

# SLDC: an open-source workflow for object detection in multi-gigapixel images

Romain Mormont, Jean-Michel Begon, Renaud Hoyoux, Raphaël Marée  
 r.mormont@student.ulg.ac.be, jm.begon@ulg.ac.be, renaud.hoyoux@ulg.ac.be, raphael.maree@ulg.ac.be  
 Systems and modeling, Department of EE & CS, University of Liège, Belgium

## Object detection and classification

- In some applications, **multi-gigapixel images** must be analysed to gather information and/or take decisions.
- The analysis usually consists in **finding some objects** in the images and to associate them with a field-related label.
- In cytology, cytopathologists analyse microscope slides in order to find malignant cells or artefacts to **diagnose diseases like cancer** (see Figure below).

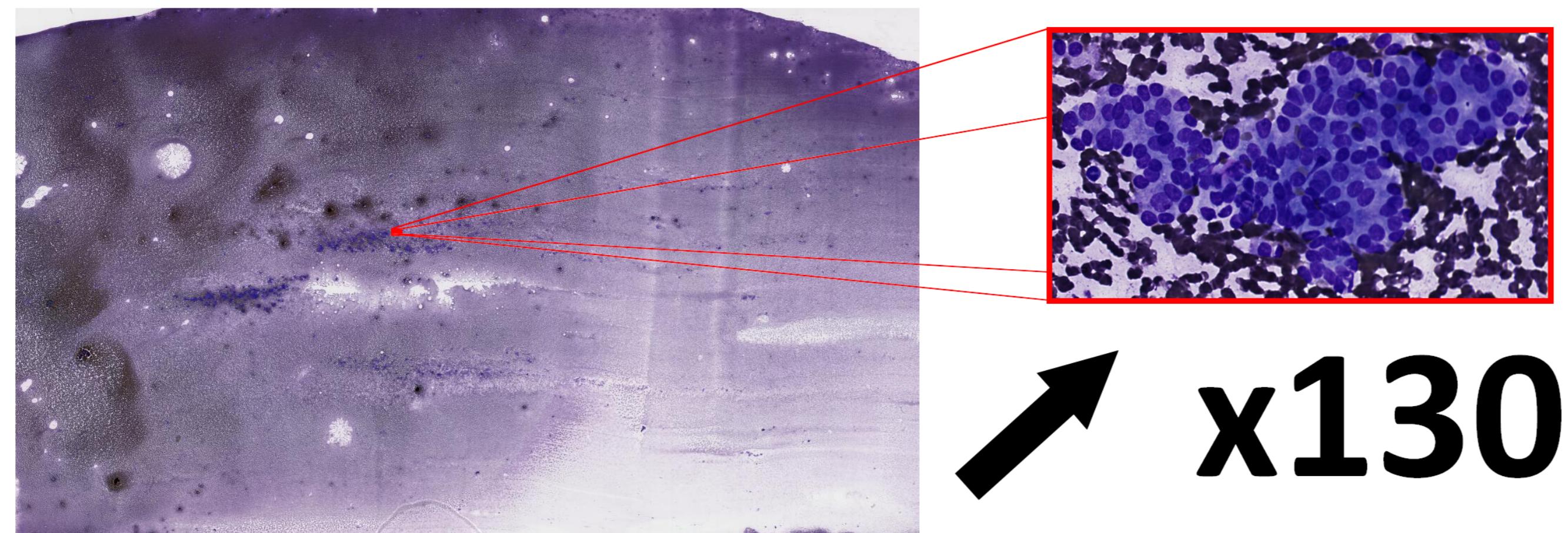


Figure: Left: a digitized high-resolution image of a microscope slide smeared with thyroid cell samples (15 gigapixels). Right: an object of interest: a proliferative architectural pattern.

