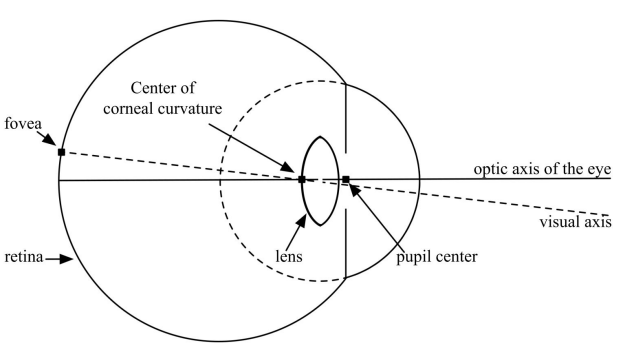
Development of a Computer Vision **eye gaze tracking** algorithm for **Virtual Reality** applications.

Presentation

CentraleSupélec

Leopold & Adrien

Eye complex geometry



Virtual Reality Applications

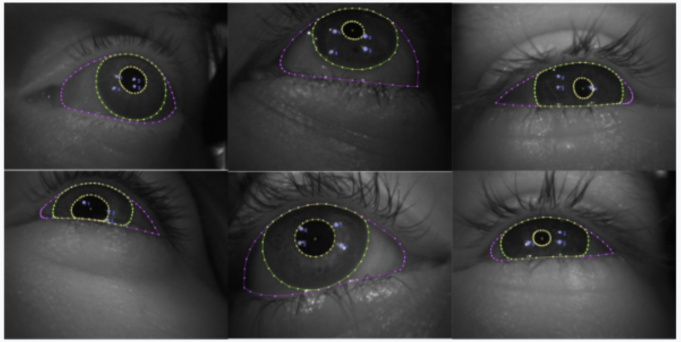
How could we use computer vision for gaze estimation in a VR context ?

Table of contents

01

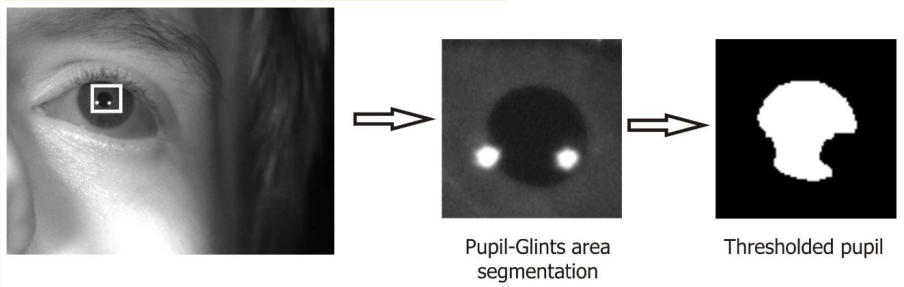
Related work

Eyes and pupils detection



- Shape-based : thresholding, Hough Transform
- Feature Based: distinctive features around the eye
- Appearance Based

