** фолдери)

- Користе се око 40 000 црно-белих слика и сваку од њих resize-ујемо на величину **80x80**.

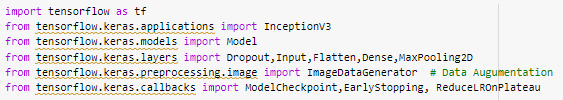


- Data augmentation се користи како бисмо повећали количину података, на пример: ротацијом, зумирањем, итд. (у овом случају смо повећале број слика на око **85 000**)

- Нормализујемо податке дељењем са 255.

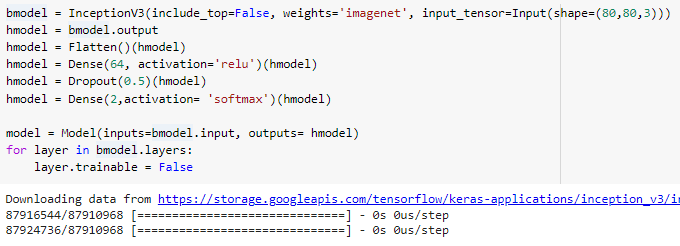
- Величина batch-а је **64** слика.

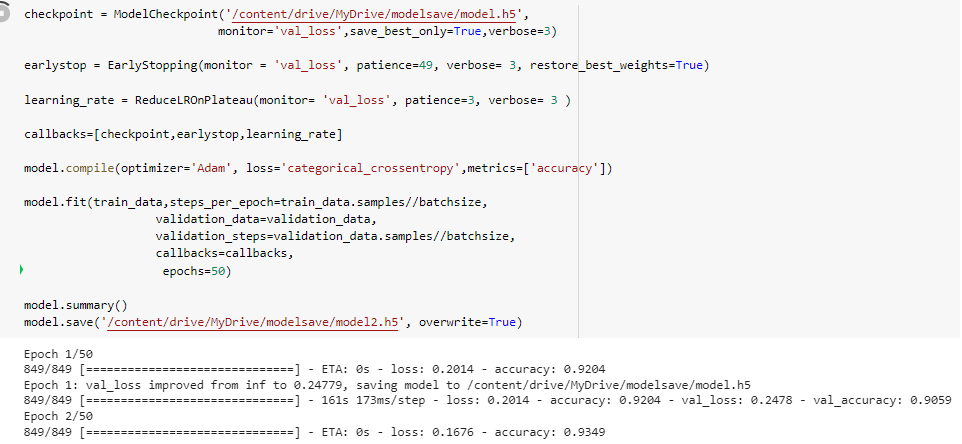
Аугментацију радимо на тренинг и валидационом скупу, док на тест скупу само нормализујемо податке.

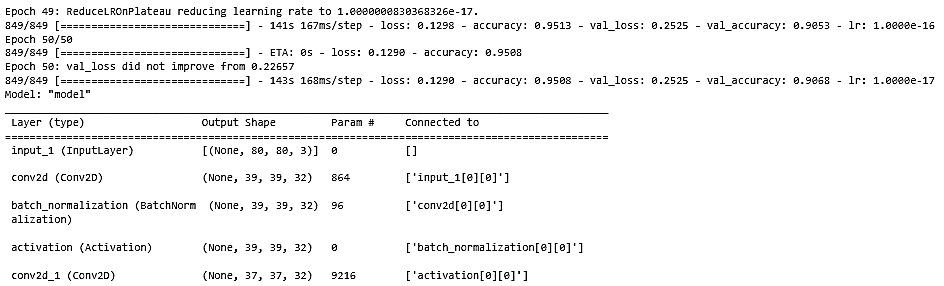




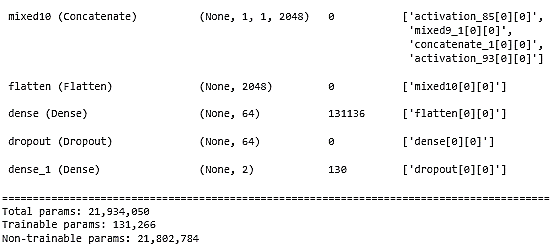
Модел чине пар слојева додатих на модел **InceptionV3**. Слојеви су замрзнути, односно, не тренирамо их. Излазни слој одбацујемо. Као излаз је стављен **софтмакс** (**Open eyes/Closed eyes**).







