Contributions to modern unsupervised learning: Case studies of multi-view clustering and unsupervised Deep Learning

Jérémie Sublime

HABILITATION À DIRIGER LES RECHERCHES
DE
SORBONNE UNIVERSITÉ

Spécialité **Informatique**

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Jérémie SUBLIME

ISEP, laboratoire LISITE, équipe DASSIP
LIPN - CNRS UMR 7030, équipe A3-ADA

Contributions to modern unsupervised learning: Case studies of multi-view clustering and unsupervised Deep Learning

Contributions à l'apprentissage non-supervisé moderne: Applications aux cas du clustering multi-vue et de l'apprentissage profond non-supervisé

Soutenue le 16 avril 2021 devant le jury composé de:

Pr. Germain Forestier	Université de Haute Alsace	Examinateur
Dr. Nistor Grozavu, HDR	Université Sorbonne Paris Nord	Examinateur
Pr. Christophe Marsala	Sorbonne Université	Examinateur (Président du jury)
Pr. Florence Rossant	ISEP	Examinatrice
Pr. Rosanna Verde	Università degli Studi della Campania	Rapportrice
Pr. Nicole Vincent	Université de Paris	Rapportrice
Pr. Cédric Wemmert	Université de Strasbourg	Rapporteur

"People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world."

Pedro Domingos

Acknowledgements

First, I would like to thank the members of my committee. They kindly accepted to be part of it and provided me with relevant feedbacks and comments. Many thanks also to Basarab, Ekaterina, Arnaud, Florence and Hélène who had the tricky task of helping me to proof-read this document. In particular, I am very grateful for the constant support and help of Basarab Matei during these past 7 years. He is a thoughtful colleague and a good friend whose help was greatly appreciated to better understands the wheels of University politics, as well as when he attempted to bring some of his mathematical order inside my chaotic world of computer scientist. It certainly made me a better Data Scientist. I also want to thank my colleagues and friends at ISEP, without whom this adventure in Higher Education would not have been the same.

This document is not a compilation of my own work. It is a synthesis of works and discussions done with many students, colleagues and fellow scientists in the last decade. Therefore, I would like to warmly thank Nistor Grozavu, Basarab Matei, Guénaël Cabanes and Pierre-Alexandre Murena: we have been working together since I still was a PhD student; Florence Rossant, Maria Trocan, Sylvain Lefebvre, Hélène Urien and Raja Chiky with whom I co-advised my first PhD students, but hopefully not the last ones; The students, interns and PhDs, I had the pleasure to work with, and without whom not much is possible in academia: Denis Maurel, Ekaterina Kalinicheva, Guillaume Dupont, Clément Royer and Nan Ding; Colleagues that even if I did not directly collaborate with all of them had an influence on my research: Patricia Conde-Cespédes, Nicoleta Rogovschi, Parisa Rastin, Juan Zamora, Dino Ienco, Patrick Wang, Ilaria Renna, Michel Pâques and Pierre Gançarski.

Last but not least, Tony for everything else.

Table of Contents

Preamble

Abstract

This document is the manuscript presented in order to obtain the *Habilitation à Diriger des Recherches* of Sorbonne University (France), prepared at ISEP Engineering School where I am currently an Associate Professor. My main professional activities of research but also teaching and administrative work, since after I defended my PhD in November 2016, are described in this document. Since research is a continuum, it may also contain elements and recalls from previous works done between 2013 and 2016.

In particular, my main axis of research is unsupervised learning, and in the first part of this manuscript I describe my contributions centered around two sub-axis: Unsupervised learning in multi-view environments, and deep learning applied to image processing (satellite and medical) in cases where no labeled data are available. These two sub-axis form the main chapters of this document and describe contributions both in terms of applications and theoretical findings. Issues such as the notion of confidence in unsupervised learning, weakly supervised learning with unreliable ground-truths, clustering stability, as well as the limitations and future evolutions of unsupervised learning are discussed in this manuscript.

Context

My interest for research in the field of machine learning started in 2012, when I was an exchange Master student in South Korea under the supervision of Professor Geun-Sik Jo. This experience was quickly followed by my enrollment as a PhD student under a short term civil servant contract at INRA. I prepared my PhD defense between AgroParisTech (Université Paris Saclay) and the University Paris 13 (now Sorbonne Paris Nord), in the context of the ANR Project COCLICO (ANR-12-MONU-0001). I spent these 3 years working on unsupervised learning method applied to remote sensing images with the goal of combining the results of several clustering algorithms to achieve better results. This duality between contributions related to clustering and contributions related to remote sensing or image analysis in a broad sense has continued after my PhD defense and explains why I have these two distinct research axis with unsupervised learning as a common point between the two.

I was then recruited as an Associate Professor at ISEP in 2016. At ISEP, I pursued my research on unsupervised learning: I continued my PhD work on multi-view clustering with a more theoretical orientation. And I also started to further develop my second axis on image analysis using unsupervised method, which quickly led me to analyzing the potential of deep learning methods in unsupervised contexts. As you can probably guess, the second axis being more of a hot topic it was a lot easier to attract money and students to work with me. Furthermore, networking from my PhD years was useful to get contact in the field of Remote Sensing which gave me a first application field. And from my colleagues at ISEP I got a second application field in medical images, and in particular a collaboration with the Clinical Investigation Center (CIC) at Paris 15-20 Hospital. The strong need for unsupervised methods due to the scarcity of annotated data for both applications, and the relative quietness of the unsupervised deep learning community probably helped too, and enabled my students and I too make original proposal in this domain.

In the mean time, I also kept a full membership in team A3-ADA at the *Laboratoire d'Informatique*

de Paris Nord (LIPN), the team in which I prepared my PhD at Paris 13 University. This ongoing collaboration kept me active in the field of multi-view clustering.

Research at ISEP

ISEP is a private Engineering School with two campuses located in Paris and Issy-Les-Moulineaux France. Being a private engineering schools, it differs from French Universities in two points:

- As an engineering school, it belongs to the category of *les grandes écoles*, a French specificity dating from Napoleon I. Schools belonging to this category are state recognized and deliver engineering degrees (a 5 years degree equivalent to a Master's degree but more focused on practical aspects that can quickly be used in a business context and less on theoretical and research aspects). Unlike Universities, engineering schools are not allowed to deliver Bachelors, Masters or PhD degrees. However in the case of the PhD degree, since many of these schools host a research laboratory, they usually train PhD students in their labs through joint PhD programs with traditional Universities. This is the case for ISEP which has a research lab associated with Sorbonne University for PhD programs.
- As a private institution, the professors are not state civil servants and the school is mostly financially autonomous for both teaching and research, so it receives fewer state funds compared to public universities that are fully sponsored by the state. However, unlike public universities, private institutions have no regulations on the students tuition fees. Finally, private institutions are usually illegible to a low number of state sponsored research grants which makes it more difficult for local academic to get funds through projects, in a context that is already very competitive.

L'Institut Supérieur d'Électronique de Paris (ISEP) was funded in 1955 as an associative structure (*Association loi 1905*). The engineering degrees delivered by ISEP are recognized by the state and the CTI (*Commission des Titre d'Ingénieurs*) since 1960. In 2015, ISEP was recognized as an EESPIG (*Établissement d'Enseignement Supérieur Privé d'Intérêt Général*), a state recognition for the school partition in higher education, and for its contributions to national and international research. It currently graduates around 350 engineers each years in a dozen of specialties linked with Information Technologies (IT).

ISEP opened its research lab in 2000, with its original research activity focused around microelectronics. The research lab has evolved a lot since, and is currently structured around 2 teams:

- ECoS (Electronics and Communications Systems): 3 Full Professors (HDR), 7 Associate Professors.
- DaSSIP (Data Science Signal and Image Processing): 3 Full Professors (HDR), 9 Associate Professors.

The DaSSIP Team to which I belong works on 3 mains axis: 1) massive and distributed data, 2) Human-Computer interactions, and 3) Image and signal processing. Within this context, my work on multi-view clustering was a fit for the first research axis on distributed data. The other part of my work on unsupervised deep learning was more related to the third axis on image processing: On

the one hand, I brought the remote sensing theme to the team and helped foster collaborations with other academic partners around this subject. On the other hand, ISEP and in particular Professor Florence Rossant had a long ongoing research collaboration with Paris 15-20 hospital which allowed me to work with Deep Learning methods in a medical image context as well.

Research at the LIPN in Team A3-ADA

The *"Laboratoire d'Informatique de Paris Nord"* (LIPN) is a computer science laboratory who was founded in 1986 under the joint control of University Paris 13 and the French CNRS (*UMR 7030*). It hosts 5 teams and around 170 members (around 80 Associate Professors and Full Professors, 8 engineers and administrative staff members, and about 50 PhD students).

The *A3* team (*Apprentissage Artificiel et Applications*) to which I belong focuses its activities around Artificial Intelligence and Machine Learning. I have been an associate member of the team between 2013 and 2017, before becoming a full member in 2018. Since the restructuring of A3 in 2020, I am a full member of research pole A3-ADA whose offices are located in the lab main building of Paris 13 University as well as in the Saint Denis annexe of *La Maison des Sciences Numériques* (LaMSN) which is conveniently located across the street from where I live.

The research activities of the ADA pole focus on unsupervised approaches for representations learning, multi-view and transfer learning. The team members address both fundamental research as well as more applied research, often supported by academic and industrial projects. Among other things, team develops collaborative unsupervised learning approaches that were the basis of most of my PhD thesis work and are now a follow up of the same work. Within this context, my work with the team is the main pillar of my research axis centered on multi-view clustering.

Structure of the document

This manuscript for my *Habilitation à Diriger des Recherches* is divided into two main parts.

The first part is my scientific synthesis which describes past, ongoing and future works. As you could probably see from this preamble, my work on unsupervised learning is divided into two axis that are quite distinct: Multi-view clustering, and Deep Learning for Unsupervised applications. They form two distinct chapters of this first part. Each of them starts with a general introduction of the domain studied and its main open issues. Then, different contributions are presented for the period 2016–2021. All contributions are discussed with respect to past and future works, as well as the connections between them. For contributions that are domain specific (medical imaging or remote sensing for instance), a brief introduction of the application is made before presenting the actual contributions. Eventually, the first part ends with an open chapter which includes some of my thoughts on the current state of my research field, and gives information about ongoing and forthcoming works as well as some avenues of research on the topics that have been addressed so far and that will probably require my attention in the next years.

The second part describes my resume with an emphasis on publications, teachings, supervised students, projects, and the scientific animation of the communities I am involved in.

Part I

Scientific synthesis

Chapter 1

Multi-view Clustering: Extending clustering techniques and clustering theory to multi-view environments

"If your research is about clustering, then you are an adventurer."

James M. Keller (WCCI 2018)

Contents

1.1 Chapter Introduction

1.1.1 Clustering

Clustering is a common machine learning task that belongs to the branch of unsupervised learning. As such, it was originally designed as an exploratory data mining task whose goal is to find data and objects that are similar, and to regroup them into groups called *clusters* [1][2]. On the other hand, it can also be used to detect outliers that don't fit into any cluster [3][4]. Due to its exploratory nature, clustering is often used as an alternative to supervised learning and classification for data sets or problems for which no data have been labeled to train and use the more common supervised models. This is done under the hypothesis that the data set to explore has underlying structures that are well-behaved (or ordered enough) to be detected by a clustering algorithm, thus leading to clusters that will hopefully match real classes of interest. In the best case scenario, a clustering algorithm is expected to find clusters that can directly be linked to real life classes of interest. If so, using a clustering algorithm would have spared data scientists a lot of time and money that should have been spent labeling (often manually) a large number of elements so that it could be fed to a classification algorithm. However, in practice, we see that in most cases the clusters found by clustering methods are rarely a great fit with the real classes of interest, and that classification algorithms will "outperform" clustering methods on real applications if provided with enough training data. This leads us to an interesting trade-off when tackling a new data science problem: spending a lot of time labeling data to train a great classification method, or going the fast road with an exploratory clustering algorithm that will most likely lead to lower quality results. It is worth mentioning that both the time spent labeling data and the probability of having a lower quality result will quickly increase as the data set is going to be more complex. Another important point is that an average clustering result can also be used as a basis to faster labeling data, and later turn to a classification algorithm.

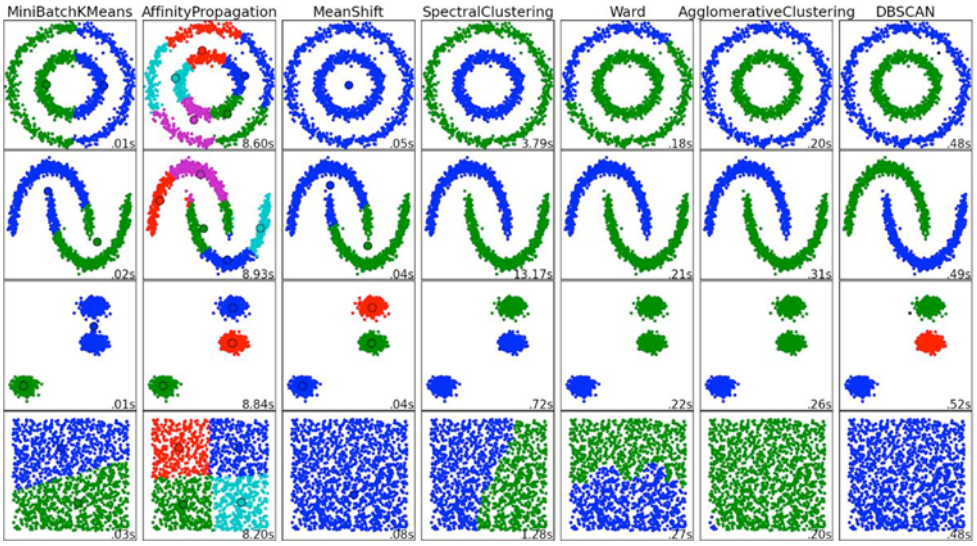


Figure 1.1: Example of several clustering methods applied to toy data sets.

Ultimately, one can say that the choice of a clustering algorithm should be made knowing

its internal model (i.e. the way it builds its clusters), because it is this model that will have to match the underlying data structures that are supposed to be "well behaved enough" when we use clustering. To this end, several models have been proposed in the literature for different types of underlying structures and different ways of building clusters [5]: Density-based clustering method [6][7][8][9][10][11] search for high density areas of data separated from each others by lower density areas, and do not assume any shape for the clusters; hierarchical clustering methods [12][13][14][15][16][17] will regroup similar data and similar group of data using a hierarchical structure similar to a dendrogram. For this type of clustering too, no assumption is made on the shape of the clusters; Prototype-based clustering methods will attempt at regrouping data around set of *prototypes* that will represent each cluster and may -or may not- have a specific data distribution expected around them (typically a Gaussian mixture). This last family of clustering algorithm contains some well known clustering method such as the K-Means algorithm [18][19] and its variations [20][21][22][23][24] or the Expectation-Maximization method [25] for the Gaussian Mixture Model (GMM); Spectral clustering methods [26][27] which turn the clustering problem into a graph partitioning problem based on the dataset similarity matrix.

An example of different clustering methods applied to several data sets is shown in Figure 1.1 to highlight some of the strenghts and weaknesses of the different families of methods.

Classical clustering algorithms that I introduced previously were for most of them nice algorithms and quite capable of tackling common data sets at the time they were designed. However, as I started my journey as a scientist in the world of unsupervised learning, I quickly found out that the clustering algorithms that my professors taught me -and the same ones I teach to my students nowadays- are in no way designed to handle modern data science problems, and in particular they struggle with the structure of modern data sets: Images, composite and hybrid data, distributed data, multi-view data, large datasets, stream datasets, all of these are extremely difficult to tackle for clustering algorithms. It also makes them interesting problems for datascientists. In this first chapter, I will mostly focus on my work on clustering applied to multi-view and distributed datasets. And in Chapter 2, I will focus more on image applications.

1.1.2 Multi-view clustering

We live in a world where data contains attributes of different nature, where information is distributed across several sites, and where multiple representations can be produced for the same data: Composite data with groups of features of different natures are ubiquitous, marketing and business host large client databases with information acquired from multiple sources, social networks are a large and ever evolving source of distributed data, many science fields such as medicine produce multi-view data from various acquisition devices, and even machine learning algorithms produce different possible representations of the same data in fields such as text mining and natural language processing [28][29].

How can clustering algorithms tackle these problems that they weren't designed for ? Well, these problems and this question gave birth to scientific communities working on multi-view clustering, collaborative clustering, distributed data clustering and unsupervised ensemble learning, all of which I am part of. Before starting with my contributions, I will make an attempt at explaining the common points, differences, and overlaps between these communities, as it is in my opinion quite confusing in the literature:

- Multi-view clustering [30][31] is concerned with any kind of clustering where the data are split into different views. It does not matter whether the views are physically stored in different places, and if the views are real or artificially created. In multi-view clustering, the goal can either be to build a consensus from all the view, or to produce clustering results specific to each views.
- Distributed data clustering [32] is a sub-case of multi-view clustering that deals with any clustering scenario where the data are physically stored in different sites. In many cases, clustering algorithms used for this kind of task will have to be distributed across the different sites.
- Collaborative clustering [33][34][35] is a framework in which clustering algorithms work together and exchange information with a goal of mutual improvement. In its *horizontal* form, it involves clustering algorithms working on different representations of the same data, and it is a sub-case of multi-view clustering with the particularity of never seeking a consensus solution but rather aiming for an improvement in all views. In its vertical form, it involves clustering algorithms working on different data samples having similar distributions and underlying structures. In both forms, these algorithms follow a 2-step process: 1) a first clustering is build by local algorithms. 2) These local results are then improved through collaboration. A better name for collaborative clustering could be *model collaboration* as one requirement for a framework to qualify as *collaborative* is that the collaboration process must involve effects at the level of the local models.
- Unsupervised ensemble learning, or cluster ensembles [36][37][38] is the unsupervised equivalent of ensemble methods from supervised learning [39]: It is concerned with either the selection of clustering methods, or the fusion of clustering results from a large pool, with the goal of achieving a single best quality result. This pool of multiple algorithms or results, may come from a multi-view clustering context [40], or may just be the unsupervised equivalent of boosting [41] methods where one would attempt to combine the results of several algorithms applied to the same data.

1.1.3 Chapter organization

The remainder of this chapter will be centered around multi-view clustering and in particular 3 key aspects that I have been working on after my PhD thesis:

1. The question of the confidence issue in unsupervised environments. It is indeed difficult to know which views are reliable, which ones contain noise, and which couples of views may or may not be complementary, in an exploratory context. Three main approaches are presented in this manuscript to approach this issues: 1) an optimization method relying on the Karush-Kuhn-Tucker conditions; 2) An approach based on non-stochastic bandits optimization that accounts for possibles changes in the quality of the views during the training process; and 3) A method relying on neural networks and mask optimization that was proposed in the context of an algorithm to reconstruct missing data based on information from other views.
2. The proposal of a universal multi-view clustering method based on a solid information theory background and which is not constrained to a very limited number of clustering algorithms.

We also show how this theoretical basis can be used for clustering fusions.

3. And finally, an ongoing work about the extension of the clustering theory of stability from Shai Ben David et al. [42] to the case of multi-view clustering.

All of the next sections correspond to individual contributions to tackle the aforementioned issues, with the exceptions of contributions being regrouped in the same section if one is the direct follow up of the other.

Please note that not all of my contributions will be mentioned in this chapter. Most notably, incremental contributions transforming existing batch algorithms into their incremental or online version have been purposefully omitted.

1.2 Tackling the issue of confidence in unsupervised multi-view environments

A common problem in multi-view environments is the issue of confidence [43][44]. While in supervised environments it is relatively easy to know which sets of features or which views are the best based on classification performances, there is no such thing in unsupervised environments. In multi-view clustering, this problem can translate into the question: *"How should we weight the views ?"* [45], and in collaborative clustering, we usually wonder about *"How do we select the best collaborators ?"* [46].

In the context of my work on multi-view clustering, most of the algorithms I have been working with -my own, but also the ones of other academics- we usually consider a system where the data $X = \{X^1, \cdots, X^J\}$ are split into J views. In a collaborative clustering context, the goal is then to find a solution $S = \{S^1, \cdots, S^J\}$ which solves Equation (1.1) below where $\mathcal{L}(X^j, S^j)$ is a local fitness function for each clustering algorithm in each view, $\mathcal{C}(S^i, S^j)$ is a consensus or agreement function between local partitions that can be asymmetrical, and each $\tau_{i,j}$ is the weight determining the weight of view j for its collaboration with view i.

$$S^* = \underset{S}{\mathbf{argmax}} \sum_{j=1}^{J} \mathcal{L}(X^j, S^j) + \sum_{i \neq j} \tau_{j,i} \cdot \mathcal{C}(S^i, S^j) \tag{1.1}$$

The difficult part of this problem is of course to determine the right weights $\tau_{i,j}$ all the while searching for an optimal solution S^*.

1.2.1 Optimization approaches to confidence in multi-view environments

To solve the problem shown in Equation (1.1), we first proposed to use an optimization process based on the Karush-Kuhn-Tucker conditions (KKT) [47]. Two contributions were made based on this solution: In [48] we applied this solution to the case of entropy-based collaborative clustering [49] and Kohonen-based [50] collaborative clustering [35][51], and in [52] the same problem is studied under the angle of non-convex optimization with a product instead of the sum inside the collaborative term.

In both publications, the problem is solved as follows: Finding the optimal $T = \{\tau_{j,i}\}_{J \times J}$ is equivalent to maximizing Equation (1.2) below where $\mathcal{C}(S^i, S^j)$ has been contracted into $\mathcal{C}_{ij}$ to

simplify the notations. We will assume that Δ is a well behave dissimilarity function so that $\mathcal{C}_{ij} \geq 0$ is always true.

$$T^* = \underset{T}{\mathbf{argmax}} \sum_{i=1}^{J} \sum_{j \neq i} \tau_{j,i} \cdot \mathcal{C}_{ij} \tag{1.2}$$

We propose the following normalization constraint over the coefficients:

$$\forall i \quad \sum_{j \neq i}^{J} (\tau_{j,i})^p = 1, \qquad p \in \mathbb{N}^* \tag{1.3}$$

The optimization problem then becomes:

$$\begin{cases} T^* = \mathbf{argmax}_T \sum_{i=1}^{J} \sum_{j \neq i} \tau_{j,i} \cdot \mathcal{C}_{ij} \\ \text{subject to} \quad \sum_{j \neq i}^{J} (\tau_{j,i})^p = 1 \quad \forall i \\ \qquad\qquad \tau_{j,i} \geq 0 \quad \forall (i,j) \end{cases} \tag{1.4}$$

The solution of this problem for $p > 1$ is given by the following proposition.

Proposition 1 *Any solution for the system* (1.4) *for* $p = 1$ *verifies:*

$$\forall j \neq i, \tau_{j,i} = \begin{cases} \frac{1}{|\mathcal{C}_{ij} = \max_k \mathcal{C}_{ik}|} \text{ if } \mathcal{C}_{ij} = \max_k \mathcal{C}_{ik} \\ 0 \text{ otherwise} \end{cases} \tag{1.5}$$

Proof We solve this problem under the KKT conditions. The five conditions form the following system:

$$\forall (i,j), i \neq j \begin{cases} (1) & \tau_{j,i} \geq 0 \quad \text{(primal feasibility)} \\ (2) & \sum_{j \neq i}^{J} \tau_{j,i} = 1 \quad \text{(primal feasibility)} \\ (3) & \lambda_{j,i} \geq 0 \quad \text{(dual feasability)} \\ (4) & \tau_{j,i} \cdot \lambda_{j,i} = 0 \quad \text{(complementarity slackness)} \\ (5) & -\mathcal{C}_{ij} - \lambda_{j,i} + \nu_i = 0 \quad \text{(stationnarity)} \end{cases} \tag{1.6}$$

We fix i. Let us suppose that there is at least one k_i so that $\tau_{k_i,i} > 0$. Such k_i must exist because of the primal feasibility condition (2). Then, (4) imposes that we $\lambda_{k_i,i} = 0$ and thus we have:

$$\nu_i = \mathcal{C}_{ik_i} \tag{1.7}$$

Then for all other values of $j \neq k_i$ two cases are possible:

Case 1: $\tau_{j,i} > 0$. In this case, we can use (5) in the same way as we did for k, we obtain: $\nu_i = \mathcal{C}_{ij}$. Using (1.7), it means that all positive coefficient $\tau_{j,i}$ have the same dissimilarity value Δ_{ij} with view i.

Case 2: $\tau_{j,i} = 0$. In this case, (5) gives us:

$$\lambda_{j,i} = \mathcal{C}_{ij} - \nu_i = \mathcal{C}_{ij} - \mathcal{C}_{ik_i} \tag{1.8}$$

And since we have $\lambda_{j,i} \geq 0$ due to the dual feasibility, then we need $\mathcal{C}_{ij} \geq \mathcal{C}_{ik_i}$, which means that k_i maximizes the consensus value.

The summary of this proposition is the following: In the context of collaborative clustering, the results should be better if each individual algorithm collaborates only with the algorithm that has the most similar solution. If several algorithms have the same most similar solution, then equal weights should be given to them and a weight of zero to the others.