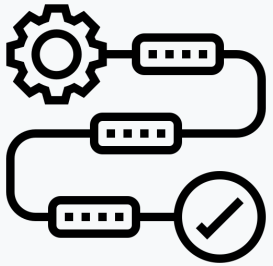
Gaze estimation



- Point based
- Shape based
- Hybrid

02

Methodology



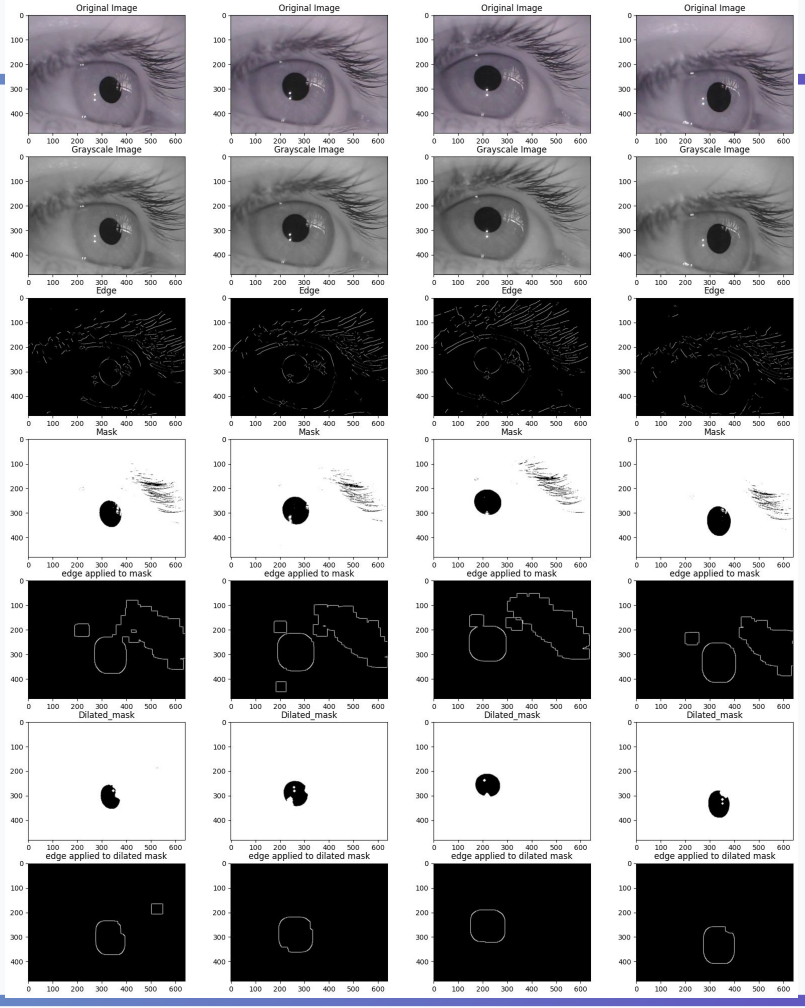
- Eye segmentation
  - Image transformation
  - Circle detection
  - Blob detection
- Gaze tracking using ML
- Gaze tracking using geometrical approach

Eye segmentation

- Why do we need to segment the eye
- Explanation of the eye segmentation technique
  - Image transformation
  - Circle detection
  - Blob detection

