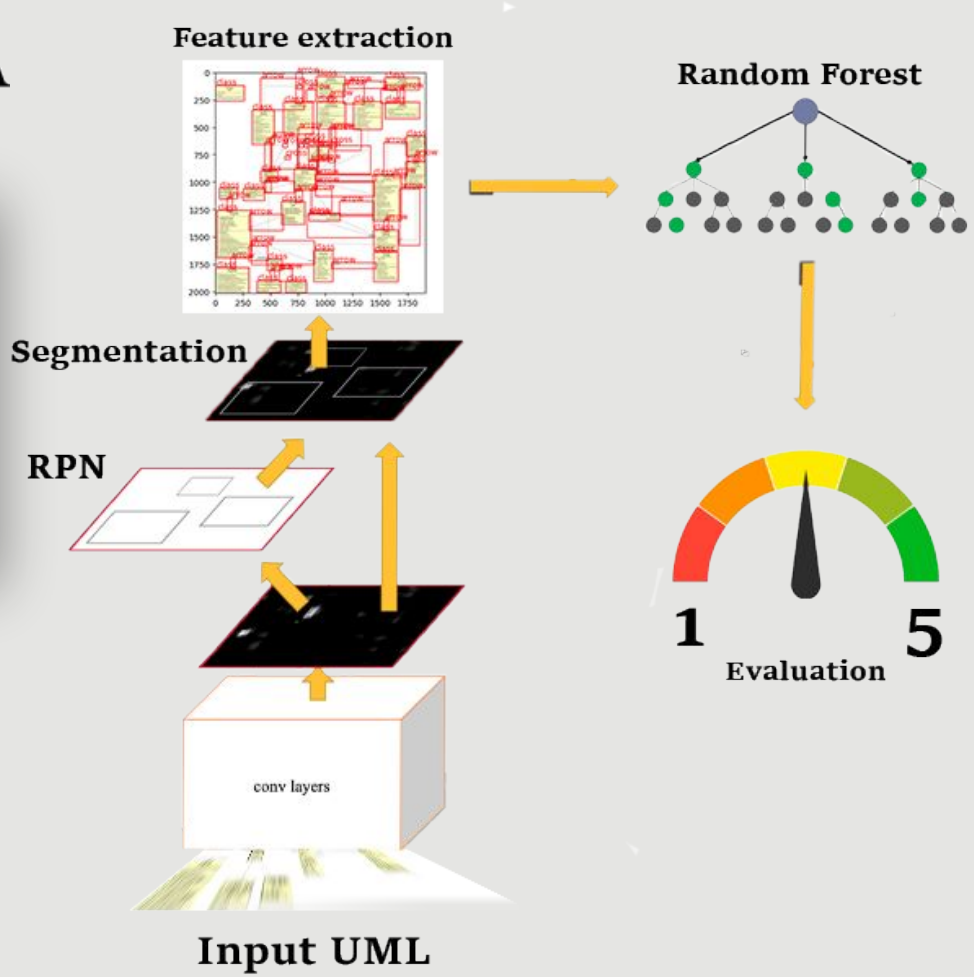
UNIVERSITÀ DI PISA

Using machine learning for automatic classification of the layout quality of UML class diagrams

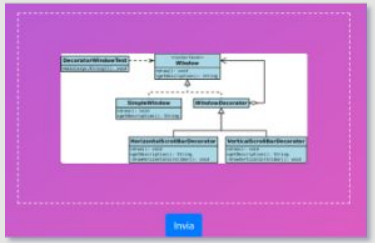
Domenico Armillotta

Supervisor: Prof. Cimino Mario G.C.A.
Alfeo Antonio Luca
Fruzzetti Chiara

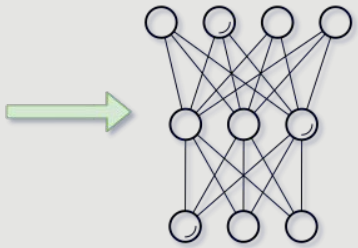


Goal of the Project :