Proposition 2 *Any solution for the system* (1.4) *for* $p > 1$ *verifies:*

$$\tau_{j,i} = \frac{(\mathcal{C}_{ij})^{\frac{1}{p-1}}}{(\sum_{k\neq i}^{J}(\mathcal{C}_{ik})^{\frac{p}{p-1}})^{\frac{1}{p}}} \tag{1.9}$$

Proof The five KKT conditions form the following system:

$$\forall(i,j), i \neq j \begin{cases} (1) & \tau_{j,i} \geq 0 \quad \text{(primal feasibility)} \\ (2) & \sum_{j\neq i}^{J}(\tau_{j,i})^p = 1 \quad \text{(primal feasibility)} \\ (3) & \lambda_{j,i} \geq 0 \quad \text{(dual feasability)} \\ (4) & \tau_{j,i} \cdot \lambda_{j,i} = 0 \quad \text{(complementarity slackness)} \\ (5) & -\mathcal{C}_{ij} - \lambda_{j,i} + \nu_i \cdot (p \cdot (\tau_{j,i})^{p-1}) = 0 \quad \text{(stationnarity)} \end{cases} \tag{1.10}$$

Let us begin by considering the case where $\lambda_{j,i} \neq 0$ in (4). Then, we would have $\tau_{j,i} = 0$ and with (5): $\mathcal{C}_{ij} = -\lambda_{j,i} \leq 0$. Since the $\mathcal{C}_{ij}$ have been defined as non-negative, this case is not possible, therefore we will only consider the case $\tau_{j,i} \neq 0$ and $\lambda_{j,i} = 0$. Then, with (5), we have:

$$\tau_{j,i} = \left(\frac{\mathcal{C}_{ij}}{p \cdot \nu_i}\right)^{\frac{1}{p-1}} \tag{1.11}$$

From Equation (1.11) and (2), we have:

$$1 = (p \cdot \nu_i)^{\frac{-p}{p-1}} \sum_{j\neq i}(\mathcal{C}_{ij})^{\frac{p}{p-1}} = (\nu_i)^{\frac{-p}{p-1}} \sum_{j\neq i}\left(\frac{\mathcal{C}_{ij}}{p}\right)^{\frac{p}{p-1}} \tag{1.12}$$

Then we can write:

$$\nu_i = \left(\frac{1}{\sum_{j\neq i}\left(\frac{\mathcal{C}_{ij}}{p}\right)^{\frac{p}{p-1}}}\right)^{-\frac{p-1}{p}} = \frac{1}{p}\left(\sum_{j\neq i}(\mathcal{C}_{ij})^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}} \tag{1.13}$$

Then by injecting the expression of ν_i into Equation (1.11), $\forall(i,j), i \neq j, p > 1$ we have:

$$\tau_{j,i} = \left(\frac{\mathcal{C}_{ij}}{\left(\sum_{k\neq i}(\mathcal{C}_{ik})^{\frac{p}{p-1}}\right)^{\frac{p-1}{p}}}\right)^{\frac{1}{p-1}} \tag{1.14}$$

$$= \frac{(\mathcal{C}_{ij})^{\frac{1}{p-1}}}{(\sum_{k\neq i}^{J}(\mathcal{C}_{ik})^{\frac{p}{p-1}})^{\frac{1}{p}}} \tag{1.15}$$

This second proposition offers a relaxed form of optimization in which the higher weights are still given to the most similar views.

1.2.1.1 Interpretations

Going deeper, we see that the degree to which one algorithm should collaborate with other collaborators that have dissimilar solutions depends on the degree of normalization p in Equation (1.3). For $p = 1$, each algorithm would only collaborate with the algorithm that has the most similar solution. If several algorithms have the same most similar solution, they would be given the same weight. When using a higher degree of normalization (Equation (1.9)), the algorithms with the most similar solutions would still be favored to optimize the likelihood of the global collaborative framework. But algorithms the solutions of which have a lesser degree of similarity would still be taken into consideration locally. In fact as p gets higher, the solutions from dissimilar algorithms would have a heavier and heavier weight, and at some point they would matter just as much as any other solution. In this later case, when the value of p is high enough, this would be equivalent to give the same weight to all the algorithms.

An simple interpretation of these results that can be applied to both collaborative and multi-view clustering is that in the absence of an external supervision (in the sense of supervised learning) to assess the quality of local views, the only valid quality criterion is to find similar structures in different views. This is exactly what our two propositions demonstrate by favoring exchanges between similar views that most likely have similar structures, with the parameter p that can be use as a "slack" parameter to give more or less freedom to explore more risky collaborations.

The limitations of this optimization model are quite obvious in the sens that it tend to reduce the diversity of partitions and views that can benefit from each others. As a matter of fact, while this conservative search for "stable structures" across the different views has been shown to be effective at detecting and neutralizing noisy views, it has also been experimentally proven -including in my own work [48][53]- that diversity and a clear quality criterion to improve (even an unsupervised one) are key elements to achieve consensus or collaborative results that do a better job than the average of the local views [46][54].

1.2.2 Non-stochastic multi-armed bandit optimization for collaborative clustering

In another contribution, we addressed the problem presented in Equation (1.1) as a bandit optimization problem.

Indeed, whether in multi-view or in collaborative clustering, it is impossible to tell in advance which views are going to bring useful information before trying to use them: On the one hand, using information from a good view will quickly improve the results, on the other hand using information from an average or low quality view will deteriorate the results and cost time before it is detected. To alleviate this problem, we proposed a contribution [55] in the form of a collaborative peer to peer clustering algorithm based on the principle of non stochastic multi-arm bandits to assess in real time which algorithms or views can bring useful information.

The main differences between this proposal and the one from the previous section are the following: 1) Instead of a purely mathematical optimization, we propose a method based on trial and errors using the multi-arm bandit algorithm. This method is closer to real cases of collaborations where it is difficult to know whether a view or algorithm can bring some positive information or not before trying to communicate. 2) We take into consideration the cost of communications between

all sites, which is a novelty since none of these earlier work considers the physical architecture of the collaborative system between data on different sites.

Furthermore, this contribution was focused on the optimization of a dual form of Equation (1.1) that is also common in multi-view learning: Instead of trying to optimize a function based on a consensus measure between partitions, we used one based on a divergence function as shown in Equation (1.16) where $\Delta(S^i, S^j)$ is a dissimilarity function -once again potentially asymmetrical- between two local partitions S^i and S^j.

$$S^* = \underset{S}{\mathbf{argmax}} \sum_{j=1}^{J} \mathcal{L}(X^j, S^j) - \sum_{i \neq j} \tau_{j,i} \cdot \Delta(S^i, S^j) \tag{1.16}$$

1.2.2.1 Multi-armed bandits

The multi-armed bandit problem [56] is a decision problem in which a learning agent (the player) must repeatedly decide between several machines, each machine i yielding a reward $r_{t,i}$ at time t. The reward is taken from an unknown distribution R_i. The goal, for the player, is to find a policy, that is a sequence of choices between machines, which maximizes the sum of rewards over T repetitions of the game. The usual formulation of the multi-armed bandit problem assumes that rewards are independent and identically distributed. However, there exists some formulation of the problem where the i.i.d rewards assumptions does not hold. The exponential weight algorithm for exploration and exploitation (Exp3) [57], was designed by Auer et al. as a general solution to the bandit problem where no assumption is made about the distribution of rewards.

Algorithm 1: Exp3 algorithm

$\gamma \in [0, 1]$
$\forall i \in [0, K], w_{i,0} = 1$
for t *from* 0 *to* T_{max} **do**
  $\forall i, p_i(t) = (1 - \gamma)\dfrac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \dfrac{\gamma}{K}$
  choose i_t with $p_i(t)$
  $x_t = R(i_t)$
  $\hat{x}_t = x_t / p_i(t)$
  $w_{i,t+1} = w_{i,t} e^{\gamma \hat{x}_t / K}$
  $\forall j \neq i, w_{j,t+1} = w_{j,t}$

The reason why we chose to use non-stochastic multi-arm bandits for our collaborative clustering problem is quite simple: in collaborative clustering due to the lack of supervision, it is impossible to know in advance which collaboration will be effective or not without trying it. Furthermore, an algorithm bringing huge gain during an iteration may not be that interesting anymore during the next run. As such, non-stochastic multi-armed bandits are a perfect tool to explore the optimization problem of picking the right inter-site collaborations in real time.

1.2.2.2 Exp3 collaborative clustering with the K-Means algorithm

For the modeling of this problem, we assume that the different views and algorithms form a weighted, un-directed, graph $\mathcal{G} = (\mathcal{V}, \mathcal{T})$ in which the set $\mathcal{V}$ represents the set of the different views, and $\mathcal{T}$

is the set of weighted edges highlighting the weighted links between the views: $T = \{\tau_{i,j}\}_{(J\times J)}$. As constraints: we assume that each view can only communicate with a limited number of other views while doing its clustering. This constraint seemed logical to us at the time to ensure that the algorithm would stop at some point.

Each view begins by performing an initial (local) clustering with the chosen algorithm. After this initial phase, each view begins to regularly choose another view to request (pull) prototypes in order to update its local clustering based on the information shared by its peer. The algorithm stops after each view has performed a finite number of exchanges, or when it has reached a certain target metric. The critical goal to achieve is of course to select the best possible views to exchange data with. The confidence matrix $T = \{\tau_{i,j}\}_{(J\times J)}$ will play this role and for each view i, we have $\sum_{j\neq i}^{J} \tau_{j,i} = 1$, the same constraint as we had previously so that the $\tau_{j,i}$ will contain the probabilities of exchanging with each other view.

This process is detailed in Algorithm 2, Let R be the maximum number of rounds of the algorithm. Let S^i be the local partition at view i and N_i the number of neighbors at site i.

Algorithm 2: Peer-to-peer Collaborative Clustering algorithm

S^i: initial clustering of site i
N_i: set of neighbors of site i
R: Maximum number of rounds
Algorithm `SendClustering`(S^i, N_i)
  $i = 0$
  while $i < R$ **do**
    $n \leftarrow$ select(N_i)
    $response \leftarrow$ send(n,S^i)
    update($response$,n,T_i, S^i)
Algorithm `OnClusterReception`(S^n, n)
  $G_i \leftarrow$ update(G^n, n S^i)

The two critical functions of this algorithm are the **update** and **select** which respectively update the local clustering depending on collaborative clustering rules, and select a neighbor node to send information to. This is done without a-priori knowledge of the other views at the beginning. However each site can learn dynamically from its neighbor by computing the agreement between its local clustering and its neighbor's.

This problem can be seen, for each view in the collaborative clustering network, as finding the most informative and economic neighbors to communicate clustering information to. The final goal is then to find an acceptable clustering solution in a reasonable amount of time while ignoring "noisy" neighboring sites who do not contribute efficiently towards correct a clustering of the data. Therefore, we formalize the problem of identifying the most informative neighbor to exchange data with as a multi-armed bandit problem, in which we model each site in the collaborative clustering network as a multi-armed bandit player, and the site's neighbors as the machines.

Let S^i be the clustering of site i, S^j the clustering of site j and k the number of clusters. A clustering is a matrix of size $n \times k$ where n is the number of individuals, where $S^i_{x,y} = 1$ if x is in cluster y in clustering i, 0 otherwise.

In order to model our collaborative clustering problem as an adversarial bandit problem we need to define a reward function indexed on the agreement between two clusterings. We chose to

use the probabilistic confusion entropy [58][59] as defined in equation 1.18, as a basis for our reward function. To compute this entropy we need to compute the confusion matrix $\Omega^{i,j}$ of clusterings S^i and S^j, as shown below for K clusters:

$$\Omega^{i,j} = \begin{pmatrix} \omega_{1,1}^{i,j} & \cdots & \omega_{1,K_j} \\ \vdots & \ddots & \vdots \\ \omega_{K_i,1}^{i,j} & \cdots & \omega_{K_i,K_j} \end{pmatrix} \text{where } \omega_{a,b}^{i,j} = \frac{|S_a^i \cap S_b^j|}{|S_a^i|} \tag{1.17}$$

This yields the following definition of entropy between sites i and j, where $P(S_a^i) = \frac{|S_a^i|}{|S^i|}$:

$$H(\Omega^{i,j}) = -\sum_{x,y} \omega_{x,y}^{i,j} \log \frac{\omega_{x,y}^{i,j}}{P(S^i)} \tag{1.18}$$

As entropy measures the level of agreement between two clustering of the same individual, in our bandits learning formulation we are rather interested in the gain induced by updating clustering S^i with clustering S^j. Let $S^{i\prime}$ be the clustering after the merge and update of S^i by S^y. We compute this gain by the difference in entropy between S^i and S^j, and $S^{i\prime}$ and S^j. This provides us the following reward equation:

$$R(i,j) = 1 - \frac{H(\Omega^{i\prime,j})}{-log(1/k)} \tag{1.19}$$

Where k is the number of clusters.

We update each local clustering depending on the level of disagreement between each local (l), and remote (r) clustering partitions based on the confusion matrix $\Omega^{l,r}$.

$$C_k^l(t+1) = C_k^l(t)\Omega_{k,k}^{l,r} + \sum_{i \neq k}^{K} C_i^r(t)\Omega_{k,i}^{l,r} \tag{1.20}$$

Where C_k^l is the local kth prototype of the local site. The new centers then provide means to compute a new clustering for the local individuals following the K-Means rule [18]:

$$S^{i\prime} = \arg\min_{l,k} ||X_l - C_k||^2 \tag{1.21}$$

From there, each site in the network computes an estimate of the global entropy across all sites, by computing the local mean entropy with all its neighbors. This estimate will eventually reach either 0.0 or a plateau in the course of the collaboration since our algorithm rewards sites that help lowering the entropy. This estimate can be used in combination with a threshold parameter α so that any site with a mean entropy lower than α will stop asking clustering information from its neighbors. In addition to this threshold, we provide each site with a maximum number of iterations R_{max} in which to ask data from neighbors. If the maximum number of turns is reached before the *alpha* threshold then the algorithm stops.

The combination of our peer to peer Collaborative clustering algorithm with the Exp3 algorithm for non-stochastic bandit learning provides the Exp3 Collaborative K-means algorithm written in Alg. 3.

This local algorithm proceeds by initializing the local clusters thanks to the K-Means algorithm, and then proceeds to a **pull** mode gossiping by requesting clustering information from a neighbor

Algorithm 3: Exp3 Collaborative K-means algorithm

Let K be the set of clusters
Let X_i be the local data-set
Let S_0 be the local clustering at round 0
Let $C_0^l = f(S_i)$ the local cluster centers at round 0
$\gamma \in [0,1], \alpha \in [0,1]$
$\forall i \in [0,N], w_{i,0} = 1$
$\forall i \in [0,N], a_{i,0} = -log(1/K)$
while $t \in 0..T_{max}$ *or* $\frac{\sum_i^N a_{i,t}}{N} < \alpha$ **do**
  Compute $p_n(t)$ as in Alg. 1
  Request G_t^r from neighbor n_t chosen from $p_n(t)$
  Compute agreement $a_{l,t} = H(G_t^n, S_t^l)$ Compute $\Omega^{l,r}$ as per Eq. 1.17
  $\forall k \in K, C_k^l(t+1) =$ update centers C_k^l as per Eq. 1.20
  $S_{t+1} =$ update clusters based on X_l, C_{t+1} as per Eq. 1.21
  $a_{l,t+1} = H(\Omega^{l,r})$
  if $a_{l,t+1} < a_{l,t}$ **then**
    $S_{t+1} = S_t$
  $r = R(l,r)$ as per Eq. 1.19
  Update w_r as in Alg. 1

chosen thanks to the selection probabilities $p_n(t)$ (line 7, 8). After the generation of a new clustering with newly updated centers, the entropy level between this new clustering and the neighbor's clustering is computed (line 14). If this new version of the clustering does not provide an improvement in entropy, then the update is discarded (line 16). In any case the reward is computed and used for updating the appropriate weight.

The critical part of the algorithm is the computation of the confusion matrix Ω. This operation has a complexity of $\mathbf{O(kn)}$ with k the number of classes and n individuals. Other operations such as the updates of centers and cluster assignments have complexities proportional to the number of clusters or the numbers of individuals. This operation happens at each turn of the algorithm, the number of turns being bounded by the stopping criterion or a limit parameter.

1.2.2.3 Conclusion

This work introduced a different approach to tackle the problem of confidence in multi-view clustering by using the same theoretical model as non-stochastic multi-armed bandits. It also ended up with the proposition of a new and original bandit-based collaborative clustering algorithm, named Exp3 Collaborative K-Means, which allows collaborators in a collaborative clustering set of sites, to identify the most appropriate site to share information with. This algorithm was applied to two data sets with various connectivity levels between collaborators, and show that thanks to the bandit learning component of the algorithm, sites that are providing useful information to clustering are consistently identified, and privileged in data exchange compared to sites with less useful information. By doing so, our method achieves better results than already existing purely mathematical optimization methods that relied only on the diversity between the different sites, which resulted in a lack of risk taking and lower chances of rapid results improvement.

1.3 Deep Cooperative Reconstruction in multi-view environments

In the introduction for this chapter, we made the assessment that one basis of multi-view clustering was that it deals with data that have representations or are split in different views. We also mentioned a few examples of such databases such as social networks and client databases. In the previous section, we mentioned the issue of confidence as being a key problem when doing multi-view clustering. However, in addition to the issue of confidence, there is another major problem when dealing with multi-view data: In practical applications all data are rarely represented in all views. This is a significant problem because most multi-view and collaborative clustering frameworks make the assumption that all objects are in all views.

To tackle this issue, in [60][44] we proposed a method called the Cooperative Reconstruction System which aims at reconstructing information missing in some views in a multi-view context using information available in the other views. As an echo to the works presented in the previous section, our algorithm also proposes an optimization method to weight the views based on their perceived usefulness to help reconstructing other views. Finally, our method considers privacy issues and therefore achieves said reconstruction without direct data transfer from one view to another.

This work is relatively close to other works in Deep multi-view learning, most notably the work of Wang et al. [61] where the authors propose a systems that learns from representations and features in a multi-view setting where only one view is available at test time, and the work of Ngiam et al. [62] where the authors propose a system that tries to reconstruct shared representations that are available from 2 views available at a given time.

1.3.1 Cooperative reconstruction system

1.3.1.1 Formalism

Let $\mathcal{X}$ be a set of individuals. Let $V_0, V_1, ..., V_n$ be a set of views, each in its own feature space $\mathcal{F}_0, \mathcal{F}_1, ..., \mathcal{F}_n$, such that $V_i : \mathcal{X} \rightarrow \mathcal{F}_i$. Let $\mathcal{X}_i \subset \mathcal{X}$ be the subset of individuals visible in view V_i. In other words $\mathcal{X}_i$ is the subset of the population for which data is available in the feature set of view V_i. We note $V_{i|j}$ the subset of V_i (in the feature space of V_i) which individuals are also present in V_j.

To its core, the *cooperative reconstruction system (CRS)* aims at learning, in view i, a reconstruction function F_i of individuals $x \notin X_i$ in view V_i, based on information provided by the other views. Therefore *ie*: $F_i : \cup_{j \neq i} \mathcal{F}_j \rightarrow \mathcal{F}_i$:

$$\tilde{x}_{u,i} = F_i(x \in X_{j \neq i}) \tag{1.22}$$

This formulation is often used in recommender systems, but in the context of multi-view systems, it ignores two critical constraints:

1. Data Security: in the context of this paper, data security is defined as the constraint of not being able to access original data if it is not from its original view. The input space of the reconstruction function should be different from the concatenation of the other views feature spaces.

2. Scalability: If a new view is added (rep. removed) to/from the system, how is learning the new representation affected by this change.

These two constraints provide new way to formulate the problem:

$$\tilde{x}_{u,i} = F_i(x \in E_{j\neq i}(X_{j\neq i})) \tag{1.23}$$

Where E_i is an encoding function on $\mathcal{F}_i$. This encoding must be designed in such a way that only the view containing the individual's original features can reconstruct the values from the encoding.

1.3.1.2 Algorithm

A global representation of our proposed cooperative reconstruction system is shown in Figure 1.2. Our system is based on several modules: first, to solve the problem of security-friendly information transfer, the system uses a set of N Autoencoders [63] –with N being the number of views–, to locally encode data to make them impossible to read from outside of their views. In a way, our proposal is similar to the architecture proposed in [64], but differs in the sense that we aim at multi-view reconstruction instead of a single consensus representation.

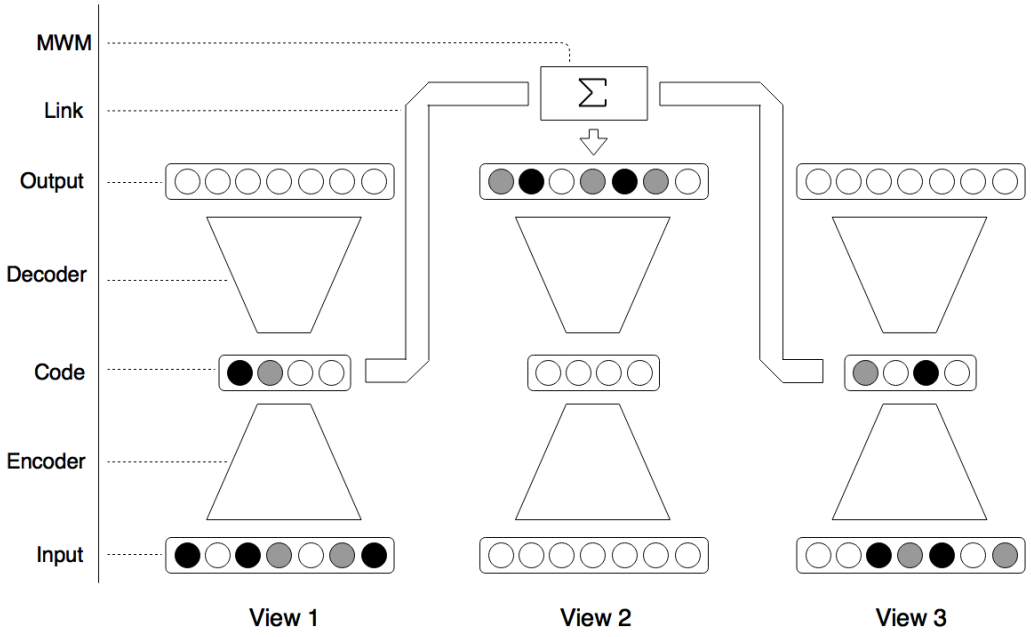


Figure 1.2: Global representation of our Cooperative Reconstruction System: an example with 3 views.

Let us use Figure 1.2 as a base to explain how our proposed method works. In this example we consider a system that only contains 3 views, but our architecture is meant to be adaptable. In this case scenario, we consider a dataset of shared individuals that have representations across the 3 views, and we have an individual or an object which exists in views 1 and 3, but has no representation in view 2. From there, the goal of our system is to reconstruct or to predict what this individual features should be in view 2.

The reconstruction system is made of the following components:

– Autoencoders that are specific to each view. These neural networks typically try to reconstruct their input data after going through a set of fully connected layers that first compress the information up to a bottle-neck (encoder part) and then decompress it between the bottleneck and the output (decoder). This component of the architecture serves two purposes:

1) Extracting the best representations of the data in each view at the bottleneck, as this is what autoencoders are good at [63]; and 2) providing a way to encrypt the data before it is send in the other views [65][66]. A more in depth presentation of how autoencoders work is proposed in Section 2.2.2.2.

– The links between the views. They are in fact another neural network made of fully connected layers and whose goal is to translated the features of the view sending information into the features of the view receiving them.

– A system of masks called "MWM" for Mask Weighting Method, that we will detail in the next section (1.3.2). It is basically use to combine the features reconstructed from several views and to weight them according to the pertinence or confidence granted to each view.

In our example, from Figure 1.2, the original features from views 1 and 3 would go through their respective encoder to be coded into better features. Then they would be sent to view 2 using the links that would translate them into code features for view 2. Then the Mask weighting Method would recombine them into a single code vector in view 2. And finally they would go through the decoder of view 2 to be reconstructed into the original features of this view.

Since the whole system is basically a big neural network, it is trained by using data that are shared across all view as a training set. All layers connection weights are optimized using gradient descent.

1.3.2 Weighting the views with smart masks

As mentioned earlier, the issue of confidence in multi-view environments without supervision is a constant problem. And this holds true for our proposed cooperative reconstruction system as well. In the case of this application the issue is to find for each view individually what are the best combinations of external views to reconstruct the local data. More practically, using the architecture presented in Figure 1.2, we want to know how to combine the reconstructions provided by the link networks in order to best reconstruct the local codes.

To solve this issue, our deep cooperative reconstruction system proposed a smart mask system: We present a method based on a set of scalar vectors $W_i = \{w_{i|j}, j \in [1..N] \setminus i\}$ such that $w_{i|j}$ is of same dimension as vectors of V_i. To get the final output $\widetilde{x}_i$ of the system in the local view i, we use the following formula:

$$\widetilde{x}_i = \sum_{j \in [1..N] \setminus i} x_{i|j} \otimes w_{i|j} \tag{1.24}$$

with $\otimes$ the pointwise vector product and $x_{i|j}$ the version of the missing data inferred using data from the view j.

The coefficients are first initialized using equal weights summing to 1 for all features. Then, they can be learned using two methods: either using gradient descent on the reconstruction error, or through an iterative update using the zero of the derivation of this latter error.

The coefficients are first initialized using equal weights summing to 1 for all features. Then, they can be learned using two methods : either using gradient descent on the reconstruction error, or through an iterative update using the zero of the derivative of this latter error.

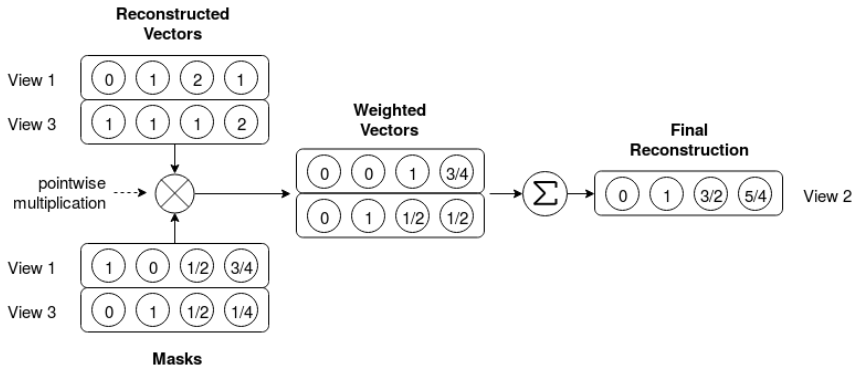


Figure 1.3: The Masked Weighting Method: here view 2 reconstructs a local code based on information from views 1 and 3, and it uses the masks previously trained to get the final weighted result.

1.3.2.1 Mask training through gradient descent

As explained earlier, the whole algorithm is trained using samples that are shared across all views. Using the system output, it is therefore possible to perform a Gradient Descent on the weights of W_i. The error we used is the mean squared error (MSE) between target data and reconstructed ones. The computation of the error E_i for the view i can be written as follows:

$$E_i = \frac{1}{|V_i|} \sum_{x_i \in V_i} ||x_i - \widetilde{x}_i||^2 \tag{1.25}$$

$$= \frac{1}{|V_i|} \sum_{x_i \in V_i} \sum_{k=1}^{\dim(V_i)} \left(x_i^k - \widetilde{x}_i^k\right)^2 \tag{1.26}$$

$$= \frac{1}{|V_i|} \sum_{x_i \in V_i} \sum_{k=1}^{\dim(V_i)} \left(x_i^k - \sum_{j \in [1..N] \setminus i} w_{i|j}^k x_{i|j}^k\right)^2 \tag{1.27}$$

where x_i^k is the k-th coordinate of the individual x_i. The differentiation of E_i w.r.t. the parameters $w_{i|j}^k$ of W_i can then be written:

$$\frac{\partial E}{\partial w_{i|j}^k} = \frac{2}{|V_i|} \sum_{x_i \in V_i} x_{i|j}^k \left(\widetilde{x}_i^k - x_i^k\right) \tag{1.28}$$

This latter formula makes it possible to update the weight $w_{i|j}^k$ using the usual gradient formula

$$\left(w_{i|j}^k\right)^{new} = \left(w_{i|j}^k\right)^{old} - \epsilon \frac{\partial E}{\partial w_{i|j}^k} \tag{1.29}$$

where $\epsilon > 0$ is the parameter defining the learning rate of the process. This update process is performed on every weight until convergence. In practice, the learning is stopped when the norm of the update value defined in Equation 1.28 goes under a threshold fixed by the user.

