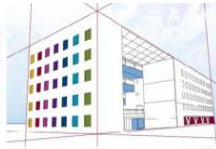




UNIVERSITÉ
TOULOUSE III
PAUL SABATIER



Université
de Toulouse



IRIT

CNRS - INPT - UPS - UT1 - UTM



Institut de Recherche en Informatique de Toulouse



EcoLab



Localisation de poisson dans une image : Conception

Clients : Alain Couzil, Bérengère Mathieu

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

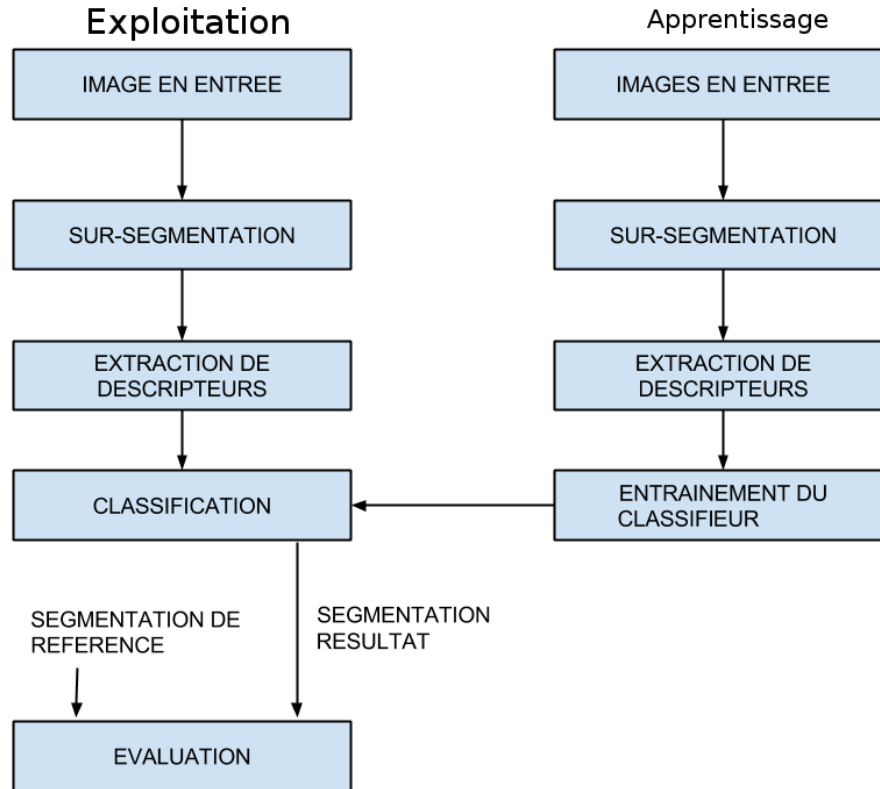
Plan de la présentation

- I. Introduction
- II. Structures de données et décomposition en classes
- III. Détails des fonctions système
- IV. Tests unitaires