The goal of the project is to create software that given an input image of a UML graph , gives an evaluation on the quality of the schema layout from 0 to 5 and feedback to the designer



Upload schema



Deep learning

Score

+

Feedback

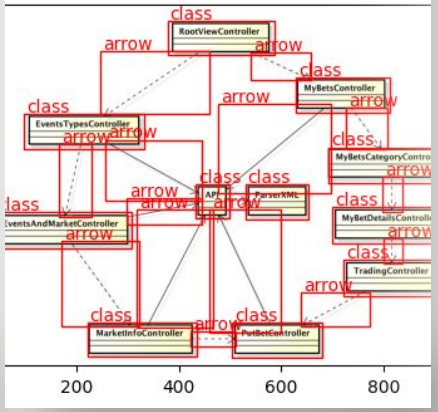
Workflow

0 : State of Art

Analyzed all the paper about this problem

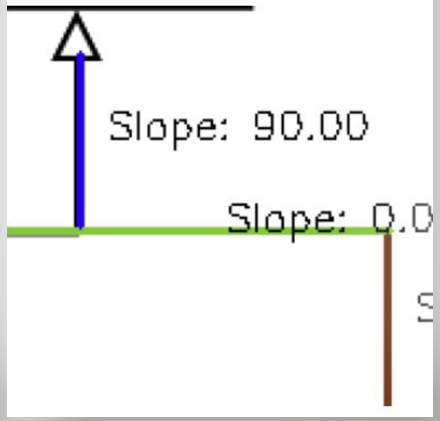
1 : Segmentation Model

Created the model used for segmentation task



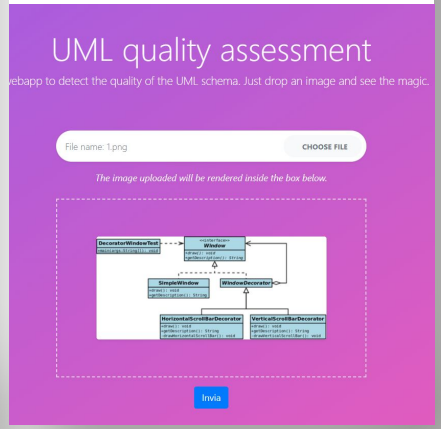
2 : Feature & classification

Extracted all the feature from image and trained the classifier



3 : Deploy final app

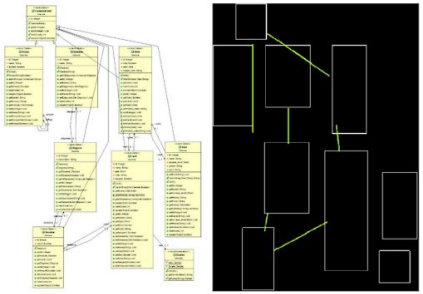
Deployed the final web app with flask



Relevant papers :

Evaluating the layout quality of UML

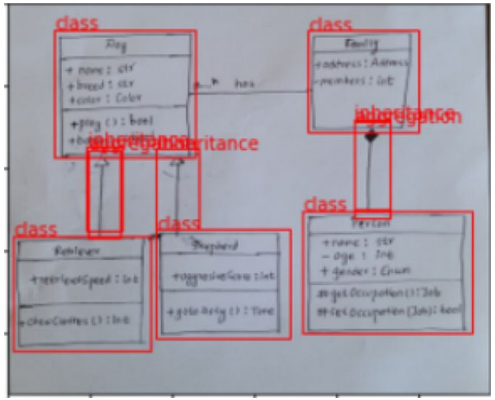
- algorithmic approach
- Dataset
- Feature