1.3.2.2 Iterative update

It is also possible to update weights based on the minimum of E_i found using Equation 1.28, which after a few developments gives us:

$$
\begin{aligned}
& \frac{\partial E_i}{\partial w^k_{i|j}} = 0 \\
\Rightarrow \quad & \frac{2}{|V_i|} \sum_{x_i \in V_i} x^k_{i|j} \left(\widetilde{x}^k_i - x^k_i \right) = 0 \\
\Rightarrow \quad & \sum_{x_i \in V_i} \Big(\left(x^k_{i|j}\right)^2 w^k_{i|j} + x^k_{i|j} \big(\sum_{j' \in [1..N] \setminus \{i,j\}} w^k_{i|j'} x^k_{i|j'} - x^k_i \big) \Big) = 0 \\
\Rightarrow \quad & w^k_{i|j} \sum_{x_i \in V_i} \left(x^k_{i|j}\right)^2 = \sum_{x_i \in V_i} x^k_{i|j} \big(x^k_i - \sum_{j' \in [1..N] \setminus \{i,j\}} w^k_{i|j'} x^k_{i|j'} \big) \\
\Rightarrow \quad & w^k_{i|j} = \frac{\sum_{x_i \in V_i} x^k_{i|j} \big(x^k_i - \sum_{j' \in [1..N] \setminus \{i,j\}} w^k_{i|j'} x^k_{i|j'} \big)}{\sum_{x_i \in V_i} \left(x^k_{i|j}\right)^2}
\end{aligned}
\tag{1.30}
$$

Equation 1.30 shows that the update of $w^k_{i|j}$ requires the values of $\{w^k_{i|j'}, j' \in [1..N] \setminus \{i,j\}\}$. Thus it is possible to define an iterative update for which the values of $\{w^{k,t}_{i|j'}, j' \in [1..N] \setminus \{i,j\}\}$ at time t are used to obtained $w^{k,t+1}_{i|j}$ at time $t+1$. This problem being convex, the iterative process is performed until convergence of the weights.

This weighting method is used because it offers several advantages:

1. With either a noisy external view or a low-quality Link, the weighting coefficients for this view will converge to a value under $\frac{1}{N-1}$ which is the value corresponding to a mean of the external views. By doing so, the method lowers the impact of the bad reconstruction on the result.

2. On the opposite, this method will favor views which might greatly improve the final reconstruction with a weight over $\frac{1}{N-1}$.

3. Contrary to a weighted mean which would assign a single scalar to a view, this method allows to favor only a subpart of an inferred vector. One can easily imagine that an external view would only allow to recover parts of the local information. Our weighting method makes it possible to automatically identify these parts during parameters training.

When W_i has been trained for all the views, the system is ready to use on missing data. An abstraction of the reconstruction process can be found on Figure 1.4.

1.3.3 Result analysis and conclusions

While we tested our method reconstruction capability on several datasets [60] with varying degrees of success (see examples with a CIFAR like dataset in Figures 1.5 and 1.6), it was not so much the quality of the reconstruction itself that we were interested in, but rather if it could be reused successfully for a subsequent Machine Learning task, be it classification or clustering.

To do so, we applied a random forest algorithm [67] to the original data and the reconstructed ones. And we compared the difference in term of classification accuracy. We found that most datasets had less that 3% of difference in classification accuracy (min 2%, avg 5%, max 7.5%) between the reconstructed and the original data. Furthermore, we also saw that there was no significant correlation between the reconstruction error and the classification accuracy. This shows that our

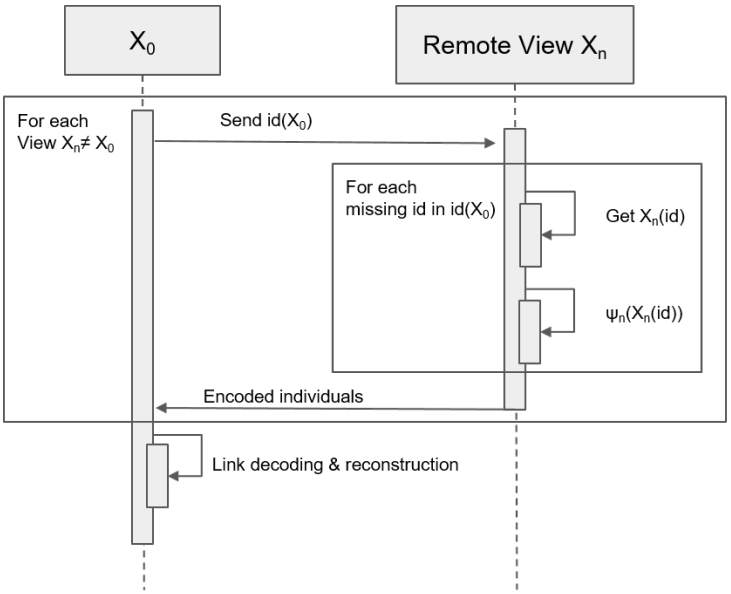


Figure 1.4: Reconstruction process: Identification of a missing item, encoding in the remote view, and reconstruction in the local view.

method was great at capturing the great underlying structure of the data to reconstruct them, and that its inability to reconstruct missing elements with a low error was mostly be due to the impossibility to reconstruct the variance around the core structure especially for features that don't have a strong correlation with any other features.

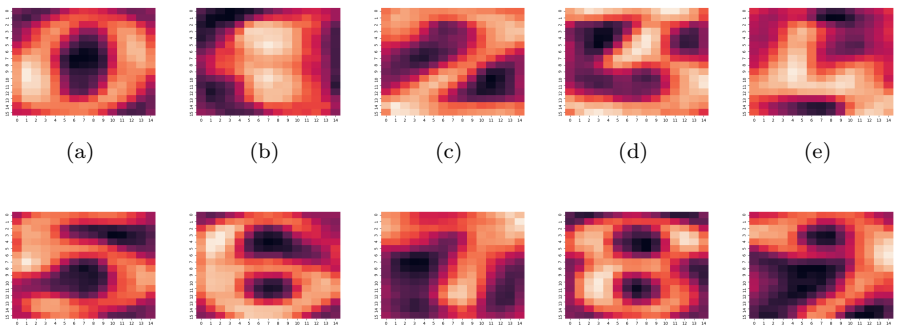


Figure 1.5: Sample of the reconstructed images available in the MFDD dataset. Some well reconstructed examples.

As a conclusion: in a global context of multiplication of multi-view data, we have presented a new system called the Cooperative Reconstruction System. The purpose of this system is to reconstruct data missing in some views by using information contained in other views. We do so without sharing the original data, thus avoiding security issues. To do this, the system relies on three modules: Autoencoders to encrypt the data under a compressed scalar vector form, fully connected deep networks -called Links- to decipher an external code in a local view, and the Masked Weighting Method, a new weighting method to combine all external reconstructions, thus obtaining the final reconstruction. The Masked Weighting Method has 3 functions: combining external information,

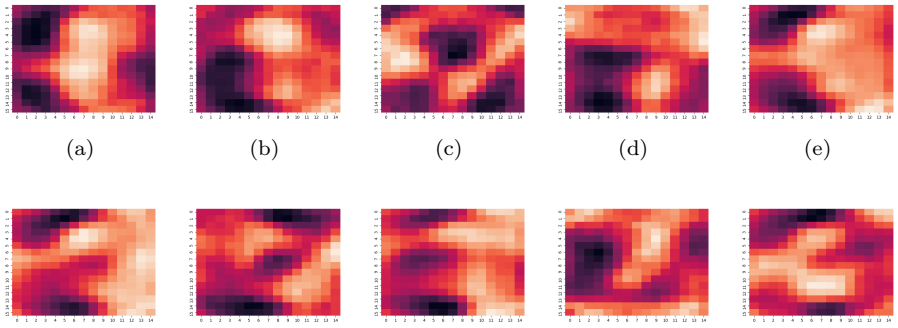


Figure 1.6: Sample of the reconstructed images available in the MFDD dataset. Some poorly reconstructed examples.

reducing the influence of views with information which could hinder the reconstruction process, and reducing the impact of missing data during the system training process.

As future works, we plan on improving the reconstructions acquired from the external views through the modification of the inter-view Links. Likewise, because of the potentially high dimensionality, the use of another error than the MSE should be considered (at least for non-image data). A feature selection process may be added to the system, thus limiting the impact of the noise features in the original data set. Another possible future extension of this work would be to work on a lighter architecture that would scale better with large data sets, or to work on an online version to alleviate the issue of scaling to large datasets.

1.4 Information theory based approach of multi-view clustering

In this section, we present the summary of two contributions that address the issue of clustering as an information compression problem. These two contributions are directly linked and rely on the same theoretical background: The first contribution [68] applies the idea of information compression as a clustering metric in the context of multi-view clustering without merging the partitions (horizontal collaborative clustering). The second contribution [69] uses the same idea for multi-view clustering but this time with the goal of merging the partitions into a single result.

1.4.1 Minimum Length description applied to clustering

1.4.1.1 Reminder: Kolmogorov complexity

A long philosophical tradition has investigated the problem of induction. Among the proposed methodologies, Ockham's razor is widely used and discussed. This simplicity principle states that, among all possible hypotheses, only the "simplest" one should be chosen to describe an observation. A more formal version of this idea has been introduced in computer science by [70] and [71] with the Minimum Description Length (MDL) principle. This principle states that the best model to select leads to a maximal compression of observed data. *Given a data set and an enumeration of theories to explain data, the best theory is the one that minimizes the sum of the length (in bits) of the theory description and the data encoded with the theory.*

The notion of *description length* originates from algorithmic theory of information and designates the minimal number of bits needed by a Turing machine to describe an object [72]. This measure is given by the tool of Kolmogorov complexity. If $\mathcal{M}$ is a fixed Turing machine, the complexity of an object $\mathbf{x}$ given an object $\mathbf{y}$ on machine $\mathcal{M}$ is defined as $K_{\mathcal{M}}(\mathbf{x}|\mathbf{y}) = \min_{p\in\mathcal{P}_{\mathcal{M}}}\{l(p) : p(\mathbf{y}) = \mathbf{x}\}$ where $\mathcal{P}_{\mathcal{M}}$ is the set of programs on $\mathcal{M}$, $p(\mathbf{y})$ designates the output of program p with argument y and l measures the length (in bits) of a program. When the argument $\mathbf{y}$ is empty, we use the notation $K_{\mathcal{M}}(\mathbf{x})$ and call this quantity the complexity of $\mathbf{x}$.

As we have defined it, the complexity of an object cannot be considered as an intrinsic property of the object since it depends on a fixed Turing machine $\mathcal{M}$. In order to overcome this weakness, the *invariance theorem* enables to define a machine-independent definition of the complexity. Although such a measure has a major theoretical impact (see for instance [73][74]), we will focus on a machine-dependent approach in the rest of this paper. Our choice is motivated by three main reasons exposed thereafter.

First, the universal complexity is not computable, since it is defined as a minimum over all programs of all machines. By choosing a precise machine, we restrict the research to a minimization over the set of programs only, which can be relatively simple depending on the chosen machine.

Second, machine dependency is a fundamental property of learning. It is intuitively obvious that all learners have their own data processing, and thus are naturally biased toward some precise tasks. For instance, human mind is designed to perceive some regularities in scenes that state-of-the-art algorithms cannot get, while they are unable to cope with pattern recognition in strings like DNA, which is now a basic task for a computer program. Since any learning method has a natural bias toward some kinds of problems, we propose here to interpret this property in terms of machine dependency: A learning algorithm corresponds to a specific choice of a Turing machine with its representation bias.

Finally, we have to notice that this assumption is a classical assumption in statistical learning theory. The restriction of the research space to classes of decision functions (hence classes of Turing machines) is even the key hypothesis in learning theory and leads to all classical definitions such as the VC-dimension in supervised learning. From our perspective, this dimension can be considered as a measure of the restriction impact. Statistical learning relies on this very assumption: because of the non-calculability of probabilities and in order to prevent overfitting (i.e. to reject distributions which do not obey the commonly admitted aim of generalization), the assumption of choosing a restricted set of hypotheses is well accepted in the machine learning field.

1.4.1.2 Notations

We consider a data set X that can be divided into J views so that: $X = \{X^1, \cdots, X^J\}$. A view correspond to a restricted version of the dataset. We make the hypothesis that all data points have a representation in each of the view. Let N be the number of data points.

We consider the collaborative setting in which we have J algorithm (one per view) denoted $\mathcal{A}^1, \cdots, \mathcal{A}^J$. We consider these algorithms to be mapping functions from the data space into an integer: $\mathcal{A}^j$ processes the data set X^j and outputs a solution vector $S^j \in \mathbb{N}^N$. In practice we consider the number of cluster to be finite and equal to K^j for any algorithm $\mathcal{A}^j$. Please note that this number can differ from one view to another. Each algorithm $\mathcal{A}^j$ is also associated with a set of parameters $\theta^i \in \Theta^i$. These parameters may also differ from one view to another and depending on

the type of clustering algorithm used.

In the following, we consider that the machine $\mathcal{M}$ is fixed. To make the equations easier to read, we will omit to specify the machine $\mathcal{M}$ in the complexity (hence we will denote by $K(\mathbf{x})$ the complexity of $\mathbf{x}$ on the chosen machine).

1.4.1.3 Local sub-machines

The purpose of the following section is to describe a class of Turing machines which is adapted to the multi-view setting.

Given multi-view data, the purpose here is to define a parameterized class of Turing machines $\mathcal{M}$ which generate the data. In a multi-source setting, and without any loss of generality, we consider that each view is encoded on a tape. We consider that data points are encoded in a given (and known) order and are separated, in such a way that the content of a tape can be uniquely decoded.

Local clustering (ie. clustering on a single view) can be interpreted as a compression of data based on external parameters. For instance, a centroid-based clustering (like K-means [18], K-medoids [75] or GTM [76]) compresses the data by "factorizing" a common position into the center. We propose to define *local sub-machines* as machines which take as input a parameter θ^j and a solution vector S^j and output the corresponding data. The length of such machines is equal to $K(X^j|S^j, \theta^j)$.

The format of these machines will depend on the nature of the clustering algorithms. It is noticeable that the framework of algorithmic learning theory authorizes a large class of data representations (and thus can be used for collaboration between different types of clustering methods). We provide a couple of examples in the thereafter:

Probabilistic models [68][77]: Clustering algorithms that rely on probabilistic models often try to model clusters by their density, usually using models such as mixtures of gaussian distributions. For this family of algorithms, the parameter θ corresponds to the parameters of the distribution. The solution vector here is the distribution in the mixture to which each point is associated.

In order to actually compute the complexity, we use the property that the complexity of a point x given a distribution p is upper-bounded by $K(x|p) \leq -\log p(x) + \mathcal{O}(1)$. In particular for a mixture of k distributions $(p_1, \cdots, p_k)$, a point x in a cluster i will be described with a complexity:

$$K(x|S, \theta) = -\log p_i(x) \tag{1.31}$$

Prototype-based models [68]: In clustering algorithms such as K-means, K-medoids, GTM, SOM [50], the parameter θ is the description of the position of the prototypes. Each data point is represented by its membership to its associated prototype (the association table being given by the solution vector S^j). The solution vector S corresponds to the solution, hence to the point-prototype association.

For these algorithms, the complexity $K(x|S, \theta)$ can be computed based on the distance between the data and the prototype.

$$K(X|S, \theta) = \sum_{i=1}^{N} K(X_i|\mu_i) + \mathcal{O}(1) \tag{1.32}$$

It is also possible to model these prototype-based models as probabilistic models with the ad-hoc variance-covariance matrices. Algorithms based on the K-Means and Fuzzy C-Means algorithms can for instance easily be modeled as a degenerate gaussian mixture model.

Density-based models [77]: Algorithms such as DBSCAN [9] and OPTICS [11] are more problematic because they do not rely on any explicit parameter θ. It is however possible to propose a description of points based on a reordering of the data set, which would then be seen as the parameter of the algorithm. The density-based models aim to find the better attachment of points inside the data set. Based on this idea, the computation of the complexity can be done as follows.

We denote by π_i the index of the parent of point i in the ordering proposed by a method such as OPTICS. Exactly as suggested for the prototype-based method, the idea will be to describe the position of a point by its relative position with respect to a reference point, which is not a prototype in this case but the parent in the ordering. Points that have no parents (hence first point of a class in the ordering) are described by their absolute position. The total complexity is then given by:

$$K(X|S,\theta) = \sum_{i=1}^{N} K(X_i|X_{\pi_i}) + \mathcal{O}(1) \tag{1.33}$$

Other models [77]: Clustering algorithms that do not match any of these category are a bit more difficult to tackle. Beyond the possibility of finding an ad-hoc formulation of the Kolmogorov complexity, two possibilities exist:

- The least satisfying one is to ignore the model (S^j, θ^j) all together and thus to consider that $K(X^j|\theta^j, S^j) = K(X^j) + \mathcal{O}(1)$
- The second solution is to artificially inject prototypes or a density-based models on top of the existing clustering and to fall back to the computation proposed by either of these models.

1.4.1.4 From global parameters to local views

We propose a decomposition of the global Turing machine into sub-machines, as exposed in Figure 1.7. In order to make the description more understandable, we invite the reader to think of machines as actual computer programs and the complexity (also called length) as the length of the program as written in a fixed programming language.

The j-th local sub-machine is in charge of producing data X^j from the clustering parameter θ^j and the solution vector S^j, received as inputs. These parameters were transferred to it from a *global configuration machine* which stores the whole configuration (ie. the complete description of all θ^js and S^js). A splitting operation is needed to transform the output $\langle \theta^1, S^1, \ldots, \theta^J, S^J \rangle$ of the *global configuration machine* into the inputs $\langle \theta^j, S^j \rangle$ of the local sub-machines. Since we use prefix codes and the index j of the parameters θ^j and S^j is explicitly given onto the tape of the global sub-machine, the complexity of this splitting operation is a constant which does not depend on the data nor on the parameters.

The *global configuration machine* receives as input the local parameters $\theta^1, \ldots, \theta^J$ and a global solution vector $\langle S^1, \ldots, S^J \rangle$. The length of this machine corresponds to the description length of the parameters θ and the cost of a concatenation (hence a constant). The complexity of the local solutions is measured by the description length of the sub-machine in charge of their generation.

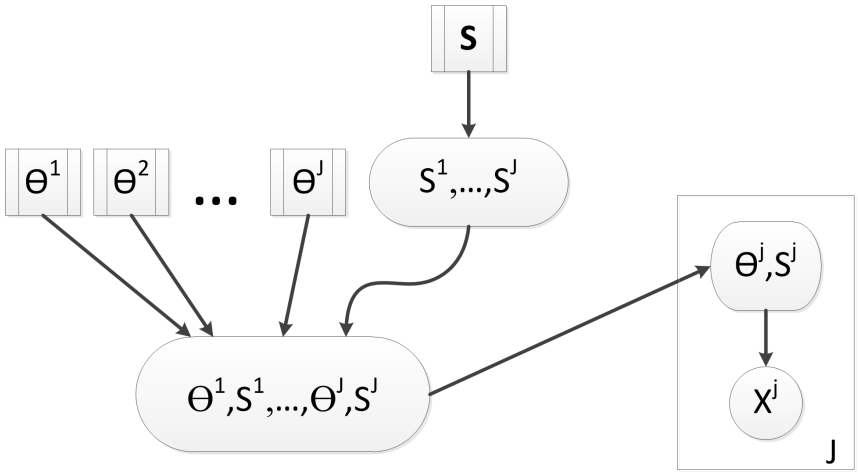


Figure 1.7: Graphical representation of the generative Turing Machine. A rounded box designates a sub-machine generating the object; a squared box designates an input; an arrow designates machine composition (the output of one machine used as input for the other machine. The plate indexed by J indicates J independent replications as for probabilistic graphical models.

A splitting operation is needed to transform the output $\langle \theta^1, S^1, \ldots, \theta^J, S^J \rangle$ of the global parameter machine into the inputs $\langle \theta^j, S^j \rangle$ of the local sub-machines. Since we use prefix codes and the index j of the parameters θ^j and S^j is explicitly given onto the tape of the global sub-machine, the complexity of this splitting operation is a constant which does not depend on the data nor the the parameters.

The key of collaboration lies in the construction of the local solutions $\langle S^1, \ldots, S^J \rangle$. This construction relies on a global unknown solution S which might be interpreted as a consensus. The nature of parameter S will be discussed later: In this section, we only consider it as a global parameter used for the construction of local solutions. For each view j, a sub-machine computes S^j from the global solution S. The length of this sub-machine is $\sum_{j=1}^{J} K(S^j|S)$. Designing the index j counts as a constant in the complexity and thus is not indicated.

1.4.1.5 Complexity of a machine

The architecture of the described machine is summed up in Figure 1.7. The machines described by such a schema constitute a parametric machine class given with parameters $\theta^1, S^1, \ldots, \theta^J, S^J, S$. The length of a machine in this class, up to an additive constant, is given by:

$$l(\mathcal{M}) = K(S) + \sum_{j=1}^{J} K(X^j|S^j, \theta^j) + K(S^j|S) + K(\theta^j) \tag{1.34}$$

Minimum Description Length principle states that the model chosen to describe data is associated to the machine of minimal length. As a consequence, the problem of interest for multi-source clustering in the proposed framework is the following:

$$\underset{\theta^1, S^1, \ldots, \theta^J, S^J, S}{\text{minimize}} \quad l(\mathcal{M}_{\theta^1, S^1, \ldots, \theta^J, S^J, S}) \tag{1.35}$$

where l is given in Equation 1.34 and $\mathcal{M}_{\theta^1, S^1, \ldots, \theta^J, S^J, S}$ designates the Turing machine in the restricted class with indicated parameters.

This minimization problem presents interesting properties: the first one is the genericity of the

formula in Equation 1.34 which has the exact same form as state of the art methods for multi-source clustering [33]. It can be divided into a local term, corresponding to the description of local views individually, and a collaborative term, measuring the inter-view interaction. The collaboration is done at the solution level, since a collaborative description of data would be too complex and would be extremely sensitive to noise, and a collaborative description of parameters θ^j would be too complex in case of heterogeneous nature of algorithms. Unlike state of the art algorithms in collaborative clustering, our method allows collaboration between algorithms of any nature and not between algorithms of a same class while considering both local and global properties.

Another interesting property of this framework is its neutrality toward the question of the consensus of the views. As discussed in the introduction, two trends emerge in multi-view clustering: On the one hand, *unsupervised ensemble learning* aims to converge to a single global solution by comparing local solutions; on the other hand, *collaborative clustering* focuses on refining the quality of local views by exploiting properties of other views. The presented framework performs equally on both tasks: the global solution S offers a consensus while the local solutions S^j correspond to refined local solutions. Depending on the context, our method can be used for both tasks, which is particularly interesting. We will see in the next subsection how the optimization can be done for both tasks.

As a final remark, we would like to insist on the *reverse* approach offered by our framework. Instead of using the available data to infer a model, we propose to use a model to generate the data. In a way, this approach is very similar to the point of view of generative graphical models.

1.4.2 Application to collaborative clustering

In this section, we explain how we optimize the objective function that we described in in Equation 1.35. In the scope of this work, we consider only the case where the solutions $S^1, \cdots, S^J$ produced by the algorithms are hard partitions. Furthermore, we focus on the case of collaborative clustering, in the sense that even if it is possible with this framework, we seek to optimize local partitions in each view rather than finding a consensus solution.

In the optimization process, the complexity $K(S^j|S)$ can be upper-bounded by $\min_{i\neq j} K(S^j|S^i)$ since the S^i are admissible values for S. With this upper-bound, the solution S is not needed any longer and can be eliminated from the problem. It is important to note at this point that this change is a purely mathematical trick and has no real foundation in terms of Turing machine description: in this setting, a local solution would be constructed from another local solution, but loops are not prohibited, which is not possible from a physical point of view.

Designing a collaborative algorithm based on the $\min_{i\neq j} K(S^j|S^i)$ upper-bound is possible, but the evaluation of the minimum value requires a comparison of all possible local solutions, which would be extremely costly. We propose to circumvent the problem by considering that the minimal value of complexity is upper-bounded by the average value of relative complexity:

$$K(S^j|S) \leq \min_{i\neq j} K(S^j|S^i) \leq \frac{1}{J-1}\sum_{j\neq i} K(S^j|S^i) \tag{1.36}$$

This simplification is coherent with the general objective of state-of-the-art methods in which the collaborative part corresponds to an average consensus measure between local solutions.

From Equation (1.34), the function to optimize therefore becomes:

$$S^* = \underset{S}{\mathbf{argmin}} \sum_{j=1}^{J} K(X^j|S^j, \theta^j) + \frac{1}{J-1} \sum_{i \neq j} K(S^j|S^i) \tag{1.37}$$

1.4.2.1 Global approach

Following the model of other collaborative and multi-view algorithms, the optimization of Equation (1.37) is done in 2 steps [35][49]:

- A **local step** during which each algorithm $\mathcal{A}^j$ processes its local view X^j and produces a first model $M^j = \langle \theta^j, S^j \rangle$ based only on the local information. These local models are used as initial values.
- A **global step** during which Equation (1.35) is optimized.

The key difficulty of the algorithm lies therefore in the global step, and in particular in the estimation of the complexity $K(S^i|S^j)$. This term is evaluated by defining a generic Turing machine which transforms a solution vector into another solution vector. The most direct idea for such a machine is to build a naive mapping from S^i to S^j. In general, such a mapping does not have any noticeable property: in particular, it is neither injective nor surjective. We propose to encode the mapping as a key-value set $\langle (1, \mathcal{R}_{j,i}(1)), \ldots, (K_j, \mathcal{R}_{j,i}(K^j)) \rangle$ (where K^j denotes the number of clusters for algorithm $\mathcal{A}^j$). The function $\mathcal{R}_{j,i}$ is called a rule and associates each cluster index of $\mathcal{A}^j$ into a cluster index of $\mathcal{A}^i$. Such a mapping is often not sufficient to offer a full description of a transformation from one solution into another: Some exceptions have to be added to describe the exact transformation. An exception is encoded as a tuple $(n, k^i) \in \{1, \ldots, N\} \times K^i$ where n is the data index, k^i the cluster index, and N the size of the dataset. An exception overwrites the transformation rule. An example of such rule mapping and their exceptions with 3 views is shown in Figure 1.8.