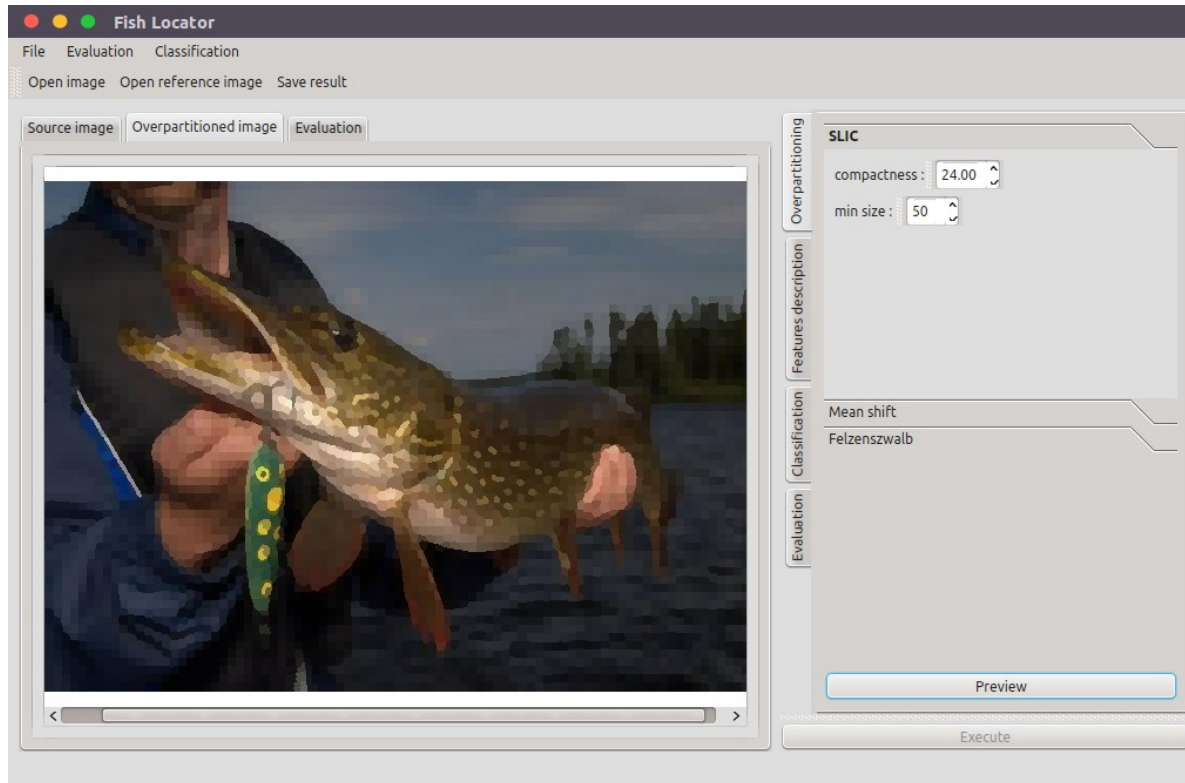
I. Introduction

- 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



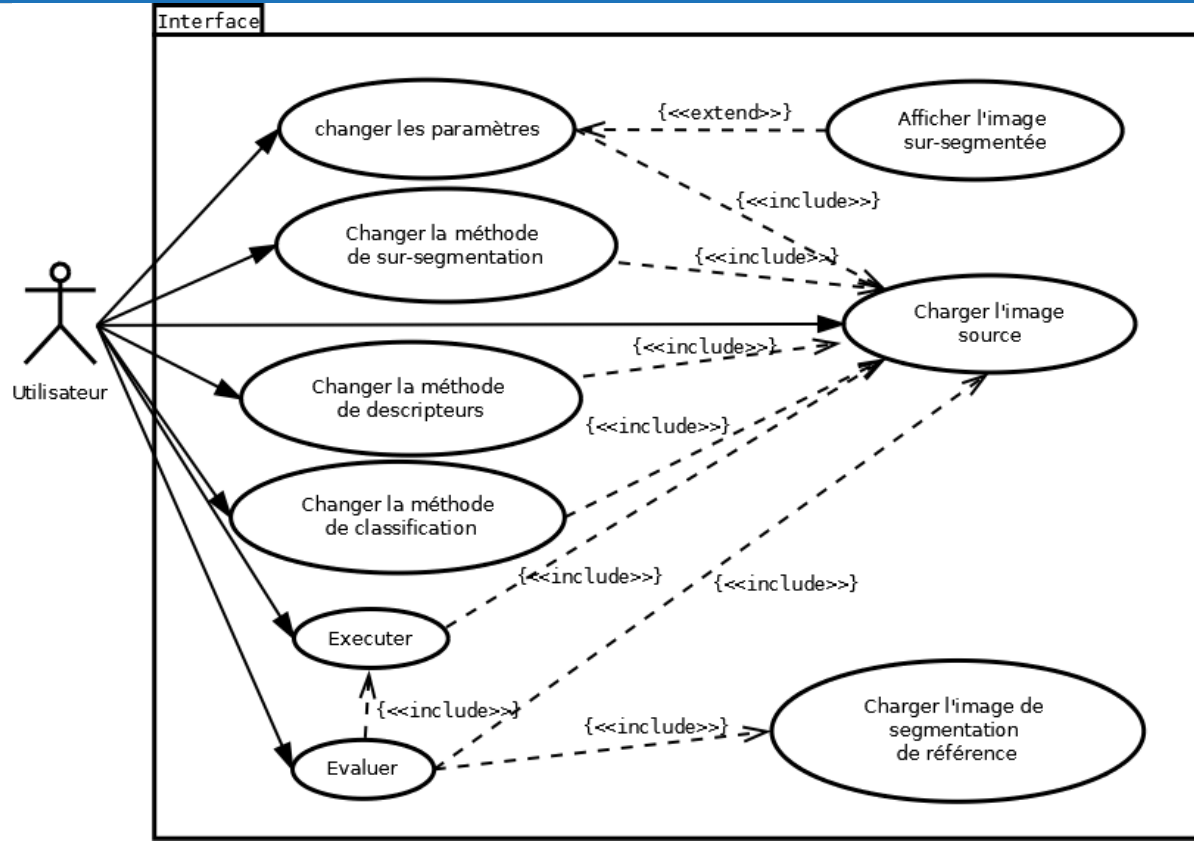
II. Structure des données

II.1. Diagramme cas d'utilisation

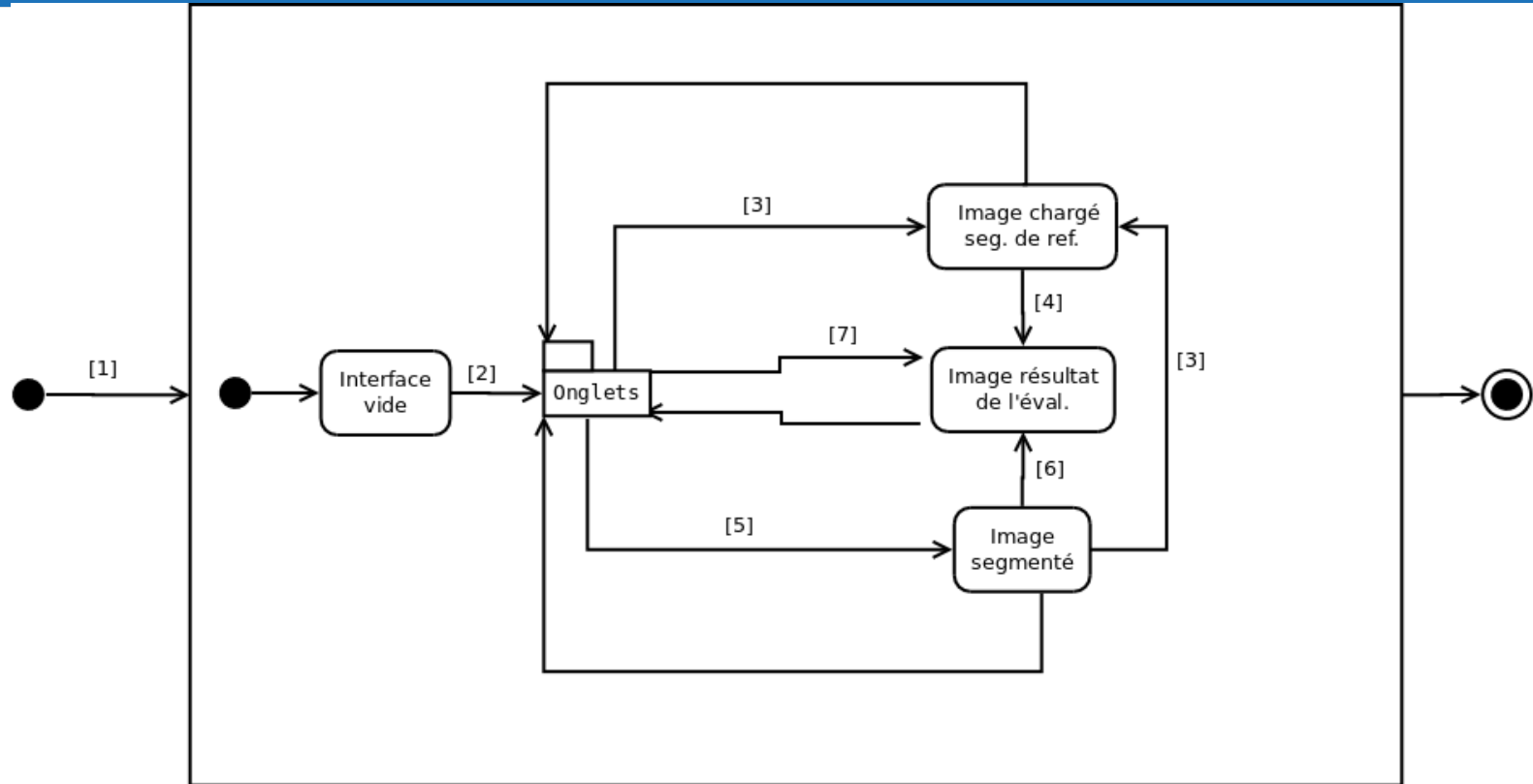
II.2. Diagramme d'états

II.3. Diagramme de classes

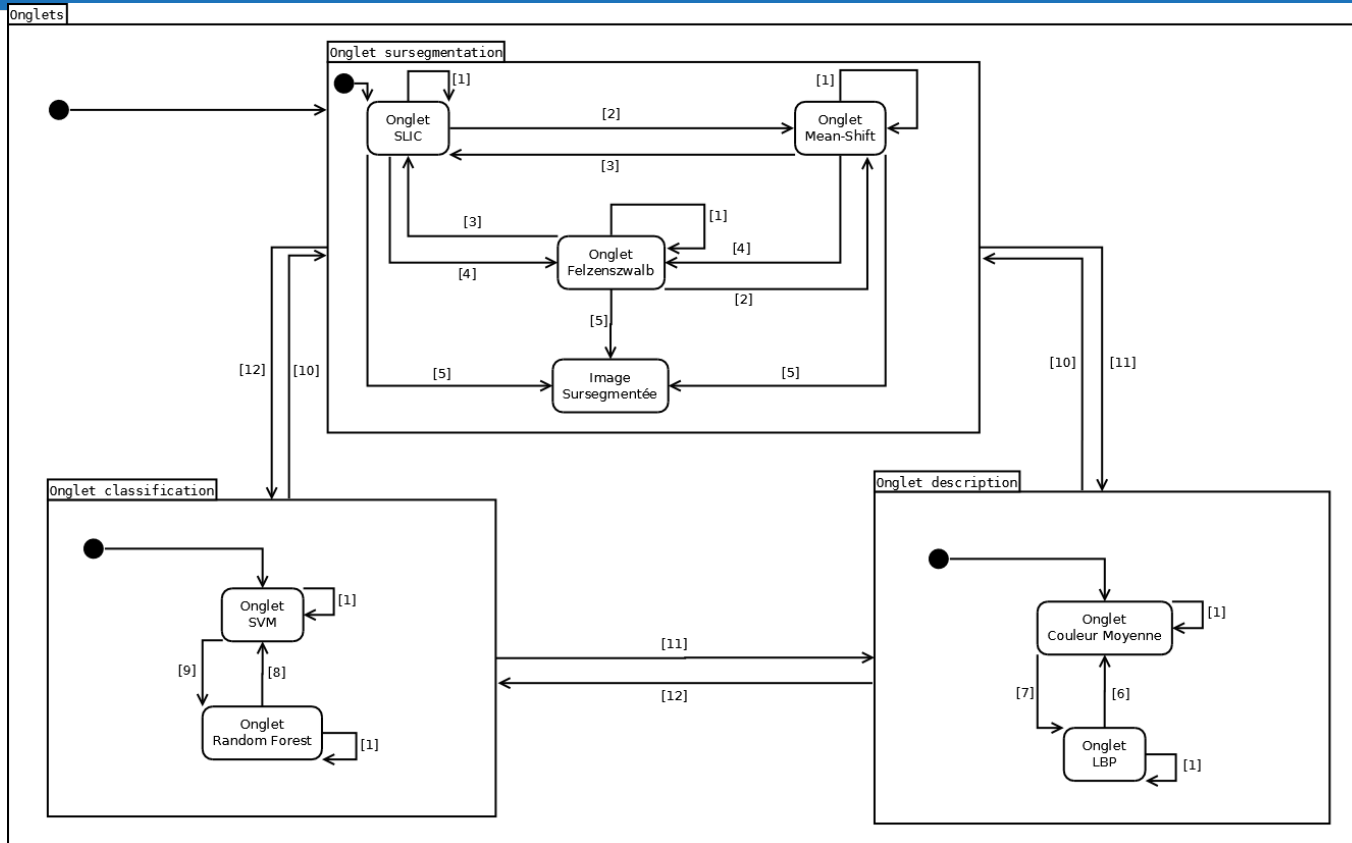
II.1. Diagramme cas d'utilisation



