



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 : Spécifications

Clients : Alain Crouzil, Bérengère Mathieu

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

Plan de la présentation

- I. Introduction
- II. Cahier des charges
- III. Vue générale du système
- IV. Planning, tests et analyse de risques

I. Introduction

- Contexte
 - Laboratoire ECOLAB
 - Authentification de poissons
- Présentation du sujet
 - Localisation dans l'image du poisson pêché
 - Pas d'objectif de résultat

I. Introduction

Le cas simple :

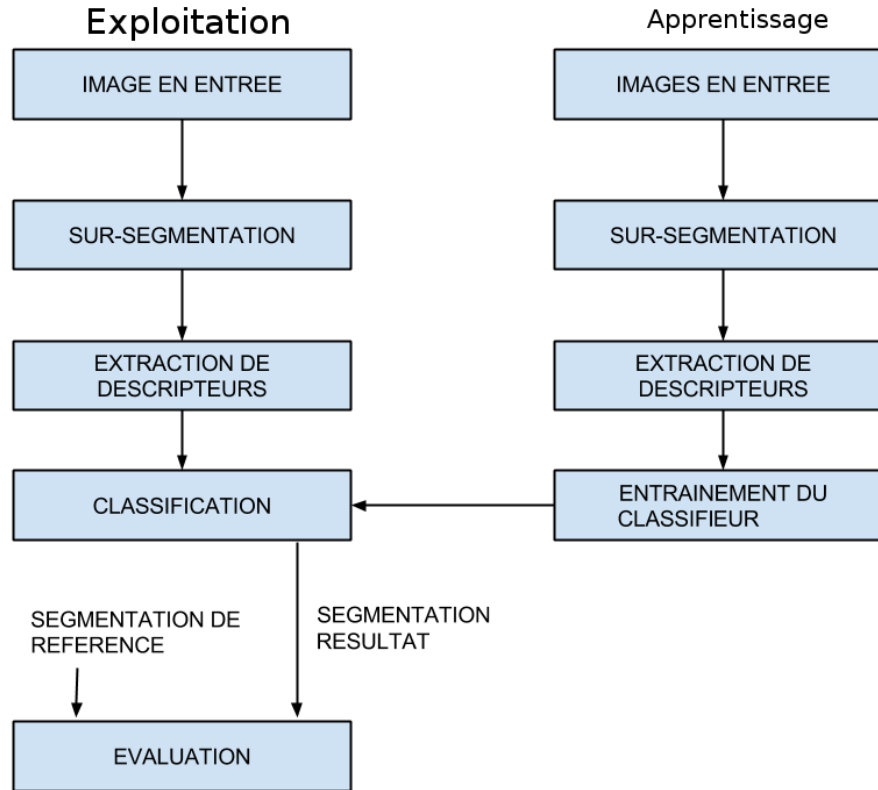


I. Introduction

Le cas compliqué :



