Design and Implementation of a system for driver cellphone usage detection during driving

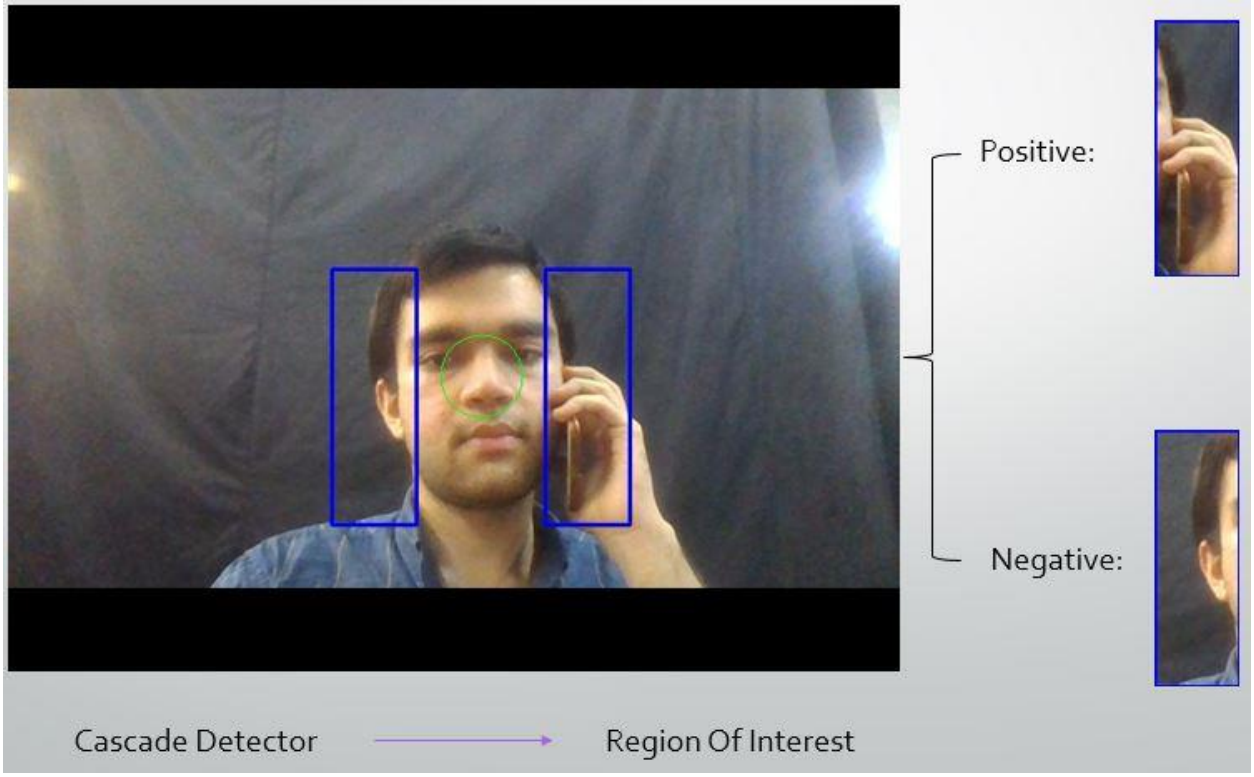
Ali Besharati Dr. Ali Nahvi

The danger of cellphone usage while driving is hidden to no body. Speaking and massaging while driving is the cause of 28% of accidents in USA. At any hour during the day, at least 5% of American drivers are using their phones which can increase the accident danger by 4 to 6 times.

The goal of this project was to design a system based on machine vision that can automatically detect and warn about mobile cellphone usage during driving.

