Project 2 - Team 3

Ellipse matching and performance assessment

1. Performance assessment of line segment detection

2. Ellipse matching

   a. Image preprocessing + Data preparation
   b. Ellipse classification
      - RandomForest
      - Boosting methods
      - Resnet
      - YOLO classification
   c. Regression on bounding boxes
   d. Regression on ellipses parameters

3. Performance assessment of the ellipse matching module

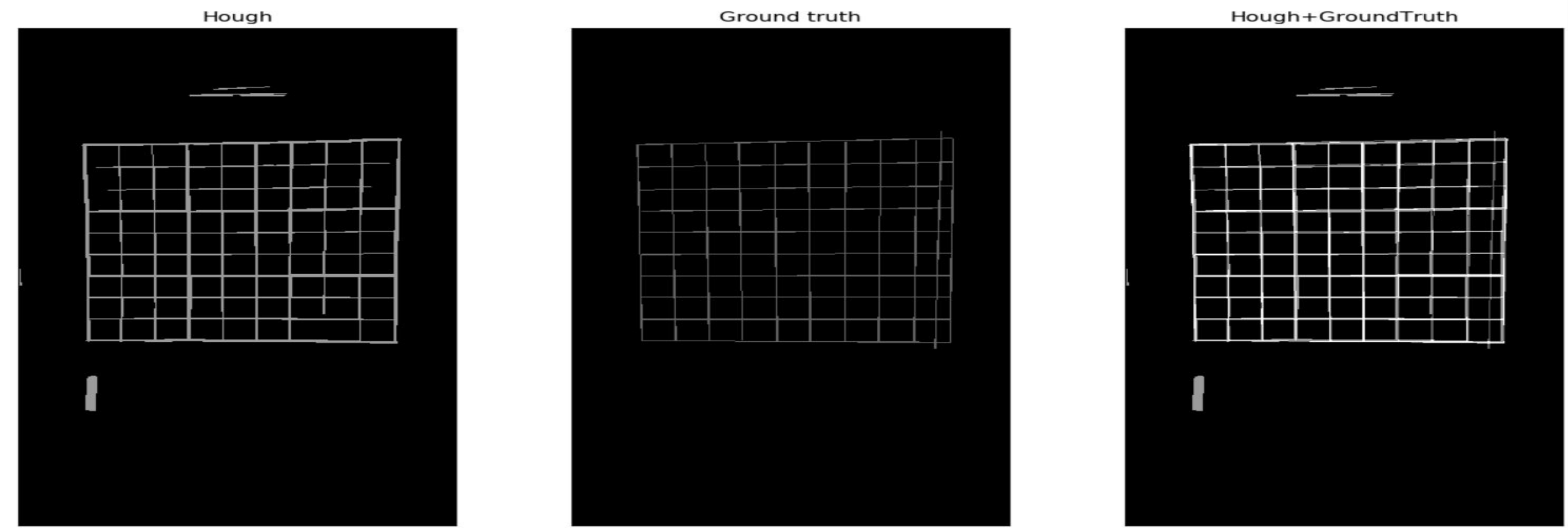
   a. Ellipse classification
   b. Bounding boxes regression
   a. Ellipse parameters regression

4. Demo

1. Performance assessment of line segment detection

1. Performance assessment of line segment detection

1) Compute hough on empty image. Use value 155
2) Compute ground truth on empty image. Use value 100
3) Add both images together
4) Check each pixel for the stored value



1. Performance assessment of line segment detection

- Using the results, created a confusion matrix.
- In our case, the thicker the lines drawn, the worst our overall ‘Good hits’ ratio was.
- The computation time was not impacted significantly by the line thickness

		Soccer				
1		0,09	0,29			87,79
		11,8	87,7			12,09
3		0,9	0,69			76,30
		22,9	75,4			23,59
7		2,2	1,04			71,90
		27,04	69,7			28,08