I. Introduction : Pipeline



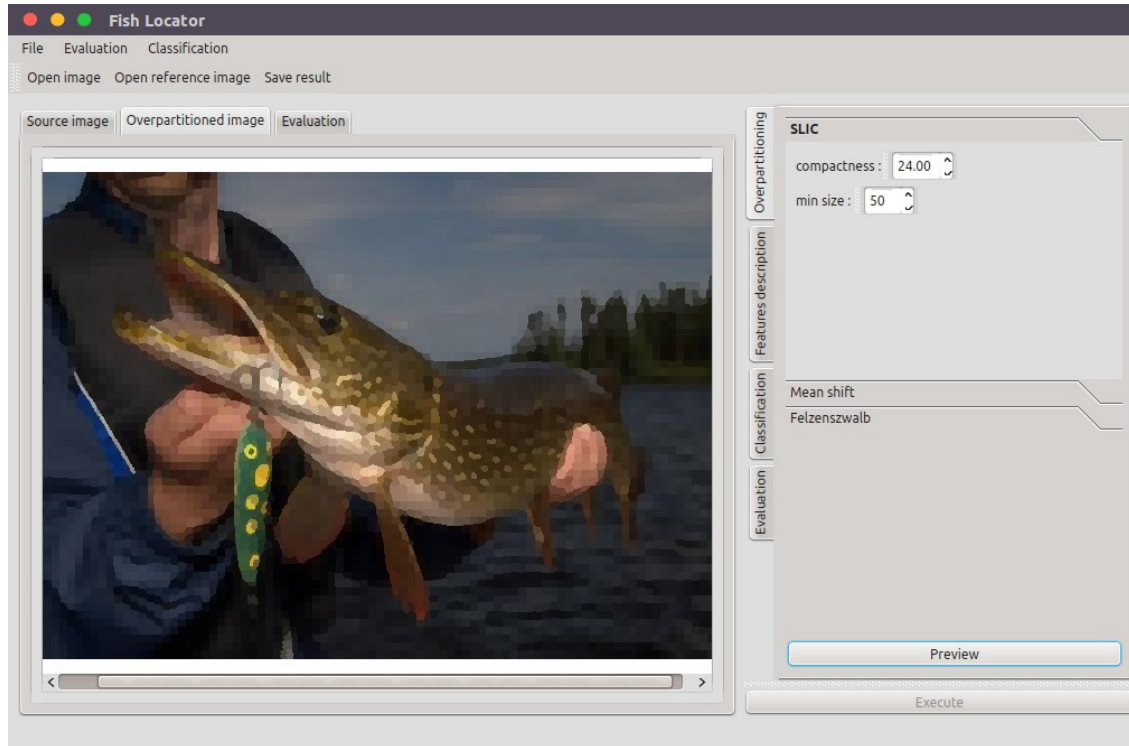
II. Cahier des charges

- Description du projet
- Analyse fonctionnelle du projet
- Analyse fonctionnelle par version

II.1. Description du projet

- 3 méthodes de sur-segmentation
- au moins 1 méthode d'extraction de descripteurs par la texture
- 2 méthodes de classification de superpixels
- entraînement des classifieurs en utilisant des données reçues au préalable
- comparaison du résultat à une segmentation de référence
- interface graphique