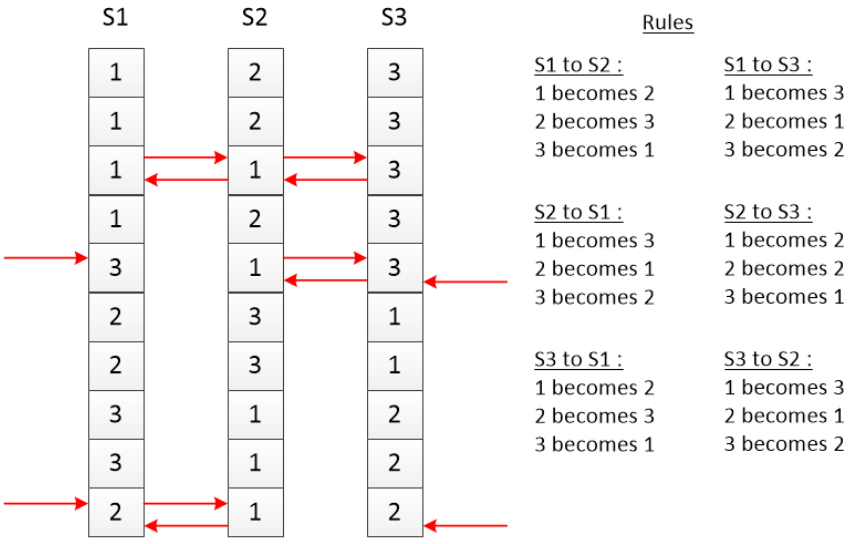


Figure 1.8: Examples of majority rules (on the right) and potential errors to correct (in red)

Using this language of rules and exceptions, we can evaluate the complexity $K(S^i|S^j)$ by measuring the length of the corresponding machine, hence the sum of the complexity of rules and

the complexity of exceptions, each of them being defined as the sum of the individual complexities of their components. The complexity of rules is then $K(\mathcal{R}_{j,i}) = K(k^j) + K(k^i)$ (cluster k^j is transformed into cluster k^i, or in pseudo-code: `if cluster ==` k^j`: return` k^i) and the complexity of an exception $K(e) = K(n) + K(k^i)$ (n-th point is in cluster k^i, or `if point == n: return` k^i). We choose to encode all elements of a same set with the same number of bits. Any element of a set of p elements can be encoded on a prefix-machine with $K(p) \leq \log p + c$ bits [72] where c is a constant. In practice, we do not take the constant into account, since we are only interested in variations of complexity.

Consequently we choose a machine defined in such a way that the description length $K(S^i|S^j)$ is equal to:

$$K^j \times \left(\log K^j + \log K^i\right) + |\mathcal{E}_{j,i}| \times \left(\log N + \log K^i\right) \tag{1.38}$$

where $|\mathcal{E}_{j,i}|$ corresponds to the number of exceptions in the mapping.

In order to define the mapping in practice, we consider the confusion matrix $\Omega^{i,j}$ (the same as in Equation (1.17), but in raw count instead of percentage) that maps the clusters of S^i to the clusters of S^j:

$$\Omega^{i,j} = \begin{pmatrix} \omega_{1,1}^{i,j} & \cdots & \omega_{1,K_j}^{i,j} \\ \vdots & \ddots & \vdots \\ \omega_{K_i,1}^{i,j} & \cdots & \omega_{K_i,K_j}^{i,j} \end{pmatrix} \text{ where } \omega_{a,b}^{i,j} = |S_a^i \cap S_b^j| \tag{1.39}$$

where K^j is the number of clusters considered by algorithm $\mathcal{A}^j$. From there an *argmax* on each line of $\mathbf{\Omega^{i,j}}$ in Equation 1.39 gives us the majority mapping rule for each cluster of $\mathcal{A}^i$ into a cluster of $\mathcal{A}^j$. Using this method, a compression is obtained by defining a general mapping transforming all labels of S^i into labels of S^j and correcting the errors afterwards. The time complexity to compute all the rules between all solutions vectors using this method is in $\mathcal{O}(N)$ for solutions vectors of length N.

Given these elements, optimizing Equation 1.37 consists in searching for the error corrections that would have the most positive impact on the collaborative term $\sum_{j \neq i} K(S^i|S^j)$ with a minimal impact on the local term $K(X^i|M^i)$. Corrections that do not improve the collaborative term or have a negative impact are ignored.

1.4.2.2 Description of the algorithm

The local optimization step consists in a parallel run of all local clustering algorithms. Because there is no collaboration in the local term in Equation 1.35, algorithms can run without any interaction. We notice that we do not aim to minimize the expression of complexity directly, but we use standard algorithms instead: The clustering algorithms are seen as research biases for the minimization of complexity.

The initial solution mapping involves a one-by-one pairing of solutions. The algorithm determines the rules by selecting the maximal cluster associations based on the confusion matrix (as explained in the previous section and in Equation 1.39). The time complexity of this step is $\mathcal{O}(N \times J^2)$. Afterwards, exceptions can be obtained easily (in linear time complexity).

The complete algorithm is detailed in Algorithm 4.

The mapping optimization step is the most complex step, but it is based on a very simple idea: It consists in searching for the error correction which would have the most significant impact

Algorithm 4: SolutionMapping

Input: A set of J clustering solutions S
Output: A set of rules $\{\mathcal{R}_{j,i}\}_{1\leq i,j\leq J}$ and exceptions $\{\mathcal{E}\}_{1\leq i,j\leq J}$
for $i = 1 \dots J$ **do**
  for $j = 1 \dots J$ **do**
    Compute $\Omega^{i,j}$
    for $k = 1 \dots K^i$ **do**
      $\mathcal{R}_{j,i}[k] \leftarrow \arg\max_l \mathbf{\Omega^{i,j}_{k,l}}$
    for $n = 1 \dots N$ **do**
      if $\mathcal{R}_{j,i}\left[S^j[n]\right] \neq S^i[n]$ **then** $\mathcal{E}_{j,i}[n] \leftarrow S^i[n]$
return $\{\mathcal{R}_{j,i}\}_{1\leq i,j\leq J}, \{\mathcal{E}_{j,i}\}_{1\leq i,j\leq J}$

on the collaborative term $\sum_{i\neq j} K(S^j|S^i)$ with a minimal impact on the local term $K(X^j|S^j,\theta^i)$. Correction that do not improve the collaborative term or have a negative impact are ignored.

In other word, it consists in removing exceptions one by one from the set $\{\mathcal{E}_{j,i}\}_{1\leq i,j\leq J}$. Removing an exception results in a single change inside a clustering solution. The system decides to remove an exception if this deletion leads to a reduction in complexity. Because a deletion modifies the solutions, the deletion order has importance in this algorithm. This issue is also encountered in the SAMARAH method [40, 78], a multi-view clustering method that aims at merging clustering partition.

Thus, the naive algorithm cannot be used here.The key idea we rely on in order to solve the problem is the independence hypothesis of the data points. Considering that all data points are described independently, the mapping optimization step can be done on all data points in parallel. It consists in removing exceptions one by one until no exception removal makes the complexity decrease. A recursive approach has been chosen to determine a solution for one data with fixed rules. The proposed algorithm, exposed in Algorithm 5.

Algorithm 5: Mapping Optimization

Input: solution vector for one point: $s = \{s^1, \cdots, s^J\}$, Rules $(\mathcal{R}_{i,j})_{i,j}$
Output: corrected vector s, complexity K
$\mathcal{E} \leftarrow \{\}$
for $j = 1 \cdots J$ **do**
  for $i = 1 \cdots J$ **do**
    if $s^i \neq \mathcal{R}_{j,i}(j)$ **then** $\mathcal{E} \leftarrow \mathcal{E} \cup \{(j,i)\}$
K $\leftarrow$ ComputeComplexity(s)
for $(j,i) \in \mathcal{E}$ **do**
  $s' \leftarrow s$
  $s'^i \leftarrow \mathcal{R}_{j,i}(j)$
  $s', K' \leftarrow$ Mapping Optimization(s,$(\mathcal{R}_{i,j})_{i,j}$)
  if $K < K'$ **then** $s \leftarrow s'$
  $K \leftarrow K'$
return s,K

The proposed algorithm removes exceptions one by one in a backtracking process. The advantage

of backtracking is that it gives an exact solution. Besides, in case two solutions have the same complexity, the solution with minimal depth in the backtracking tree is selected.

At each step, the algorithm has access to a finite list of exceptions and removes the bad exceptions: from one step to another, the complexity can only decrease. Because the number of possible solutions is finite and the total complexity is necessarily non-negative, the algorithm must converge in a finite number of steps. Hence, no stopping criterion has to be given.

It is important to mention that this resolution system for the case of horizontal collaborative clustering completely discards the issue of confidence that we mentioned in the previous sections. Indeed, the equivalent of the view weights in Equation (1.38) is the $\frac{1}{J-1}$ that came naturally as part of the bounding process. As such the views all have the same weights. In a system where the views could have different weights, the mapping algorithm described in 5 would not work anymore, and would have to be replaced by a more complex algorithm. As one can see, this is therefore a strong limitation of this proposal.

1.4.3 Application to clustering fusion

In this subsection, we propose to tackle the problem of clustering fusion in a multi-view context. This work published in [69], re-uses most of the notions introduced in section 1.4.1 about MLD and Kolmogorov complexity.

1.4.3.1 Formalism and problem description

Let us consider a data set $X = \{x_1, \cdots, x_N\}$ of N data points. The Multi-view clustering task considers that the information regarding to each data point in X comes from multiple sources called views. After performing a clustering algorithm over each view several partitions are generated. Let us define this set of partitions as $\mathcal{S} = \{S^1, \cdots, S^J\}$.

A partition S is a set of $|S|$ disjoint sets $\mathbf{c} \in \mathcal{X}$ (the Power set of X) called clusters of the data set X. Let us define an agreement function Ω between two clusters as a mapping $\Omega : \mathcal{X} \times \mathcal{X} \to [0, 1]$ which attains lower values for clusters having a smaller overlap and higher values for clusters sharing more elements of X. In this work we employ the Jaccard similarity function to measure agreement between two clusters.

For a point $\mathsf{p} \in X$, its cluster in any partition $A \in \mathcal{S}$ is denoted by $\mathcal{N}_{\mathsf{p}}^A$ and it is defined as:

$$\mathcal{N}_{\mathsf{p}}^A = \{\mathsf{x} \in \mathcal{X} | \exists \mathbf{c} \in A \wedge \mathsf{p} \in \mathbf{c} \wedge \mathsf{x} \in \mathbf{c}\}$$

Given a cluster $\mathbf{c}$ and a partition B, the function that maps $\mathbf{c}$ to the cluster in B with the largest overlap is called maximum agreement function and it is defined as follows:

$$\Phi_B(\mathbf{c}) = \underset{\mathbf{e} \in B}{\mathbf{argmax}}\, \Omega(\mathbf{c}, \mathbf{e}) \tag{1.40}$$

1.4.3.2 Algorithm description

Our goal is to combine several partitions in order to build a final consensus. To this end, in our method we perform successive pairwise fusion procedures between partitions following a bottom-up strategy until we reach a single partition. This procedure is depicted in Algorithm 6.

Algorithm 6: Main procedure for building the consensus partition.

Input: A set $\mathcal{P}$ of m partitions over the data $\mathcal{X}$.
Output: A consensus partition.
$\mathcal{Q} \leftarrow []$ `/* exceptions after each merge operation */`
while $|\mathcal{P}| > 1$ **do**
  $A, B \leftarrow \underset{A^*, B^* \in \mathcal{P}}{\textbf{argmin}}\, K(A^*|B^*) + K(B^*|A^*)$
  $C \leftarrow \textbf{merge}(A, B, \mathcal{Q}, W)$
  add C into $\mathcal{P}$
  remove A, B from $\mathcal{P}$
`/* Solving points marked in last item from` $\mathcal{Q}$ `*/`
$\xi_D \leftarrow$last partition's exceptions added to $\mathcal{Q}$
foreach $\mathsf{p} \in \xi_D$ **do**
  $\mathcal{N}_{\mathsf{p}}^D \leftarrow \underset{\mathbf{c} \in D}{\textbf{argmax}}\, W_D(\mathsf{p}, \mathbf{c})$
return D

Without loss of generality, when a fusion step is performed between two partitions A and B, a new partition C is created. Since the successive partition fusions are performed by following the maximum agreement criteria between clusters as stated in Equation (1.40), it is possible that some data points do not fit to this rule and hence be marked as exceptions during the execution of the merge operation. The set of data points marked as exceptions before the creation of partition C is denoted by ξ_C, formally,

$$\xi_C = \{p \in \mathcal{X} | \mathcal{N}_{\mathsf{p}}^A \cap \Phi_B(\mathcal{N}_{\mathsf{p}}^A) = \emptyset \cup \mathcal{N}_{\mathsf{p}}^B \cap \Phi_B(\mathcal{N}_{\mathsf{p}}^B) = \emptyset\} \tag{1.41}$$

Please note that the exceptions as they are defined in Equation (1.41) are the same set of exceptions as the one computed for the collaborative approach, and they can be computed using the same Algorithm 4.

Then, when partition C is created, each point $\mathsf{p} \in \xi_C$ receives a weight $W_C(\mathsf{p}, \mathbf{c})$ for every cluster $\mathbf{c} \in C$. This weight is made up by the relative weights that both source partitions A and B contribute, namely $\omega_A(\mathsf{p}, \mathbf{c})$ and $\omega_B(\mathsf{p}, \mathbf{c})$. Without loss of generality, the contribution of each source partition is given by:

$$\omega_A(\mathsf{p}, \mathbf{c}) = \begin{cases} \Omega(\mathbf{c}, \mathcal{N}_{\mathsf{p}}^A) & \text{if } \mathsf{p} \notin \xi_A \\ \Omega(\mathbf{c}, \Phi_A(\mathbf{c})) \cdot W_A(\mathsf{p}, \Phi_A(\mathbf{c})) & \text{if } \mathsf{p} \in \xi_A \end{cases} \tag{1.42}$$

Thus, the final weight $W_C(\mathsf{p}, \mathbf{c})$ for each point $\mathsf{p} \in \xi_C$ in each cluster $\mathbf{c} \in C$ is given by:

$$W_C(\mathsf{p}, \mathbf{c}) = \frac{\omega_A(\mathsf{p}, \mathbf{c})}{2} + \frac{\omega_B(\mathsf{p}, \mathbf{c})}{2} \tag{1.43}$$

A more detailed description of this merging process is depicted in Algorithm 7. It is important to indicate that once a point is marked as an exception, it remains so through all the subsequent fusions. After the last fusion, each of these exception data points are assigned to one of the final clusters by picking the one whose membership weight is the highest. This exception resolution is described between lines $7-9$ in Algorithm 6 where $K(A|B)$ is the simplified writing of $K(S^A|S^B)$, the Kolmogorov complexity of partition A knowing partition B as defined in Equation (1.38).

Algorithm 7: Merge procedure that fuses two partitions into a new one identifying also problematic points as exceptions.

Input: Partitions $A, B \in \mathcal{P}$ s.t. $|A| > |B|$
, list with previous merge exceptions $\mathcal{Q}$ and weight function for previously created partitions W
Output: New partition C and a set of marked points along with their scores $\forall \mathbf{c} \in C$.

$\mathcal{M} \leftarrow []$
foreach $\mathbf{a} \in A$ **do**
  add $\Phi_B(\mathbf{a})$ into $\mathcal{M}[\mathbf{a}]$
foreach $\mathbf{b} \in B$ **do**
  `/* b can be associated to more than one cluster in A */`
  add $\mathbf{b}$ into $\mathcal{M}[\Phi_A(\mathbf{b})]$
$C \leftarrow \emptyset$ `/* The new partition to be returned */`
foreach $\mathbf{a} \in A$ **do**
  $\mathbf{c} \leftarrow \emptyset$
  foreach $\mathbf{b} \in \mathcal{M}[\mathbf{a}]$ **do**
    $\mathbf{c} \leftarrow \mathbf{c} \cup (\mathbf{a} \cap \mathbf{b})$
    $\mathbf{a}' \leftarrow \mathbf{a}$
    $\mathbf{a} \leftarrow \mathbf{a} - \mathbf{b}$ `/* updates cluster a */`
    $\mathbf{b} \leftarrow \mathbf{b} - \mathbf{a}'$ `/* updates cluster b */`
  add $\mathbf{c}$ into C
`/* generating the list of marked points by the current fusion */`
$\xi_C \leftarrow \emptyset$
foreach $\mathbf{a} \in A$ **do**
  if $|\mathbf{a}| > 0$ **then**
    add each $\mathsf{p} \in \mathbf{a}$ into ξ_C
  foreach $\mathbf{b} \in \mathcal{M}[\mathbf{a}]$ *s.t.* $|\mathbf{b}| > 0$ **do**
    add each $\mathsf{p} \in \mathbf{b}$ into ξ_C
add ξ_C into $\mathcal{Q}$
foreach $\mathsf{p} \in \xi_C$ *and* $\mathbf{c} \in C$ **do**
  $W_C(\mathsf{p}, \mathbf{c}) = \frac{\omega_A(\mathsf{p}, \mathbf{c})}{2} + \frac{\omega_B(\mathsf{p}, \mathbf{c})}{2}$
return C

1.4.4 Conclusions on the use of Kolmogorov complexity as a universal multi-view clustering tool

In this section, we have presented a new perspective on the problem that is multi-view clustering. Inspired by algorithmic information theory, we reduced the problem to a model selection over a well-defined set of Turing machines. Compared to state of the art methods, our methodology is based on a well-known theoretical background and does not rely on arbitrary heuristics to define the objective function to optimize. This makes our model an ideal proposal compatible with any type of clustering algorithm: In its collaborative version without fusion, it relies on the partitions and the type of clustering model used, that as we have seen can often be modeled in term of Kolmogorov complexity. Our model is even stronger for the multi-view consensus form aiming at merging partitions, where it only relies on the local results and needs absolutely no information on the type of clustering algorithm that was used.

However, this compression approach to multi-view clustering is not without weaknesses: First and as we have already mentioned, it does not consider the issue that some views might be noisy

or simply incompatible. Second, the local machines are not optimal fits for clustering algorithms outside the family of density based, prototype based and distribution based algorithm. Then, we have seen that the fusion problem is not that easy to solve, and an expert eye can probably see in the algorithm than the order in which the clusters and partitions are merged can change the result depending on the view used a a pivot. Last but not least, this approach relies on the idea that the process can be done in two steps: local clustering first, collaboration or merging then. While this is a strength because of the freedom to choose whatever local algorithm we want, it is also a weakness because it makes it very difficult to make any comparison with other state of the art multi-view methods that adopt a global framework that skip the local step altogether and goes directly for the global multi-view clustering (e.g.[79][80][45]).

1.5 Stability analysis of multi-view clustering

In this section, we tackle a more theoretical aspect of multi-view clustering: the question of its stability [81]. Indeed, for regular clustering, stability is an important notion that if we roughly explain it, describes the ability of a clustering algorithm to find consistently the same structures over several samples of the same data. In a way, stability is an unbiased quality measure not of a clustering partition over a data set, but of a clustering algorithm over a data set.

Since it is a key notion in regular clustering, in this section we gibe some early considerations and findings as to how this notion may be extended to the case of multi-view clustering under its different forms. This work done with Pierre-Alexandre Murena and Basarab Matei has not been published in any review or conference yet, and this section is very heavily inspired from section 15.3 of Pierre-Alexandre's PhD Thesis [77] which you should read for more in depth details on aspects of this work that I was not a part of.

1.5.1 Reminders on clustering stability

We begin our analysis by an introduction of the original definition of stability for classical clustering. The notions exposed in this section are introduced in the original work of Ben David et al. [42]. Please note that while there are similarities in the way we define individual clusters, this formalism below is not exactly the same as the one defined in section 1.4.3.1.

Let us consider a data space $\mathbb{X}$ endowed with a probability measure P. If $\mathbb{X}$ is a metric space, let l be its metric. In the following, let $S = \{x_1, \cdots, x_m\}$ be a sample of size m drawn i.i.d. from $(\mathbb{X}, P, \Sigma)$.

A *clustering* $\mathcal{C}$ of a subset $X \subseteq \mathbb{X}$ is a function $\mathcal{C} : X \to \mathbb{N}$ which to any of said subset X associates a solution vector in the form of matching clusters $S = \mathcal{C}(X)$. As one can see, this definition can be linked to the one of *clustering partition* proposed in 1.4.3.1 with the same formalism, and in the present case a clustering is a partitioning of the entire data space and not only of the observed dataset.

Individual *clusters* are defined by:

$$\mathcal{C}_i = \mathcal{C}^{-1}(\{i\}) = \{x \in X; \mathcal{C}(x) = i\} \tag{1.44}$$

Finally, *clustering algorithm* $\mathcal{A}$ is a function that computes a clustering of X for any finite

sample $\mathcal{S} \subseteq X$, so that $\mathcal{A} : X \to \mathcal{C}$.

To define stability, we need to compare different clustering solutions, and therefore to define clustering distance.

Definition 1 (**Clustering distance**) Let $\mathcal{P}$ be a family of probability distributions over some domain $\mathbb{X}$. Let Σ be a family of clusterings of $\mathbb{X}$. A clustering distance is a function $d : \mathcal{P} \times \Sigma \times \Sigma \to [0, 1]$ that for any $P \in \mathcal{P}$ and any clusterings $\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3$ satisfies:

1. $d_P(\mathcal{C}^1, \mathcal{C}^1) = 0$
2. $d_P(\mathcal{C}^1, \mathcal{C}^2) = d_P(\mathcal{C}^2, \mathcal{C}^1)$ (*symmetry*)
3. $d_P(\mathcal{C}^1, \mathcal{C}^3) \leq d_P(\mathcal{C}^1, \mathcal{C}^2) + d_P(\mathcal{C}^2, \mathcal{C}^3)$ (*triangle inequality*)

Please note that clustering distances as we have defined them are not required to satisfy $d_P(\mathcal{C}^1, \mathcal{C}^2) = 0 \Rightarrow \mathcal{C}^1 = \mathcal{C}^2$, which is not true with most clustering distances that are commonly used.

Earlier, we have introduced stability as the ability of a clustering algorithm to find consistently the same structures over several samples of the same data. From there and using the notion of clustering distance defined above, it is possible to formally define clustering stability as follows:

Definition 2 (**Stability of a clustering algorithm**) Let P be a probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a clustering algorithm. The stability of the algorithm $\mathcal{A}$ for the sample of size m with respect to the probability distribution P is:

$$stab(\mathcal{A}, P, m) = \mathop{\mathbb{E}}_{\substack{X_1 \sim P^m \\ X_2 \sim P^m}} [d_P(\mathcal{A}(X_1), \mathcal{A}(X_2))] \tag{1.45}$$

From there, the stability of algorithm $\mathcal{A}$ with respect to the probability distribution P is:

$$stab(\mathcal{A}, P) = \limsup_{m \to \infty} stab(\mathcal{A}, P, m) \tag{1.46}$$

We say that algorithm $\mathcal{A}$ is stable for P, if $stab(\mathcal{A}, P) = 0$.

1.5.2 Stability applied to multi-view clustering

1.5.2.1 Case of consensus algorithms without an intermediate local step

In the case of multi-view algorithms that don't have an intermediate local step (i.e. they produce the final consensus result directly from the different views), all definitions given for regular clustering remain unchanged. The only difference is that we consider an algorithm that starts from a multi-view space instead of a single view one. In other word, we consider that the total space $\mathbb{X}$ can be decomposed into the product $\mathbb{X}^1 \times \cdots \times \mathbb{X}^J$ of J views spaces $\mathbb{X}^j$.

1.5.2.2 Case of horizontal collaborative clustering

In the case of collaborative clustering however, things are different since we do not seek a consensus solution anymore: we are in a multi-view context in which each view tries to improve its own local clustering by exchanging with the other views. Therefore, we need to redefine the notions of

clustering algorithm and clustering partitions in this context, and see if the definition of stability can be adapted.

Since we are still in a multi-view context, we still consider that the total space $\mathbb{X}$ can be decomposed into the product $\mathbb{X}^1 \times \cdots \times \mathbb{X}^J$ of J views spaces $\mathbb{X}^j$.

Definition 3 (**Collaborative clustering**) A collaborative clustering is defined as a combination of local clustering in the following sense: A collaborative clustering $\mathcal{C}$ of the subset $\mathcal{X} \subseteq \mathbb{X}$ is a function $\mathcal{C} : \mathcal{X} \to \mathbb{N}^J$, where the i-th local cluster for view j, denoted $\mathcal{C}_i^j$ is defined as:

$$\mathcal{C}_i^j = \{x \in \mathcal{X}; (\mathcal{C}(x))^j = i\} \subseteq \mathbb{X} \tag{1.47}$$

A collaborative clustering algorithm $\mathcal{A} = \langle \mathcal{A}^1, \cdots, \mathcal{A}^J \rangle$ is a function which computes a collaborative clustering based on local clustering algorithms $\mathcal{C}^j$ on $\mathbb{X}^j$.

Definition 4 (**Collaborative clustering Algorithm**) Let $\mathbb{A}^j$ be the set of clustering algorithms on $\mathbb{X}^j$. Let $\mathcal{C}$ be the set of collaborative clusterings on $\mathcal{X} \subseteq \mathbb{X}$. And let Σ be the set of finite partitions of $\mathcal{X}$. Then, a collaborative clustering algorithm is defined as a mapping $\mathbb{A}^1 \times \cdots \mathbb{A}^J \times \Sigma \to \mathcal{C}$.

If we consider the whole process with the local and the collaborative step, this simplifies into a collaborative clustering algorithm $\mathcal{A}$ being a function $\mathcal{A} : \mathcal{X} \subseteq \mathbb{X} \to \mathcal{C}$.

It is worth mentioning that in general the projection of the clustering obtained by a collaborative algorithm onto one of the views j is distinct from the original clustering result obtained by the local algorithm $\mathcal{A}^j$ for the same view: If $\mathcal{C} = \mathcal{A}(X)$, then in general we have that $\mathcal{C}^j \neq \mathcal{A}^j(X^j)$.

Definition 5 (**Concatenation of local clustering algorithms**) The concatenation of local clustering algorithms $\mathcal{A}^1$ to $\mathcal{A}^J$, denoted by $\bigoplus_{j=1}^J \mathcal{A}^j$ is defined as follows: If $\mathcal{C}$ is the global clustering induced by $\mathcal{A} = \bigoplus_{j=1}^J \mathcal{A}^j$ on a dataset X, then:

$$\forall x \in \mathbb{X}, \forall j \in [1..J], \qquad \mathcal{C}^j(x^j) = \left(\mathcal{A}^j(X^j)\right)(x^j) \tag{1.48}$$

This defines the concatenation of local clustering algorithms as a collaborative algorithm that does nothing, and produces the exact same results as the ones obtain by the local algorithms $\mathcal{A}^j$.