- Thanks to modern technologies, microscope slides are often digitized to be further analysed on computers. But **slides are huge**:  $\sim 100K \times 100K$  pixels typically
- Due to the lack of proper tool, those slides are **usually analysed manually**!
- Machine learning and image processing** could provide a great alternative to a pure-human approach.

⇒ Those problems can be expressed as problems of **object detection and classification**.

## SLDC framework

SLDC is an **open-source Python framework** created for accelerating development of large image analysis workflows.

### How ?

- It **encapsulates problem-independent logic** (parallelism, memory limitation due to large images handling,...)
- It provides a **concise way of declaring problem dependant components** (segmentation, object classification,...)

## Features

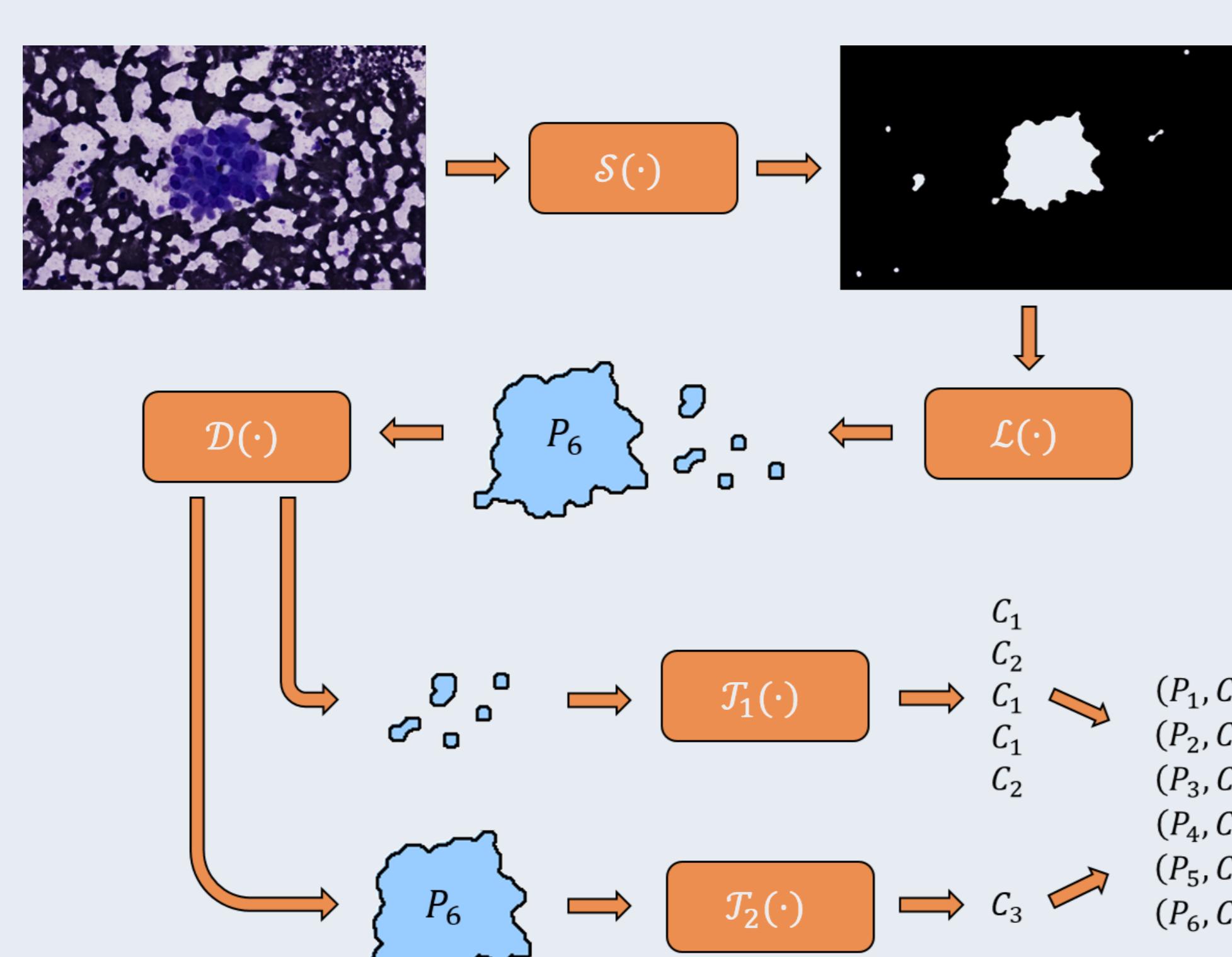
- Tile-based**: images are splitted into tiles which are loaded into memory one after another. A full image is never loaded into memory at once.
- Parallel**: the framework can take advantage of parallelism to reduce its execution time. Parallelism is available at several levels (tiles, objects, images,...).
- Talkative**: a customizable logging system provides a rich real-time feedback about the execution.
- Integrable**: thanks to Python, integration with other libraries such as scikit-learn (ML), open-cv (IP), PyCuda (GPU),... is effortless.
- Convenient**: builder components provide an easy way of constructing complex workflows.

