

University of Liège  
Faculty of Engineering



Master Thesis

---

**A workflow for large-scale computer-aided cytology  
and its applications.**

---

*Author* : Romain Mormont

*Supervisor* : Prof. Pierre Geurts

Master thesis submitted for the degree of  
MSc in Computer Science and Engineering

Academic year 2015-2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Object detection in large images</b>	<b>2</b>
2.1	General problem . . . . .	2
2.1.1	Formulation . . . . .	2
2.1.2	Implementation issues . . . . .	2
2.1.3	Related works . . . . .	2
2.2	Cytology . . . . .	2
2.2.1	An object detection problem ? . . . . .	2
2.2.2	The thyroid case . . . . .	2
<b>3</b>	<b>A generic workflow : Segment Locate Dispatch Classify</b>	<b>3</b>
3.1	Principle . . . . .	3
3.1.1	History and first developments . . . . .	3
3.1.2	Scope . . . . .	3
3.1.3	Algorithm . . . . .	3
3.1.4	Framework . . . . .	4
3.2	Implementation . . . . .	4
3.2.1	Technologies . . . . .	4
3.2.2	Software architecture . . . . .	4
3.2.3	How to use the framework . . . . .	4
<b>4</b>	<b>SLDC at work : the thyroid case</b>	<b>5</b>
4.1	Cytomine . . . . .	5
4.2	Implementation issues . . . . .	5
4.3	Implementation . . . . .	5
4.4	Performance analysis . . . . .	5
4.4.1	Detection . . . . .	5
4.4.2	Execution time . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>6</b>
	<b>List of Tables</b>	<b>8</b>
	<b>List of Figures</b>	<b>9</b>
	<b>Bibliography</b>	<b>10</b>

# Summary

# Acknowledgement

## Chapter 1

# Introduction

## Chapter 2

# Object detection in large images

### 2.1 General problem

#### 2.1.1 Formulation

The problem consists in finding objects of interest in images

#### 2.1.2 Implementation issues

What issues an implementor could face when trying to implement object detection in large images

#### 2.1.3 Related works

What solutions are usually presented in the litterature to solve those problems (shallow overview as this is a wide topic)

### 2.2 Cytology

#### 2.2.1 An object detection problem ?

Why it is an instance of the "object detection in large images" problem

#### 2.2.2 The thyroid case

Explanation of the Thyroid case

## Chapter 3

# A generic workflow : Segment Locate Dispatch Classify

### 3.1 Principle

#### 3.1.1 History and first developments

The Segment-Locate-Dispatch-Classify (SLDC) workflow was first imagined by Jean-Michel Begon as a generalization of the work on thyroid nodule malignancy detection made in [Deb13]. In the context of this master thesis, the author had implemented a processing workflow for detecting cells with inclusion and proliferative patterns (see ?? (thyroid)) in digitized thyroid punctions slides. The cells and patterns were detected by segmenting the images and then classified using machine learning techniques. As explained in the Section ?? (thyroid), the patterns could themselves contain cells with inclusion, so the author implemented a second processing workflow to detect those cells in the patterns. The idea behind the second workflow was similar to the first, that is segmenting the patterns and then classifying the detected object.

From those workflows, a common pattern emerged : performing detection using a segmentation algorithm and then classifying the detected objects using machine learning techniques. This pattern was at the root of the SLDC workflow idea.

#### 3.1.2 Scope

The idea was to implement a ?? plug-and-play meta-algorithm ?? based on this that a developer could customize to solve any object detection and classification problem.

#### 3.1.3 Algorithm

The workflow is an object detection and classification algorithm. Therefore, it takes as input an image and outputs a set of objects of interest found in this image. Especially, the algorithm is expected to return information about the shape and location of those objects. In addition to those information, a classification label is computed using a classifier and associated with every object. Due to the possible variability of the elements to detect in the images, the workflow can use different classifiers to predict the target label. Therefore, before being classified, an object is first dispatched to one of these classifiers.

## Formalization

As suggested by its name, the workflow is composed of four steps.

**Segment** The first step consists in finding objects of interest in the input image, that is, finding grouping of pixels composing those objects. This step can be formulated as a segmentation problem where each pixel is associated a binary value indicating whether it belongs or not to an object of interest. Given the input image  $\mathcal{I}$ , this step therefore produces a binary image  $\mathcal{B}$  of which the pixel  $b_{ij}$  is set to 1 if the pixel  $p_{ij}$  of  $\mathcal{I}$  is located in an object of interest. Otherwise,  $b_{ij}$  is set to 0. Formally, we have a segmentation operator  $\mathcal{S}(\cdot)$  :

$$\mathcal{S}(\cdot) : \mathcal{I} \rightarrow \mathcal{B}$$

**Locate** The second step consists in extracting the information about the shape and location of the objects of interest from the binary image  $\mathcal{B}$  and to convert this information into a convenient format such as a polygon (see Section 3.1.3 for more details about this representation). Given the binary image  $\mathcal{B}$  and given that it contains  $K$  distinct regions of interest, this step produces  $K$  polygons where the polygon  $P_i$  represents the geometric contour of the  $i^{th}$  object of interest. Formally, we have a location operator  $\mathcal{L}(\cdot)$  :

$$\mathcal{L}(\cdot) : \mathcal{B} \rightarrow \{P_i | i \in \{1, \dots, K\}\}$$

**Dispatch** At this step, the workflow must dispatch each polygon to its corresponding classifier. To achieve this goal, it uses a the

## Classify

### On encoding location and format using a polygon

#### Chaining workflows

##### 3.1.4 Framework

Describe what we expect from a framework implementing the workflow (parallism, easy to use,...)

## 3.2 Implementation

### 3.2.1 Technologies

Describe and justify the choice of Python and of the various dependencies

### 3.2.2 Software architecture

Detail of the software architecture

### 3.2.3 How to use the framework

A toy example : finding disks in an image with grey background and guessing whether they're black or white



## Chapter 4

# SLDC at work : the thyroid case

### 4.1 Cytomine

Presentation of cytomine

### 4.2 Implementation issues

Presentation of implementation issues related to the thyroid case (image size, over HTTP, image quality, human annotation vs computer annotation, presence of inclusions in patterns, dispatching ...)

### 4.3 Implementation

Actual implementation of the processing using the workflow

### 4.4 Performance analysis

#### 4.4.1 Detection

#### 4.4.2 Execution time

**Chapter 5**

**Conclusion**

# Notations

## Image :

$\mathcal{I}$	An image
$w$	The width of an image
$h$	The height of an image
$c$	The number of channels of an image
$\mathcal{I}_{hw}$	An two dimensional image of width $w$ and height $h$
$p_{ij}$	A pixel at row $i$ and column $j$ of a two dimensional image
$\mathcal{B}$	A binary image
$b_{ij} \in \{0, 1\}$	A pixel of a binary image
$P$	A polygon

## Machine learning :

$T(\cdot)$	A classifier
$C$	A classification label

# List of Tables

# List of Figures

# Bibliography

- [Deb13] Antoine Deblire. “Segmentation et classification automatiques de cytoponctions de la thyroïde.” fr. MA thesis. Université de Liège, Liège, Belgique, 2013, p. 74.