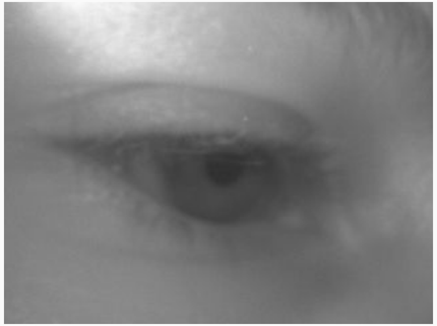
		Key			
	True positive(%)	False negative(%)			Good hits (%)
	False positive(%)	True negative(%)			Bad hits(%)

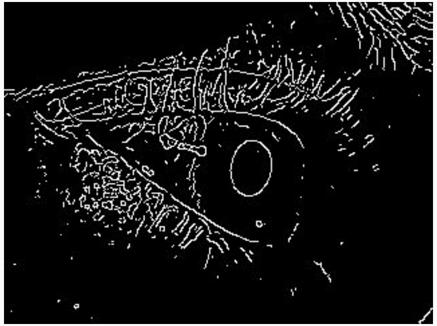
2. Ellipse matching

2.a Image preprocessing + Dataset preparation

<u>Importance of preprocessing</u>

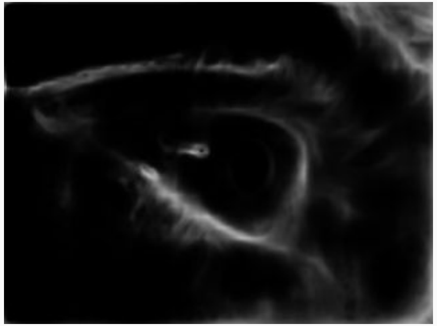
- SL algorithms: Learn the input-output relation
- They must be not erroneous and not noisy
- Otherwise, training phase is more difficult
- Here => Removal of the images without labels





<u>Image processing tried</u>

- Grayscale images
- Canny edge detection
- HED edge detection
- Binarization



2.a Image preprocessing + Dataset preparation

Classification

	Scikit-Learn methods	**ResNet**
Input	Resized/Reshaped grayscale images	Resized color images
Output	Number of ellipses	Number of ellipses

Regression

	Parameters	**Bounding boxes**
Input	Reshaped grayscale images	Color images
Output	xc,yc,MA,ma,θ	xmin,ymin,xmax,ymax

2.b Ellipse classification - RandomForest

Decision tree

- Simple but powerful estimator
- Easily interpretable

but…

- Overfit very quickly on training set

Ensemble methods

RandomForest

- Combine the predictions of several trees
- Add randomness to avoid overfitting

<u>Advantages:</u>

- Easy to build
- Default parameters sufficient to have great performances

<u>Drawbacks:</u>

- Lots of hyperparameters to tune when improving the results (nb_estimators, max_depth, warm_start,...)
- Difficulties to classify images with 2 or 3 ellipses

2.b Ellipse classification - Boosting methods

Motivations :

- Focus on the hard-to-predict elements to increase the accuracy
- Put some emphasis on the few elements of class 2 and 3

Algorithms considered :

- Adaboost
- Gradient Descent Boosting
- XGBoost

2.b Ellipse classification - Boosting methods (Adaboost)

Adaboost

- Takes a lot of weak learners, barely better than random guessers
- Creates the next learner by taking the error the previous one made into account
  - ➢ Focus on the hard cases
- Final vote amongst classifiers on the correct class

Results

- Results reaching 80 percent of accuracy for low amount of classifiers
- Higher amounts never finished
- Good results for common classes, misqualifies rare ones

2.b Ellipse classification - Boosting methods (Adaboost)

Advantages:

- Easy to build
- Default parameters sufficient to have great performances
- Less subject to overfitting

Drawbacks:

- Can be sensitive to noisy data and outliers.