II.2. Analyse fonctionnelle du projet



II.2. Analyse fonctionnelle du projet

Exemple d'image source



II.2. Analyse fonctionnelle du projet

Une segmentation de référence, fournie par le client



II.2. Analyse fonctionnelle du projet

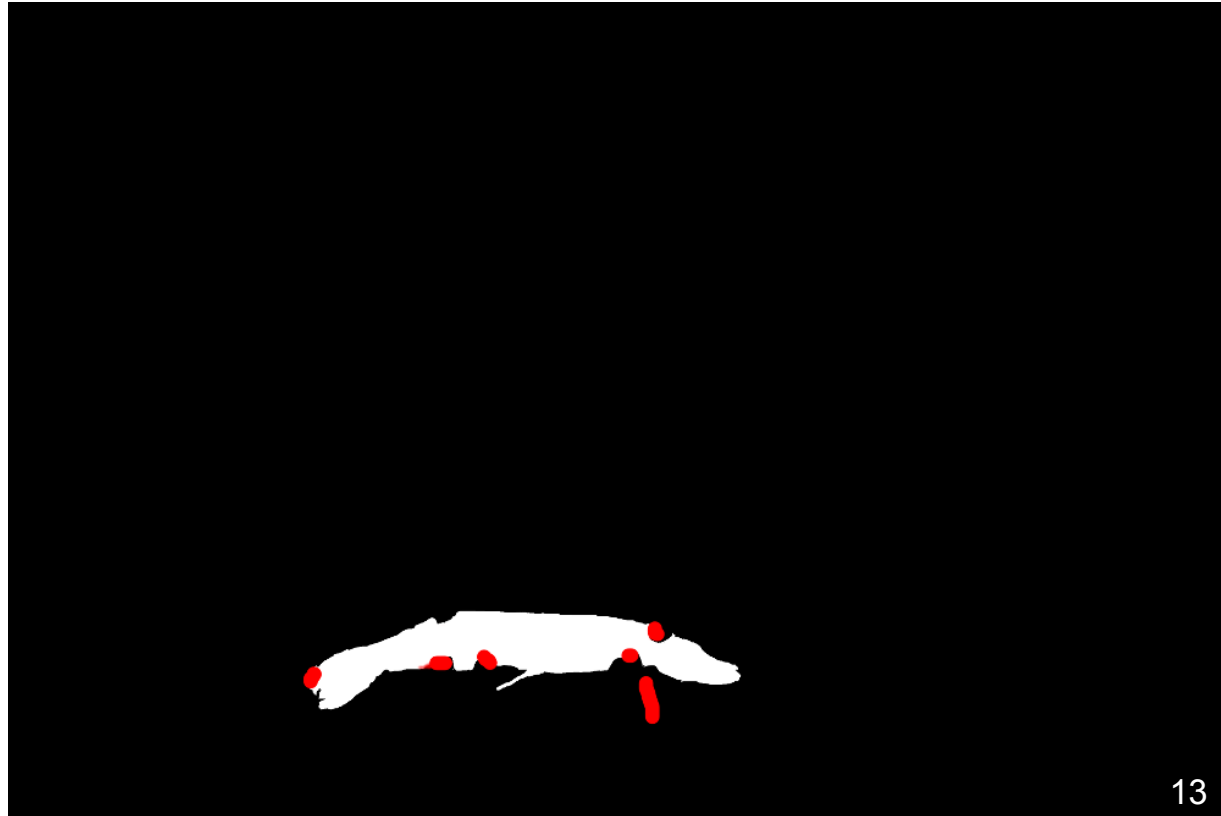
Une segmentation
résultat de l'application



II.2. Analyse fonctionnelle du projet

Pixels mal classés en rouge :

faux positifs, et faux négatifs



II.2. Analyse fonctionnelle du projet

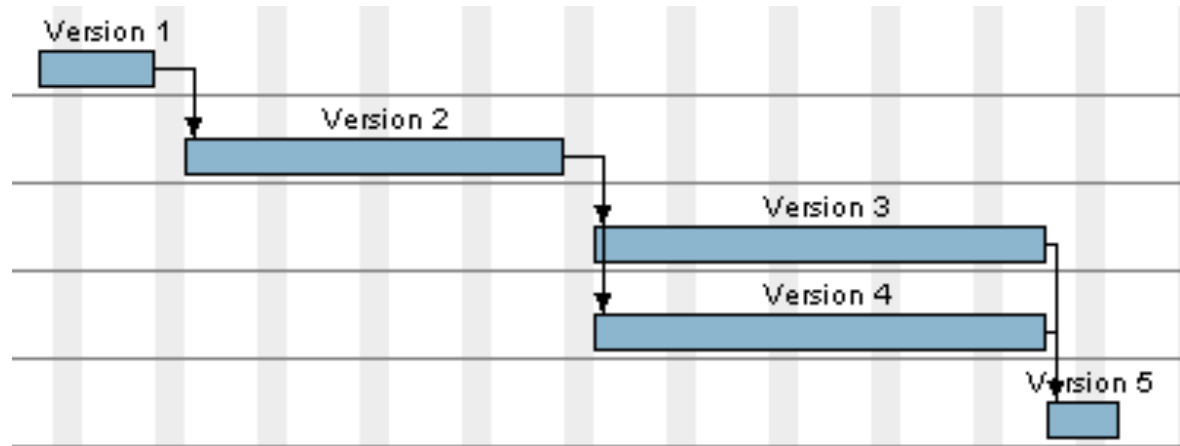
Système d'affichage par calques :