## How SLDC works

Given an input image, the framework produces a set of polygons and classification labels representing the objects of interest. In order to do so, it executes the following steps:

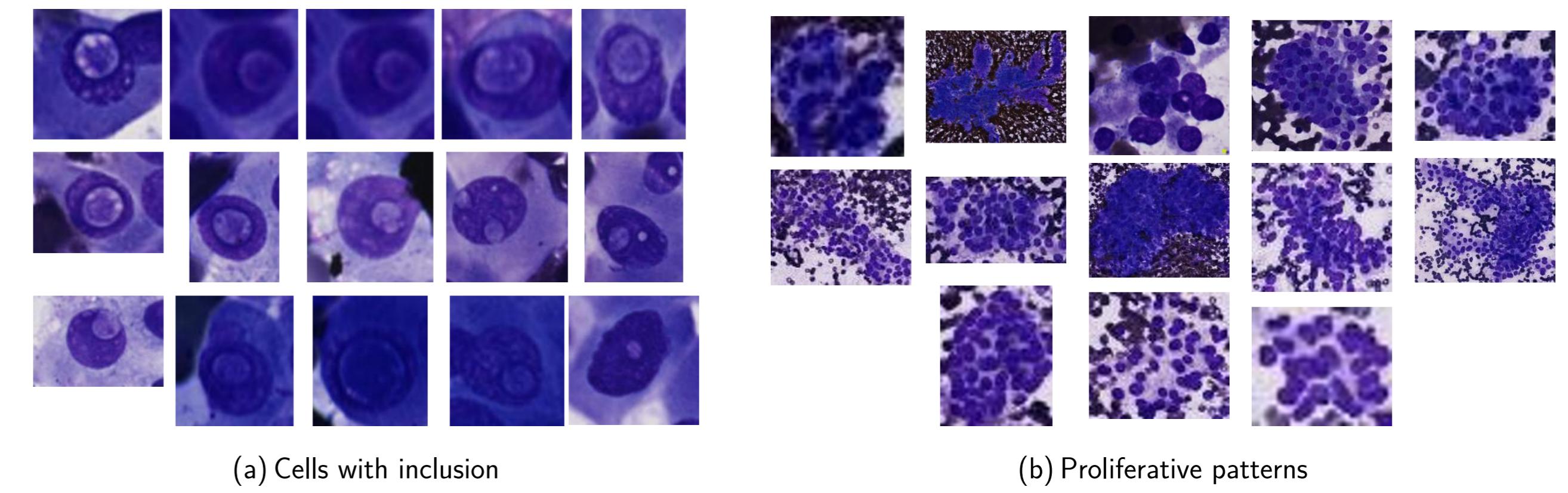
- Segment  $S$** : a segmentation procedure is applied on the input image (top-left) and produces a binary mask (top-right) locating the objects of interest in the image.
- Locate  $L$** : polygons representing the objects found in the image are extracted from the binary mask.
- Dispatch  $D$** : polygons are dispatched to their most appropriate classifier using some dispatching rules  $r$ .
- Classify  $T_i$** : a classifier produces classification labels for the polygons/objects it is passed.