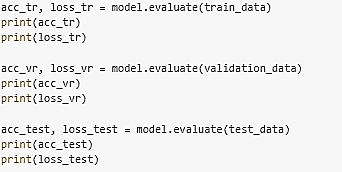
**Тачност на тренинг скупу након 50 епохa је 95%**

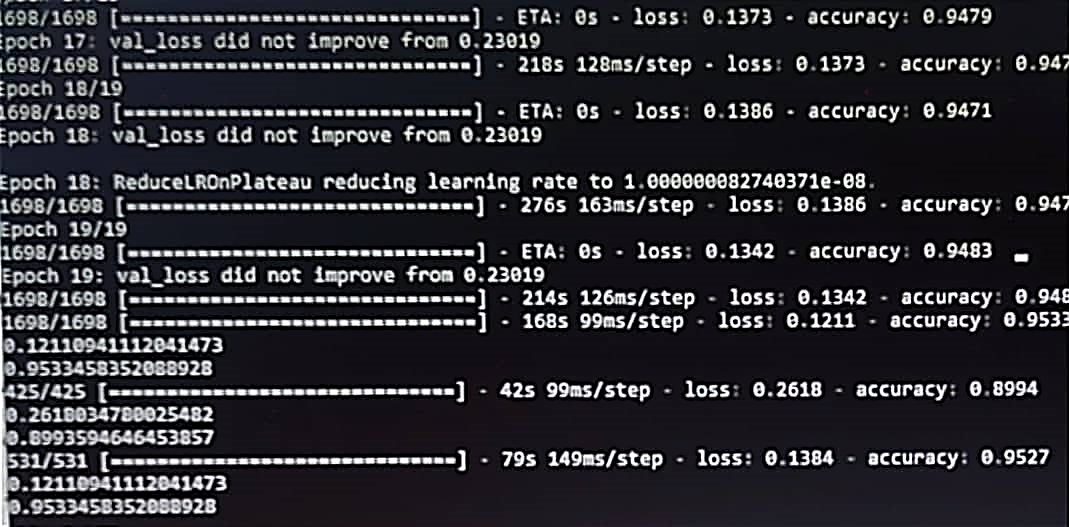


**Додати слојеви на InceptionV3,**

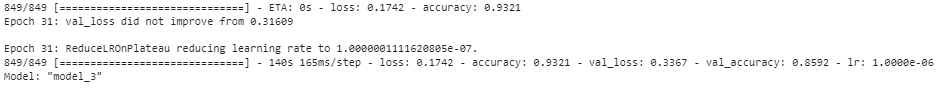
**као и укупан број, број тренираних и замрзнутих параметара.**

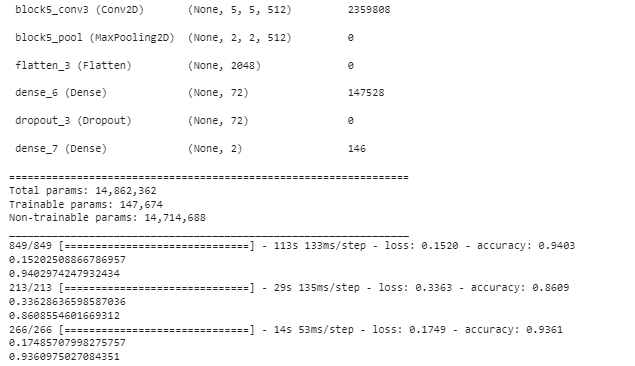
**Сада је потребно утврдити како наш програм ради и на тестном скупу.**





Са VGG-16 моделом би тачност као и фукнција трошкова горе изгледали.





**Литература**

1. Подаци <http://mrl.cs.vsb.cz/eyedataset>
2. Материјали <https://elektrotehnika.github.io/dl>
3. Курсеви на Coursera-и : <https://www.coursera.org/specializations/deep-learning>

<https://www.coursera.org/learn/neural-networks-deep-learning?specialization=deep-learning><https://www.coursera.org/learn/deep-neural-network?specialization=deep-learning><https://www.coursera.org/learn/machine-learning-projects?specialization=deep-learning>

https://www.coursera.org/learn/convolutional-neural-networks?specialization=deep-learning