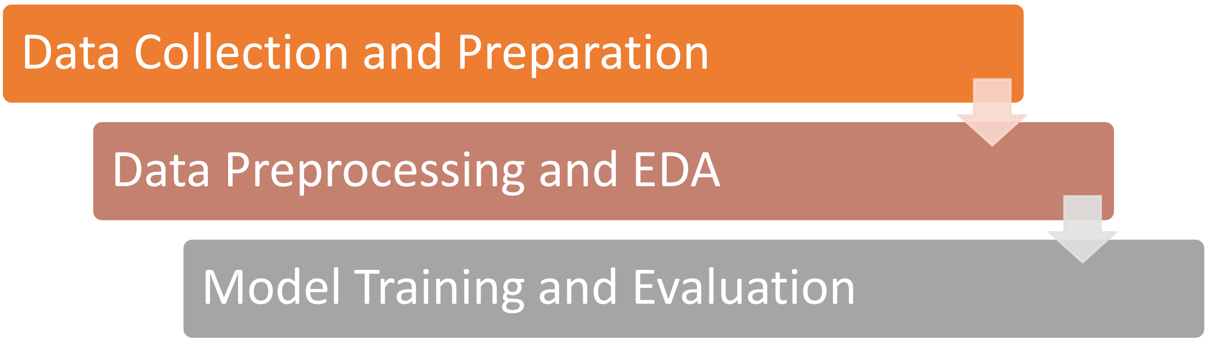
Drowsiness Detection using Transfer Learning

Process Flow:



Data Collection and Preparation

Data Preprocessing and EDA

Model Training and Evaluation

Data Collection and Preparation

Chosen Dataset: http://mrl.cs.vsb.cz/eyedataset

Other Datasets:

http://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html - Data is not labelled.

https://research.fb.com/programs/openeds-challenge - only open eyes data is available

The chosen dataset (mrl eye dataset) consists of around 15k eye images of 37 subjects. I have chosen only images of two subjects (Subject 10 and Subject 22) as the required data is only around 1k images and also transfer learning requires less data to attain the good accuracies.

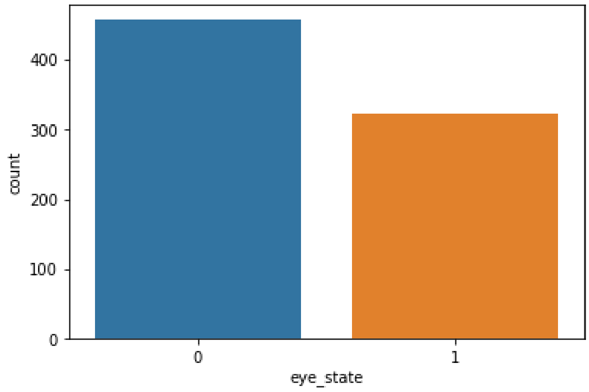
Image names are extracted using DOS commands and using python the image names are further split to extract the features of the image. Please refer to **Combined.csv** and **Images** file for the data.

eye_state, glasses, reflections, image_quality are the features extracted from the image names. Please refer to annotation.txt file for detailed explanation of features.

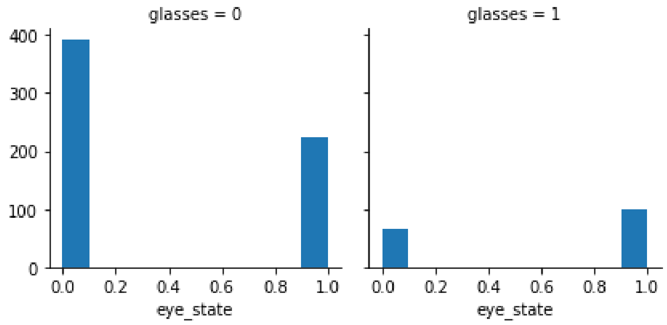
Data Pre-processing and EDA

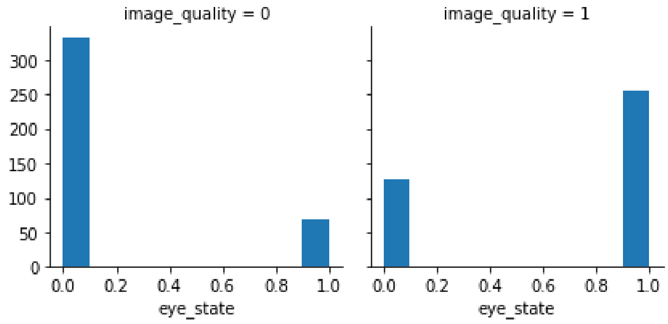
The shape of images in the transfer learning model is 24*24 and the extracted images are of different shapes. So the images are resized and converted to gray scale.

There are a total of 781 images with 324 open eye images and 457 open eye images.



There are also images with and without glasses and different image qualities in both the eye states.





For data statistics please refer to S10 and S22 in **stats_2018_01.xls**

Model Training and Evaluation

Transfer learning technique is used to re-train the model for detecting drowsiness in the images.

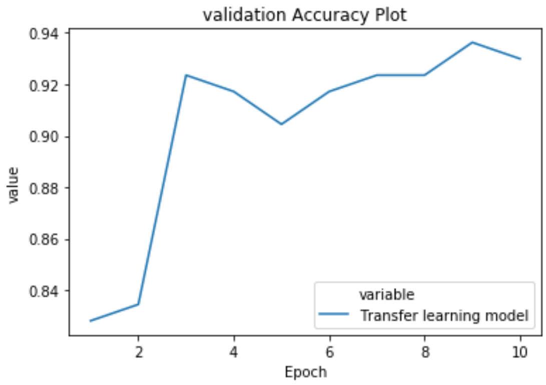
Model weights of the pretrained model are used and the model is trained on 624 samples and tested on 157 samples (80-20 split). Please refer to **cnnCAT.h5** file for model weights.

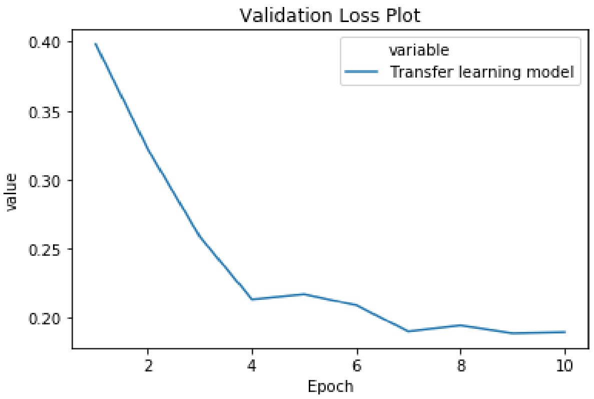
The pretrained model consists of 3 convolutional layers with one fully connected layer.

Attained ~92% accuracy with the hyperparameters - batch size = 64, epochs = 10, Relu activation function.

The re-trained weights are saved in **Retrained_sleepy_non_sleepy_64_10.h5** file.

The accuracy and loss function plots are as follows.





Real-Time Applications of Drowsiness Detection

1. Driver Alertness detection
2. Student Alertness detection

References:

https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/

http://mrl.cs.vsb.cz/eyedataset