All problem-independent concerns being encapsulated by the framework, developers only have to define the segmentation  $S$ , the dispatching rules  $r$  and the classifiers  $T_i$ .



## SLDC at work: thyroid nodule malignancy diagnosis

To diagnose thyroid cancer, cytopathologists screen microscope slides smeared with thyroid nodule cell samples. The illness is diagnosed when two types of objects are found on those slides:



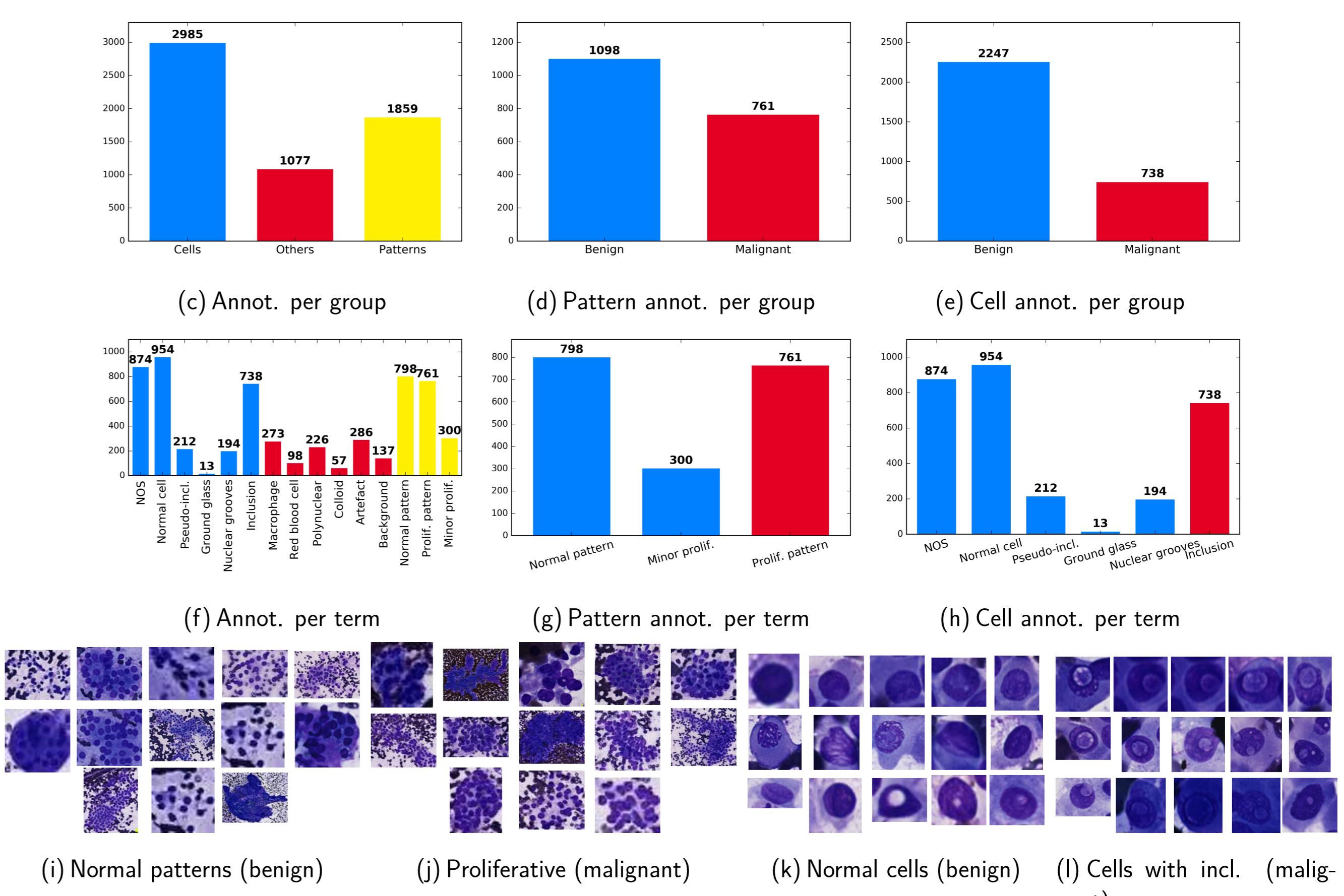
(a) Cells with inclusion

(b) Proliferative patterns

### Data

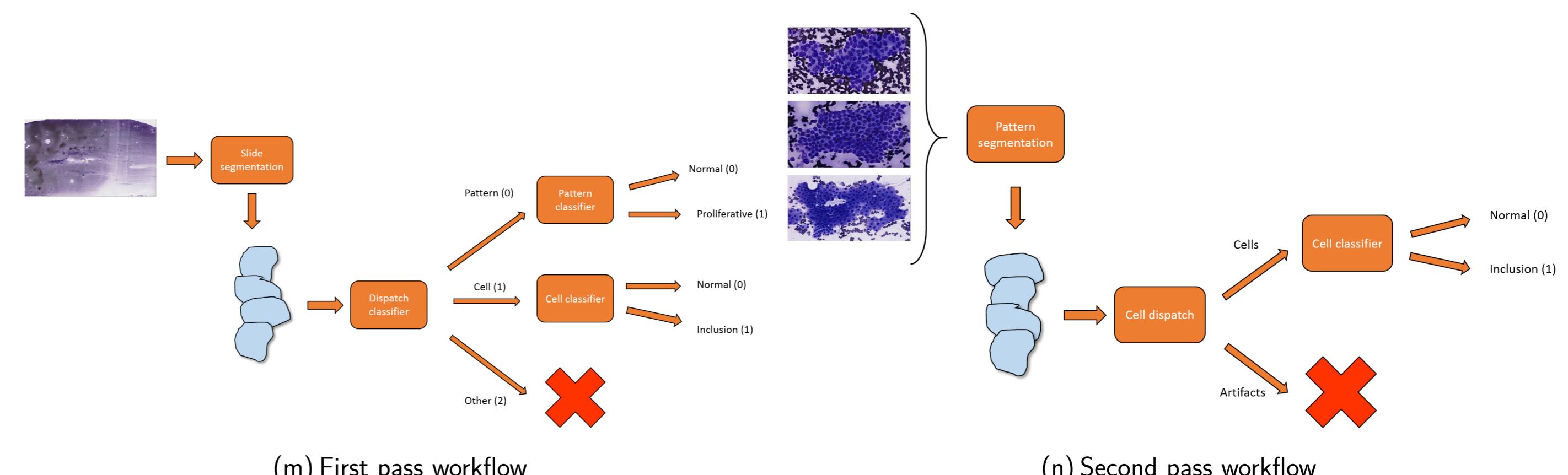
The dataset is stored on the Cytomine [Mar+16] web platform. It consists in:

- 84 images** with size ranging from 4 to 18 gigapixels
- 68 annotated images**
- 5921 labelled annotations** made by cytopathologists from ULB (Team of Pr. Isabelle Salmon, Dept. of Pathology)



