



UNIVERSITÉ  
TOULOUSE III  
PAUL SABATIER



Université  
de Toulouse



IRIT

CNRS - INPT - UPS - UT1 - UTM



Institut de Recherche en Informatique de Toulouse



EcoLab



# Localisation de poisson dans une image : Recette

**Clients** : Alain Couzil, Bérengère Mathieu

**Présenté par** : Benjamin Nevers, Fateh Benmerzoug, Julie Chéoux, Marina Bertolino, Vincent Laborde

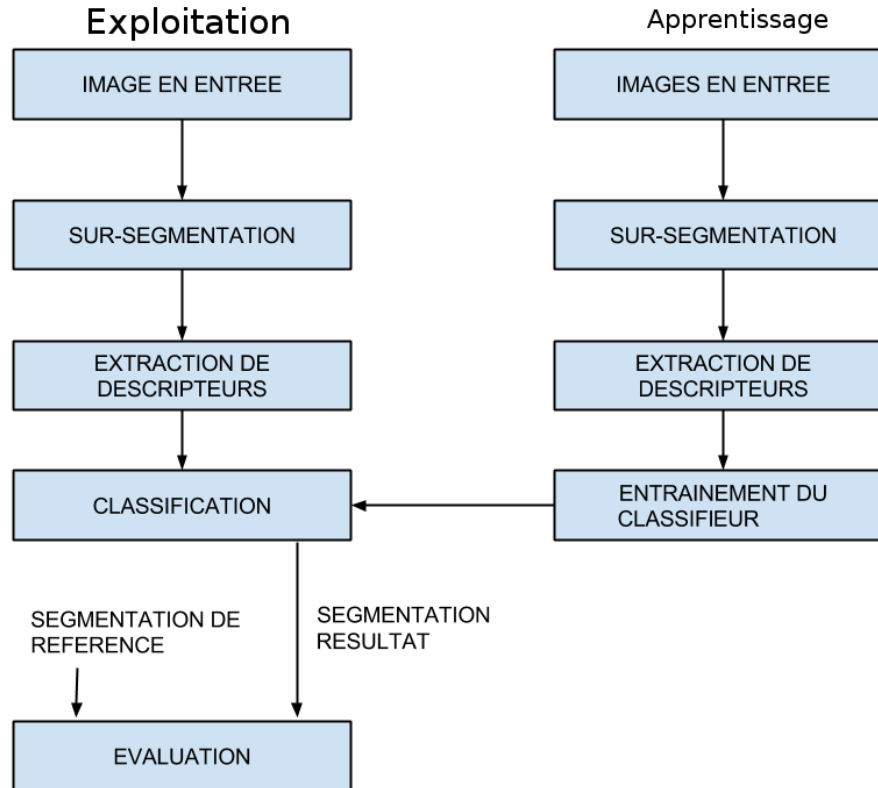
# Plan de la présentation

- I. Introduction
- II. Travail effectué par modules
- III. Tests
- IV. Présentation de l'application
- V. Conclusion et perspectives