Since $\mathbb{N}^J$ is isomorphic to $\mathbb{N}$, a collaborative clustering can be interpreted as a clustering of $X \subseteq \mathbb{X}$. Consider the isomorphism $\nu_J : \mathbb{N}^J \to \mathbb{N}$, which we will denote ν when the value of J is obvious. Then, the mapping $\nu \circ \mathcal{C}$ is a clustering of $X \subseteq \mathbb{X}$.

Using this equivalence, the notion of clustering distance that we defined previously holds for collaborative clustering.

Proposition 1 *Let $\mathbb{X} = \mathbb{X}^1 \times \cdots \times \mathbb{X}^J$ be a domain, and the d^j clustering distance on $\mathbb{X}^j$. We define the function $d : \mathcal{P} \times \mathcal{S} \times \mathcal{S} \to [0, 1]$ such that $d_P(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{J} \sum_{j=1}^J d_{P_j}^j(\mathcal{C}_1^j, \mathcal{C}_2^j)$. Then d defines a clustering distance on $\mathcal{X}$. We call it the canonical collaborative clustering distance.*

Proof The clustering distance properties follow from the linearity in terms of d^j and from the properties of the local clustering distances.

We now introduce the notion of *novelty*, a desired property of any collaborative clustering algorithm to do more than just concatenating the local solutions. This represents the ability of a collaborative algorithm to produce solutions that could not have been found locally.

Definition 6 (**Collaborative clustering novelty**)

Let P be probability distribution over $\mathcal{X}$. The novelty of the algorithm $\mathcal{A}$ for the sample size m with respect to the probability distribution P is

$$nov(\mathcal{A}, P, m) = \mathop{\mathbb{P}}_{X \sim P^m} \left[\mathcal{A}(X) \neq \bigoplus_{j=1}^{J} \mathcal{A}^j(X^j) \right] \tag{1.49}$$

where $\mathcal{A}(X)$ is the global collaborative or multi-view clustering and $\bigoplus_{j=1}^{J} \mathcal{A}^j(X^j)$ is the concatenation of all local clusterings.

Then, the novelty of algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$nov(\mathcal{A}, P) = \limsup_{m \to \infty} nov(\mathcal{A}, P, m) \tag{1.50}$$

$\mathcal{A}$ satisfies the novelty property for distribution P if $nov(\mathcal{A}, P) > 0$

Yet, while novelty is a desirable property, in collaborative clustering -and in multi-view clustering in general-, there is also a need that the results found a the global level after the collaborative step remain *consistent* with the local data when projected onto the local views. This leads us to the notion of consistency:

Definition 7 (**Collaborative clustering consistency**)

Let P be probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a collaborative clustering algorithm. The consistency of the algorithm $\mathcal{A}$ for the sample size m with respect to the probability distribution P is

$$cons(\mathcal{A}, P, m) = \mathop{\mathbb{E}}_{X \sim P^m} \left[d_P \left(\mathcal{A}(X), \bigoplus_{j=1}^{J} \mathcal{A}^j(X^j) \right) \right] \tag{1.51}$$

The consistency of algorithm $\mathcal{A}$ with respect to the probability distribution P is

$$cons(\mathcal{A}, P) = \limsup_{m \to \infty} cons(\mathcal{A}, P, m) \tag{1.52}$$

Intuitively, consistency measures the distance of the global clustering produced by the collaboration to the clustering produced by concatenation of local algorithms.

Two things can be said about novelty and consistency: The first one is that obviously these notions are very specific to the case of collaborative clustering and unsupervised ensemble learning (as we will see after), as it is obvious that without intermediary local clusterings, these notions simply don't exist. The second thing is that it is noticeable that there is a link between consistency and novelty, and that novelty is actually a particular case of consistency based on the clustering distance defined as follows:

$$\forall \mathcal{C}_1, \mathcal{C}_2, \quad d_P^{\mathbb{I}}(\mathcal{C}_1, \mathcal{C}_2) = \mathbb{I}(\mathcal{C}_1 \neq \mathcal{C}_2) \tag{1.53}$$

It can be verified easily that the function $d_P^{\mathbb{I}}$ is clustering distance.

However, it should be noted that there is no direct link from consistency to novelty. As such, the intuitive idea that a 0-valued consistency implies equal clusterings (hence no novelty) is wrong.

As a consequence, consistent algorithms are not necessarily concatenations.

Proposition 2 *Let P be probability distribution over $\mathcal{X}$. Let d be a clustering distance. Let $\mathcal{A}$ be a collaborative clustering algorithm.*

Then: $cons(\mathcal{A}, P) = 0 \Rightarrow nov(\mathcal{A}, P) = 0$ is incorrect.

Proof Clustering distances do not satisfy $d_P(\mathcal{C}_1, \mathcal{C}_2) = 0 \Rightarrow \mathcal{C}_1 = \mathcal{C}_2$. Lacking this property, the implication is incorrect.

Coming back to the notion of stability, with Proposition 1 of this section, and Proposition 13 of section 15.3 from Pierre-Alexandre's PhD Thesis [77] which states that "*If P has a unique minimizer $\mathcal{C}^*$ for risk R, then any R-minimizing collaborative clustering algorithm which is risk converging is stable on P*", we can see that the notion of stability as it was defined in Definition 2 holds true for collaborative clustering algorithms and can be treated in the same way when it comes to stability analysis.

A first result can be shown about the concatenation of clustering algorithms. Proposition 3 below states that a concatenation of local algorithms is stable provided that the local algorithms are stable.

Proposition 3 *Suppose that the local algorithms $\mathcal{A}^j$ are stable for distance $d^j_{P_j}$. Then the concatenation of local algorithms $\mathcal{A} = \bigoplus_{j=1}^{J} \mathcal{A}^j$ is stable for canonical distances.*

Proof Let X_1 and X_2 be two samples drawn from distribution P. Then we have :

$$d_P(\mathcal{A}(X_1), \mathcal{A}(X_2)) = \frac{1}{J} \sum_{j=1}^{J} d^j_{P_j} \left((\mathcal{A}(X_1))^j, (\mathcal{A}(X_2))^j \right) \tag{1.54}$$

$$= \frac{1}{J} \sum_{j=1}^{J} d^j_{P_j} \left(\mathcal{A}^j(X_1^j), \mathcal{A}^j(X_2^j) \right) \tag{1.55}$$

From the linearity of the expected value, it comes that:

$$stab(\mathcal{A}, P, m) = \frac{1}{J} \sum_{j=1}^{J} stab(\mathcal{A}^j, P^j, m) \tag{1.56}$$

Hence the stability of $\mathcal{A}$.

This result is rather intuitive, since the concatenation corresponds to a collaborative algorithm that does nothing. From this point of view, it is expected that the unmodified results of stable local algorithms will remain stable. More interestingly, using the notion of consistency, the same result can be applied to get a more generic result:

Theorem 1 *Let $\mathcal{A} = \langle \mathcal{A}^1, \cdots, \mathcal{A}^J \rangle$ be a collaborative clustering algorithm. Then the stability of $\mathcal{A}$ relatively to the canonical distance is upper-bounded as follows:*

$$stab(\mathcal{A}, P) \leq cons(\mathcal{A}, P) + \frac{1}{J} \sum_{j=1}^{J} stab(\mathcal{A}^j, P^j) \tag{1.57}$$

Proof Let X_1 and X_2 be two samples drawn from distribution P. Since the canonical distance satisfies the triangular inequality, we have:

$$d_P(\mathcal{A}(X_1), \mathcal{A}(X_2)) \leq d_P\left(\mathcal{A}(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1)\right) \tag{1.58}$$

$$+ d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2)\right) \tag{1.59}$$

$$+ d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2), \mathcal{A}(X_2)\right) \tag{1.60}$$

Then, by taking the expected value of this expression, we obtain:

$$stab(\mathcal{A}, P, m) \leq 2 \times \mathbb{E}_{X \sim P^m}\left[d_P\left(\mathcal{A}(X), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X)\right)\right] \tag{1.61}$$

$$+ \mathbb{E}_{X_1, X_2 \sim P^m}\left[d_P\left(\left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_1), \left(\bigoplus_{j=1}^{J} \mathcal{A}^j\right)(X_2)\right)\right] \tag{1.62}$$

which is the result we wanted.

This result has the advantage of being generic since it makes no assumption on the nature of the collaboration process. It also has the direct consequence that any consistent collaborative algorithm working from stable local results is stable for the canonical distance. However, this corollary is quite limited since the consistency assumption is extremely strong and does not apply to most practical cases where the collaborative process is expected to find results that differ from the simple concatenation of the local results from each views.

1.5.2.3 Case of unsupervised ensemble learning

Finally, we move to the case of unsupervised ensemble learning where several clusterings are to be merged into a single consensus clustering. It is obvious from this definition that we have a hybrid situation between the case of consensus clustering without intermediate results and the case of horizontal collaborative clustering that we have presented earlier.

Nevertheless, it is easy to see that unsupervised ensemble learning is closer to collaborative clustering with notions such as novelty and consistency being relevant here. As such, all definitions, propositions and theorems defined for collaborative clustering can be applied to the case of unsupervised ensemble methods with the main following differences:

- The total data space considered is $\prod^J \mathbb{X}$, the original data space $\mathbb{X}$ duplicated into the J views.
- Since an unsupervised ensemble method $\mathcal{A}$ produces a single partition, a specific distance function must be defined to compare $\mathcal{A}$ and a concatenation of local results $\bigoplus_{j=1}^{J} \mathcal{A}^j$ as these are not objects of the same nature here. This can easily be done by duplicating the consensus partition J times.

Please note that for the second difference, going the other way by merging the local solutions instead of concatenating them is a bad idea.

1.5.3 Conclusion

The results presented in this section have been done with the goal of presenting a unified theoretical framework from stability in an unsupervised multi-view context. As you can see, this is a very preliminary work that does not yet yields results that can be used in practice.

Nevertheless, it is our hope that as this work advances we may be able to practically analyze the dozens of multi-view algorithms and collaborative framework that exist in the literature so that we can know where they stand regarding the important property that is stability. Indeed, most of classical clustering methods have been deeply analyzed from a theoretical point of view: their convergence properties, complexity, deterministic nature, and stability are known properties that are useful to pick a clustering algorithm. However, many of these properties still remain barely scratched from most of what has been proposed in multi-view clustering, especially for horizontal collaborative clustering and the many available methods for clustering fusion and unsupervised cluster ensembles.

1.6 Chapter Conclusion

In this chapter, I have presented 5 years of work on clustering techniques in multi-view environments after my PhD thesis. During these years, through several collaborations and with the help of PhD students, several aspects of this field have been tackled.

First we have tackled the issue of confidence in multi-view clustering. We have used several approaches all with the pros and cons. The purely mathematical approach using KKT optimization [47] was effective in the sense that it favors the stability of the structure found. But we saw it also completely undermines the principle of multi-view learning by discouraging the search for novel results that couldn't be found locally. Furthermore, some of our subsequent works [53] have empirically confirmed that this approach based solely on the notion of inter-view partition similarity was not the most effective one. The non-stochastic multi-armed bandit approach [57] on the other hand has the advantage of allowing a lot more exploration and take into consideration the potentially changing usefulness of the views through the learning process. The main inconvenient of this second approach was its complexity and difficulty to scale with a large number of view.

We tackled the issue of view confidence and usefulness from another angle with the deep cooperative reconstruction system [60]. With this algorithm we approached the problem of reconstructing missing data based on information from other views. By doing so, we saw that it is possible to optimize the weight given to each view in deep learning processes simply by using gradient back-propagation. The results for the reconstructed data were not always as impressive as we had hoped. But it is worth mentioning that all reconstructed data led to good scores for subsequent classification tasks, thus highlighting that despite average reconstructions the latent representations were most likely correct.

We have then presented an information theory based approach for multi-view clustering [68]. It relies on the notion of Kolmogorov complexity and clustering being a compression task. Using this principle, we have shown how the Kolmogorov complexity of given partitions can be computed using

different formulations depending on the family of clustering algorithm that was used to produce these results. By using such representations and combining it with different error resolutions algorithms, we have shown how this idea can be efficiently used for both multi-view clustering and unsupervised ensemble learning (clustering fusion).

Finally, we worked on the extension of the notion of clustering stability to multi-view applications. As you have seen, these are still preliminary results. Regardless, it is my strong opinion that this work is important, as it should help many people working in multi-view clustering to know what are the properties of their algorithms: are they stable ? Can they produce novel results compared with the local views ? And many other questions.

Progresses have been made, and I hope that all these works can be useful to other people in the community. Nevertheless, it is easy to see that there is still a lot to do and that scholars working on clustering are indeed adventurers: multi-view or not, clustering remains a difficult machine learning specialty where everything is ill-defined (when not downright quirky), where results are difficult to evaluate and compare, and where the simple problem of picking a clustering algorithm for a new task is difficult.

Within this context, it seems to me that -while I hope that all contribution of this manuscript will be useful,- the continuation of our work on stability is something that must be done as this might allow to lift the fog that surrounds multi-view unsupervised learning by providing some extra clues at the properties of algorithms available in the literature.

Chapter 2

Conciliating powerful, but data hungry algorithms with applications where labeled data are scarce: An attempt at Deep Learning in unsupervised environments

"Deep Learning makes no sense for unsupervised applications. You should definitely work on fuzzy quantum topological collaborative clustering through optimal transport. I foresee promising applications and great findings!"

A visionary colleague (2017)

Contents

2.1 Chapter Introduction

2.1.1 Deep learning in unsupervised environments ?

By the end of my PhD thesis, most of my work had been focused on multi-view clustering aspects and I had a somewhat limited experience with applications in the field of remote sensing. At the time, it is fair to say that I was not really knowledgeable about state of the art algorithms for image processing. And yet, I had a real interest for this field, especially with Deep Learning being on the rise: Deep neural networks based on convolutional neural networks [82] such as VGGNet [83], Inception [84] and ResNet [85][86] were all the rage at this time. And in a word where GPU were becoming more common, the deep learning rising tide had already pretty much destroyed everything done by any other family of machine learning algorithms: Medical image analysis for disease detection [87], facial recognition [88], video analysis for autonomous driving [89], gesture recognition [90], handwriting recognition [91], and many more; Deep Learning and neural networks were everywhere and were dominating everything when it came to image or video analysis.

Like with more regular datasets, my modest experience with image analysis had been mostly unsupervised. And all of sudden, my publications about multi-view clustering for satellite image analysis [92][93][49][94] felt small. After all, I was facing the best machine learning algorithms of their time equipped with variants of the K-Means algorithm [18] applied to segments extracted using watersheds [95]. Yet, the more I was reading about these top-notch deep learning methods, the more it became apparent to me that nearly all of these methods were designed for supervised learning tasks and required enormous amounts of carefully chosen and labeled data to produce the great results that we were all so proud of. And so I wondered: What do we do when we don't have labeled data ? Can we do clustering with deep learning algorithms ? Can it find interesting things in images if you don't feed it thousands of examples of what is interesting and what is not ? Except for the lonely W-Net algorithm [96] for unsupervised image segmentation and clustering (and it came out only in 2017), I did not find much answers in the literature. Of course, there were also the autoencoders [63] that had been around for a while, butthey are mostly dimensionality reduction algorithms and don't do clustering or predictions on their own.

To sum up, in one hand I had antiquated clustering algorithms that were not great for image processing, and on the other hand I had the mighty and powerful deep learning algorithms that despite their performances would not work for my unsupervised problems because I had no labeled data to feed them. As for my peers, they had varying views on the subject of unsupervised deep learning ranging from "*It cannot be done*" and "*We people of unsupervised learning are not concerned by your deep networks*", to weird interpretations of Pr. Yann Le Cun's cake analogy on self-supervised learning[1] and that such unsupervised deep madness -if possible- would most certainly result in creating Skynet[2]. I was eager to prove them wrong. And thus began my journey to figure out if it was possible to do some Deep Learning image analysis in unsupervised environments.

[1] `https://medium.com/syncedreview/yann-lecun-cake-analogy-2-0-a361da560dae`
[2] `https://en.wikipedia.org/wiki/Skynet_(Terminator)`

2.1.2 Chapter organization

Unlike the previous chapter in which each section was a contribution or a group of subsequent contributions, the organization of this chapter is different. We follow a thematic organization: Section 2.2 features contributions made for remote sensing applications, and Section 2.3 introduces some more contributions in the field of medical image analysis.

Like for the previous chapter, not all contributions are detailed in this manuscript.

2.2 Time series analysis of satellite images using unsupervised deep learning methods

Most elements presented within this section are part of the work realized with my former PhD student Ekaterina Kalinicheva and can also be found in her PhD thesis manuscript [97]. This work was also part of a broader involvement of ISEP within the *CES Détection des changements génériques*[3] (expert committee on detecting generic changes) led by Professor Pierre Gançarski.

2.2.1 The remote sensing context

2.2.1.1 Introduction to remote sensing

With the development of satellite technologies and the improvements in various space programs, it is now possible to acquire images from pretty much anywhere in the world thanks to the ever increasing number of Earth observation satellite on orbit around Earth. This gave birth to powerful tools for Earth observation which allow the study of any area of interest at any time and without direct physical interaction. It goes without saying that remote sensing as a science advanced in parallel with the progresses in image processing algorithms, including the rise of deep learning algorithms: While only several decades ago, remote sensing image analysis was a difficult, time-consuming and a manual task; nowadays, image processing algorithms suited for image processing make it possible to analyze the image data using computer in no time.

While they can involve either aircraft, drones or satellites, remote sensing images are still most commonly acquired by artificial satellites with various sensors on board. The sensors on board these satellites can be divided into two categories: active and passive. Active instruments use their own source of energy to interact with an object, while passive instruments use the energy emitted from a natural source, in particular, from the sun.

Active sensors usually refer to *Synthetic Aperture Radar* (SAR) that measures the surface roughness. The main idea of this approach is to measure surface back-scattering: the portion of the radar signal that is redirected back by the target.

Passive sensors on the other hand are mostly made of optical sensors that measure the amount of sun energy reflected by the target. It is this type of sensor that is used in this section. From these optical sensors, we exploit different radiation wavelengths or frequencies from the electromagnetic spectrum (some of which match with visible colors). Figure 2.1 shows the electromagnetic spectrum with the corresponding wavelengths.

[3] `https://www.theia-land.fr/ceslist/ces-detection-des-changements-generiques/`

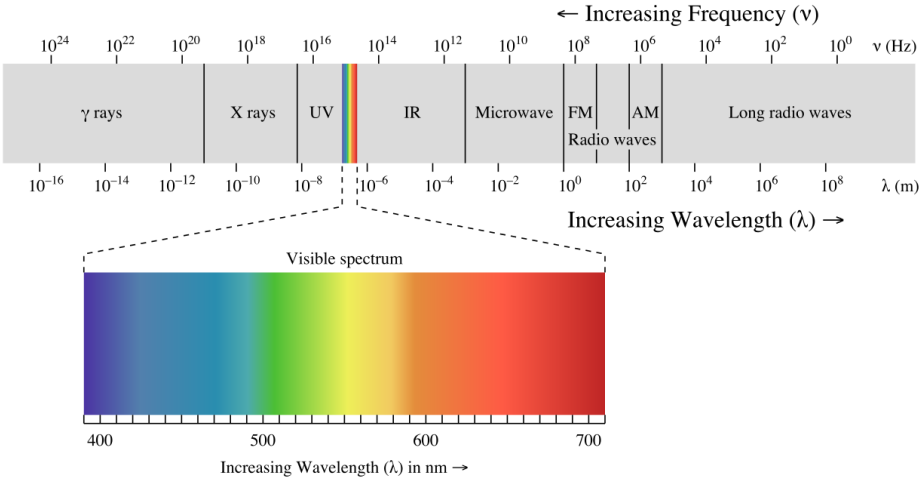


Figure 2.1: Electromagnetic spectrum.

The spectrum is divided in different spectral ranges: We distinguish the visible spectrum (the radiation perceived by human eye), infrared, ultraviolet, etc. The Earth surface reflects different types of radiation, depending on its coverage. Figure 2.2 shows that vegetation absorbs the visible radiation and reflects near-infrared radiation, and it is the same for bare soil. At the same time, water surfaces reflects all the visible wavelengths.

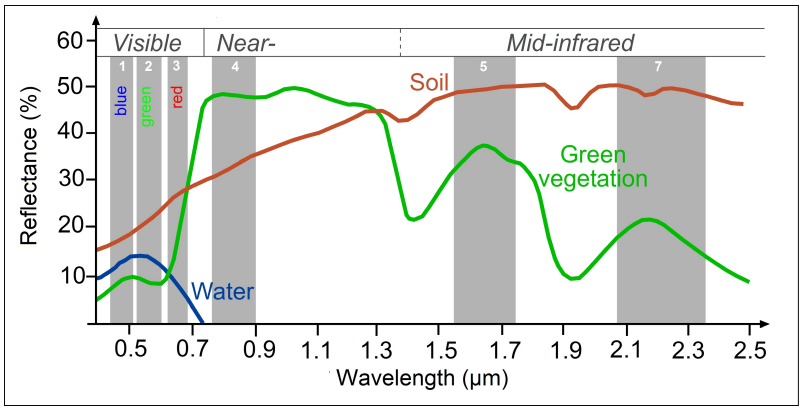


Figure 2.2: Spectral signatures of the water, green vegetation and soil within the different windows of the electromagnetic spectrum.

Every satellite image acquired by an optical sensor is characterized by:

- Its number of spectral bands. By contrast with RGB images, satellite images can have more than 3 bands.
- Its spectral resolution: the spectral width of each spectral band or the capacity of sensor to define fine wavelength intervals for each band.
- A spatial resolution: the pixel size of each band.
- A radiometric resolution: the range of image bits. It reflects the capacity of an instrument to distinguish differences in object reflectance.

We distinguish between mono-, multi- and hyperspectral images: Monospectral images contain a unique spectral band and the image pixels are characterized by a single value. Multispectral images contain from 3 to 10 spectral bands, while for hyperspectral images their number is higher that 10. Their pixels are characterized by vectors of radiometric values from each band. Monospectral images usually contain a single panchromatic band. In the meantime, multispectral images should have at least green, red and near infrared (NIR) bands as they are the most informative.

Finally, it is worth mentioning that Deep Learning has already revolutionized the analysis of remote sensing images in several ways. The most straightforward one is that it makes hand-crafted feature design and selection unnecessary. This is all the more interesting with increasing spectral and temporal resolutions since it allows alleviating the conception of specific attributes between bands or epochs. The second one is that the division of a standard processing pipeline into feature extraction, classification, and regularization steps becomes probably obsolete. Deep Learning algorithms do it all.

2.2.1.2 Challenges of satellite images

Compared with regular RGB images, satellite images present a number of challenges that make them particularly difficult to tackle:

- Multispectral images tend to contain more than 3 channels. Because of this, all algorithms designed for RGB images need to be modified to take into account the extra channels. But these extra-channels also mean that visualizing these images can be tricky and require the use of specific software. This can also make result interpretation more difficult in some cases.
- Satellite images especially recent ones can be huge and reach sizes that make them difficult to handle by image processing algorithms without clipping them or having a lot of RAM and computation power. This and the extra-channels can be a problem for high complexity algorithms.
- They have all sorts of defects and issues: cloud masking part of the images, shadows projected by buildings, optical distortions due to atmospheric conditions, over-saturated pixels due to high reflectance objects, etc. While some of these problems can be solved or diminished in prep-processing, many will remain in the final images.
- Satellite images contain lots of objects, and different scales of objects and areas that may or may not be interesting depending on the application [93]. This makes any segmentation process difficult, especially if it is unsupervised, because the scale of interest has to be defined: Are wee looking for city areas or individual houses ? Are we searching for vegetation areas, or do we want to sort out different types of vegetation ?

Finally, it is worth mentioning that while there is an abundance of available satellite images, very few of them are provided with labels. The reason for that are quite simple and have been discussed above: First these images present many different objects at different scales. And as such, labeling them greatly depend on the application and is a task that often must be done manually and by experts of whatever we are looking to detect. This means a costly and time consuming pre-processing to label these images. Second, due to the variety of landscapes, different image resolutions, and various seasonality effects, data that have been labeled or model that have been

train on given images can rarely be used for other images with slightly different landscapes. As a result, unless you either tackle data from a pre-existing project or you have time, money and people to manually label images and produce a ground-truth, a new project with remote sensing images often means that you won't be able to use supervised learning algorithms. Finally, even when they are available there is always a reliability issues with the produced ground-truth [98], as these are -again- difficult images and experts tend to not always agree as to what should be labeled how.

In this context, having image processing algorithms that work in an unsupervised environment becomes handy.

2.2.1.3 Satellite image time series

Satellite image times series (SITS) are used for numerous applications: the analysis and preservation of the stability of ecosystems, the detection of phenomena such as deforestation and droughts, the study of economical and urban development of cities and agglomerations, the analysis of vegetation states and changes for different agricultural purposes, etc.

All of these applications were made possible thank to the regular acquisition of satellite images of the same areas from programs such as SPOT-5 or Sentinel. However, in addition to the challenges mentioned with individual remote sensing images, time series add their own challenges that can make change detection, time series analysis and time series prediction even more difficult:

- The temporal resolution of satellite image time series can vary greatly from one mission to another, and the time between exploitable images may not be constant (irregular acquisition or simply cloudy images).
- Many areas will have strong seasonality effects which can lead to drastic changes in landscape (vegetation changes for instance, or snow in extreme cases). But some of these seasonal changes can also take the form of a constant cloud cover during several months in tropical areas with a wet season. In a best case scenario, at least the luminosity will vary between images.
- All the artifacts and defects mentioned for single images will still happen but may vary from image to images: shadows and saturated pixels will not be the same depending on the time of the year an image was taken and also depending on the weather conditions.

2.2.2 Detecting non-trivial changes using joint-autoencoders

In this subsection, we tackle the problem of detecting non-trivial changes between two remote sensing images of a time series. This work is extracted from some of our published work [99] and proposes an unsupervised neural network based on auto-encoders that can detect non-trivial changes between two consecutive images of the series.

2.2.2.1 Problematic and State of the art

As we have mentioned in the introduction, satellite image times series are difficult data to tackle. In particular, many applications concerning these time series require to detect specific changes between images. This is however a difficult task because of the need to define what a *"meaningful" or "non-trivial" change* is, and because satellite image time series are known to be riddled with all sort of seasonal changes and defects that we have already mentioned. In our case, our goal is to

detect changes in urban areas (road or building construction for instance), or changes in land cover (e.g.: forest being replaced by crops), all the while ignoring all seasonal changes in the vegetation, as well as changes in luminosity and other artifacts between two images. As such, in this work, we define a *non-trivial change* all the changes in land cover that are not caused by seasonal effects (vegetation changes), lighting issues or artifacts.