image source + segmentation + pixels mal classés

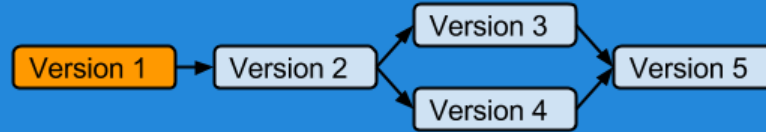


II.3. Analyse fonctionnelle par version

- Cycle itératif
- 5 versions dépendantes

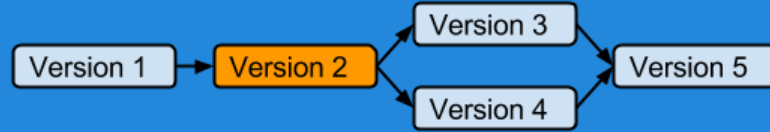


II.3. Versions



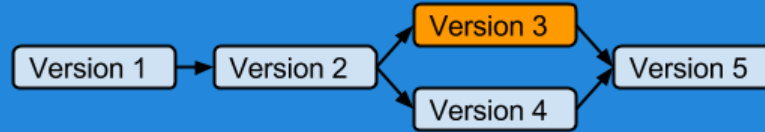
- Version 1
 - Réaliser l'interface graphique
 - Charger et afficher une image
- Version 2
 - Intégrer les différentes méthodes de sur-segmentation
 - Afficher les superpixels à l'aide de leur couleur moyenne

II.3. Versions



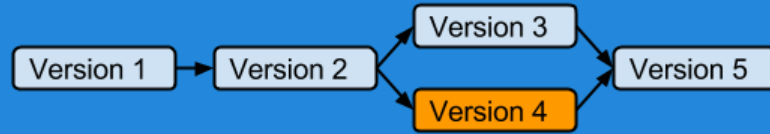
- Version 1
 - Réaliser l'interface graphique
 - Charger et afficher une image
- Version 2
 - Intégrer les différentes méthodes de sur-segmentation
 - Afficher les superpixels à l'aide de leur couleur moyenne

II.3. Versions



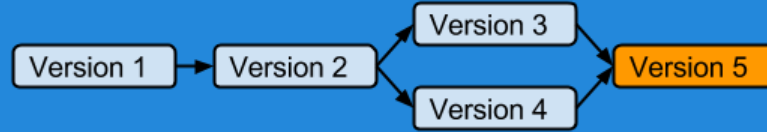
- Version 3
 - Intégrer les méthodes de classification supervisée et entraîner les classifieurs
 - Afficher le résultat : pixels du poisson en blanc, le reste en noir
- Version 4
 - Implémenter un descripteur de texture : LBP

II.3. Versions



- Version 3
 - Intégrer les méthodes de classification supervisée et entraîner les classifieurs
 - Afficher le résultat : pixels du poisson en blanc, le reste en noir
- Version 4
 - Implémenter un descripteur de texture : LBP