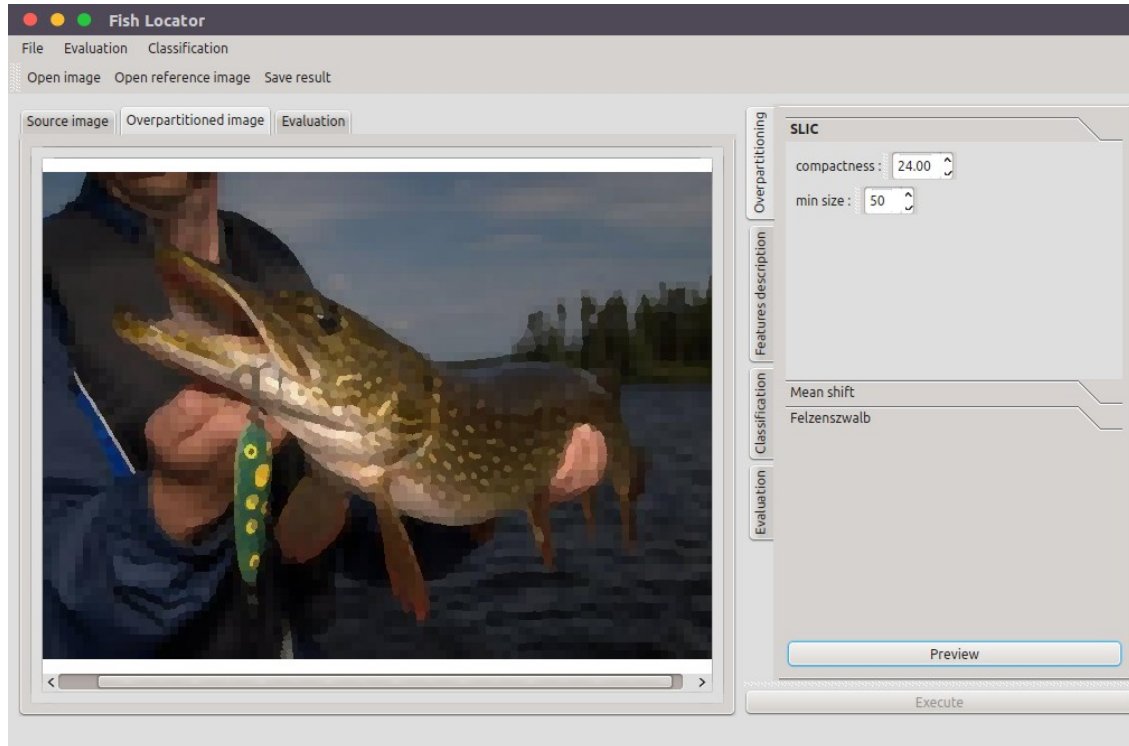
# I. Introduction

- Contexte
  - Laboratoire ECOLAB
  - Authentification de poissons
- Présentation du sujet
  - Localisation dans l'image du poisson pêché
  - Simplification des méthodes de Fulkerson et Liu

# I. Introduction : Pipeline



# I. Introduction : Interface



# I. Introduction : Application

- Une interface graphique ;
- Trois méthodes de sur-segmentation ;
- Deux méthodes d'extraction de descripteurs par la texture ;
- Deux méthodes de classification de superpixels ;
- Deux méthodes d'évaluation.