2.b Ellipse classification - Boosting methods (Gradient Boosting)

Gradient Boosting

- Every new model is built in a way to reduce the global error
- Following a metric (loss)
- Greedy algorithm
- Tend to overfit if too much estimators

Results

- Results reaching 94% percent of accuracy for 600 classifiers, LR of 0.9
- Less populated classes fairly well evaluated
- 2nd best algorithm in this task

2.b Ellipse classification - Boosting methods (Gradient Boosting)

Advantages:

- Easy to build
- Default parameters sufficient to have great performances
- Should be on paper better than RF

Drawbacks:

- Training generally takes longer
- Prone to overfitting

2.b Ellipse classification - Boosting methods (XGBoost)

XGBoost

- Variant of Gradient boosting
- Great success in Kaggle competitions
- Especially suited for tabular or structured data

Results

- Results failed to breach through the 90% accuracy threshold
- LGBM

2.b Ellipse classification - Boosting methods (XGBoost)

Advantages:

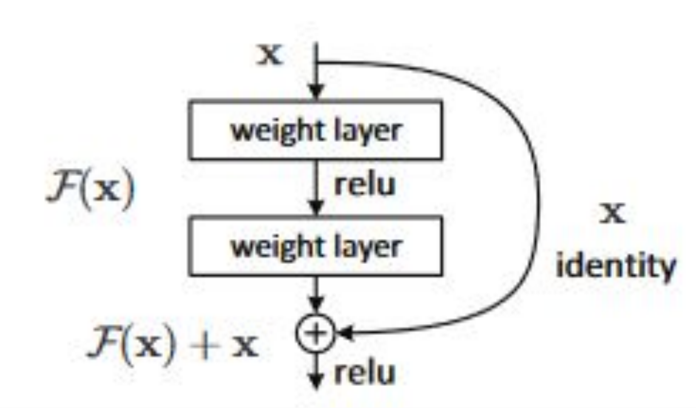
- Parallel processing
- Run a cross-validation at each iteration

Drawbacks:

- Same as gradient descent

2.b Ellipse classification - ResNet

- Winner of the ImageNet challenge in 2015
- Solves the problem of the vanishing gradient in deep neural networks
- Use shortcuts between inputs and outputs of layers
- Allows huge networks (up to more than 100 layers)
- Especially suited for computer vision tasks



Advantages:

- Able to fit on very complex datasets
- Obtain the best performances from all the methods tested
- Easily built using keras' pre-built models

Drawbacks:

- Very high variance
- Difficulties to classify images with 2 or 3 ellipses
- May easily overfit

2.b Ellipse classification - YOLO

Using the YOLO algorithm (details on YOLO in the bounding box regression task), possibility to detect the number of ellipses on soccer images:

1. Perform bounding box regression on soccer images
2. Knowing the number of bounding box, immediately infer the number of ellipses

Advantages:

- Solves the problem of the multiple ellipses images (same accuracy on them than on other images)
- No additional effort required following regression

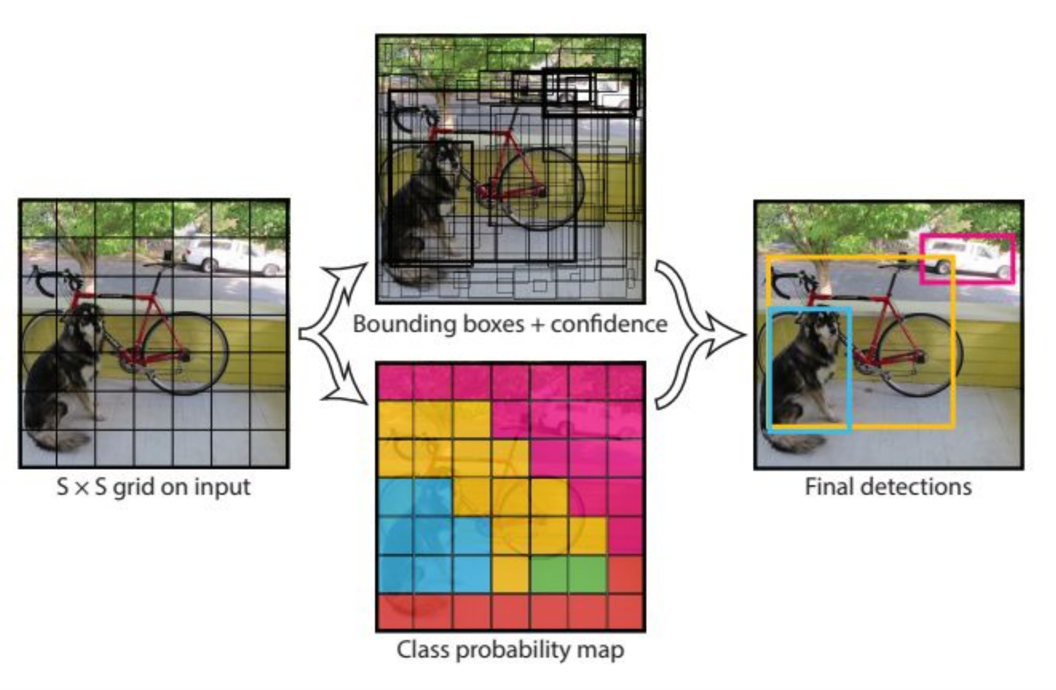
Drawbacks:

- Global accuracy smaller than other algorithms (~ 80%)
- Cannot be used for bounding box regression tasks

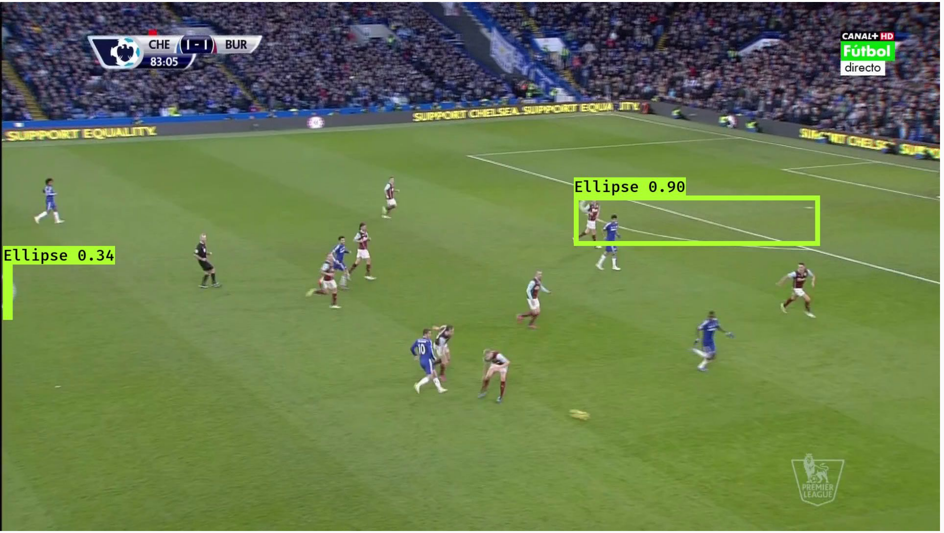
2.c Regression on bounding boxes

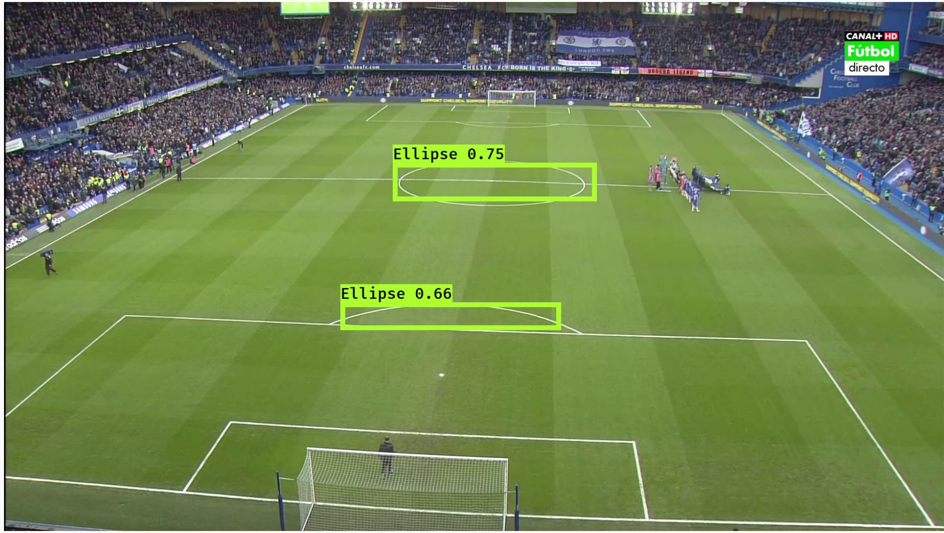
- No results with Tree based methods

- Good results with YOLO
  - Generate boxes for each tile
  - Compute class probability map
  - Combine information