1

Parsing from UML Class

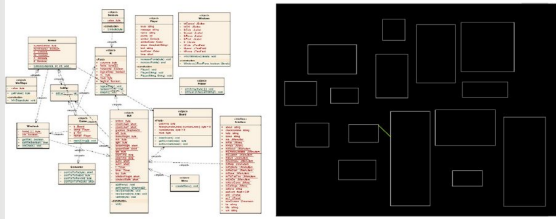
- R-CNN with Detecto approach



2

OPENCV COUNTOURN

- Similar to Paper 1
- Algorithmic approach

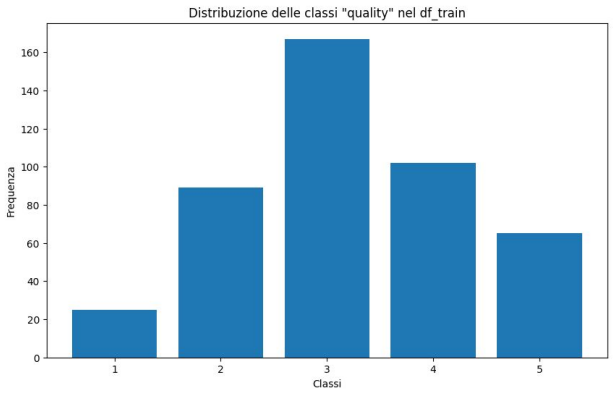


3

Dataset taken from Paper 1

Number of image	651

Classes



Number of element
classes = 7.728
arrows = 7.455
cross = 1.689

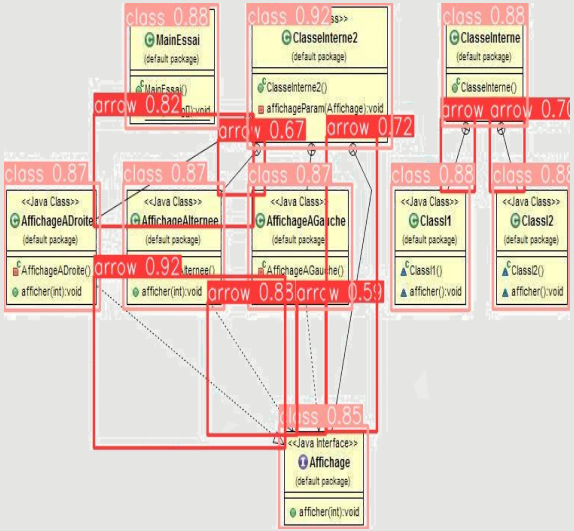
Technique used :