# Plate-forme de développement

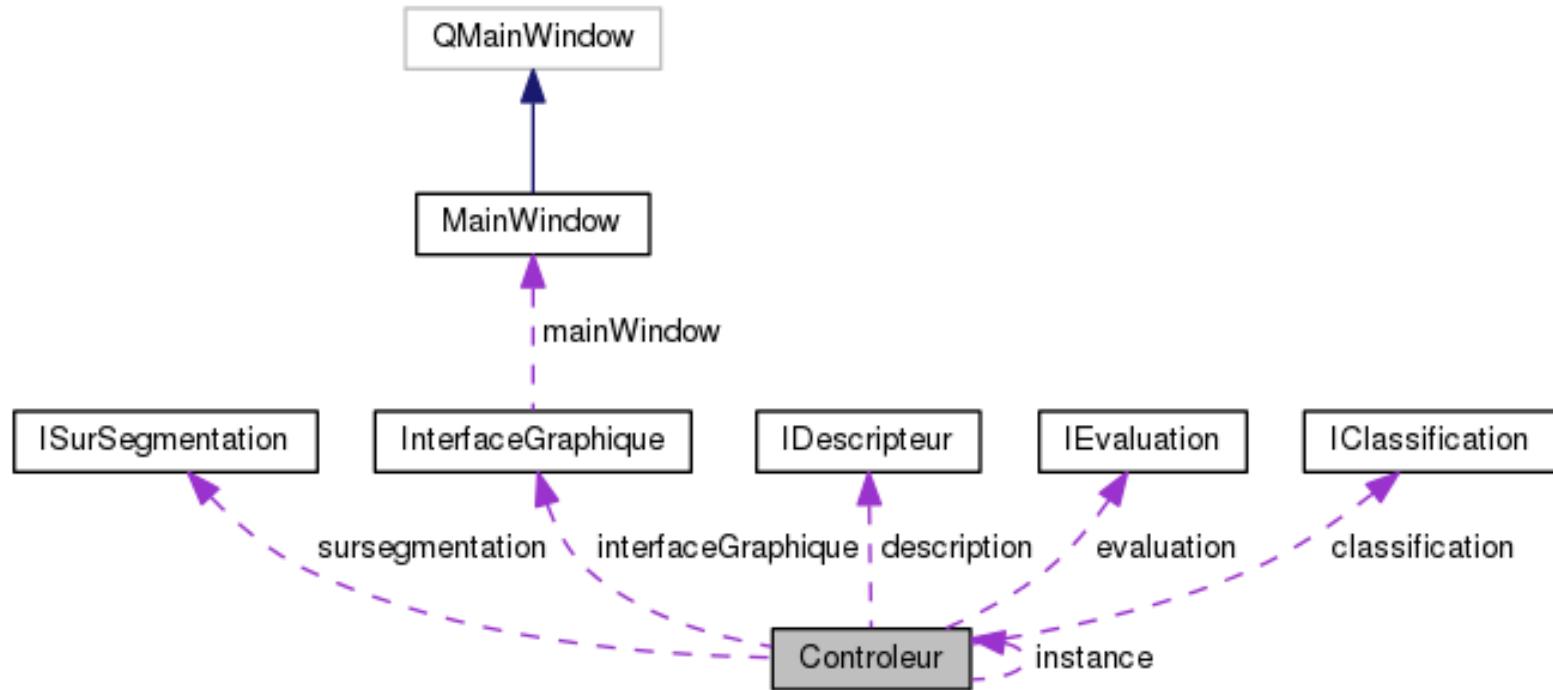
- Langage C++
- Code existant
  - Les méthodes de sur-segmentation sont fournies par le client.
- Différentes bibliothèques
  - qt, libsvm, alglib.

# II. Travail effectué par modules

1. Le module Contrôleur
2. Le module Sur-segmentation
3. Le module Description
4. Le module Classification
5. Le module Évaluation
6. Le module Interface Graphique



## II. Interaction entre les modules

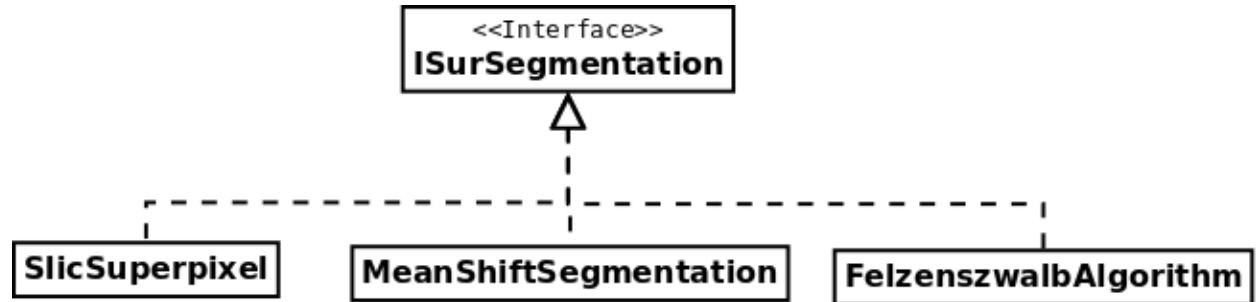


# II.1. Module Contrôleur

- Gère les interactions entre les modules ;
- Contient les fonctions “Set” et “perform” ;
- Fonctionne aussi en mode console.