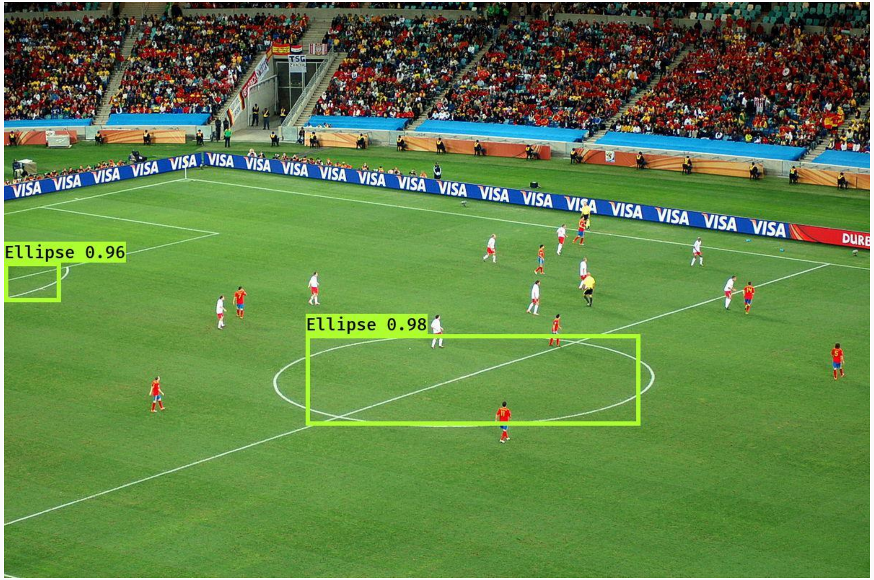


2.c Regression on bounding boxes - Results



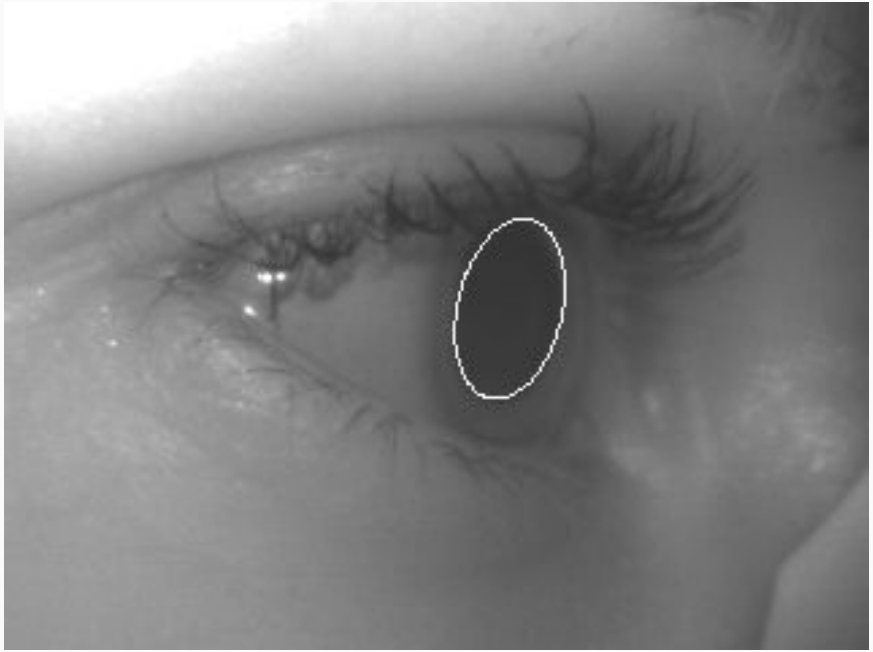


2.c Regression on bounding boxes - Results

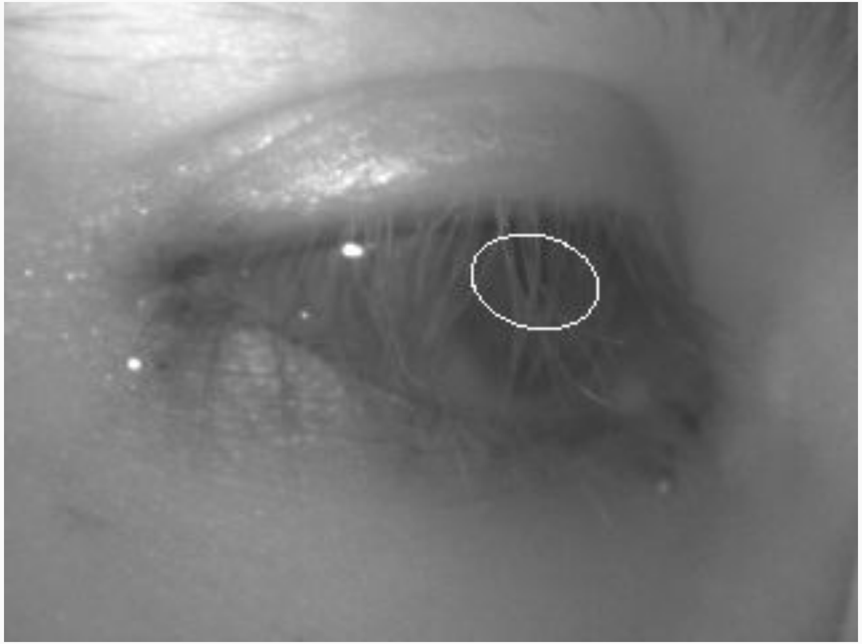


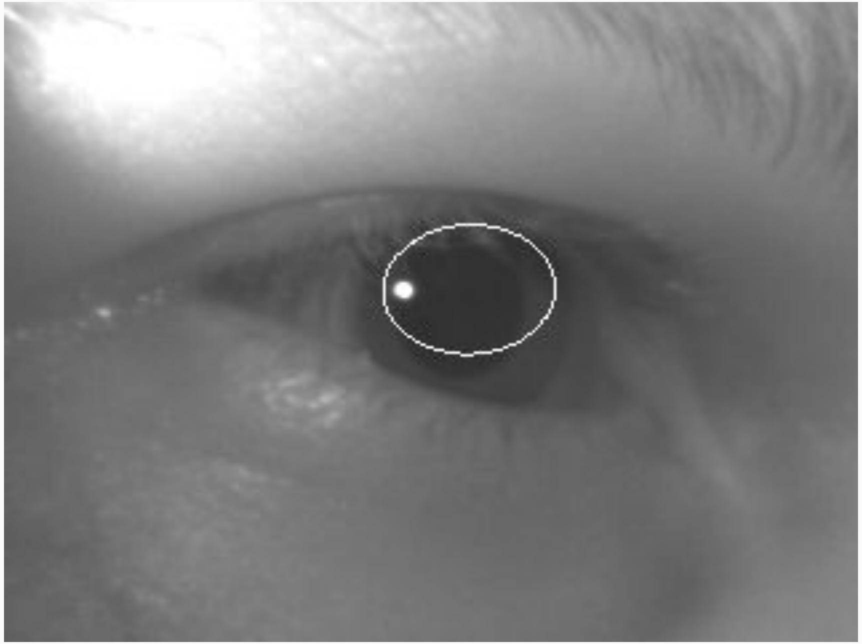
2.d Regression on ellipse parameters

- Input: Reshaped grayscale image in a one-dimensional vector
- Output: 5 parameters of the ellipse, obtained with fitEllipse
- Use of RandomForestRegressor
- Tuning over number estimators and max_depth