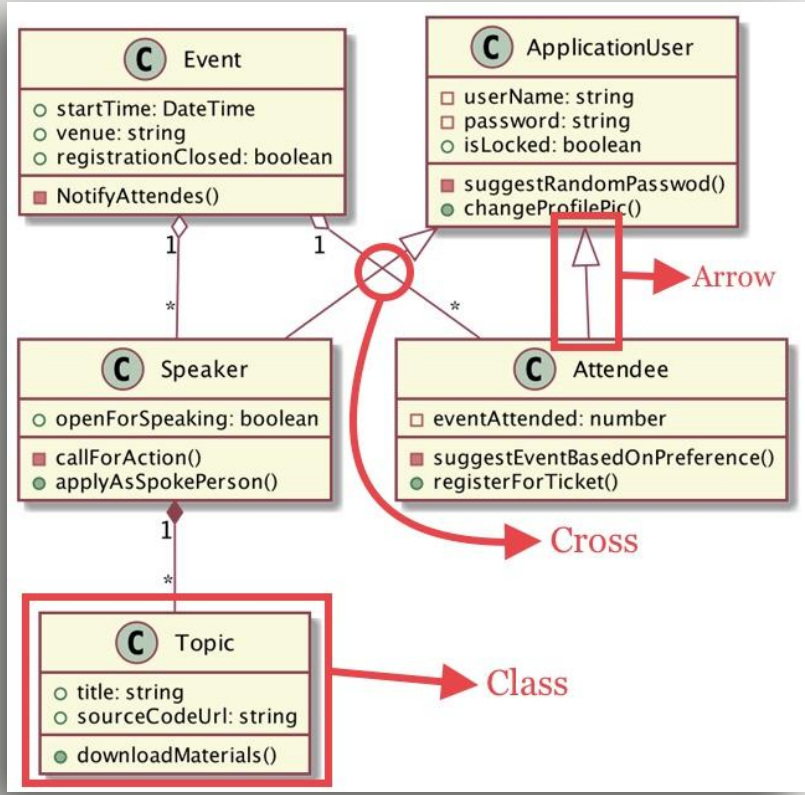
- SMOTE
- AUGMENTATION
- GENERATIVE AI

What to segment

Segmented item :

- Class
- Arrow
- Cross





Model trained for segmentation

YOLO V8

Less accurate but slightly faster than Detecto

1

FASTER R-CNN

Very accurate but issue with complex arrows

2

MIXED OPENCV + FASTER R-CNN

Accurate and fast

3

Picked the best : Mixed Approach

Classes and crosses : Faster R-CNN

Arrow : Line Detection

Precision : 94%

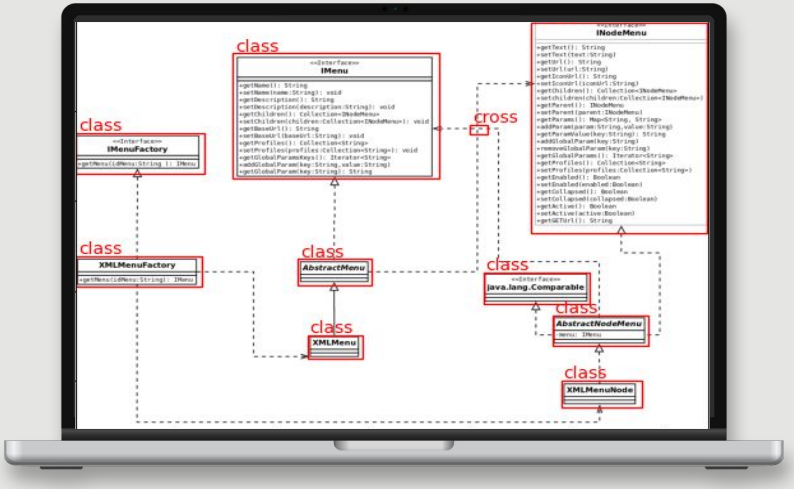
Discarded approach :

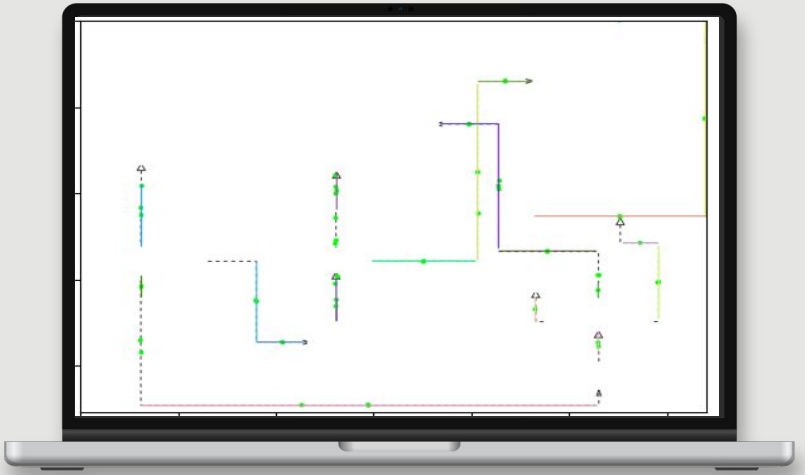
❌ YOLO : lower performance

Accuracy 0.91

❌ Faster R-CNN :