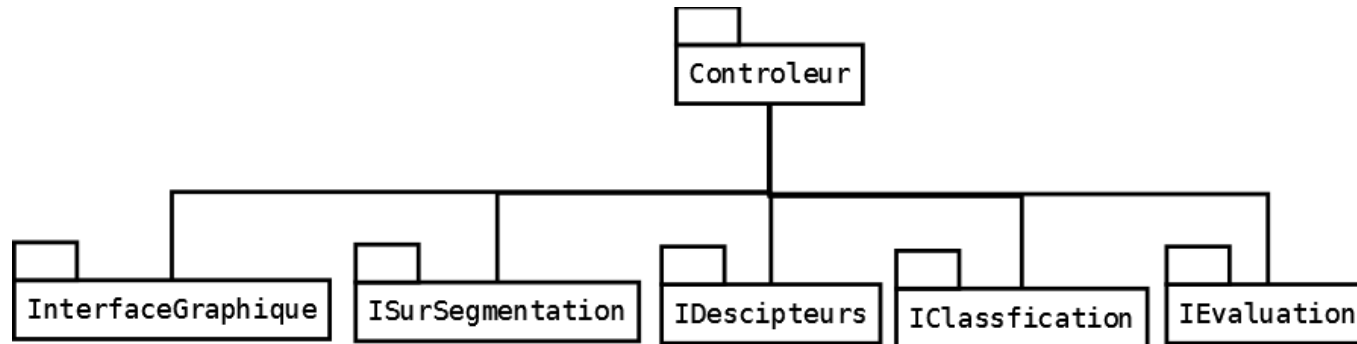
II.2. Diagramme d'états



II.2. Diagramme d'états



II.3. Diagramme de classes



III. Détails des fonctions système

- **Le module Contrôleur**
- **Le module Interface Graphique**
- **Le module Sur-segmentation**
- **Le module Description**
- **Le module Classification**
- **Le module Évaluation**

III.1. Module Controleur