Different algorithms for change detection have been proposed in the literature. For example, in [100] the authors use PCA and hybrid classification methods to detect changes in urban areas. On the other hand, in [101] the authors propose a siamese neural network for supervised change detection in open source multi-spectral images. In [102], the authors propose a supervised change detection architecture based on based on U-Nets [103]. Similarly, in [104], the authors propose another and better supervised architectures based on convolutional neural networks (CNN) and that shows very good performance to separate trivial changes from non-trivial ones. As one can see, the main issue with these methods is that they are all supervised and need labeled data.

A few unsupervised methods exist in the literature for change detection, but many of them are primitive (based on image differences and thresholds) and can hardly be applied to high resolution remote sensing images with complex objects. For this reason, in this state of the art, we will only mention the few ones that are related to satellite image analysis. To improve the quality of unsupervised change detection between two images, the fusion of results from different algorithms is often proposed [105]. At the same time, automatic methods for selection of changed and unchanged pixels are used to obtain training samples for a multiple classifier system [106]. Following this paper, the authors of [107] propose the improved backpropagation method of a deep belief network (DBN) for change detection based on automatically selected change labels. In this work, the authors use an RBM-based (Restricted-Boltzmann Machines) model to learn the transformation model for a couple of VHR co-registered images. RBM is a type of stochastic artificial network that learns the distribution of the binary input data. It is considered to be simpler than convolutional and autoencoder-based neural networks, and works very well with Rectified Linear Units activation functions [108].

Nevertheless, classic change detection approaches do not separate trivial (seasonal) changes from non-trivial ones (permanent changes and changes that do not follow seasonal tendency). This weakness can drastically complicate the interpretation of change detection results for regions with high ratio of vegetation areas. In fact, when analyzing two images belonging to different seasons of the year, almost all the area will be marked as change and further analysis will be needed to identify meaningful changes (non-trivial).

In [109], a regularized iteratively weighted multivariate alteration detection (MAD) method for the detection of non-trivial changes is proposed. This method was based on linear transformations between different bands of hyperspectral satellite images and canonical correlation analysis. However, the spectral transformation between multi-temporal bands was too complex. For these reasons, deep-learning algorithms which are known to be able to model non-linear transformations, have proved their efficiency to solve this problem and have been proposed as an improvement of this architecture in [110].

Alternatively, in [111], the authors propose a non-neural network-based, but still unsupervised, approach which relies on following segmented objects through time. This approach is very interesting but remains difficult to apply for cases where the changes are too big from one image to another.

Our approach described in Section 2.2.2.3 re-uses some of the abilities and ideas from the RBM architecture proposed in [110] and updates it by using the advantages of autoencoders (on which we give a few reminders in Section 2.2.2.2 below).

2.2.2.2 Autoencoders

The original autoencoders was introduced by Hinton et al. [63]. The principle of the autoencoder is that this fully unsupervised network take some data or an image as an input, and attempts to reconstruct it at the output. It is a network with fully connected layers (at least in the original version) made of two main parts: An encoder in which the number of nodes is slowly reduced until we reach a bottleneck. And a decoder which is built in a symmetrical fashion to the encoder. The learning is done using simple back-propagation of the difference between the expected output and the predicted output. This clever principle shown in Figure 2.3 has several advantages: The information compression caused by the bottleneck often results in the removal of noise between the original image and the reconstructed one. Robust features can often be extracted at the bottleneck of the network, features which can then be used for clustering or classification. And finally, and this is important for our applications, autoencoders are particularly good at learning textures.

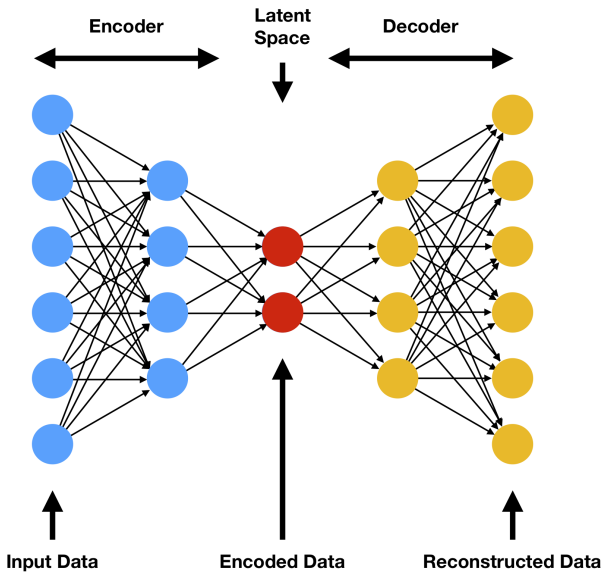


Figure 2.3: Basic architecture of an autoencoder made of an encoder going from the input layer to the bottleneck and the decoder from the bottleneck to the output layers.

Among the different neural network models, autoencoders have found application in many domains. In image processing, autoencoders are widely used for image segmentation [112][103], image compression [113], image reconstruction [114], for feature extraction [115, 116] and clustering [117][118].

2.2.2.3 Proposed architecture for non-trivial change detection

For our non-trivial change detection problem, we take advantage of the autoencoder ability to learn robust features and to map textures. Our proposal is the following: We will train a joint-autoencoder

to predict an image Im_{n+1} based on the image Im_n, (and Im_n based on Im_{n+1} in the other direction), see Figure 2.4. The autoencoder will learn all seasonal changes and global changes in luminosity from one image to another simply by mapping the textures since autoencoders are good at it. Some of the artifacts should also be removed by its denoising ability. And from there, we simply exploit its inability to do anything much than mapping common textures so that it won't be able to map and reconstruct all of the non-trivial changes that we are interested in.

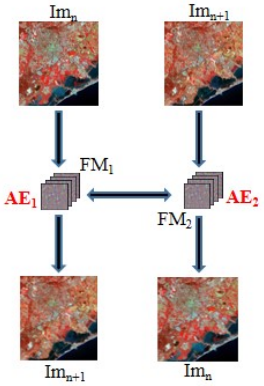


Figure 2.4: Joint Autoencoder

In other words, with use both the strength of the autoencoder to map texture so that we can get rid of seasonal changes (in the vegetation for instance) and we use the weakness that it can't predict the future so that the non-trivial changes will show up in places the autoencoder has a high reconstruction error.

It is worth mentioning that many approaches rely on the analysis of image differences. We tried this approach with regular autoencoders [119], and while we achieved some results, it was overall less good than the joint approach from the original images.

The full architecture of our model is shown in Figure 2.5. The detailed steps of the algorithm are the following:

1. We start with a **pre-training phase** where the autoencoder is not joint yet: a single autoendoer is pre-trained based on patches of Im_n and Im_{n+1}. At this step, all patches are used for self-reconstruction and there is no reconstruction from one image to the other yet. The goal of this step is to pre-train the network with the various textures.

2. The auto-encoder pre-trained during the previous phase is duplicated and joint as shown in Figure 2.4. This is the **fine-tuning phase** where Im_n tries to predict Im_{n+1} and vice-versa.

3. Once the network is trained comes the **reconstruction phase**: From Im_n the autoencoder builds Im'_{n+1} and from Im'_{n+1} we get Im'_n.

4. The reconstruction error (RE) map is computed between each image and the autoencoder prediction. Then, an average error map is built from the reconstruction error of Im'_n and Im'_{n+1}.

5. Finally, we apply Otsu thresholding [120] to build the final map of non-trivial changes.

For a time series of size S, the training in step 1 should be done on the full series and not only each pair of images, and steps 2 to 5 should be repeated for all pairs of images so that all binary change maps of non-trivial changes can be built.

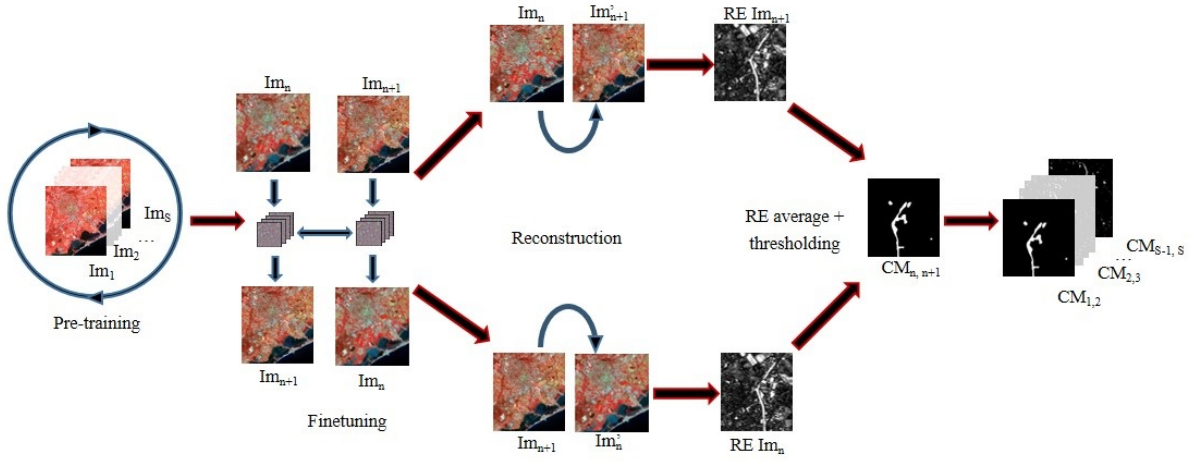


Figure 2.5: Full architecture: unsupervised detection of non-trivial changes.

Different variant of this architecture presented in the table below have been used in several of our publications [121][99][122] for change detection or for the clustering of the non-trivial changes we detected. For all architectures, the Mean square error (MSE) of the reconstruction is used for the back-propagation process. All convolution layers, except for the last layer of the decoder part, have batch normalization. The most common parameters for all convolutional layers were: kernel size=3, stride=1, padding=1. The patch size p^2 may vary depending on the resolution of the image.

AE type	Fully-conv. AE	Conv. AE	Conv. AE + DEC
encoder	Conv(B, 32)+ReLU Conv(32, 32)+ReLU Conv(32, 64)+ReLU Conv(32, 64)+ℓ_2	Conv(B, 32)+ReLU Conv(32, 32)+ReLU Conv(32, 64)+ReLU Conv(64, 64)+ReLU Lin($64 \times p^2$, $12 \times p^2$)+ReLU Lin($12 \times p^2$, $2 \times p^2$)+ℓ_2	Conv(B, 32)+ReLU Conv(32, 32)+ReLU Conv(32, 64)+ReLU Conv(64, 64)+ReLU Maxpool(p) Lin($64 \times p^2$, $32 \times p^2$)+ReLU Lin($32 \times p^2$, $4 \times p^2$)+ℓ_2
decoder	Conv(64, 64)+ReLU Conv(64, 32)+ReLU Conv(32, 32)+ReLU Conv(32, B)+Sigmoid	Lin($2 \times p^2$, $12 \times p^2$)+ReLU Lin($12 \times p^2$, $64 \times p^2$)+ReLU Conv(64, 64)+ReLU Conv(64, 32)+ReLU Conv(32, 32)+ReLU Conv(32, B)+Sigmoid	Lin($4 \times p^2$, $32 \times p^2$)+ReLU Lin($32 \times p^2$, $64 \times p^2$)+ReLU Unpooling(p) Conv(64, 64)+ReLU Conv(64, 32)+ReLU Conv(32, 32)+ReLU Conv(32, B)+ reLU

Table 2.1: Model architectures with B the number of bands and p the patch size. Models 1 and 2 are for change detection, and model 3 is for clustering.

Our experiments applied to SPOT-5 images of the French city of Montpellier have shown fully convolutional joint-autoencoders (1st column of Table 2.1) and simple convolutional joint-autoencoders (2nd column of the same table) to have overall similar performances, with a slight

performance edge for simple convolutional joint-autencoder due to its higher number of parameters, and better training times for the fully convolutional model.

2.2.3 Case study of the 2011 Tohoku tsunami

In this subsection, we present a case study of the previously presented change detection method applied to the case of the 2011 Tohoku tsunami [122].

If you are interested, there are other applications of more or less advanced machine learning to damage surveys of geohazards [123][124][125][126], however many of them heavily rely on supervised learning or the intervention of human experts as part of the active process of image analysis.

A broad review of remote sensing based approaches for damage assessment after the Tohoku tsunami is available in [127]. And another quite exhaustive review focusing on Machine Learning and artificiel intelligence methods applied to this particular disaster is proposed in [128].

2.2.3.1 Dataset

The 2011 earthquake off the Pacific coast of Tohoku was the result of a magnitude 9.1 undersea mega-thrust earthquake that occurred on Friday March 11th of 2011 at 2:46 p.m. local time (JST). It triggered powerful tsunami waves that may have reached heights of up to 40 m and laid waste to coastal towns of the Tohoku's Iwate Prefecture, traveling up to 5 km inland in the Sendai area [129].

The goal of our case study was to see if our previously proposed change detection deep learning could be used on a real scenario: In this case we wanted to map de damages caused by the Tohoku tsunami. To this end, we used images from the ASTER program. We kept the Near-Infrared, Red and Green bands with a resolution of 15 m.

The optical images we used are from March 19th 2011, November 29th 2010 and July 7th 2010 (Figure 2.6), see Table 2.2 and where chosen for different reasons: The march 19th image was the first available image for this area after the disaster and is cloud free. As for the two other images, the one from November 29th was the first available before the disaster but is quite cloudy, so we had to seek an earlier image hence the image from July 7th 2010.

The correction level of the images is 1C—reflectance of the top of atmosphere. It means that reflectance values are not corrected for the atmospheric effects.

Table 2.2: Images characteristics.

	Date	**Clouds**	**Program**	**Resolution**	$\boldsymbol{H \times W}$**, pixels**
Im_{b1}	24/07/2010	<1%, far from the coast			
Im_{b2}	29/11/2010	≈15%, over the coast	ASTER	15 m	2600×1000
Im_a	19/03/2011	none			

The first difficulty was therefore to choose what to do with the before images: On the one hand the November image with clouds was closer and had the advantage of having similar states of vegetation than the March one. On the other hand, the July image was clearer but had a very different vegetation, dry rivers and a different summer reflectance that may affect the detection of the damages. The details can be seen in Figure 2.7. It seemed to us that both images had their advantages, but that the November should have been our priority in the absence of clouds. Since our algorithm cannot used two "before" images anyways, we decided to combine the two images by

using masks that would replace cloud areas from the November 2010 image by elements of the July 2010 image.

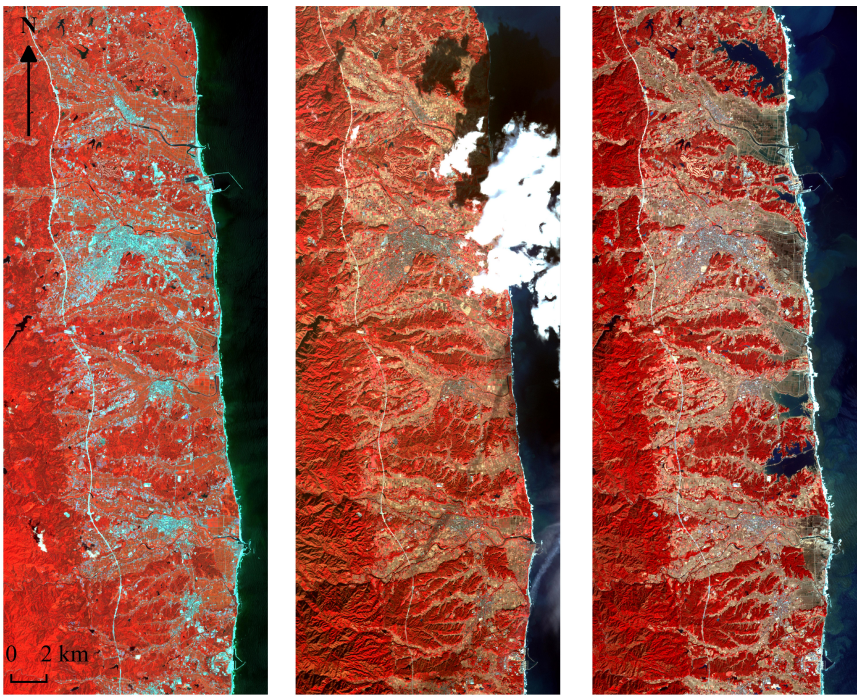


Figure 2.6: Images taken over the damaged area, (**a**) 7 July 2010, (**b**) 29 November 2010, (**c**) 19 March 2011.

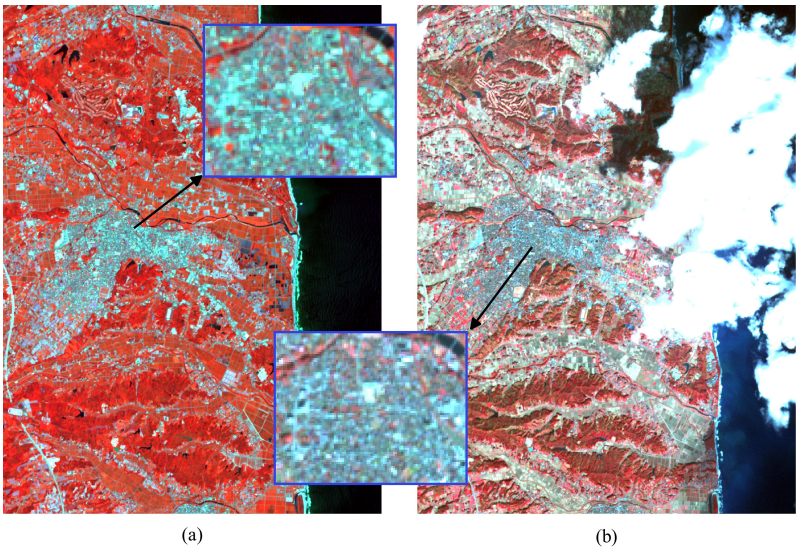


Figure 2.7: ASTER images taken on (**a**) July 2010 and (**b**) November 2010. Image (**a**) was taken in sunny conditions that caused much higher pixel values for urban area pixels (zoomed) than for image (**b**). For example, the value of the same pixel of this area is equal (83, 185, 126) for (**a**) and (37, 63, 81) for (**b**). Moreover, a great part of image (**b**) is covered by clouds and their shadow.

2.2.3.2 Adding clustering on top of change detection

To detect the changes damages caused by the tsunami, we used the same joint-autoencoder principle than the one proposed in the previous section: the goal was to ignore seasonal changes and to detect the damages as non-trivial changes. However, for this application there was an extra step that we were interested in: sorting the damages into different categories, or in other words running a clustering algorithms on the detected non-trivial change areas. This leads us to the architecture process shown in Figure 2.8.

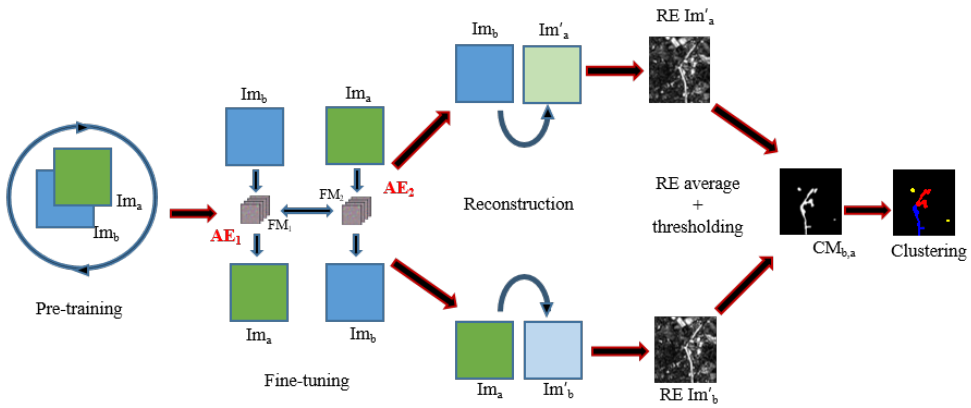


Figure 2.8: Unsupervised detection of non-trivial changes and clustering.

To do so, we used Deep Embedding clustering (DEC) [118], a clustering technique that seeks clusters from the bottleneck nodes of autoencoders. The main steps of the DEC algorithm are the following:

1. Pre-train an AE model to extract robust features from the patches of concatenated images in an embedding space.

2. Initialize the centers of clusters by applying classical K-Means algorithm [18] on extracted features.

3. Continue training the AE model by optimizing the AE model and the position of the centers of clusters, so the last ones are better separated. Perform label update every q iterations.

4. Stop when the convergence threshold t between labels update is reached (usually $t = 0.5\%$).

In our case, we used two of the autoencoder architectures described in Table 2.1 that share a common training of their convolutional layers: the joint fully convolutional joint-autoencoder is used for the change detection part (column 1), and the DEC version of the regular autoencoder is used for the clustering part (column 3).

To detect the changes on 15 m resolution ASTER images we use patch size $p = 7$ pixels that was chosen empirically. In the case if images were perfectly aligned, $p = 5$ would be enough, but since we have a relatively important shift in these data, we add margins by using larger patches.

As we have two before images, we pre-train the model on the patches extracted from 3 images with the cloud mask applied (the cloud mask is extracted automatically with the K-Means algorithm

using 2 clusters on the encoded images). Once the model is stabilized, we fine-tune it for 2 couples of images Im_{b1} / Im_a and Im_{b2} / Im_a and we calculate the RE for both couples in order to produce change maps $CM_{b1,a}$ and $CM_{b2,a}$. We replace the masked part of $CM_{b2,a}$ by $CM_{b1,a}$ to obtain the final change map $CM_{b,a}$. We combined the results of two couples of images as the results produced by Im_{b2}/ Im_a are a priory more correct as the acquisition dates of the images are closer than for Im_{b1} / Im_a. It is explained by the fact that the seasonal changes and other changes irrelevant to the disaster are less numerous.

During the last step we perform the clustering of obtained change areas to associate the detected changes to different types of damage (flooded areas, damaged constructions, etc.).

2.2.3.3 Results

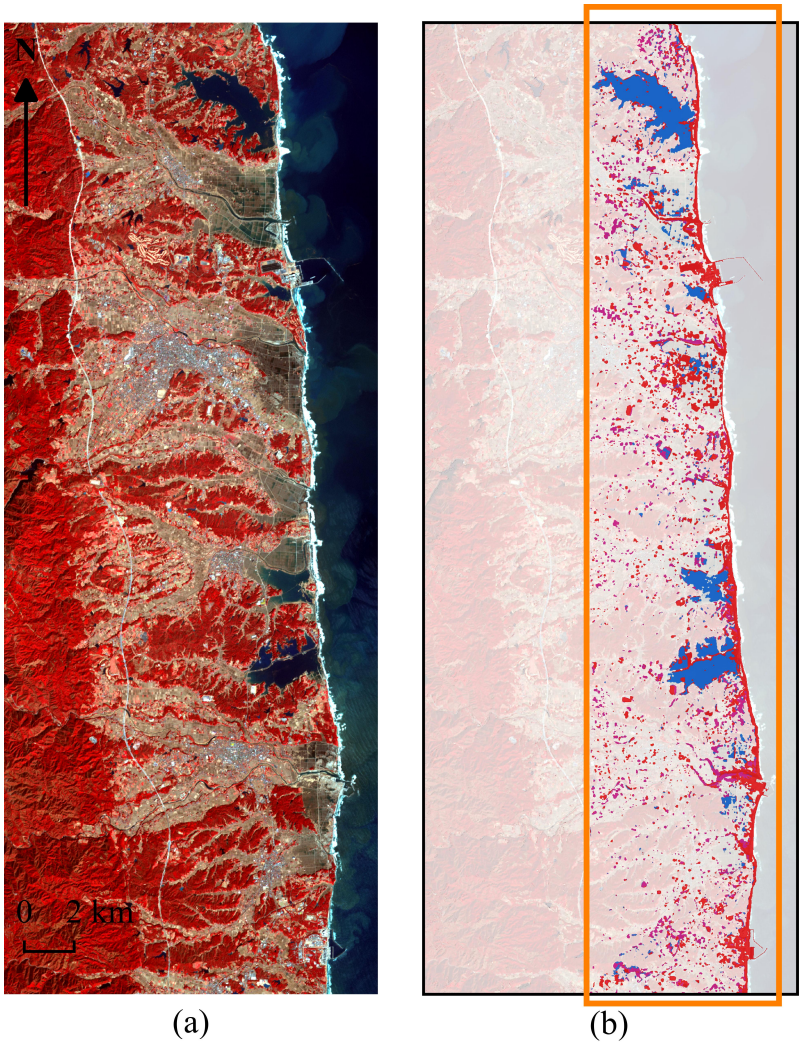


Figure 2.9: **(a)** Extract of the original post-disaster image **(b)** Clustering results with 4 clusters from the DEC algorithm. On the left is the post-disaster image, on the right the clustering with the following clusters: (1) In white, no change. (2) In blue, flooded areas. (3) In red, damaged constructions. (4) In purple, other changes.

The results of our proposed method are shown in Figure 2.9 where we show a projection of the result on the coastal area of interest that was used as an application area. In this image, the blue cluster matches for flooded area, the red cluster destroyed buildings, and the purple cluster is other types of changes that we weren't able to assigned to any obvious type of damage.

Obviously we had no ground truth to check the accuracy of our prediction, but we tried to make one for a few coastal areas. The results can be seen in Figures 2.10, 2.11, 2.12 and 2.13 where we focus on some clusters: flooded areas and destroyed buildings. In these figures, we also compare our method with the K-Means algorithm applied to the same original bottleneck features as the ones used by the DEC algorithm (DEC is actually initialized using KMeans) and with an improved version of the RBM model from [110].

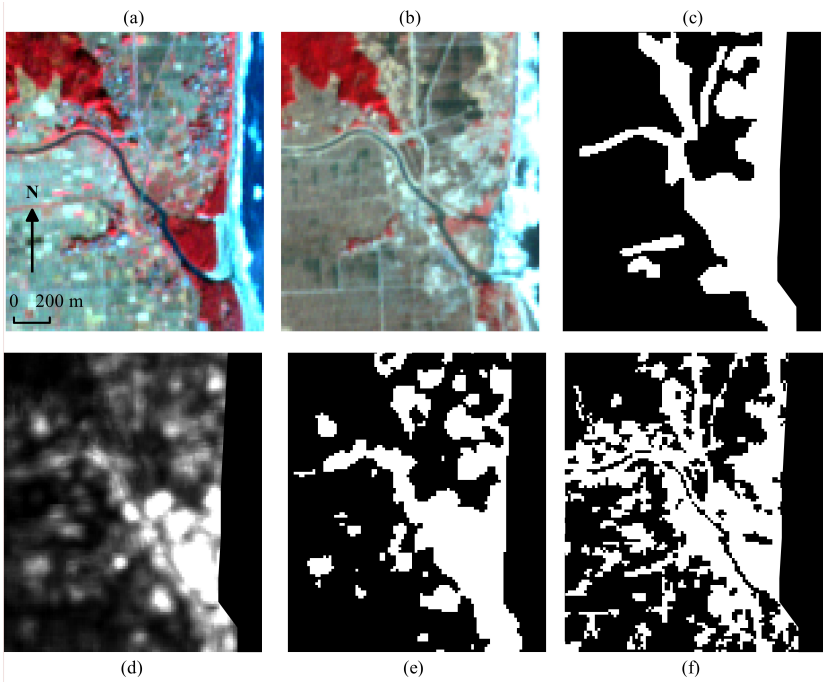


Figure 2.10: Change detection results. (**a**) image taken on 29 November 2010, (**b**) image taken on 19 March 2011, (**c**) ground truth, (**d**) average RE image of the proposed method, (**e**) proposed method CM, (**f**) RBM.