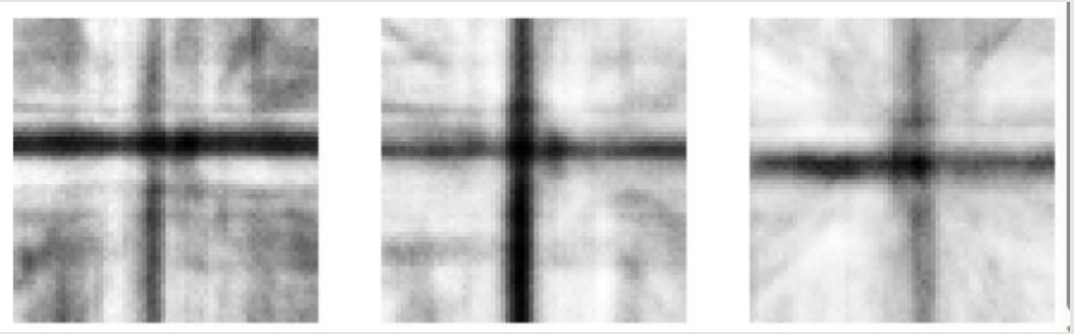
Not precise in the case of complex group of lines





Generation of crosses

’cross’ class is underrepresented in the dataset.
An attempt has been made to generate new data to fill this gap , an approach commonly used in literature is the Variational Autoencoder (VAE)



Images are not as sharp as to be usable in the training phase due to small dataset that constrained the model’s ability to learn robust data representations.

Feature Extraction from segmented image

ARROW

- angles
- orthogonality
- line bends
- avg length
- length variation
- longest line
- shortest line
- number of arrow

CLASS

- coverage
- ratio
- size variation
- size average
- number
- proximity
- distribution
- ratio class/arrow

CROSS

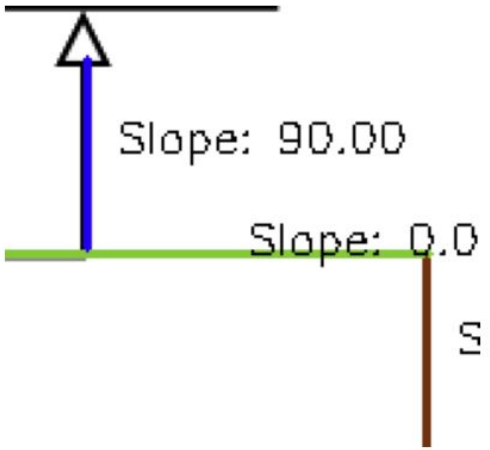
- number of cross
- crossing angles

IMAGE

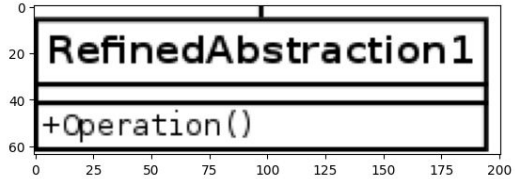
- image ratio
- contrast
- Image Blur
- Color Histogram
- Complexity

Feature Extraction from segmented image

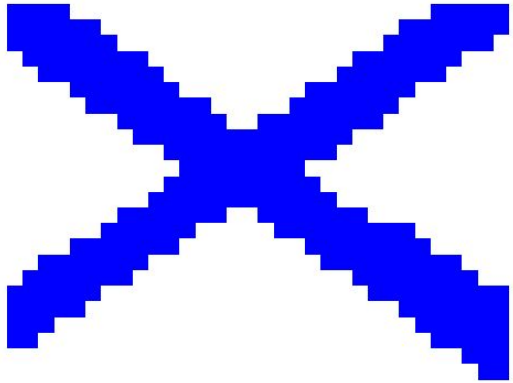
ARROW



CLASS



CROSS



More than 6 different approach discarded

Classifier Trained :