Ce module gère toutes les interactions entre les différents modules de l'application.

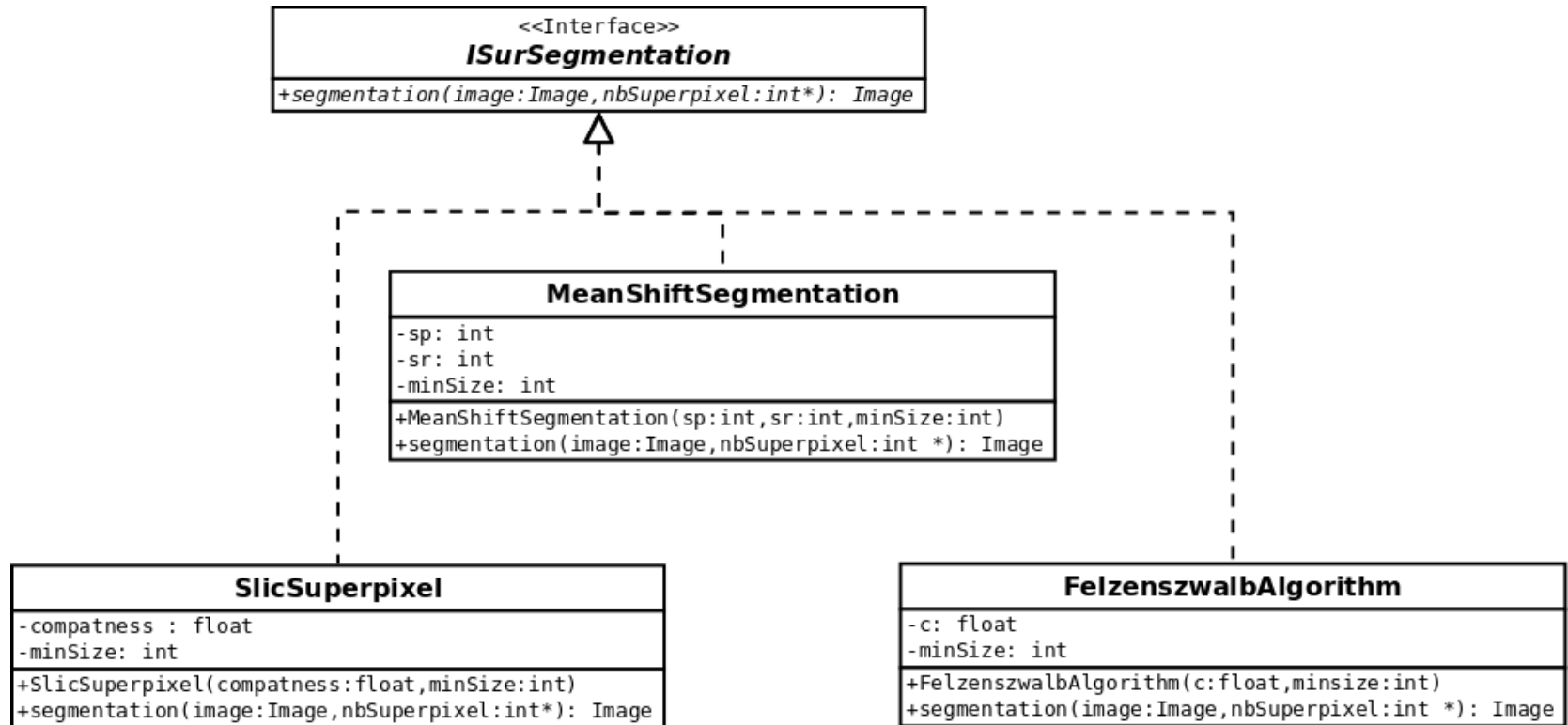
Controleur
<code>+sursegmentation: ISurSegmentation*</code>
<code>+classification: IClassification*</code>
<code>+description: IDescripteur*</code>
<code>+evaluation: IEvaluation*</code>
<code>+interface: InterfaceGraphique*</code>
<code>+imageSource: Image*</code>
<code>+oversegmentedImage: Image*</code>
<code>+segmentedImage: Image*</code>
<code>+refImage: Image*</code>
<code>+init(): bool</code>
<code>+setImage(srcImage:Image*): void</code>
<code>+performSurSegmentation(): void</code>
<code>+performClassification(): void</code>
<code>+performEvaluation(): void</code>
<code>+performDescription(): void</code>