## II.2. Module Sur-segmentation

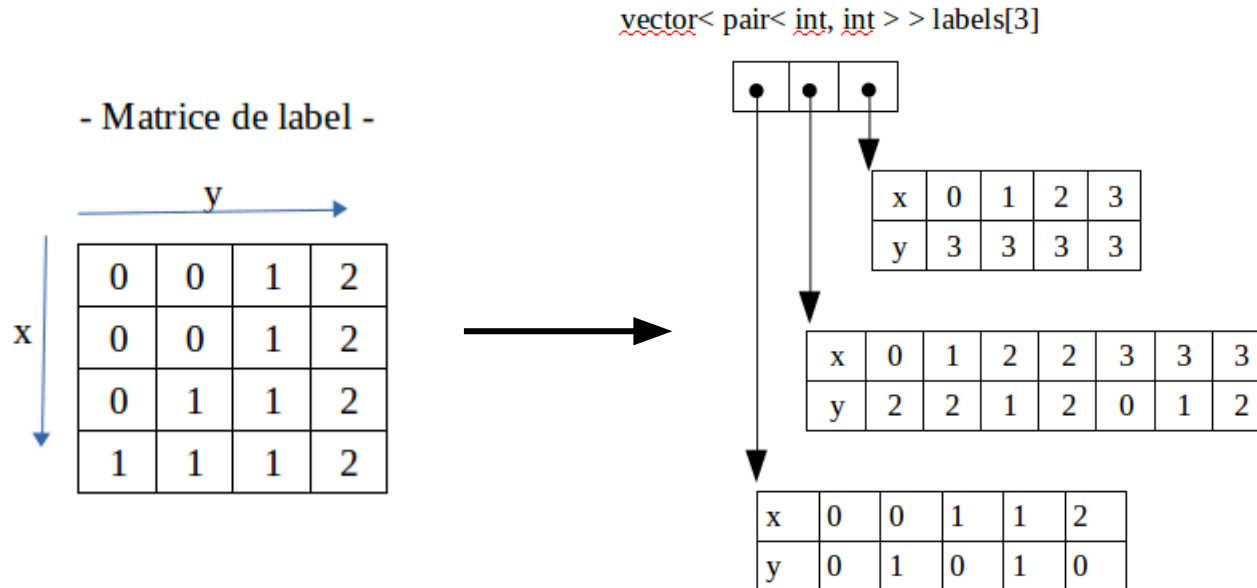
La sur-segmentation sert à découper l'image chargée à partir de l'interface graphique, en plusieurs sous-régions de pixels homogènes appelés superpixels.



# II.2. Module Sur-segmentation

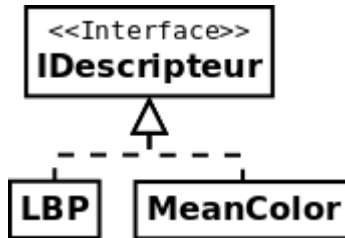
Pour faciliter la manipulation de données :

- *transformeImageLabel* : `vector<pair<int, int>> * labels`.



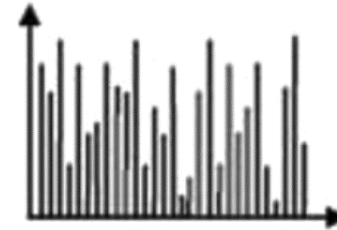
# II.3. Module Description

Un descripteur nous permet de représenter la région d'un pixel par sa texture ou sa couleur, afin d'obtenir une information sur ce pixel et son voisinage.



Pour LBP :