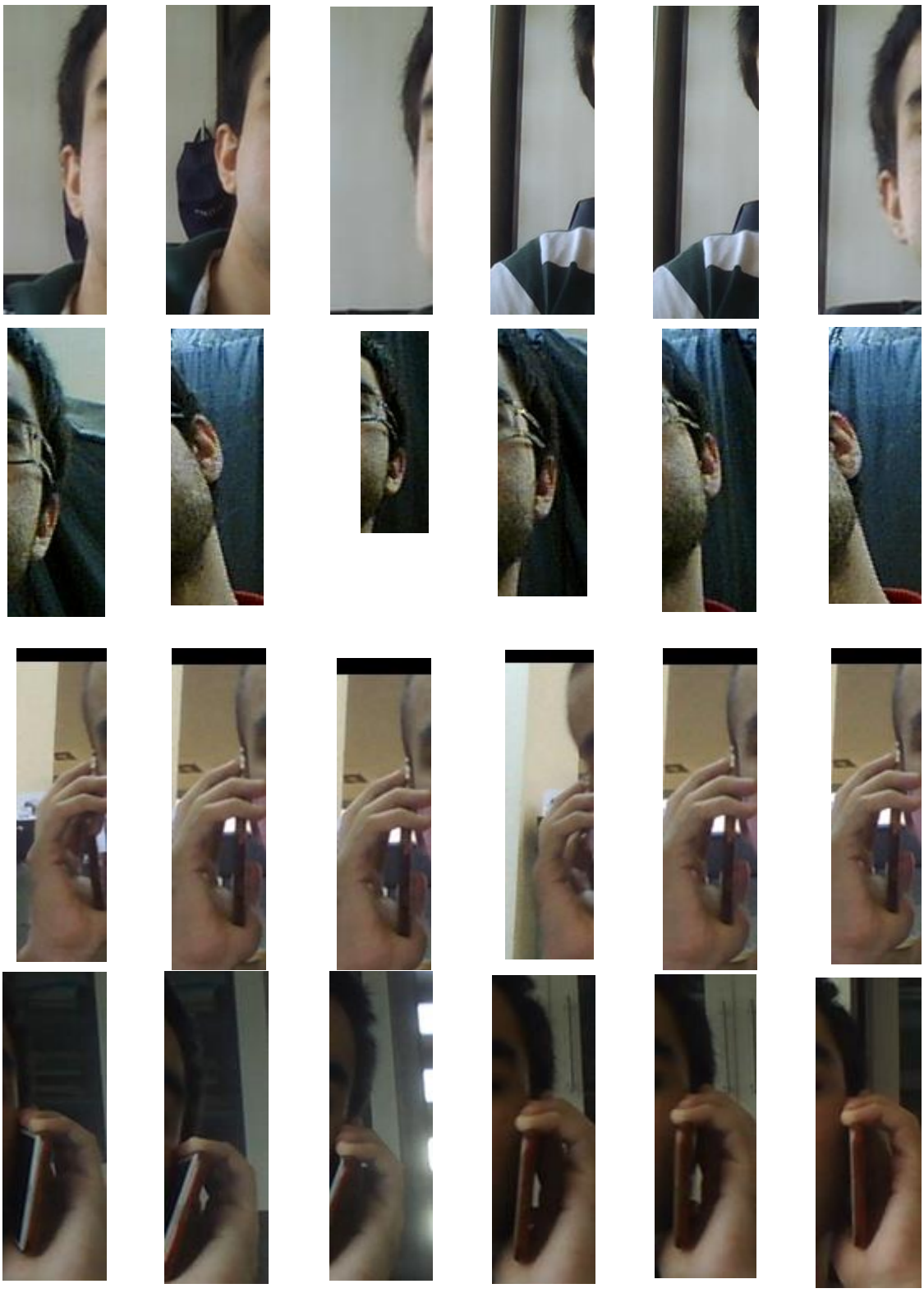
Image Acquisition:

For each instance of time, we locate the position of the driver's face using a HAAR cascade detector, then we extract and crop the positions of their ears (potential locations of the cellphone in case of usage). [code:collect_ear_dataset.py]



The samples of the dataset are provided in the following figure.

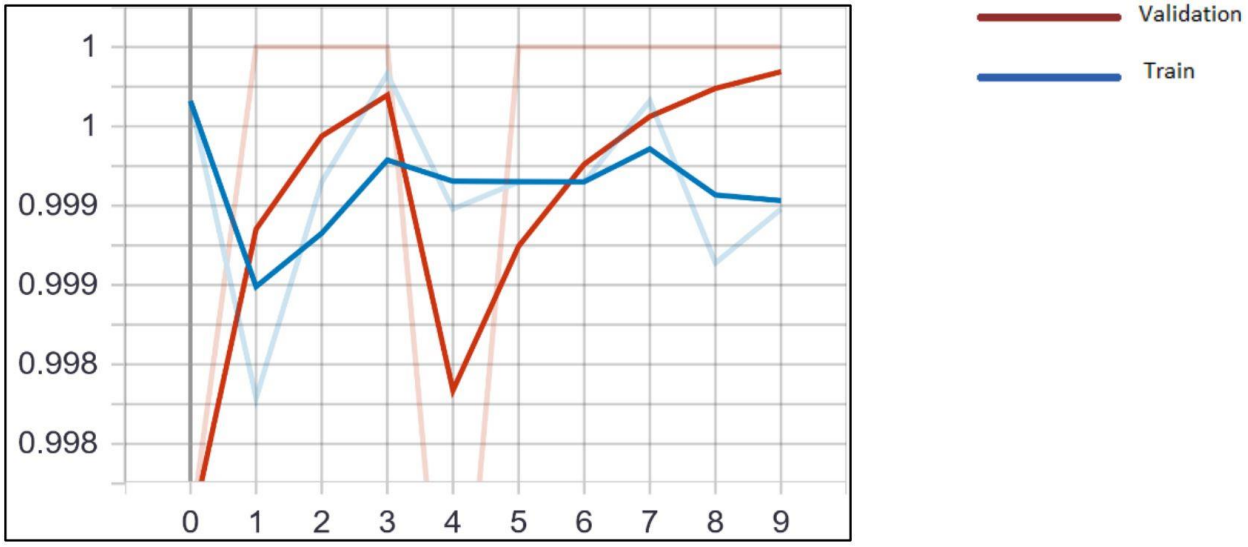
The first 2 rows depict negative images and the second 2 rows show positive images.