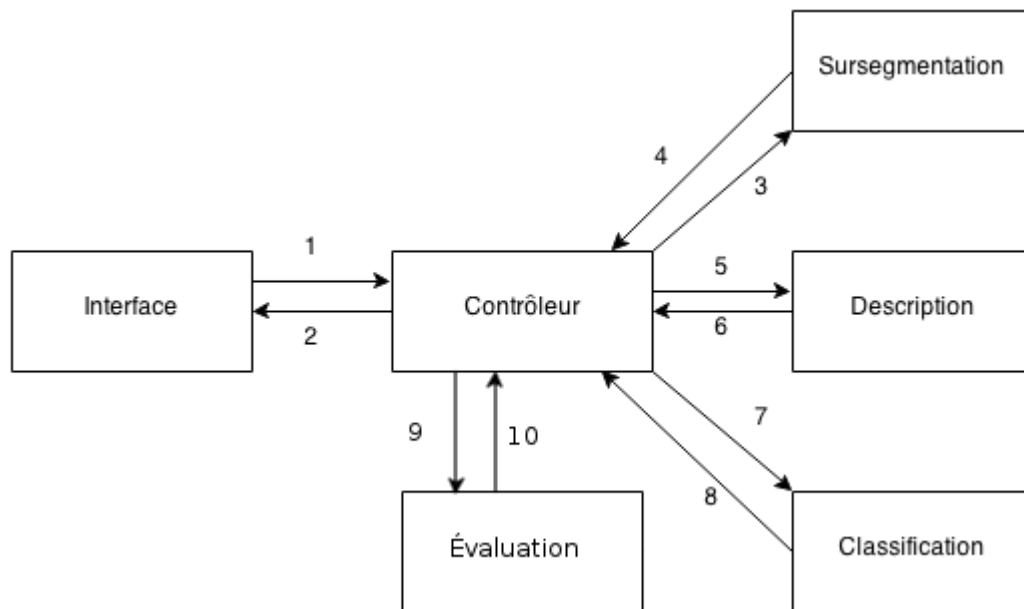
II.3. Versions



- Version 5
 - Charger la segmentation de référence fournie par l'utilisateur
 - Calculer le pourcentage de pixels corrects avec plusieurs méthodes d'évaluation

III. Vue générale du système

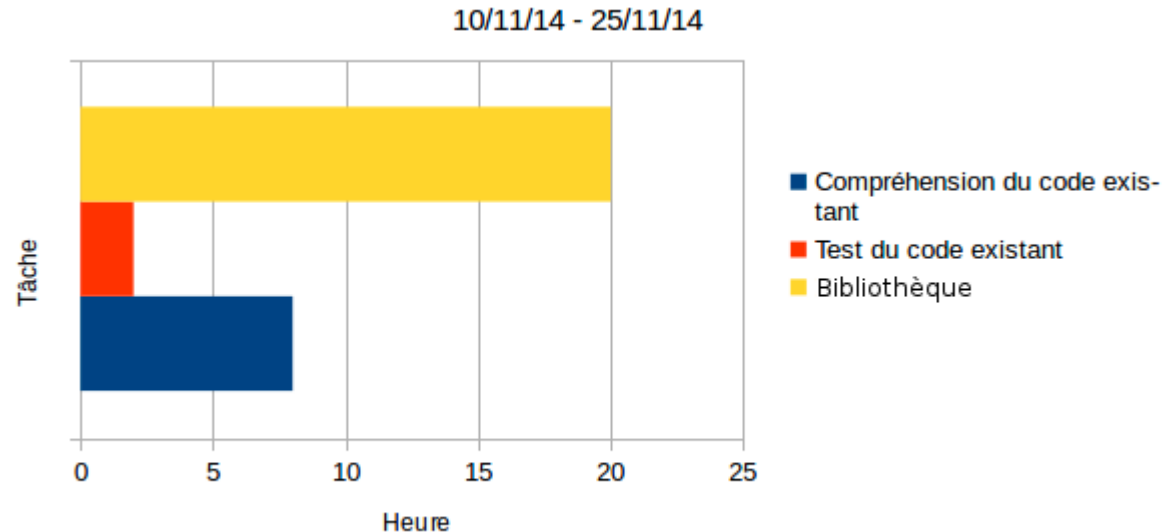
1. interaction interface
2. affichage résultats
3. envoi image source
4. réception des superpixels
5. envoi des superpixels
6. réception des descripteurs
7. envoi des descripteurs
8. image segmentée
9. segmentations
10. pourcentage pixels bien classés