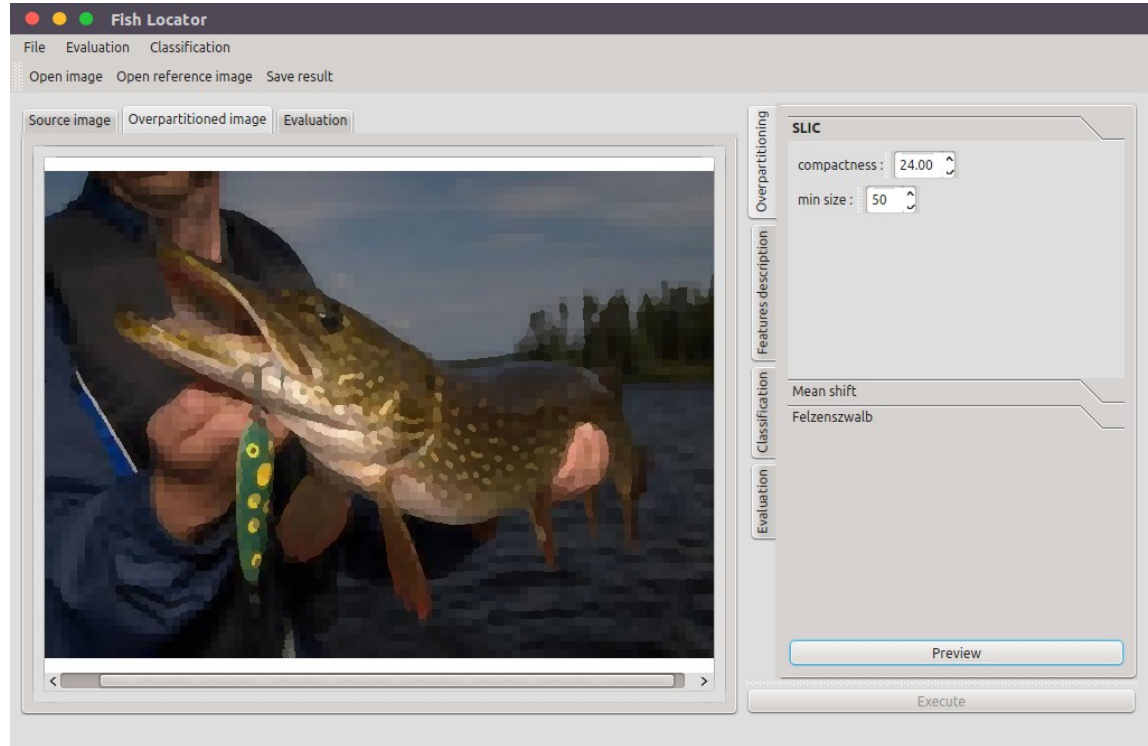
- un descripteur = un histogramme ;
- invariance en rotation et en luminance .



Histogramme représentant un descripteur d'un superpixel

# II.3. Module Description

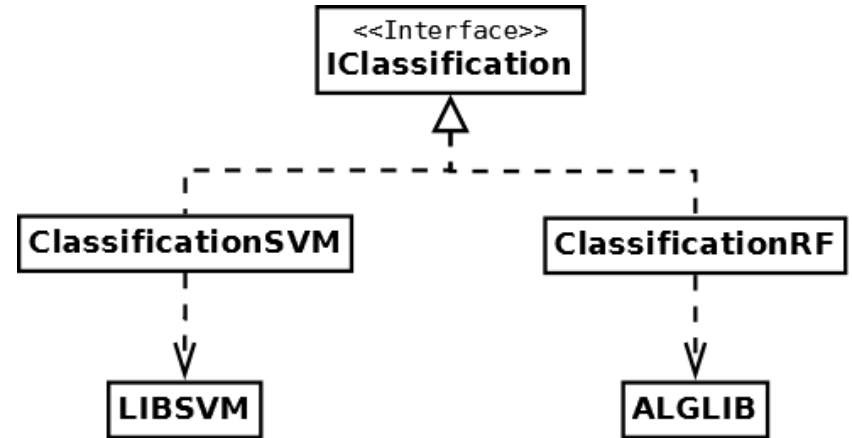
Mean color



# II.4. Module Classification

La classification permet de déterminer si un superpixel correspond au brochet sur la photographie ou non.