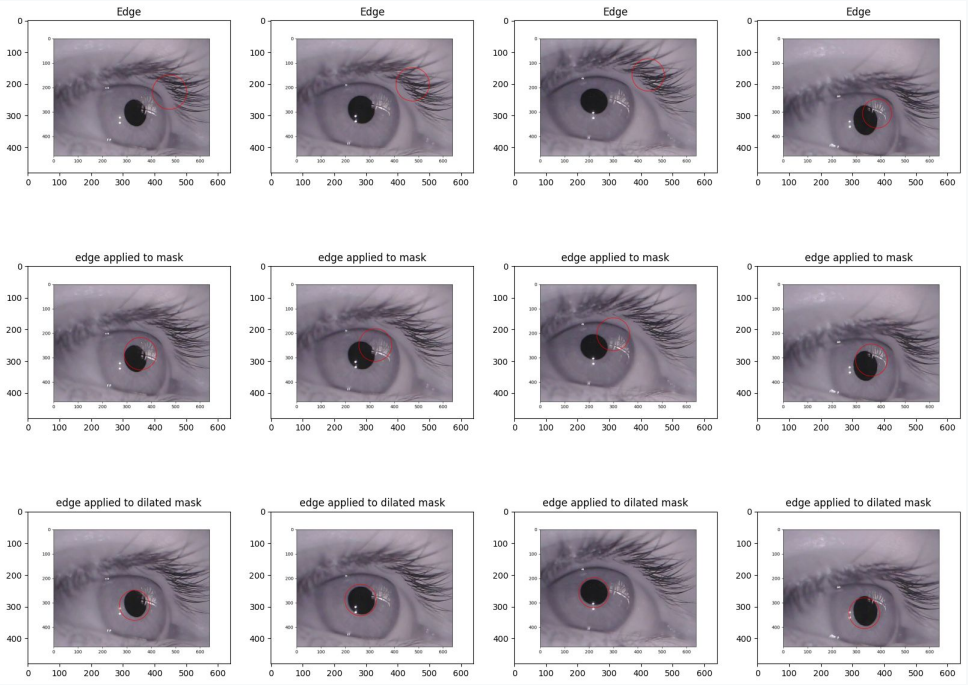
Image transformation

Example of an image where the eyelashes are more intense due to make-up (mascara)



Different transformations applied to 4 randomly selected images of the dataset.

Circle detection

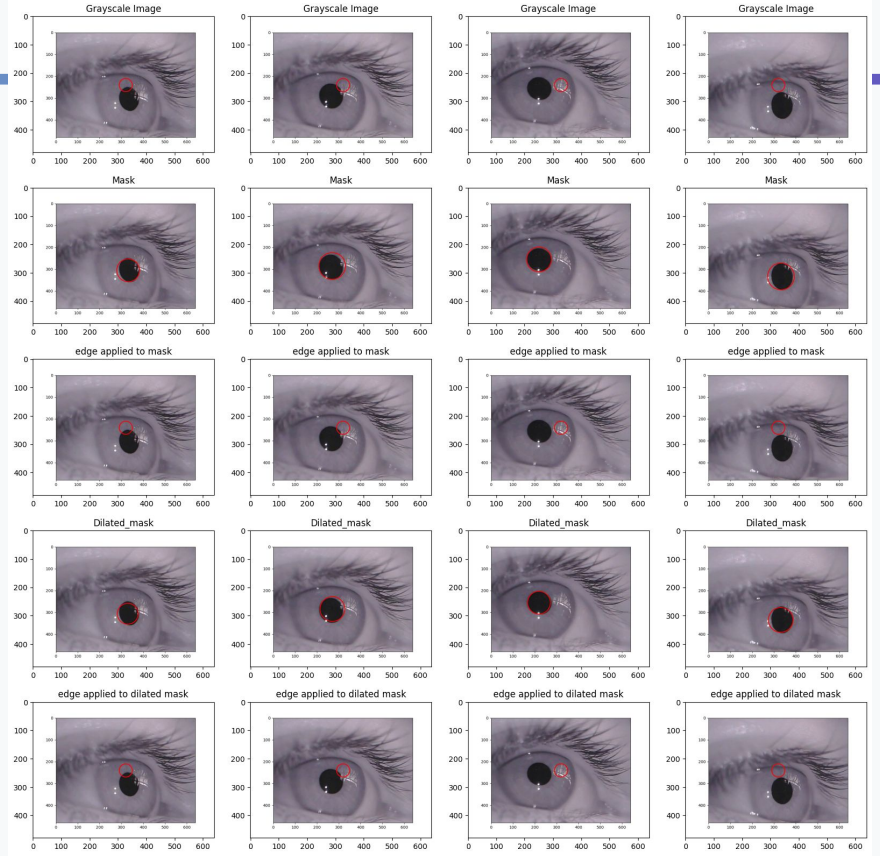


Circles detection algorithm tested on multiples transformed images.