Classifier	Precision	Recall	F1	Deviation
KNN	0.48	0.44	0.46	1.02
Random Forest	**0.65**	**0.59**	**0.62**	**0.75**
Decision Tree	0.47	0.52	0.49	1.00
Naive Bayes	0.38	0.38	0.38	0.92
Neural Network	0.22	0.30	0.25	1.22
Ensemble : SVM , Gradient Boosting & Random Forest	0.50	0.38	0.42	0.87

Performance achieved after dataset balancing with SMOTE , feature importance analysis, and hyperparameter tuning with grid search on n_estimators,max_depth,min_samples_split,min_samples_leaf,max_features (in the case of Random Forest)

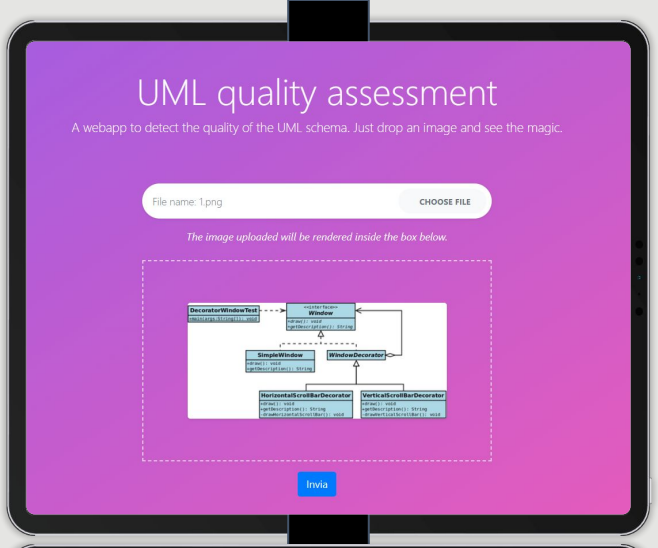
Web deploy with Flask

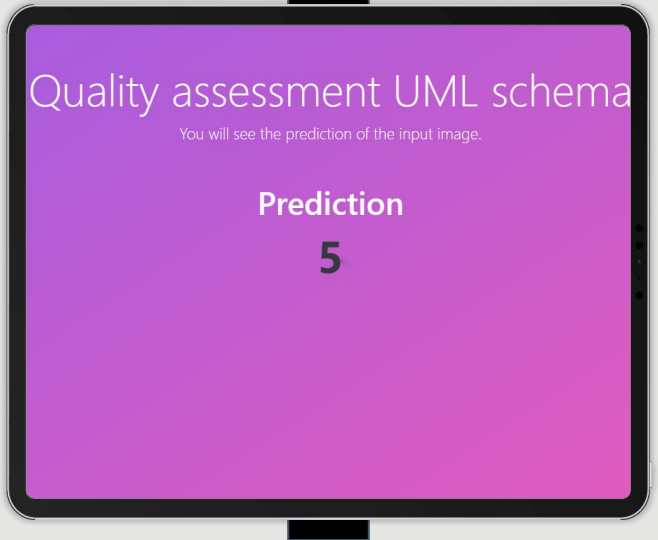
Flask allowed deployment and the latency time for each image is less than 4 seconds

Or textual

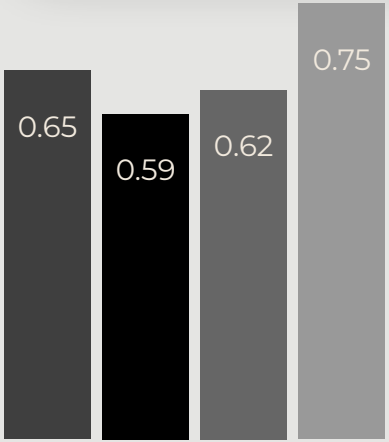
In this case is possible to load multiple schema and receive for each the rating and the feedback.

```
############ Image 0  ----> RATING : 5. ############
The Rectangle proximity is already optimal for image 0.
The Number of rectangles is already optimal for image 0.
The Line bends is already optimal for image 0.
The Number of lines is already optimal for image 0.
The Longest line is already optimal for image 0.
The Average line length is already optimal for image 0.
############ Image 1  ----> RATING : 1. ############
The Rectangle proximity should be decreased for image 1.
The Number of rectangles should be decreased for image 1.
The Line bends should be decreased for image 1.
The Number of lines should be decreased for image 1.
The Longest line should be decreased for image 1.
The Average line length should be decreased for image 1.
############ Image 2  ----> RATING : 2. ############
The Rectangle proximity should be decreased for image 2.
The Number of rectangles should be decreased for image 2.
The Line bends should be decreased for image 2.
The Number of lines should be decreased for image 2.
The Longest line should be decreased for image 2.
The Average line length should be decreased for image 2.
```