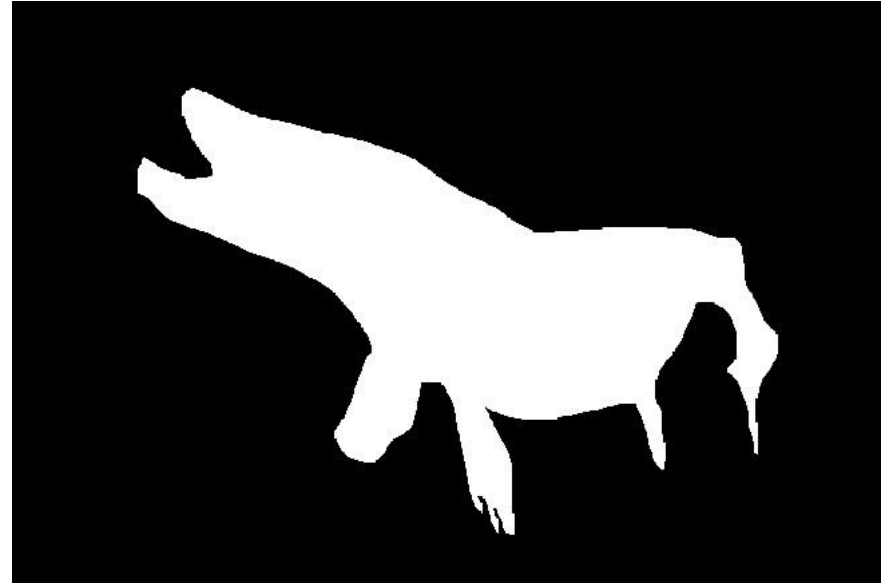
- Contient les méthodes :  
load, save, train, classify.
- Deux phases :  
entraînement, classification.
- Retour :  
vector<int> représentant la classe de chaque superpixel.



## II.4. Module Classification



Image source



segmentation de reference



# II.5. Module Évaluation

- Évalue les méthodes utilisées pour segmenter ;
- Comparaison avec une segmentation de référence ;
- 2 types d'évaluations :
  - Classique ;
  - Non classique.
- Résultats :
  - Image (noir/blanc/rouge) ;
  - Précision, rappel, pourcentage de pixels corrects.