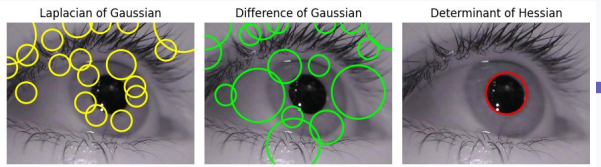
- Convert to grayscale
- Smooth the image with a gaussian filter
- Threshold to create a binary images
- Dilate the image
- Apply Edge Canny detector
- Iterate over different radius values:
  - Create a binary mask of a circle with the current radius value.
  - Use the mask to extract the circular region from the edged image.
  - Count the number of edge pixels in the circular region.
  - If the number of edge pixels > threshold, record the coordinates of the center of the circle and its radius.
- Return the list of circles with their center and radius
- Draw the circles on the original image

Blob detection

- Convert to grayscale.
- Threshold the image.
- Apply dilation.
- Compute the Hessian matrix to detect blobs in the image.
- Compute the eigenvalues and eigenvectors of the Hessian matrix.
- Set a threshold on the eigenvalues to remove small or insignificant blobs.
- Identify blobs by extracting connected components in the image that have eigenvalues above the threshold.
- Return the center of mass and the radius of each blob
- Draw the circles on the original image



Blob detection algorithm tested on multiples transformed images.



Gaze tracking using ML

- Train a model to relate the center coordinates of the eye to the gaze direction.
- multi-output regression from scikit-learn library:
  - RandomForestRegressor
  - LinearRegression
  - Ridge
  - Lasso
  - DecisionTreeRegressor
  - Support Vector Regression (SVR)

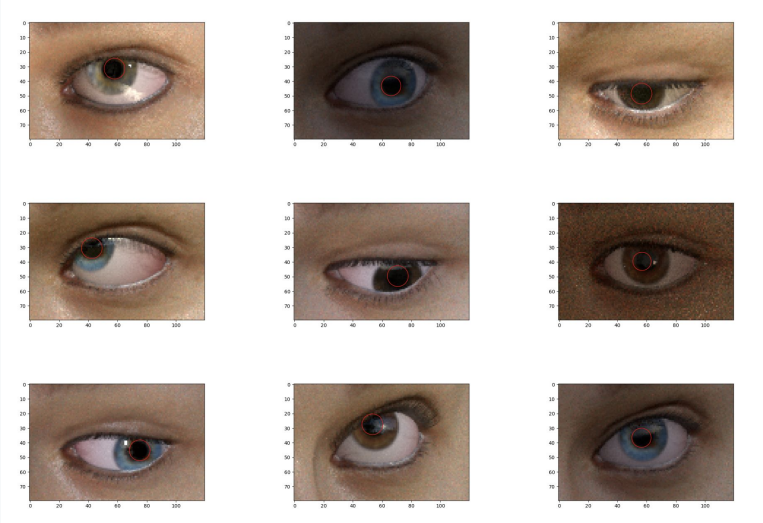
03

Evaluation

- Dataset tested
  - LPW
  - UnityEyes
- Results
  - Execution times
  - Accuracy of the segmentation and pupils coordinates prediction
  - Accuracy of the gaze estimation using ML

Dataset tested

- Labeled Pupils in the wild (LPW)
  - Videos of individuals filmed inside a VR headset.
  - Participants of different ethnicity, both outdoor and indoor.
  - Variants such as subjects wearing makeup or contact lenses.
  - Annotations of the center of the pupils tracked along the video
  - In total 12,000 images.



- UnityEyes
  - Collection of synthesized eye region images created using a novel method
  - Based on high-resolution 3D face scans and use real-time approximations
  - Full control over environments, cameras, and precise calculation of gaze
  - Ideal solution, but not able to generate images on compatible computers