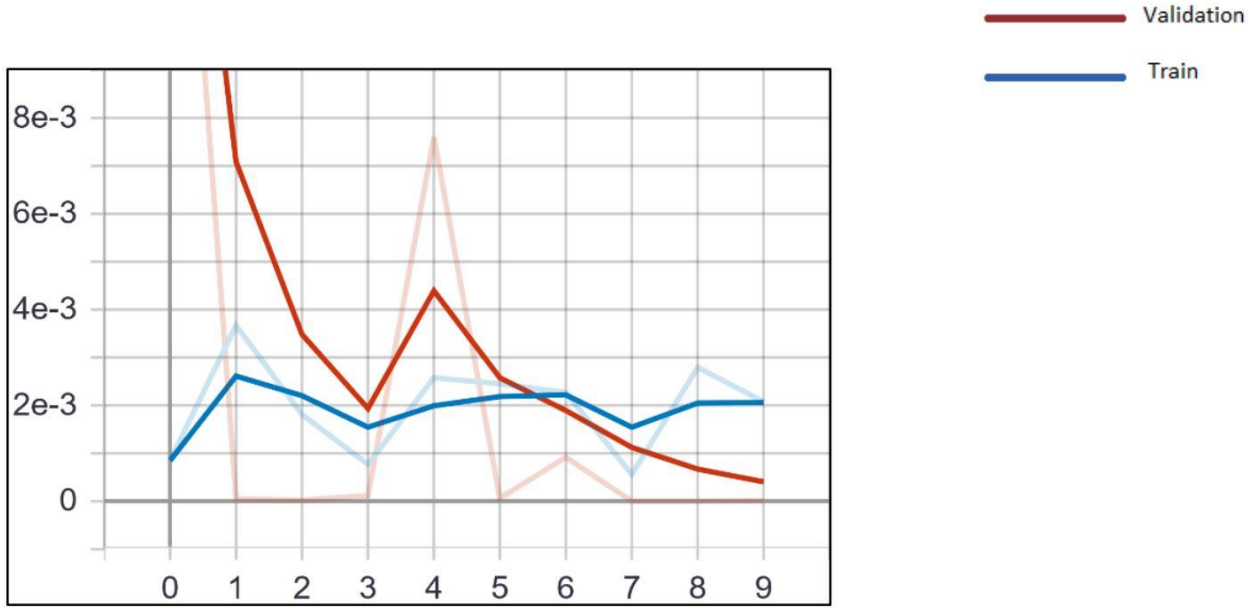
The total size of the dataset reached nearly to 8000. We made 3500 positive and 4500 negative images from various test subjects.

Classification:

For this purpose, the 3 colored images were reduced to black and white ones to lessen the unnecessary data volume. Then, several neural networks were designed. At this stage, I tried many architectures using different hyper parameters (number of layers, neurons, types of activation functions and optimizers, and learning rate) but none of them yielded the desirable results (maximum 70% acc) [CNN.ipnb]. After all, I decided to overcome my lack of experience in designing neural nets by using transfer learning [Transfer Learning.ipynb]. For that goal the EfficientNet architecture alongside the swish_act[1] activation function was used. The final result was 87% classification accuracy.



Model Accuracy



[1] It fights the vanishing gradient problem in very low and very high values

Model Loss

For the implementation of the process, we added the created classifier inside the imager acquisition loop to evaluate each image [Implementation.py]. The processing speed on an ordinary machine is less than 1 fps which is not a major problem since a driver most probably uses the phone for more than 3-5 seconds.