2.d Regression on ellipse parameters - Results





Tackle the angle problem

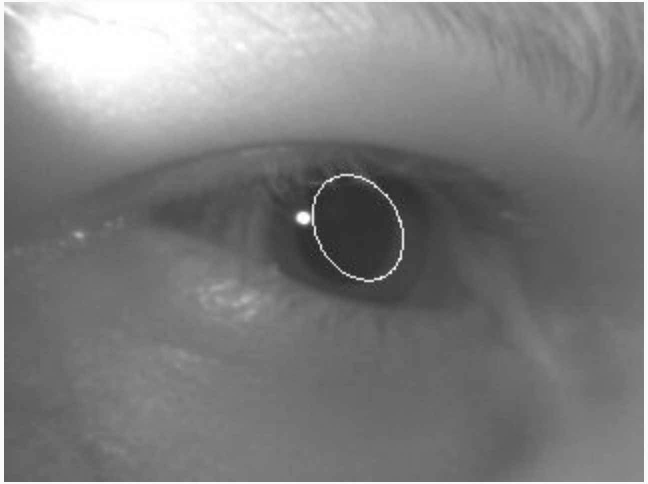
- Variation on the predicting models : RF, MLP, Boosting, were tested.
  - Poor improvements were observed on all tentatives

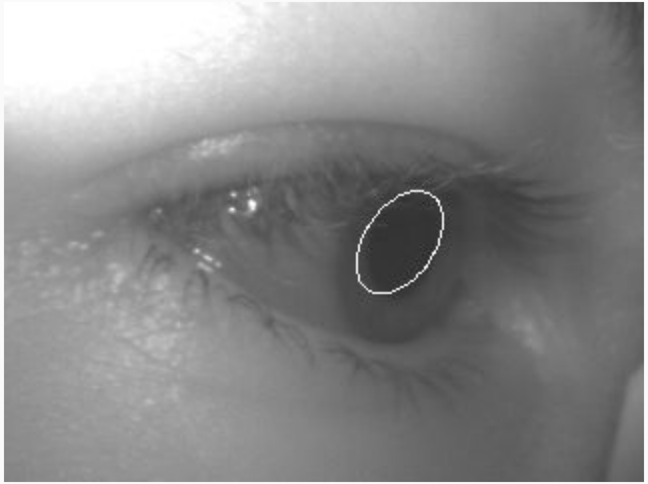
- Decided to train a full model solely for the angle regression

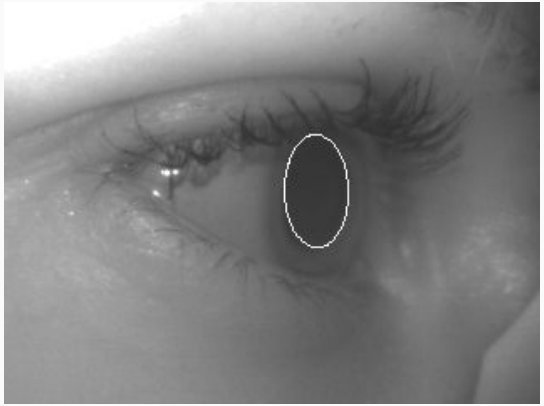
  - MLP, RF, Adaboost, XGBoost, LGBM, Gradient Descent and SVM were tested
  - Still not great, 40° error remains at least
  - Huge models of several thousand estimators haven't helped
  - Study of the evolution of accuracy with respect to size of training set
    - gets better, but price is too few testing elements