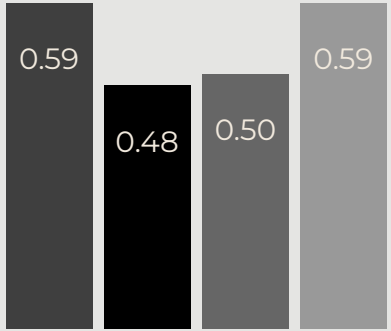


Final Performance :



My model

Using Faster rcnn + random forest



State of Art

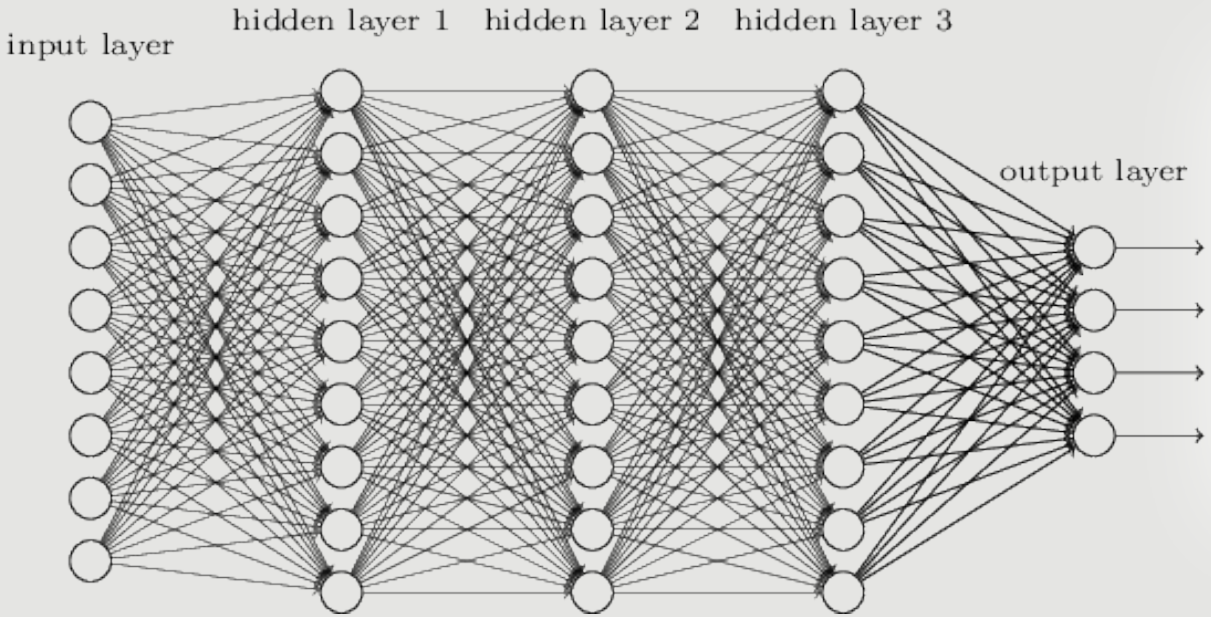
Using contourn approach + Weka random forest

Accuracy

Recall

F1

Deviation


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

Thanks!