III.2. Module Interface graphique

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

InterfaceGraphique
<pre>+showImage(image:Image): void +showSuperPixelImage(image:Image): void +showRes(image:Image): void +showSegRef(image:Image): void +showEval(image:Image): void +showLayer(layerIndex:int,image:Image): void +hideLayer(layerIndex:int): void +getSuperPixelParams(): vector<float> +getDescriptionParams(): vector<float> +getClassificationParams(): vector<float></pre>

III.3. Module Sur-segmentation



III.3. 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.



III.3. Module Sur-segmentation

SlicSuperpixel
-compatness : float -minSize: int
+SlicSuperpixel(compatness:float,minSize:int) +segmentation(image:Image,nbSuperpixel:int*): Image

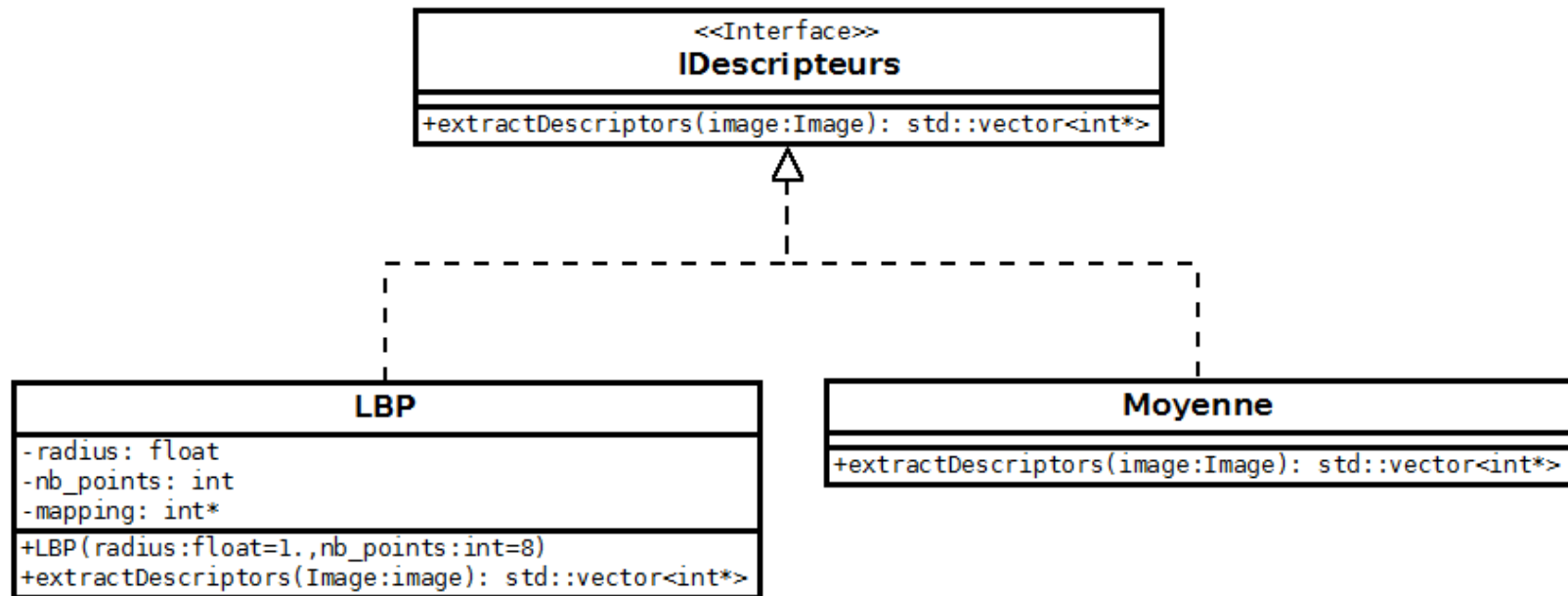
III.3. Module Sur-segmentation

MeanShiftSegmentation
-sp: int -sr: int -minSize: int
+MeanShiftSegmentation(sp:int, sr:int, minSize:int) +segmentation(image:Image, nbSuperpixel:int *): Image