2.d Regression on ellipse parameters

Tackle the angle problem : Results







- Reduction of the error from 45 to 10°
- …. but results not trustworthy

3. Performance assessment of the ellipse matching module

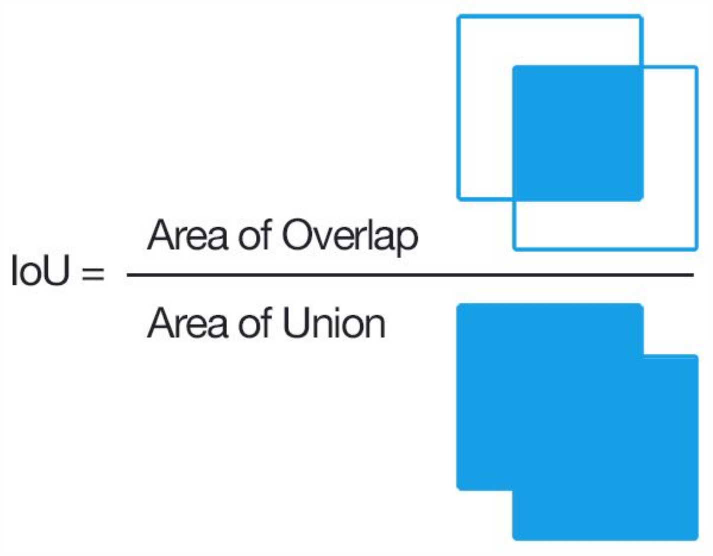
3.a Ellipse classification

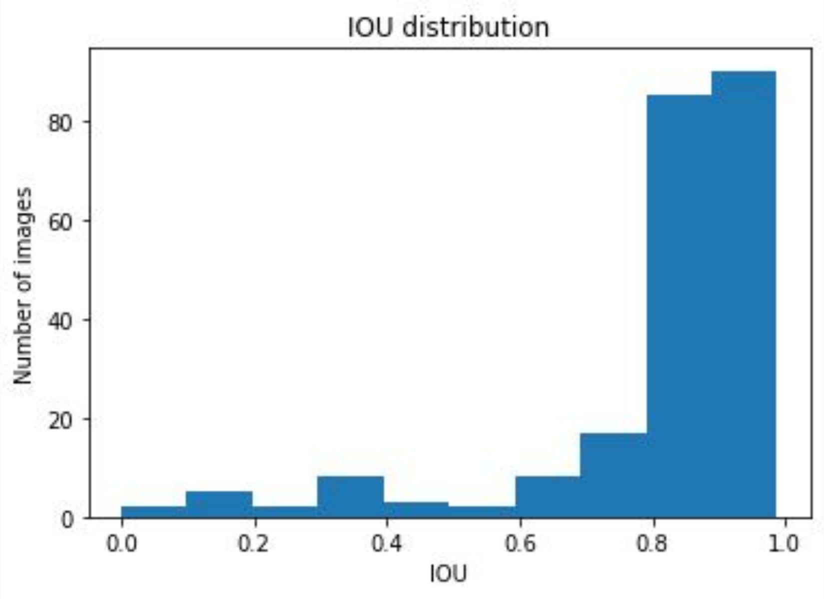
Main metric: **accuracy**

As images with 2 or 3 ellipses are minority classes, accuracy is not relevant for them, so those will be considered as a “detail”

Algorithm	**Accuracy**
ResNet	90% - 96%
RandomForest	~ 94%
Gradient boosting	~ 94%
Adaboost/XGBoost	80 < x < 90%

3.b Bounding boxes regression





Mean : 0.8097
Median 0.8689

3.c Ellipse parameters regression

5 metrics : one per parameter.

Average absolute error, in pixels or degrees. Together gives a good idea of how accurate the model is and gives insight on what parameters are the best and most badly qualified

Algorithm	**Results for the angle**
Gradient descent	50/10°
Random Forest	45°
Boosting	59°
MLP	56°

General results for the other parameters were, using Random Forests :

x : 4.57
y : 5.28
main axis : 5.4
minor axis : 7.26

4. Demo