# II.5. Module Évaluation

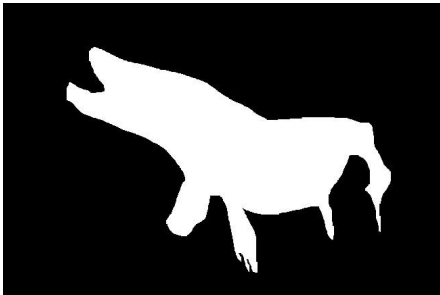
Image source



Segmentation résultat



Segmentation de référence



Résultat visuel de l'évaluation

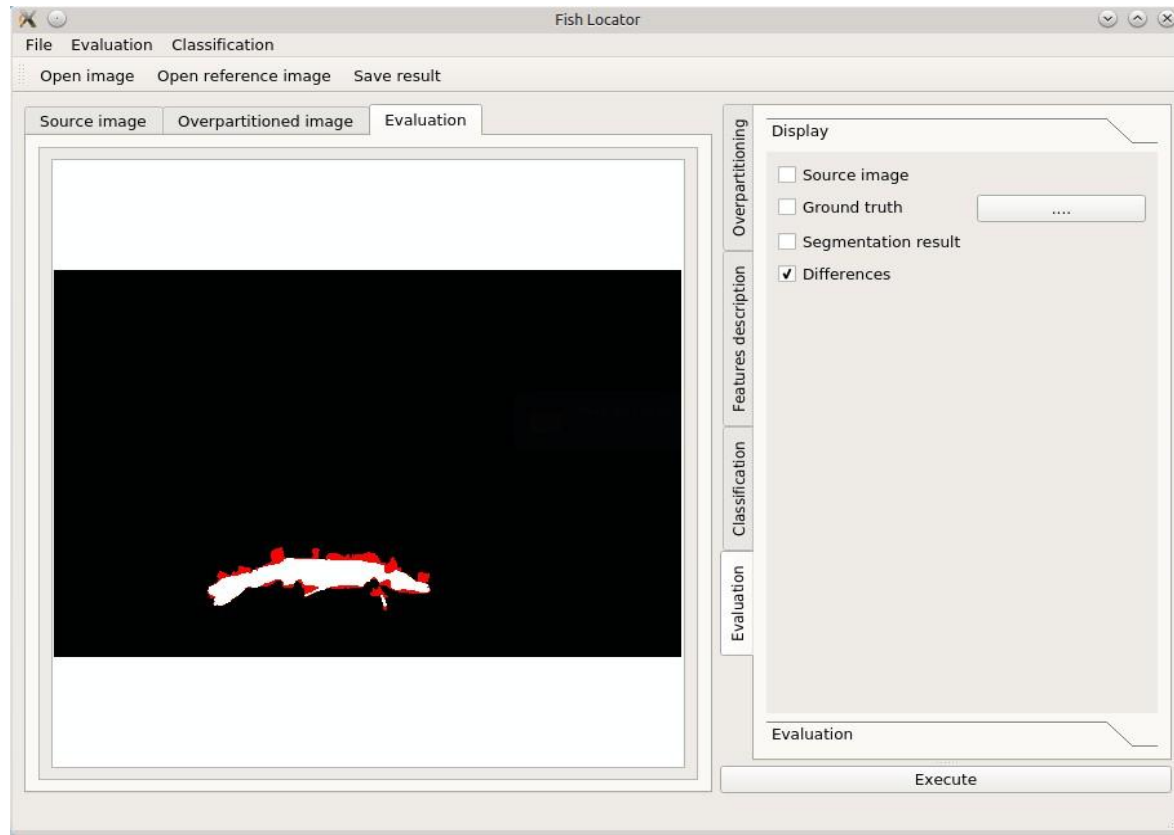
# II.6. Module Interface graphique

L'interface permet de mettre en relation l'utilisateur et le contrôleur par le biais d'une multitude de widgets.

Permet de :

- Charger une image ;
- Sélectionner une des méthodes à appliquer ;
- Choisir les paramètres des méthodes ;
- Sauvegarder un entraînement ;
- Sauvegarder le résultat (image + fichier xml) ;
- Afficher par calques.

# II.6. Module Interface graphique

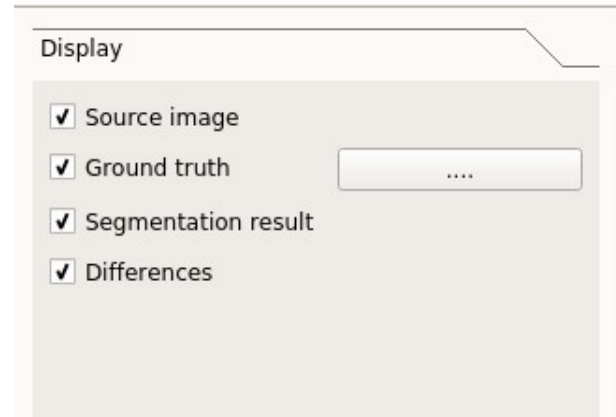


# III. Tests

- **Module Contrôleur :**
  - Nous avons vérifié qu'il faisait bien appel aux bonnes fonctions suivant les méthodes choisies dans l'application ;
  - Nous avons vérifié que le programme fonctionne en mode console.

# III. Tests

- Module Interface graphique :
  - Tester l'édition des différents paramètres ;
  - Tester l'appel des fonctions concernées lors des clics sur les boutons ;
  - Tester les différents boutons preview, save, load, execute, evaluate ;
  - Tester l'affichage avec les différents checkbox de l'onglet Display.



# III. Tests

- **Module Sur-segmentation :**
  - Comparer les résultats du code existant avec nos résultats ;
  - Tester sur de petites matrices la fonction `transformeImageLabel`.
  
- **Module Description (utilisation de code jetables) :**
  - Classe `LBP` : vérifier le résultat de l'histogramme obtenu, avec des petites matrices ;
  - Classe `MeanColor` : comparer les calculs obtenus avec le résultat attendu.