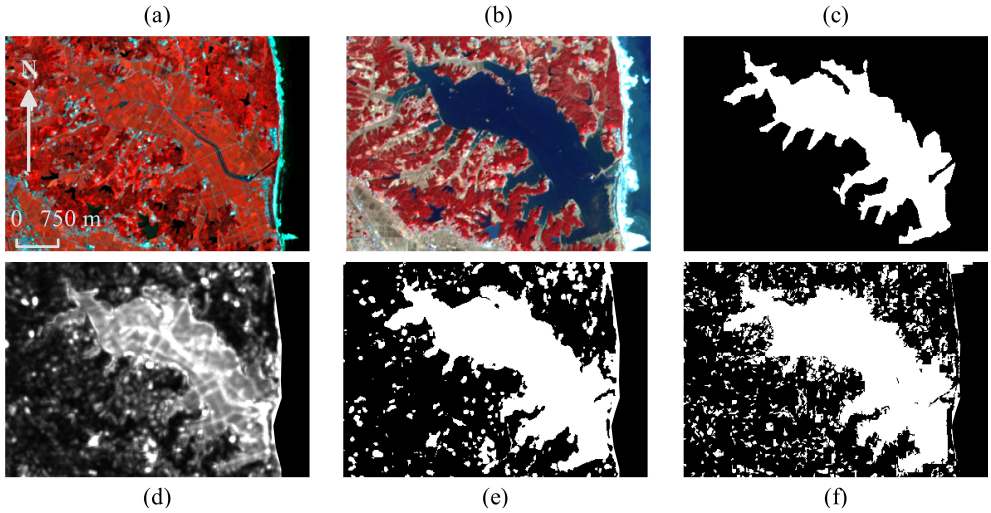


Figure 2.11: Change detection results. (**a**) image taken on 7 July 2010, (**b**) image taken on 19 March 2011, (**c**) ground truth, (**d**) average RE image of the proposed method, (**e**) proposed method CM, (**f**) RBM.

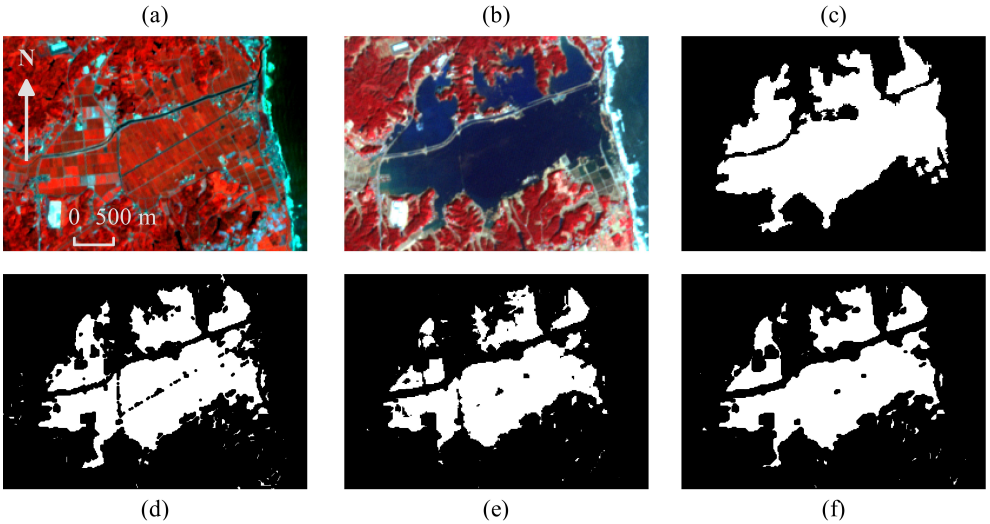


Figure 2.12: Clustering results, flooded area. (**a**) image taken on 7 July 2010, (**b**) image taken on 19 March 2011, (**c**) ground truth, (**d**) K-Means on subtracted image, (**e**) K-Means on concatenated encoded images, (**f**) DEC on concatenated encoded images.

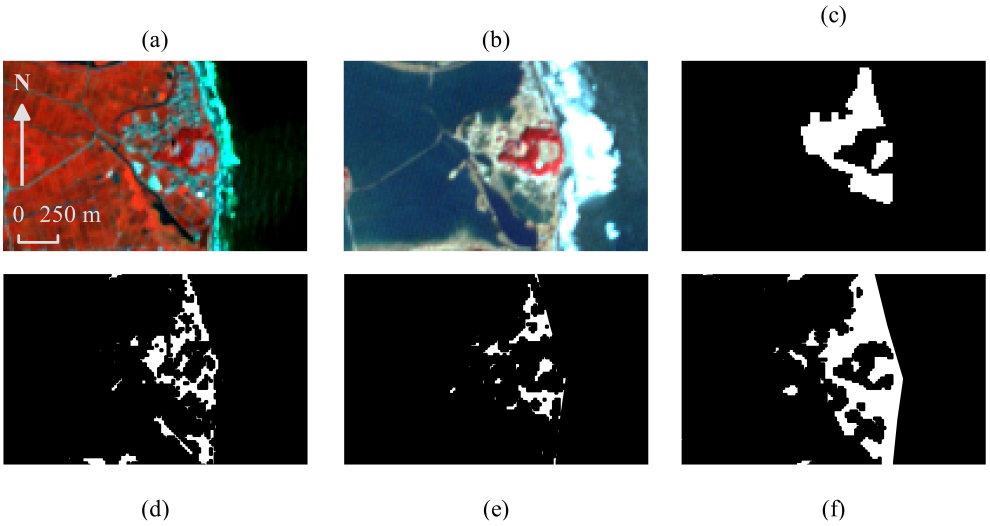


Figure 2.13: Clustering results, destroyed constructions. (**a**) image taken on 7 July 2010, (**b**) image taken on 19 March 2011, (**c**) ground truth, (**d**) K-Means on subtracted image, (**e**) K-Means on concatenated encoded images, (**f**) DEC on concatenated encoded images.

These experiments have highlighted some of the strengths and weaknesses of our proposed methods.

First, we saw that despite being unsupervised, our algorithm is very strong to detect non-trivial changes even with relatively low quality images that are far apart in time, as well as cloud coverage issues and changes in luminosity. We achieve an accuracy around 85% [122] which is comparable with supervised methods from the state of the art. This is a very strong point with an unsupervised algorithm.

Then, we also saw that the clustering phase had more mixed results, which was to be expected from an unsupervised approach. This is due to several phenomena:

- The small errors from the change detection step were propagated to the clustering step.
- It is very difficult for an unsupervised method to find clusters that perfectly match expected expert classes. Our proposed method was good enough to detect flooded areas; however damaged constructions were a lot more difficult to detect and resulted in the creation of a cluster that mixed the modified shoreline and damaged constructions.
- As mentioned previously, the ground truth was built from investigation report and manual labeling of the focus areas which means that our ground truth is far from perfect outside of these focus areas.

However, despite these difficulties, our proposed pipeline relying on joint-autoencoders for change detection and the DEC algorithm for the clustering part achieve very good results for water detection, and fair results for damaged constructions detection with high recall results.

Finally, while the application area and the data quality are different, it is worth putting our results into perspective while comparing them with the ones from [126] where the authors proposed a state-of-the-art method for the same application of the Tohoku tsunami. The main differences are that (1) they use a supervised neural network and thus require labeled data, which we do not, and

(2) they have higher quality satellite images of a different area that are not publicly available. Still, our unsupervised method achieved comparable results with them both visually and quantitatively [122].

2.2.4 Time series analysis using an unsupervised architecture based on Gated Recurrent Units

In this subsection, we go beyond the simple analysis of changes between two images. We propose a method [130] to study full time series. Our method is a clustering technique aiming at analyzing and sorting different types of change behaviors in image time series. It was originally designed for satellite image time series, but we see no reason why this could not be adapted to other types of image time series.

2.2.4.1 Global architecture

The architecture we propose was originally develop to segment non-trivial change behavior through time. But when combined with more regular space-time segmentation methods [111], it can identify spatio-temporal entities in satellite image time series and associate them to 3 different types of temporal behaviors: : no change area, seasonal changes and non-trivial changes. No change areas are mostly presented by spatio-temporal entities that have the same spectral signature over the whole SITS, such as city center, residential areas, deep water, sands, etc. Trivial (seasonal) changes correspond to cyclic changes in vegetation prevailing in the study area. Finally, non-trivial change areas are mostly represented by permanent changes such as new constructions, changes caused by some natural disasters, crop rotations and the vegetation that do not follow the overall seasonal tendency of the study area. Furthermore, as we have explained earlier non-trivial changes are basically outliers that don't follow the general trend, and as such in our approach we propose to cluster them into several categories.

The proposed approach is composed of several steps. Let R_S be a time series of S co-registered images Im_1, Im_2, ..., Im_S acquired at timestamps T_1, T_2, ..., T_S. The algorithm steps are the following (See Figure 2.14 for a graphical summary):

- We start by applying the bi-temporal non-trivial change detection algorithm that we presented in Section 2.2.2.3 to every couple of consecutive images Im_n-Im_{n+1} ($n \in [1, S]$). However, the changes detected by doing so are contextual to each pair of images. It is therefore necessary to refine the $S-1$ binary change maps $CM_{1,2}$, $CM_{2,3}$, ..., $CM_{S-1,S}$ over the whole series using logical constraints that can discriminate 3 types of changes: false-positive changes, one-time anomalies and real changes (see Section 2.2.4.2), only the later of which we are interested in.
- We extract the spatio-temporal change areas by applying the change masks to the corresponding images of the time series.
- Then we perform image segmentation within these change areas to obtain changing objects.
- Afterwards, the change objects located in the same geographic area are grouped in temporal evolution graphs [131][111].

– Finally, we cluster the obtained graphs using the features extracted from the change areas. We use a summarized representation of graph structure - synopsis - as input sequences of hierarchical agglomerative clustering [13].

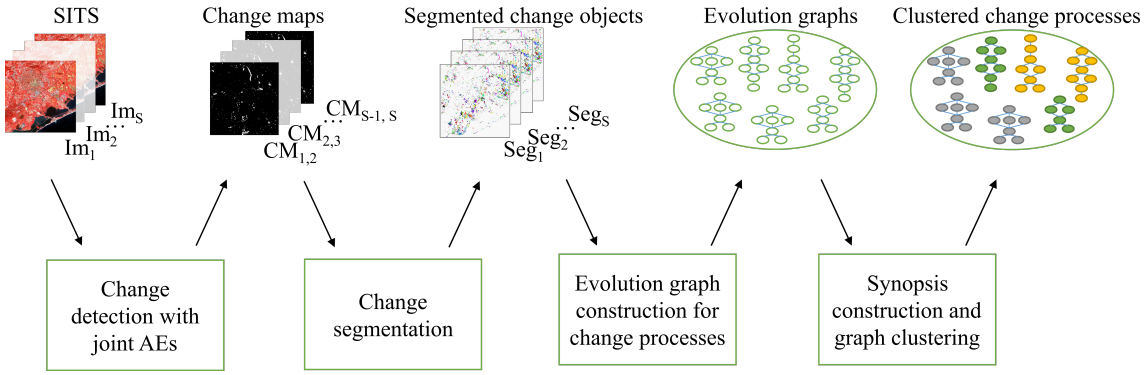


Figure 2.14: Proposed framework for time series clustering of change behaviors.

2.2.4.2 Detecting contextual changes

Since we have already detailed how the joint-autoencoder for non-trivial change detection works, we jump directly to the analysis of these changes in a multi-temporal context. Please note that bi-temporal non-trivial changes can be interpreted as contextual anomalies [132] as they depend on the overall change tendency in the couple of images considered. Their interpretation might therefore change when moving from bi-temporal to a multi-temporal context.

To introduce multi-temporal context when detecting changes that appear between timestamps T_n and T_{n+1}, we propose to check if the detected change polygons (areas of changes) have been detected in other change maps, see Figure 2.15.

If a change polygon P_{ch} does not have spatial intersection with any polygon(s) of $CM_{n-1,n+1}$, it may belong to different types of temporal behavior:

– If P_{ch} does not have any spatial intersection with any polygon(s) from $CM_{n-1,n}$, it is marked as false positive (FP) as it was most likely caused by some image defaults or was wrongly detected by the algorithm.

– If P_{ch} has a spatial intersection with any polygon(s) from $CM_{n-1,n}$ and with polygon(s) in at least one other change map of the series, it is marked as a part of an irregular change process.

– Finally, if P_{ch} has intersection only with polygon(s) from $CM_{n-1,n}$ and does not have any intersection with polygons from other change maps, it is marked as a one time anomaly that happened at timestamp T_n. In this case, all change polygons from $CM_{n-1,n}$ that have intersection with P_{ch} are also marked as one time anomalies.

Note that here we use a threshold t_{int} that defines the minimum percentage of spatial intersection of P_{ch} with other change polygon(s), otherwise, it is considered that there is no intersection.

Our model works under the hypothesis that every change process belonging to the same geographical location is continuous. For example, if a pixel i, j has been classified as change in $CM_{1,2}$, $CM_{2,3}$ and $CM_{4,5}$, it should be also marked as change in $CM_{3,4}$ (see Figure 2.16).

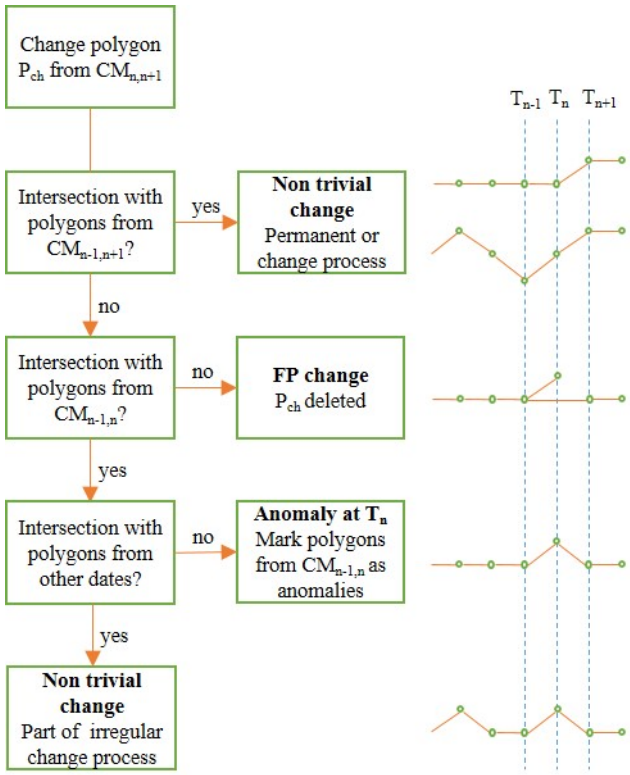


Figure 2.15: Correction of detected bi-temporal contextual anomalies accordingly to multi-temporal context.

Once the correction in done for the whole time series, we apply the union of change maps $CM_{n-1,n}$ and $CM_{n,n+1}$ to extract the change areas for every image Im_n. Obviously, we apply only one change map for the first and last images of the SITS.

2.2.4.3 Building the evolution graphs

The next step is to segment the different objects in the detected change areas. First, all change masks $CM_{n,n+1}$ are applied with the matching images Im_n-Im_{n+1}. Then, since each image (except the first and last one) have two masks, the segmentation is done within the union of the two change masks.

To do so, we use a graph-based tree-merging segmentation algorithm [133] due to its ability to produce relatively large segments without merging different classes together. Large segments facilitate further construction and interpretation of evolution graphs as the shapes of some change segments may have important variations from one image to another when they are over-segmented.

It is worth mentioning that no-change areas can also be segmented at this point (using any of the timestamps as a reference) to fill in the blanks and achieve a full 3D segmentation.

Based on the segments, we build the evolution graphs by adapting the method proposed in [131] and [111] (see Figure 2.17 for an example). The initial approach is the following: given a SITS and its associated segmentation, we choose a set of objects that corresponds to the spatial entities we want to monitor. This set of objects is used as bounding boxes. A bounding box can come from any timestamp and is connected to the objects covered by its footprint in the previous and next timestamps. A bounding box and the objects connected to it form an evolution graph.

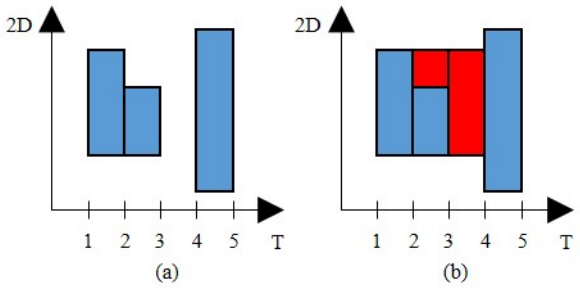


Figure 2.16: Transformation of a discontinuous change process into a continuous one. (a)- discontinuous change process, (b)- corrected blue polygons correspond to detected change objects, red polygons are added to transform a discontinuous change process into a continuous one.

Each evolution graph can have only one bounding box and has to be continuous. Every object of a graph represents a node and overlapping values between two objects at two consecutive time-stamps are the edges. Objects at timestamp T_n can be connected only to objects from T_{n-1} or T_{n+1}, a timestamp that contains a bounding box can only have a single object corresponding to this bounding box.

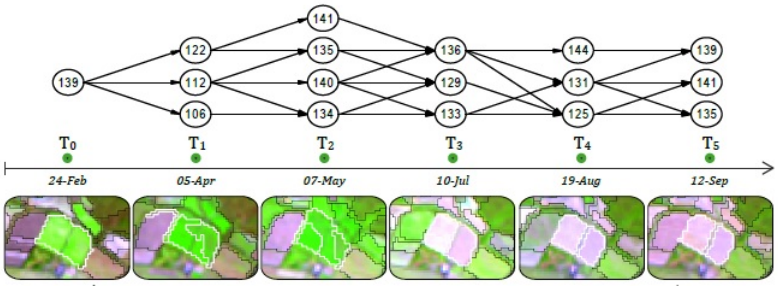


Figure 2.17: Example of an evolution graph (Guttler et al. 2017 [131])

In our method, we construct evolution graphs in such a manner that every graph contains only coherent information. In other words, every bounding box is connected only to its best matching segments comparatively to neighbor bounding boxes and each segment can belong only to one or no evolution graph. In order to construct graphs that contain only objects belonging to the same phenomena, we use different parameters for the construction of evolution graphs that are independent of one another: at least τ_1 percent of the object should be inside the bounding box footprint, and the intersection with the object should represent at least τ_2 percent of the bounding box footprint:

$$\tau_1 = \frac{Pix(O) \cap Pix(BB)}{Pix(BB)} \tag{2.1}$$

$$\tau_2 = \frac{Pix(O) \cap Pix(BB)}{Pix(O)} \tag{2.2}$$

The first parameter τ_1 is the most important and allows to select only the objects that are covered the most by BB footprint. The second parameter τ_2 is used to keep the objects filling only certain percentage of the footprint.

Due to pixel shift and to some false positive changes, we may observe many parasite objects in

the evolution graphs. These objects usually correspond to crop fields. If a timestamp of an evolution graph is solely made of a parasite object, it can influence further graph interpretations. To minimize the number of parasite objects, we introduce a parameter τ_3 that represents minimum ratio of coverage between two consecutive timestamps.

$$\tau_3 = \frac{\sum_1^q Pix(O_i^{n+1})}{\sum_1^r Pix(O_j^n)}, \tag{2.3}$$

where q and r are the number of objects at timestamps T_{n+1} and T_n respectively, $Pix(O_i^{n+1})$ is i-th object at timestamp T_{n+1} and $Pix(O_j^n)$ is j-th object at timestamp T_n.

2.2.4.4 Graph synopsis and feature extraction

Table 2.3: Feature extraction model.

	Feature extraction
encoder	Conv(B,32)+ReLU
	Conv(32,32)+ReLU
	Conv(32,64)+ReLU
	Conv(64,64)+ReLU
	MaxPooling(kernel=3, stride=3)
	Conv(64,128)+ReLU
	Conv(128,128)+ReLU
	MaxPooling(kernel=3)
	Linear(128,64)+ReLU
	Linear(64,32)+ReLU
	Linear(32,f)+ℓ_2-$norm$
decoder	Linear(f,32)+ReLU
	Linear(32,64)+ReLU
	Linear(64,128)+ReLU
	UnPooling(kernel=3)
	Conv(128,128)+ReLU
	Conv(128,64)+ReLU
	UnPooling(kernel=3, stride=3)
	Conv(64,64)+ReLU
	Conv(64,32)+ReLU
	Conv(32,32)+ReLU
	Conv(32,B)+ReLU

To cluster the extracted evolution change graphs, we compute each graph synopsis as in [111]. A synopsis summarizes each graph's information and makes it possible to compare them with each other. A synopsis Q is defined as a sequence of the same length as the corresponding evolution graph. Each timestamp T_n of sequence Q contains the aggregated values of graphs objects at this timestamp. The influence of each object at the aggregated value at timestamp T_n is proportional to its size and calculated as follows:

$$Q_n = \frac{\sum_1^r Pix(O_j^n) \cdot v_j}{\sum_1^r Pix(O_j^n)} \tag{2.4}$$

where Q_n is the synopsis value at timestamp T_n, $Pix(O_j^n)$ is the size of a j-th object at timestamp T_n ($j \in [1, r]$, where r is the total number of objects within the evolution graph E at timestamp T_n) and v_j is the corresponding mean of object value.

The feature are extracted using a deep convolutional denoising autoencoder the architecture of which is presented in Table 2.3. These encoded features are used to represent the synopsis. The extraction steps are the following:

1. We extract patches of size p for every pixel of every image of SITS to train convolutional AE model.
2. We divide the extracted patch dataset into training set and validation set (67% and 33%). The validation set is used for early stopping and prevent overfitting [134].
3. We train the AE model in such manner that every patch from the training dataset is firstly encoded in a feature vector and then is decoded back to the initial patch. We use the mean square error of the patches reconstruction to optimize the model at each iteration.
4. The early stopping algorithm is applied at every epoch and check the loss value when fitting the validation set. If the validation loss does not improve during a given number of epoch, the model is considered stable and the training is stopped.
5. We use the encoding part of the AE to encode every patch of SITS change areas in a feature vector.

2.2.4.5 Graph Clustering using GRU and hierarchical agglomerative clustering

In the presented framework, we propose to use Gated Recurrent Units (GRU) [135] combined with autoencoders to extract features from the graphs of different objects in a time series, and then hierarchical agglomerative clustering [13] to cluster the obtained evolution graphs.

GRU is a recurrent neural networks-based (RNN) type of neural network that is able to process time series in order to extract some meaningful information. Unlike many other approaches for time series analysis, RNN is able to deal with varying sequence lengths.

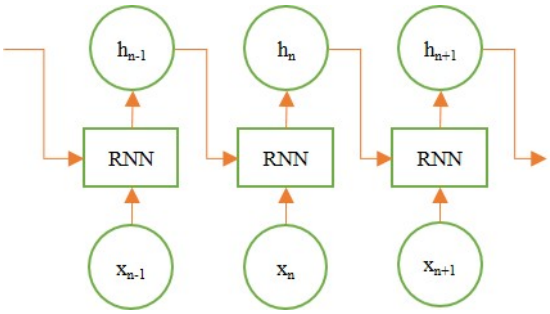


Figure 2.18: The classical RNN model.

In general, recurrent neural networks are built as follows: let $X = \{x_1, x_2, ..., x_n, ..., x_S\}$ be a sequence composed of S timestamps. the network computations are realized in such way that each timestamp T_n is associated with a hidden state h_n (see Figure 2.18) which represent the accumulative

value of the previous hidden states of the sequence. The final hidden state h_S characterizes the whole sequence and is used afterwards as series descriptors for classification or clustering. The main problem of RNN is that the value of each hidden state h_n depends only on the value of previous hidden state h_{n-1}, hence, RNN networks may suffer from a long term memory problems caused by vanishing gradient and does not consider long term dependencies. To solve this issue, more complex Long Short-Term Memory (LSTM) networks were introduced [136]. Contrary to RNN, LSTM contains input, output and forget gates as well as memory cell c_n at each timestamp that makes it possible to retain meaningful information from all previous steps and, as a consequence, the value of h_n depends on all previous hidden states of the sequence and not only on h_{n-1}. Later, to facilitate the computation and implementation of the LSTM model, GRU networks were developed [135] for Natural Language Processing (NLP) tasks. GRU contains only update and reset gates thus allowing the model to be trained faster with a lower memory consumption. GRU were successfully adapted for remote sensing applications and proved to give a higher accuracy than LSTM networks in this research area [137][138, 139]. Originally recurrent neural networks were not able to capture spatial information. However, convolutional recurrent neural networks were later introduced [140] and are able to process videos or image time series.

Back to our original problem, in our case we want to cluster the graph synopsis that we have previously built. To do so, we use a GRU autoencoder (GRU AE) which combines the advantages of an autoencoder architecture with time series analysis properties: During the encoding pass, GRU AE extracts the accumulated hidden state of the sequence h_S at the last timestamp. The last hidden state is then self-concatenated S times (see [141]) and passed to the decoding part that aims to reconstruct the inversed initial sequence $X_{inv} = \{x_S, ..., x_n, ..., x_2, x_1\}$. As it is usually recommended to set hidden state size large (>100), we add fully-connected layers before the GRU AE bottleneck to compress the size of hidden state to ameliorate the further clustering results. The overall GRU AE schema is presented on Figure 2.19. Finally, we apply hierarchical clustering to the bottleneck of GRU AE to obtain the associated change clusters.

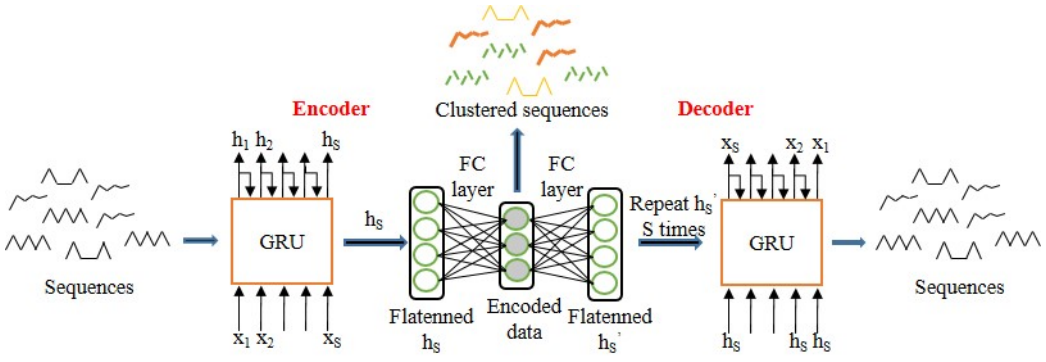


Figure 2.19: GRU AE clustering model.