Results

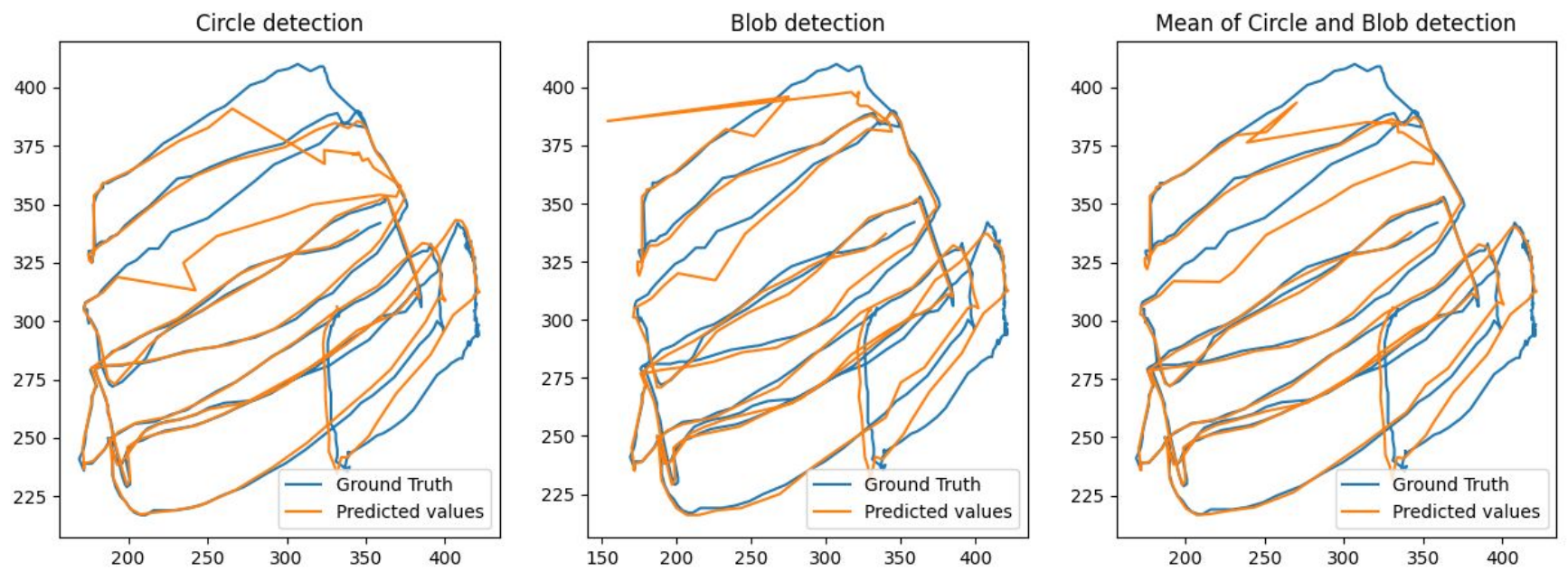
- Execution time
- Accuracy of the segmentations and center of pupils coordinates predictions
- Accuracy of the gaze estimation using ML

Execution times

	Implemented from scratch	From SkImage libraries
Circle detection	~ 30 min	~ 2 min 05
Blob detection	~ 25 min	~ 2 min 50

Execution times of different algorithms on 200 frames to segment and estimate the coordinates of the pupils

Accuracy of the center of pupils coordinates predictions



Comparison of estimated pupil coordinates using Circle Detection, Blob Detection, and Mean of Circle and Blob detection algorithm against the ground truth for a video of an eye looking in different directions.

Accuracy of the center of pupils coordinates predictions

Metric	Circle detection	Blob detection	Mean of both methods
MAE	5.77	5.00	5.11
MSE	92.2	93.1	64.6
RMSE	9.62	9.65	8.03
R2	0.975	0.980	0.982

Metrics applied to the same data than the previous slide for Circle, Blob and mean of both methods pupils estimation.

Accuracy of the gaze estimation using ML

	MAE	MSE	RMSE	R2
Random forest regression	0.093	0.024	0.155	0.69
Linear regression	0.155	0.035	0.188	0.410
Ridge	0.155	0.035	0.188	0.410
Lasso	0.17	0.045	0.211	0.324
Decision tree regression	0.109	0.04	0.201	0.484
SVR	0.153	0.035	0.186	0.473

Metrics for different regressor used in the multi-output regression algorithm from the train-test split evaluation of true data

Conclusion

Take home message