IV.1. Planning du projet

- 5 grandes étapes divisées en tâches

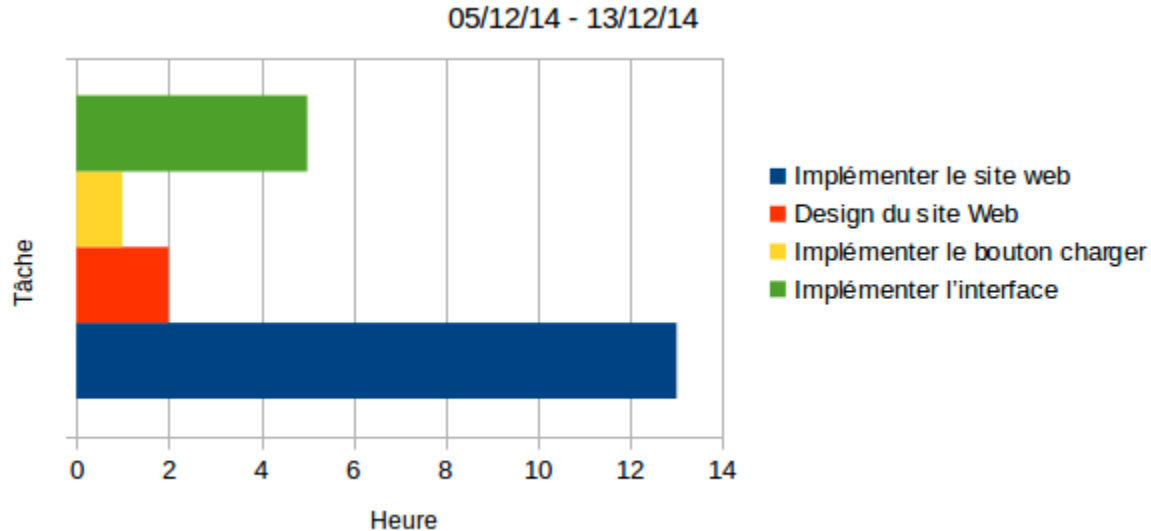
Étape 1 : prise en main



- ~30 heures

IV.1. Planning du projet

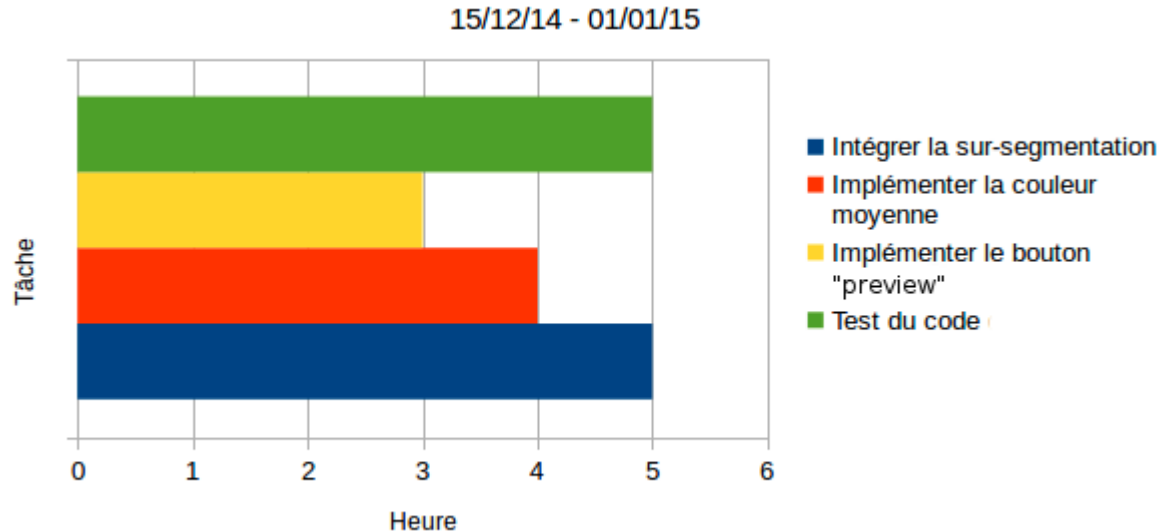
Étape 2 : Version 1 + site web



- ~21 heures

IV.1. Planning du projet

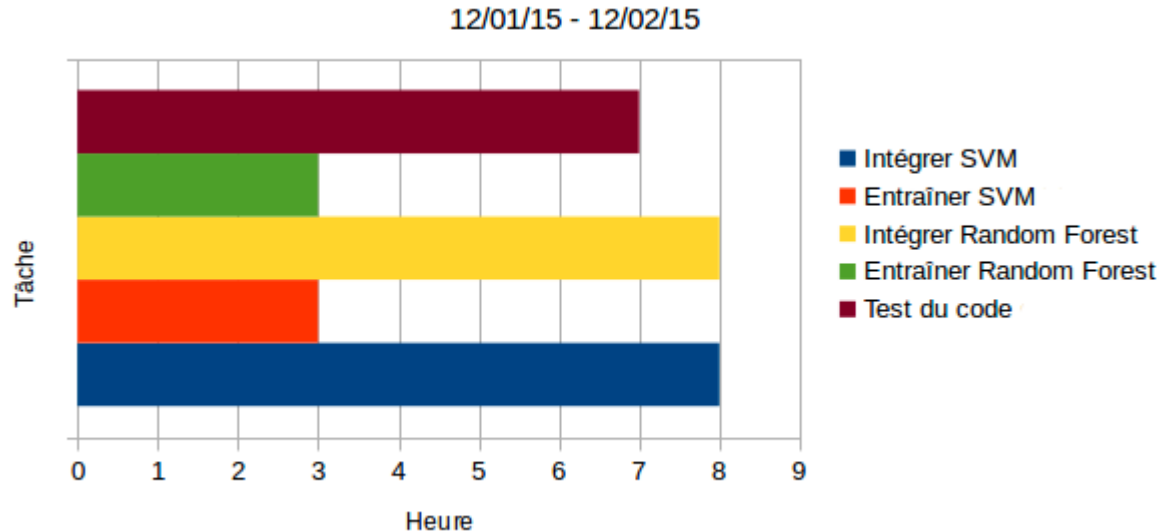
Étape 3 : Version 2



- ~17 heures

IV.1. Planning du projet

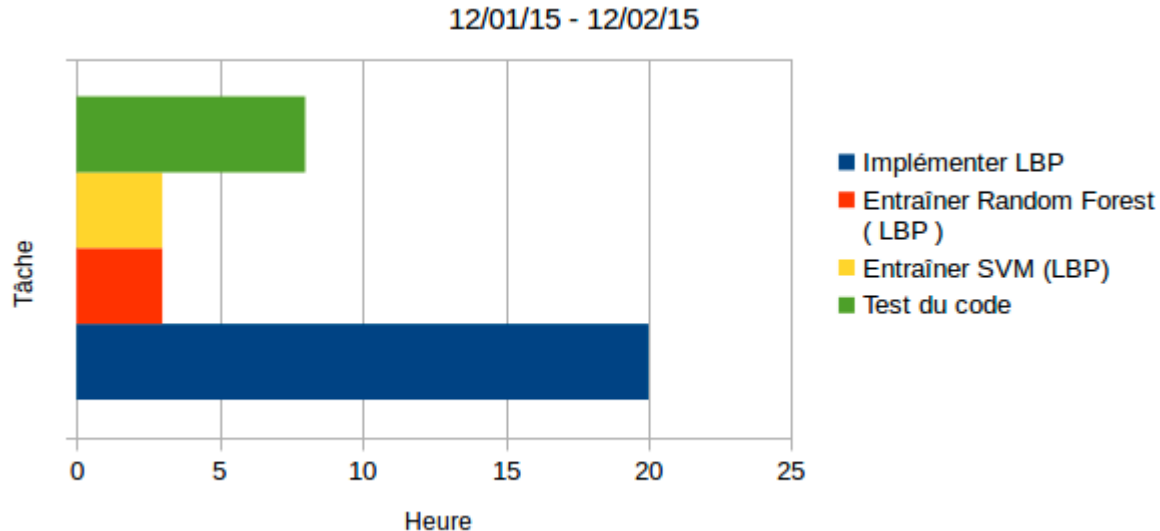
Étape 4 : Version 3 et version 4



- ~29 heures

IV.1. Planning du projet

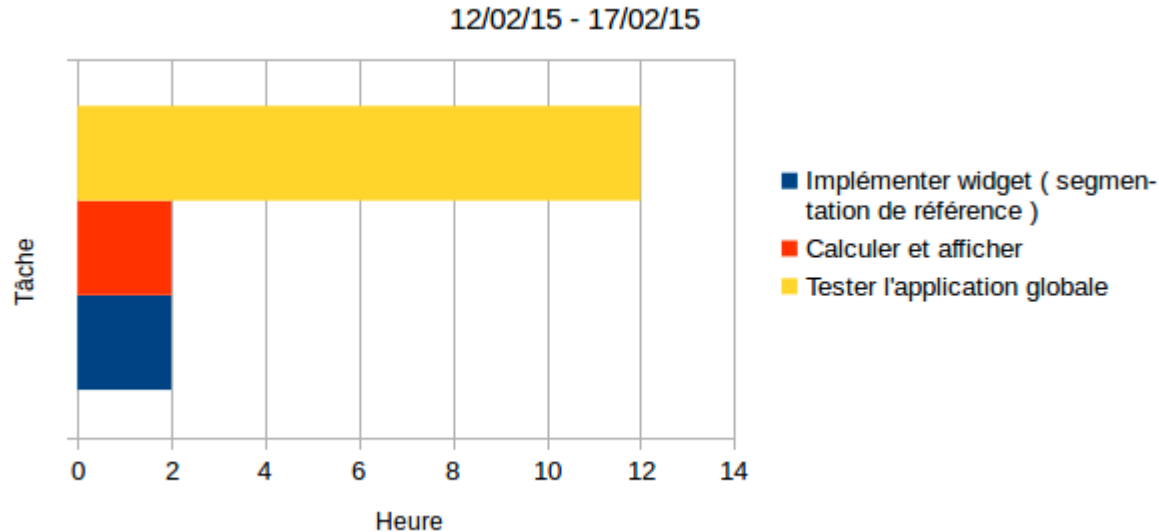
Étape 4 : Version 3 et version 4



- ~34 heures
- total = ~63 heures

IV.1. Planning du projet

Étape 5 : Version 5



- ~16 heures

IV.2. Tests

- Les superpixels (code déjà existant)
- Classifieur SVM avec les fichiers de LibSVM
- Méthode LBP implémentée avec celle d'OpenCV
