# Technologies for Autonomous Vehicles
# Assignment 1

Cavallotti Edoardo s308473, Maggipinto Luigi s319874

March 2024

## 1 Introduction

Mediapipe is an open source Machine Learning tool developed by Google used firstly in the field of computer vision: if it is used in combination with OpenCV, it allows to process video streams to detect landmarks on faces (by means of mesh) and subject behaviour with respect to head and eyes gaze/attitude. The aim of this project is to use the provided tools to realize a software able to supervise the driver and detect dangerous behaviors caused by distraction or drowsiness. Indeed, from statistical analysis, it's known that these are the most common causes of car accident and a system to prevent them would be of great importance.

## 2 EAR and PERCLOS

$EAR$ stands for Eye Aspect Ratio and represents a rough measure of the openness of the eye as a ratio between height and width. $PERCLOS$ stands for Percentage of Closure and can be seen as a measure of openness of eyes over time. We compute the $EAR$ for each eye by means of the following formula

$$\frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2\|P_1 - P_4\|}$$

then we get the mean value between right and left eye as $EAR_m$. This value is necessary for the following computations.
$OBS$: the $EAR$ values, provided by the algorithm, are no longer valid if the head is much rotated (since several landmarks become hidden by face geometry and some measurement can't be performed): indeed, these values can corrupt the maximum and minimum we are considering for the percentages: to cope with this problem we simply neglect the new acquired values and keep the maximum and minimum previously taken when head angles overcome certain thresholds.

For the computation of the $PERCLOS$ we implemented a state machine made of four states and for each of them we measured the elapsed time: eyes open

more than the 80%, eyes open between 80% and 20%, eyes open under 20% (passing through 0%), eyes open between 20% and 80% (passing through 0%): in order to distinguish the second and the fourth cases we need a state machine to take into account the previous state we arrive from. Once the entire cycle is completed, i.e. transition from $state_4$ to $state_1$ is made, timers are reset and the counting for $t_1$ is started again.
A side note is now necessary: we implemented a robust measuring of $PERCLOS$ in order to avoid misinterpretation of eye blinking as a complete cycle of $PERCLOS$. Indeed, when one blinks, eyes go through the four steps that we had considered in the computation of $PERCLOS$, but within very short periods of time for $t_2$, $t_3$ and $t_4$. So we set a minimum threshold to check whether these steps take a reasonable time or not: by means of this validation we can safely measure $PERCLOS$ and detect effectively dangerous situations, distinguishing from common eye blinks (in which the time of the states could be not really accurate). The openness of the eyes is, of course, computed using the $EAR_m$. The formula used to the $PERCLOS$ computation which takes into account the four measured times is

$$\frac{t_3 - t_2}{t_4 - t_1}$$

The normal behaviour is that $t_1$ is much longer than the other times since eyes are kept open.

SATURATION COUNTER: If the PERCLOS, in percentage, exceeds the 100%, we have saturated the value to 100; consequentially, if the value goes below the threshold of 0%, we have saturated the value to 0. In addition, we have displayed a counter showing how many times, in a row, the Perclos has been saturated. Graphically, this is useful to see if the Perclos is refreshed or not.

TIMER: when the $EAR$ goes below the 20% a timer is activated, and until it stays below that threshold, the timer is checked to be under 10 seconds: in case it overcomes this limit, a warning message is displayed signaling that driver is drowsy. Otherwise, if eyes are reopened before the timer expires, the timer is reset.

# 3 Head and Eye Gaze

Once we had compute the vector representing the direction of eyes and nose, we have the *Roll*, *Pitch* and $Yaw$ angles for both eyes and for the face: actually we use only the angles for the head since the eyes' ones are very noisy and unreliable. Furthermore, we use different thresholds to check whether the angle permits to have a good looking on the street.

Here we present the list of threshold values used for the head Euler angles:

- MAX ROLL (rightwards) → 50°
- MIN ROLL (leftwards) → -45°

- MAX YAW (leftwards) $\rightarrow$ 75°
- MIN YAW (rightwards) $\rightarrow$ -47°
- MAX PITCH (upwards) $\rightarrow$ 60°
- MIN PITCH (downwards) $\rightarrow$ -20°

If any of these thresholds is overcome, a message of warning is displayed signaling that driver is distracted.

Another common case of our interest, that we can't detect by using the head's angles, is represented by the situation in which the driver is looking downward with his eyes (i.e. when he is staring at the phone) for a reasonable amount of time. This entails a reduction of $EAR$ and, as soon as this happens, a timer is started: if this $EAR$ reduction lasts for more than two seconds, a message of warning is displayed signaling driver's distraction, but if timer keeps going beyond ten seconds, then a message of drowsiness alert is displayed, as we discussed before.

### 3.1 Eye Gaze alternative method

Instead of using eyes angles, we can exploit the relative position of the center of the eye with respect to the position of the pupil: if the distance exceeds a certain threshold, it means the eyes aren't looking forward, so the driver might be distracted. We take into account also the case in which the driver rotates (reasonably) the head but keeps the sight on the road by looking forward: in this case center of eyes and pupils coincide and this permits to avoid useless and redundant warnings.

In case of low resolution, i.e. when eyes height and width are below 65 pixels, we compute eyes center and pupil positions in 2D: otherwise, when the resolution is higher, i.e. eyes are closer to camera, the computation is made from 3D values.

## 4 Conclusion

The code has been developed and tested on a MacBook Pro 2019 with a Webcam: 720p@30fps. During the testing, we observe the influence of noise on measurement and distance computations; a difficulty we encountered was the setting of good thresholds based on the sensibility of the specific case: indeed, accordingly to the distance of the face from the camera thresholds must be changed to cope with different number of pixels.

NOTE: In order to achieve a good performance of the algorithm, some choices about thresholds are different from the ones provided in the assignment, i.e. the request on the message printing if $EAR$ gets below 80% has been changed into 60% closed (40% open). Also, for what regards the head and eyes gaze with respect to rest position, threshold angles have been chosen differently after many attempts [3].