# III. Tests

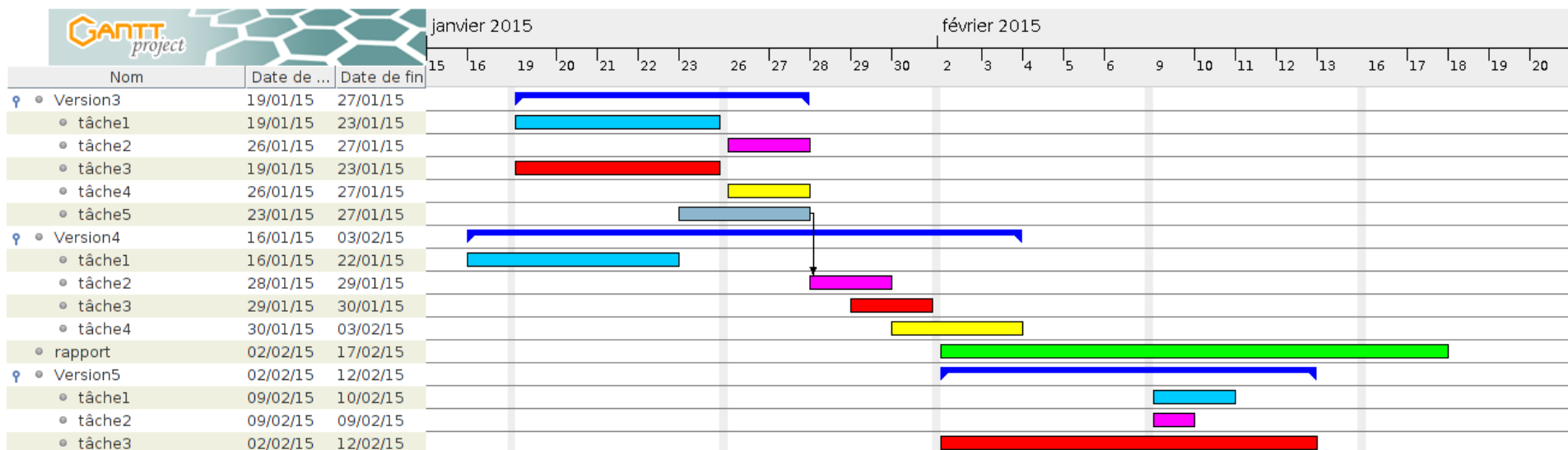
- **Module Classification :**
  - Analyser les informations renvoyées par les fonctions des bibliothèques ;
  - Retourne bien 0 ou 1 (c'est-à-dire “pas brochet” ou “brochet”).
- **Module Évaluation :**
  - Mettre la même image en référence et en résultat (~ 100%) ;
  - Mettre deux images complètement différentes (~ 0%).



# IV. Présentation de l'application



# Planning mis à jour



# Livrables

- Code source ;
- Manuel d'utilisation ;
- Documentation Doxygen ;
- Site Web.

# Le site web

## Chef d'oeuvre M2IM : Localisation de poissons



### PRÉSENTATION

Dans le cadre du master 2 Image et Multimédia, nous avons participé à l'UE (unité d'enseignement) chef d'oeuvre, encadré par Mathias Paulin, David Vanderhaeghe et Christophe Collet, dans laquelle il nous a été demandé de développer un projet en groupe implémentant plusieurs méthodes décrites dans des articles de recherche. Ce projet de fin d'études nous a permis de mettre en pratique les connaissances acquises lors nos cours à l'université.

Dans le cadre de ses travaux avec EcoLab (laboratoire écologie fonctionnelle et environnement), l'équipe TCI (Traitement et

### À PROPOS DE NOUS

Nous sommes 5 étudiants en Master 2 d'Informatique Image et Multimédia à l'université Paul Sabatier de Toulouse :

- Fateh Ben Merzoug
- Marina Bertolino
- Julie Chéoux

# Conclusion et perspectives

Le travail demandé a été fait :

- Les versions ont été validées par nos clients.

Perspectives :

- Augmenter le nombre d'images pour l'entraînement ;
- Possibilité d'ajout de méthodes (HOG, SIFT...) ;
- Prochaine étape → authentification individuelle ;
- Généralisation du projet.