### Workflow

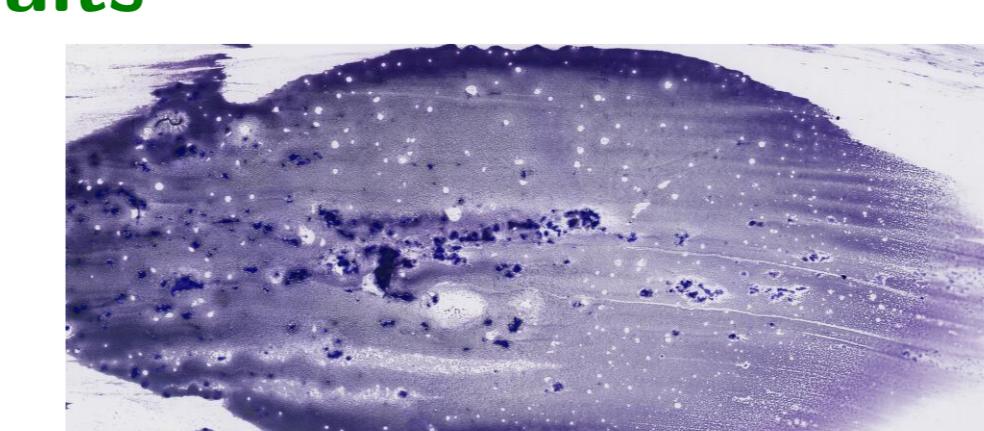
To actually detect those objects, two passes are performed on the images. The first consists in extracting standalone cells and architectural patterns. The second consists in extracting the cells contained in the detected patterns.



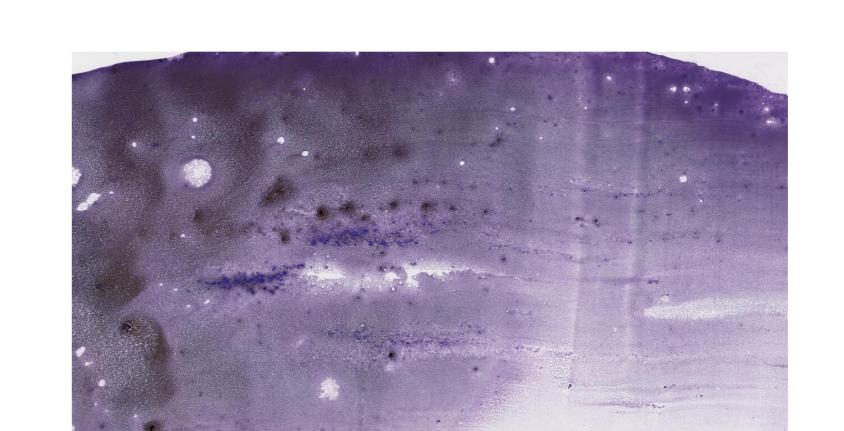
### Classification

Classification is performed based on the detected objects' crop images using **random subwindows** and **extremely randomized trees** [MGW16].

### Results



Size: 163840 × 95744  
 Time (1st pass): 12 min 24 sec  
 Time (2nd pass): 7 min 10 sec  
 Objects found: 76133  
 Cells found: 69820  
 Patterns found: 6313  
 Jobs: 64  
 Max memory usage: 179.855 Go



Size: 131072 × 57856  
 Time (1st pass): 4 min 38 sec  
 Time (2nd pass): 2 min 04 sec  
 Objects found: 18882  
 Cells found: 17802  
 Patterns found: 1080  
 Jobs: 64  
 Max memory usage: 159.414 Go

## Conclusion

- The **SLDC framework is production ready**, feel free to use it: <https://github.com/waliens/sldc>
- The thyroid workflow can still be improved (classifiers and segmentations)

## Bibliography

- [Mar+16] Raphaël Marée et al. "Collaborative analysis of multi-gigapixel imaging data using Cytomine". In: *Bioinformatics* (2016), btw013.
- [MGW16] Raphaël Marée, Pierre Geurts, and Louis Wehenkel. "Towards generic image classification using tree-based learning: An extensive empirical study". In: *Pattern Recognition Letters* 74 (2016), pp. 17–23. ISSN: 0167-8655. DOI: <http://dx.doi.org/10.1016/j.patrec.2016.01.006>. URL: <http://www.sciencedirect.com/science/article/pii/S0167865516000179>.