III.3. Module Sur-segmentation

FelzenszwalbAlgorithm
-c: float -minSize: int
+FelzenszwalbAlgorithm(c:float,minsize:int) +segmentation(image:Image,nbSuperpixel:int *): Image

III.4. Module Description



III.4. Module Description

Le module Description permet d'attribuer à chaque superpixel un vecteur de valeurs calculé suivant un des algorithmes choisis par l'utilisateur, ainsi que ses paramètres associés.



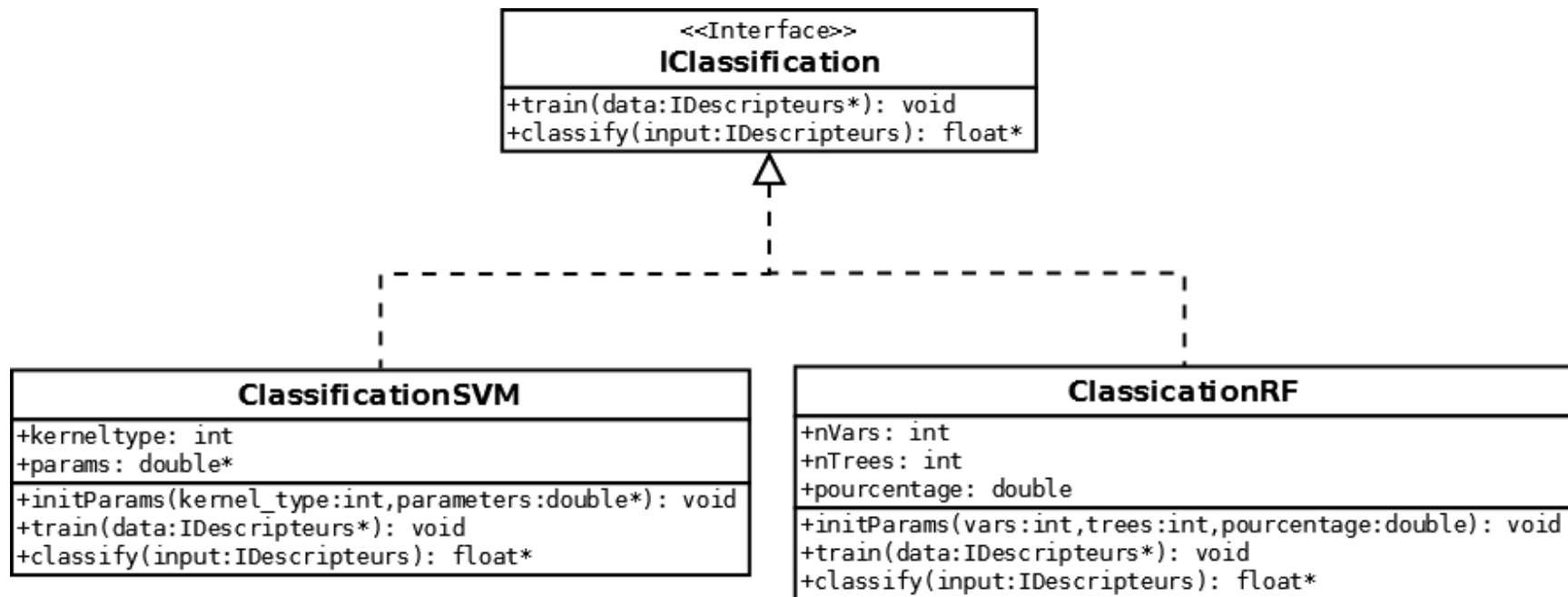
III.4. Module Description

LBP
-radius: float -nb_points: int -mapping: int*
+LBP(radius:float=1.,nb_points:int=8) +extractDescriptors(Image:image): std::vector<int*>

III.4. Module Description

Moyenne
<code>+extractDescriptors(image:Image): std::vector<int*></code>

III.5. Module Classification



III.5. Module Classification

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



III.5. Module Classification

kernel_type : cet attribut de type "int" représente le type du noyau utilisé par l'algorithme SVM

ClassificationSVM
+kerneltype: int +params: double*
+initParams(kernel_type:int,parameters:double*): void +train(data:IDescripteurs*): void +classify(input:IDescripteurs): float*

— Poly : params sera composé de : c0 (le coefficient initial), degree (le degré du polynôme) et gamma, qui est un multiplicateur.