- Général → modification du paramétrage des méthodes et analyse des rendus

IV.3. Analyse de risques : faible

- Une méthode de sur-segmentation ne fonctionne pas :
 - Impossible de sur-segmenter
 - Possibilité d'utiliser une des deux autres méthodes
- Ne pas avoir au moins une segmentation de référence à temps :
 - Impossible d'évaluer nos résultats
 - Nous ne pourrions que vérifier visuellement les résultats
- Manque de données pour entraîner les classifieurs :
 - Le programme renvoie des résultats mal classés
 - Essayer de se procurer d'autres images (demande au client)
- Une bibliothèque ne fonctionne pas comme on l'espérait :
 - Difficulté de s'en servir
 - Ré-implémenter les parties concernées

IV.3. Analyse de risques : **moyen**

- Le seul membre affecté à une partie du projet est absent :
 - Impossibilité d'effectuer la tâche
 - Toujours mettre au moins deux membres par tâche

IV.3. Analyse de risques : élevé

- La structure de données des descripteurs n'est pas compatible avec un classifieur :

Impossibilité de classifier

Modifier les structures

- La méthode de descripteurs par texture LBP n'est pas implémentée à temps :

Impossibilité d'utiliser cette méthode

Bibliothèque openCV contient la méthode LBP

- Mauvaise estimation des durées :

Manque de temps pour implémenter toutes les fonctionnalités

Développer les fonctionnalités les plus cruciales d'abord

IV.3. Analyse de risques : élevé

Les fonctionnalités les plus cruciales :

- une des méthodes de sur-segmentation fonctionnelle
- la méthode d'extraction de descripteurs fonctionnelle
- une des méthodes de classification fonctionnelle
- la phase d'apprentissage fonctionnelle
- une des méthodes d'évaluation fonctionnelle