— RBF : params sera uniquement composé du paramètre gamma

III.5. Module Classification

ClassificationRF
+nVars: int +nTrees: int +pourcentage: double
+initParams(vars:int,trees:int,pourcentage:double): void +train(data:IDescripteurs*): void +classify(input:IDescripteurs): float*

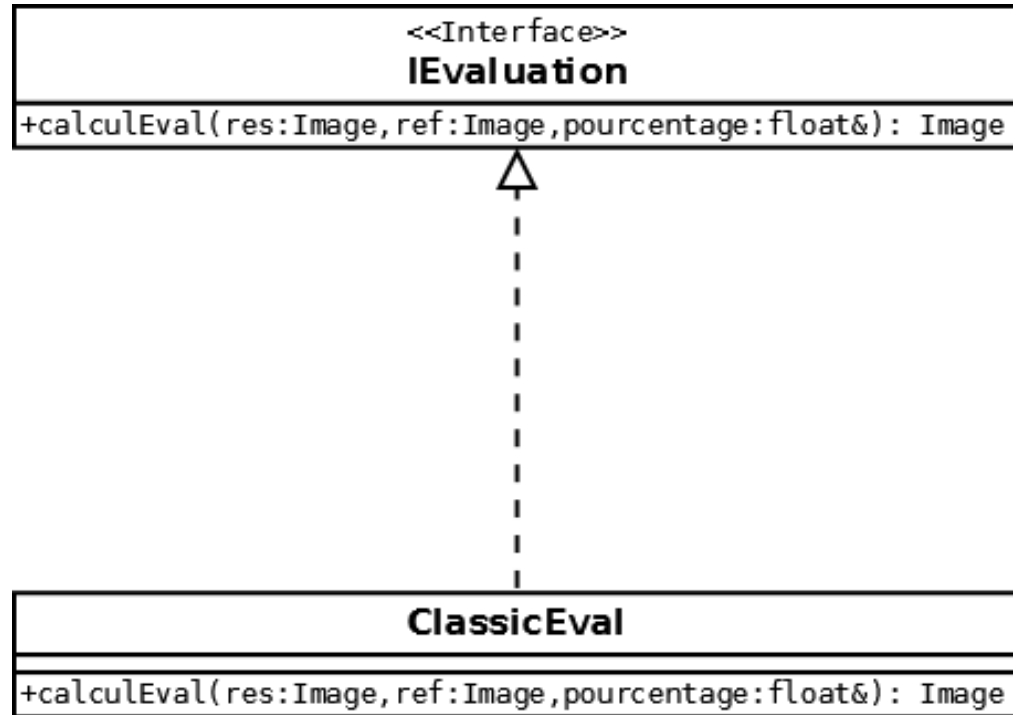
III.6. Module Evaluation

Interface: IEvaluation

- calculEval(res: Image,
 ref: Image,
 pourcentage: foat&) : Image

La classe: ClassicEval

- Implémente l'interface
- Compare chaque pixel



IV. Tests unitaire

- Module Contrôleur :
 - Tester préalablement tout les autres et vérifier si le module permet l'interaction
- Module Interface graphique:
 - Tester l'édition des différents paramètres.
 - Tester le chargement et l'affichage des images.
 - Tester l'affichage avec le système de couches de transparence.
 - Tester l'appel des fonctions concernées lors des clics sur les boutons.
 - Tester les différents boutons preview, execute, evaluate.

IV. Tests unitaire

- Module Sur-segmentation:
 - Comparaison visuelle car le code a déjà été testé.
- Module Description:
 - La classe LBP:
 - Comparer les résultats avec la méthode existante dans openCV.
 - La classe Moyenne:
 - Comparaison visuelle car le code a déjà été testé.

IV. Tests unitaire

- Module Classification:
 - Les classes classificationSVM et classificationRF
 - Tester grâce à des fichiers tests fournis avec la bibliothèque>
- Module Évaluation:
 - Mettre la même image en référence et en résultat (~ 100%)
 - Mettre deux images complètement différentes (~ 0% et une image entièrement en rouge)