As the input sequences have varying length, some data preparation is necessary, so the GRU is able to correctly process it. Data preparation is performed for every training batch individually, after the input GRU dataset has been created. For every batch B_i, we perform the following steps (see Figure 2.20):

1. We define the maximum sequence length d of B_i.

2. We zero-pad the end of all the sequences of B_i, so they have the same length d.
3. The padded sequences are passed to the encoder, for each sequence its final hidden state h_S is obtained.
4. As indicated before, we use the cloned h_S as the input of GRU layer in the decoding part, where h_S is repeated S times.
5. We apply the inverted padding mask to the cloned h_S sequence that is fed to GRU layers of the decoder.
6. The output of the decoder should resemble to the inverted padded input sequence.

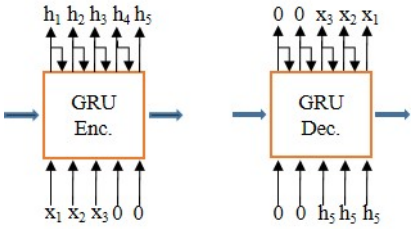


Figure 2.20: Padding of data sequences. In this example, the initial sequence x_1, x_2, x_3 has the length of 3 timestamps and the maximum sequence length per batch is $d = 5$. For the simplicity of representation, we do not consider the number of features of each sequence.

While the padding of the encoder input sequence allow us to proceed batches with varying length sequences, the padding of the decoder input improves the model quality, especially, it lowers the influence of sequence lengths on the extracted encoded features.

We do not divide the sequence data into training and validation datasets as the nature of some change sequences may be unique. For these reason, we control the training loss changes between two consecutive epochs to prevent the model over-fitting.

The model configuration is presented in Table 2.4, where f is number of features of the input sequences, *hidden_size* is the length of hidden state vector, d is the maximum sequence length per batch, *f_hidden* is the size of encoded hidden state vector.

Table 2.4: GRU model.

	Sequence feature extraction
enc.	GRU(f,hidden_size, dropout=0.4) (2 layers) Linear(hidden_size,f_hidden)+ℓ_2-*norm*
dec.	Linear(f_hidden,hidden_size)+ReLU Repeat hidden state d times Apply inversed padding mask GRU(hidden_size,f, dropout=0.4) (2 layers)

2.2.4.6 Result example

In Figure 2.21 below, we show an example of a graph built on a Sentinel-2 dataset with images from the French city of Montpellier taken between 2005 and 2008 as the city was building a stadium.

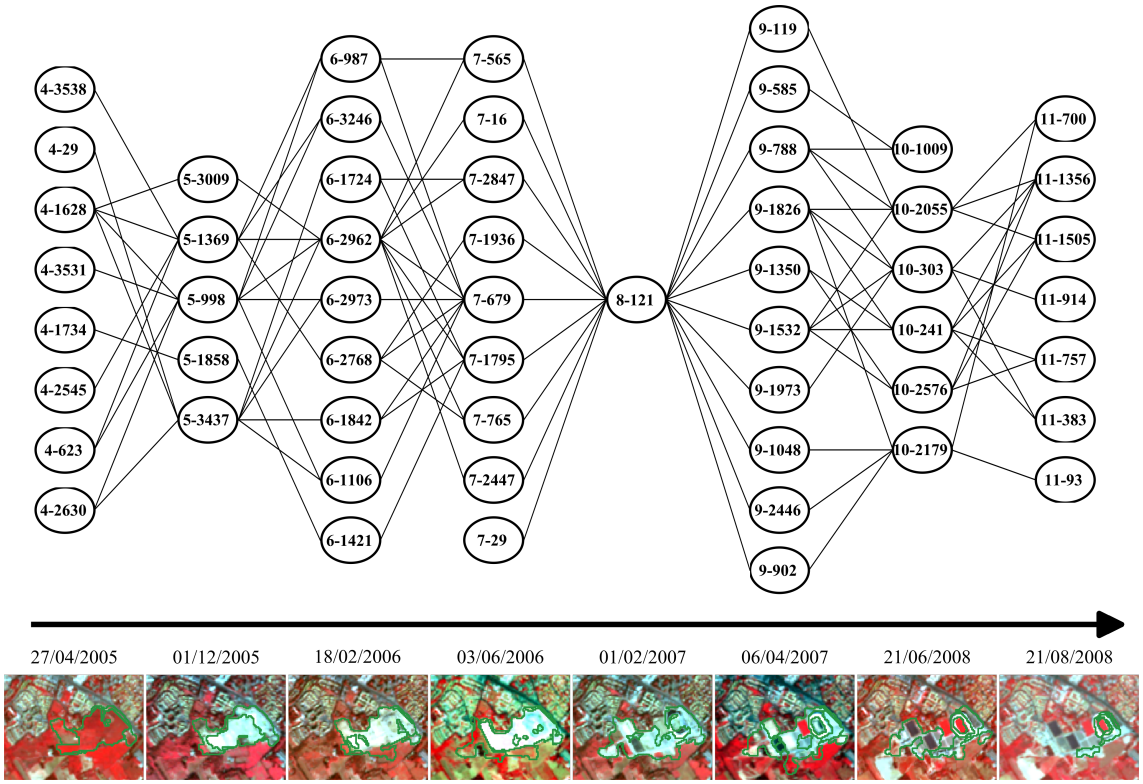


Figure 2.21: Example of an evolution graph: construction of a Stadium.

2.2.5 Conclusions for remote sensing applications with unsupervised learning

As you have seen in this section, we have shown that it is indeed possible to process remote sensing images with deep learning algorithms in an unsupervised context. We have seen both the strength of this type of approach for change detection using joint autoencoders, but also the limits both in term of practical application as the cluster mapping to the real classes is complex, and with the analysis of full time series where the architecture is very complex and prone to error accumulation.

Regardless, despite the fact that there no state of this art for this problem, we obtained encouraging results and provided a new a unique framework for the end-to-end change detection and modeling in satellite image time series. Furthermore, while our method has many steps, our computation times remain reasonable.

A more complete conclusion will be given in Section 2.4 which will analyze the contributions made in this chapter and put the results of this section into perspective with the ones of similar algorithms applied to medical images that are presented in the next section.

2.3 Unsupervised deep learning applied to time series of Age Related Macular Degeneration lesions

Since we managed to show that it was possible to use deep learning in an unsupervised context successfully for applications in the field of remote sensing, we wanted to know if it was possible to do the same in the field of medical imaging. Indeed, these two types of images share many common

features: lighting, blur and distortion issues, different scales of interest with complex structures, alignment problems, saturated pixels, etc. However, medical images are also different and simpler in several ways: they are small, they have less channels, the variety of objects is lower, and when it comes to time series analysis, evolution usually go one way.

Luckily for me, ISEP has a long standing collaboration with the Clinical Imaging Center 1423 of Paris Quinze-Vingts Hospital for the study of various eye disease pathologies through image processing techniques. As such, this section will show some adaptations of the previously presented algorithms as well as some novel ideas applied to medical images. This work was done in collaboration with Florence Rossant and Michel Pâques through internships and 2 ongoing PhD thesis.

2.3.1 Age Related Macular Degeneration time series

Dry age-related macular degeneration (ARMD or sometimes AMD), a degenerative disease affecting the retina, is a leading cause of intractable visual loss. It is characterized by a centrifugal progression of atrophy of the retinal pigment epithelium (RPE), a cellular layer playing a key role in the maintenance of the photoreceptors. In Figure 2.22, we show a simplified schema of the human eye structure.

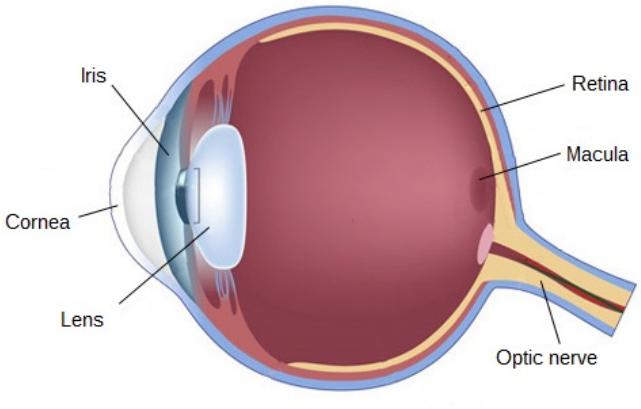
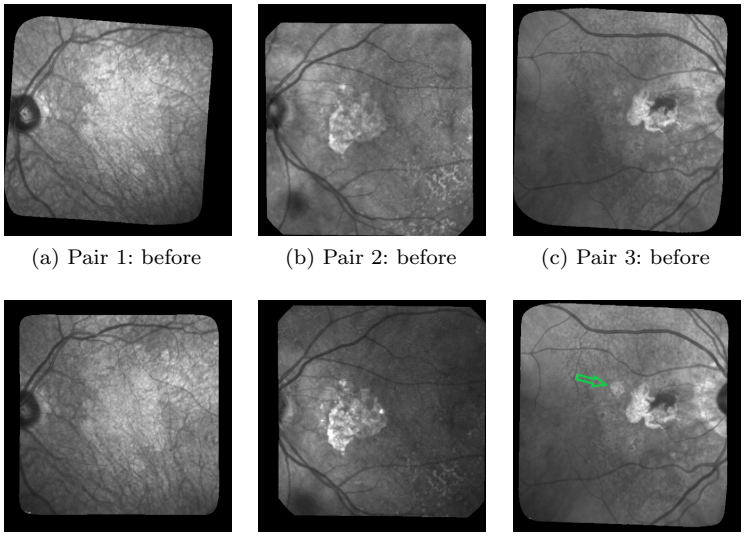


Figure 2.22: Schema of the eye structure

Since ARMD first appear in the central part of the eye (the macula) which contains the highest density of photoreceptors, partial blindness may rapidly occur as the disease progresses. The disease may be diagnosed and monitored using fundus photographs: ophthalmologists can observe pathological features such as drusen that occur in the early stages of the ARMD, and evaluate the geographic atrophic (GA) progression in the late stages of degeneration (see Figure 2.23).



(a) Pair 1: before (b) Pair 2: before (c) Pair 3: before

(d) Pair 1: after (e) Pair 2: after (f) Pair 3: after

Figure 2.23: 3 of pairs of images acquired six months apart, the geographic atrophic lesions are the bright areas. The green arrow in (**f**) shows a new lesion.

Automatic analysis of fundus images with dry ARMD is of high medical interest [142] and this has been an important research field for two decades, for diagnosis [143] or follow up [144, 145] purposes. Imaging modalities are most often color eye fundus images [146][147][148], fundus autofluorescence (FAF) [149][150][145], and, to a lesser extent, confocal scanning laser ophthalmoscopy (cSLO) in infrared (IR), or optical coherence tomography (OCT) [151]. In our study, we process cSLO images in IR as this modality is comfortable for the patient, and it has higher resolution and higher contrast than color imaging, an older technology.

Figure 2.23 shows three pairs of consecutive images, taken at 6-month intervals. The lesions (GA) are the brighter regions in the fundus and around the optical disk. Automatic processing to follow up these areas is obviously very challenging given the quality of the images: uneven illumination, saturation issues, illumination distortion between images, GA poorly contrasted with retinal structures interfering (vessel, optical disk), blur, etc. The difficulty also lies in the high variability of the lesions in terms of shape, size, and number. The lesion boundary is quite smooth in some cases (b and e) and very irregular in others (a and d). At any time, new spots can appear (as shown by the green arrow between c and f) and older lesions can merge. All these features make the segmentation task very difficult, and especially long and tedious to perform manually. It is worth noting that even experts cannot be sure of their manual delineation in all cases.

Studying the ARMD lesions evolution is of great interest to ophthalmologists as the existing models for the disease progression are rather crude and the underlying phenomenons causing age-related macular degeneration have not been fully identified yet. As such, studying the growth pattern of the lesions can be very useful not only to better understand the disease progression, build predictive models, but also to assess the effectiveness of potential trial drugs. The main difficulty so far for any of these applications lies in the numerous issues mentioned in the previous paragraph which make the images very difficult to tackle: Even experienced ophthalmologists often disagree on where the lesion start and stop in these images. As such, there are very few reliable datasets that are labeled and can be used for supervised learning.

2.3.1.1 State of the art on change detection and disease progression applied to ARMD

The common point between all the applications that we have mentioned earlier is that they deal with the progression of the lesions through time. In other word, we are back to a change detection problem in an image time series. Since this type of image is simpler than satellite images (in the sense that they contain a lot less classes of interests), there are two ways to tackle the problem that are commonly found in the literature:

- The *segmentation first approach*: All images are segmented individually and we use the segmentations as a basis to observe the disease progression.
- The *Difference approach* which consists in comparing pairs of images using different methods. It is this approach that we used for remote sensing images in the previous section.

First Approaches Applied to ARMD and Other Eye Diseases: In [146], the authors proposed an approach where they first segment all healthy regions to get the lesions as the remaining areas. This approach requires segmenting separately the blood vessels, which is known to be a difficult task. This method involves many steps and parameters that need to be supervised by the user. In [145], Ramsey et al. proposed a similar but unsupervised method for the identification of ARMD lesions in individual images: They use an unsupervised algorithm based on fuzzy c-means clustering. Their method achieves good performances for FAF images, but it performs less well for color fundus photographs. We can also mention the work of Hussain et al. [152] in which the authors are proposing another supervised algorithm to track the progression of drusen for ARMD follow-up. They first use U-Nets [103] to segment vessels and detect the optic disc with the goal of reducing the region of interest for drusen detection. After this step, they track the drusen using intensity ratio between neighbor pixels.

Other traditional more machine learning approaches have also been used for GA segmentation such as the k-nearest neighbor classifiers [150], random forests [148] ([67]), as well as combinations of Support Vector Machines and Random Forests [153]. Feature vectors for these approaches typically include intensity values, local energy, texture descriptors, values derived from multi-scale analysis and distance to the image center. Nevertheless, these algorithms are supervised: they require training the classifier from annotated data, which brings us back to the difficulty of manually segmenting GA areas.

Related to other medical images, in [154] the authors show that the quantization error (QE) of the output obtained with the application of Self Organized Maps [50] is an indicator of small local changes in medical images. This work is also unsupervised but has the defaults that the SOM algorithm cannot provide a clustering on its own and must be coupled with another algorithm such as K-Means [18] to do so. Furthermore, since there is no feature extraction done, this algorithm would most likely be very sensitive to the lighting and contrast issues that are present in most eye fundus time series. Lastly, the use of SOM based methods on monochromatic images is discouraged since no interesting topology may be found from a single attribute.

Finally, the literature also contains a few user-guided segmentation frameworks [155][156] that are valuable when it is possible to get a user input.

Differential Approaches Applied to ARMD: The following works are most related to our proposed algorithm as they are unsupervised algorithms applied to various eye disease images, including

ARMD: In [157], Troglio et al. published an improvement of their previous works realized with Nappo [158] where they use the Kittler and Illingworth (K&I) thresholding method. Their method consists of applying the K&I algorithm on random sub-images of the difference image obtained between two consecutive eye fundus images of a patient with retinopathy. By doing so, they obtain multiple predictions for each pixel and can then make a vote to decide the final class. This approach has the advantage that it compensates for the non-uniform illumination across the image; however, it is rather primitive since it does not actually use any Machine Learning and rely on different parameters of the thresholding method to then make a vote. To its credit, even if it achieves a relatively weak precision, it is fully unsupervised like our method. In [147], the authors tackle a similar problematic to ours where they correct eye fundus images by pairs, by multiplying the second image by a polynomial surface whose parameters are estimated in the least-squares sense. In this way, illumination distortion is lessened and the image difference enhances the areas of changes. However, the statistical test applied locally at each pixel is not reliable enough to get an accurate map of structural changes.

As one can see, quite a few method exist in the literature. Table 2.5 summarizes these different approaches, plus some other from fields other than medicine: it specifies whether or not they are supervised (and need labeled data), if they are based on segmentation first on individual images or on pairs of images, their main underlying principle, and their original field of application.

Authors	Supervised	Images Used	Algorithm	Application
Troglio et al. [157][158]	No	Pairs	K&I Thresholding	ARMD
Marrugo et al. [147]	No	Pairs	Image correction	ARMD
Köse et al. [146]	semi	Individual	Raw segmentation	ARMD
Ramsey et al. [145]	No	Individual	Fuzzy C-Means	ARMD
Hussain et al. [152]	Yes	Individual	U-Nets	ARMD
Burlina et al. [159][160]	Yes	Individual	pre-trained CNN	ARMD
Kanezaki et al. [161]	No	Individual	CNN	Image Processing
Sublime et al. [99][122]	No	Pairs	Joint-AE & KMeans	Remote Sensing
Celik et al. [162]	No	Pairs	PCA & KMeans	Remote Sensing

Table 2.5: Summary of the state-of-the-art methods for change detection.

In sections 2.3.3 and 2.3.4, we present two of our contributions that explore both the *segmentation first* and the *difference approach* using deep learning methods in an unsupervised context.

2.3.1.2 ARMD Dataset presentation

Our images whose main characteristics can be found in Table 2.6 were all acquired at the Quinze–Vingts National Ophthalmology Hospital in Paris, in cSLO with IR illumination. This modality has the advantage of being one of the most common and cheapest legacy method of image acquisition for eye fundus images, thus allowing to have lots of images and to follow the patients for several years. However, it is infrared only and therefore all images are monochromatic and may contain less information than images acquired from other techniques with multiple channels (that are less common for this type of exam and more difficult to find in numbers).

Number of patients	15
Number of image time series	18
Average number of images per series	13
Total number of images	336
Acquisition period	2007–2019
Average time between two images	6 months

Table 2.6: Description of the data.

Patients have been followed-up during a few years, hence we have a series of retinal fundus images, sometimes for both eyes (hence the number of series and patients being different in Table 2.6), showing the progression of the geographic atrophies. The average number of images in each series is 13. The images are dated from 2007 for the oldest to 2019 for the most recent. All pictures are in grayscale and vary greatly in size, but the most common size is 650×650 pixels.

As mentioned previously, we notice many imperfections such as blur, artifacts and, above all, non-uniform illumination inside the images and between them (see Figure 2.24). All images were spatially aligned with i2k software [4].

2.3.2 Image preprocessing

Regardless of the approach, *segmentation first*, or *difference based*, a first preprocessing step was necessary.

To solve this issue, we first use a new method to compensate for illumination distortion between the images (not published yet). This algorithm is based on an illumination/reflectance model and corrects all images of a series with respect to a common reference image. Uneven illumination generally remains present in every processed image (Figure 2.24), but the smooth illumination distortions are compensated. The calculus of the absolute value of the difference between two consecutive images demonstrates the benefit of this algorithm (Figure 2.24, last column).

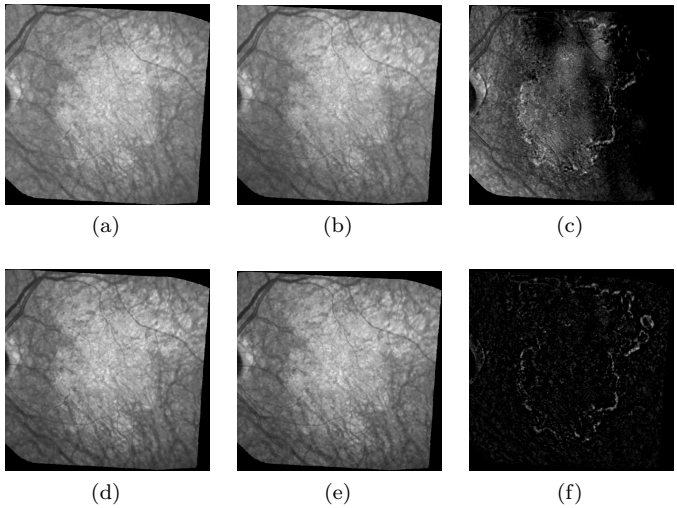


Figure 2.24: Example of illumination correction. The three images on the top row represent the two original consecutive images (**a**) and (**b**), and their raw difference in absolute value (**c**); on the bottom row: the same images after illumination correction (**d**) and (**e**), and the new difference (**f**).

[4] `https://www.dualalign.com/retinal/image-registration-montage-software-overview.php`

Then, we have another problem to solve: as one can easily see, the area of useful data does not fill the entire image which is surrounded by black borders. The automatic detection of these black zones in each image gives a mask of the useful data, and the intersection of all masks the common retinal region where changes can be searched for. As can be seen in Figure 2.25, we solve this problem by using the Inpainting function of the library *scikit-image* [163] to complete this background. This inpainting function is based on the biharmonic equation [164][165], and it exploits the information in the connected regions to fill the black zones with consistent gray level values.

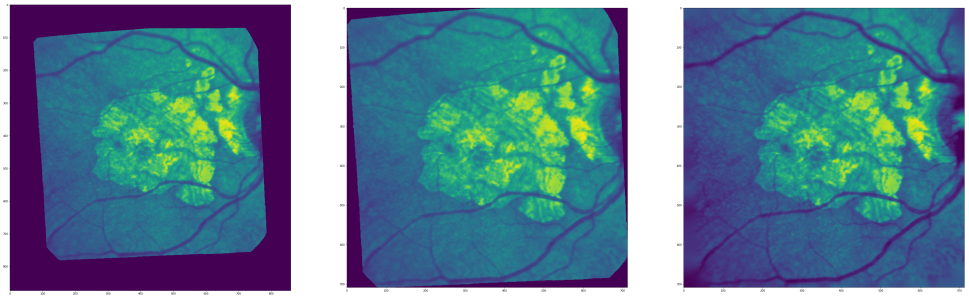


Figure 2.25: From left to right: Original image in false colors, cropped image, cropped image with inpainting

2.3.3 Lesion segmentation using W-Nets

This section presents some results achieved during the Clément Royer's internship [166].

2.3.3.1 From U-Nets to W-Nets

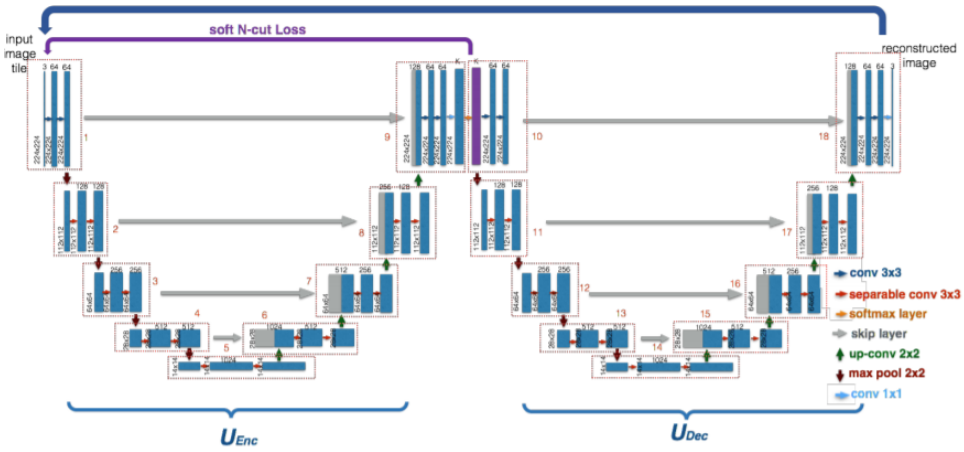


Figure 2.26: The W-Net architecture we used on our ARMD images: Both reconstruction and N-cut loss functions are shown

As mentioned in the introduction, our goal was to propose a segmentation method that could be applied to pre-processed images of ARMD patient, with the goal that these segmentations could be used as a basis for further analysis, if they are good enough.

With that in mind, we quickly search for a deep learning method that in an unsupervised context could segment the image and correctly form a cluster that would contain the regions that we are interested in: the geographic atrophies. Because U-Nets [103] were used in the literature for on

ARMD lesions [152], we quickly decided to try applying W-Nets [96] to our images, since this is the unsupervised equivalent of U-Nets.

The main architecture of W-Nets is shown in Figure 2.26. This network belongs to the family of autoencoders and alternatively optimizes 2 loss functions until convergence: The reconstruction loss over the whole network (which is normal for an auto-encoder), and a soft N-cut loss function which is used for the segmentation part.

2.3.3.2 Results

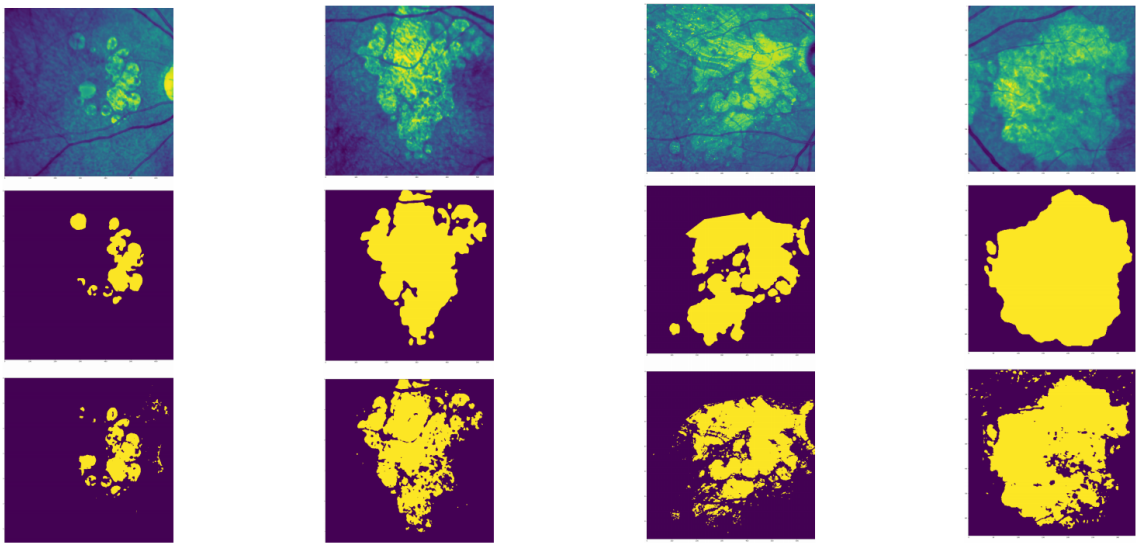


Figure 2.27: Row 1: original image in false colors; Row 2: ground truth; Row 3: W-Net result

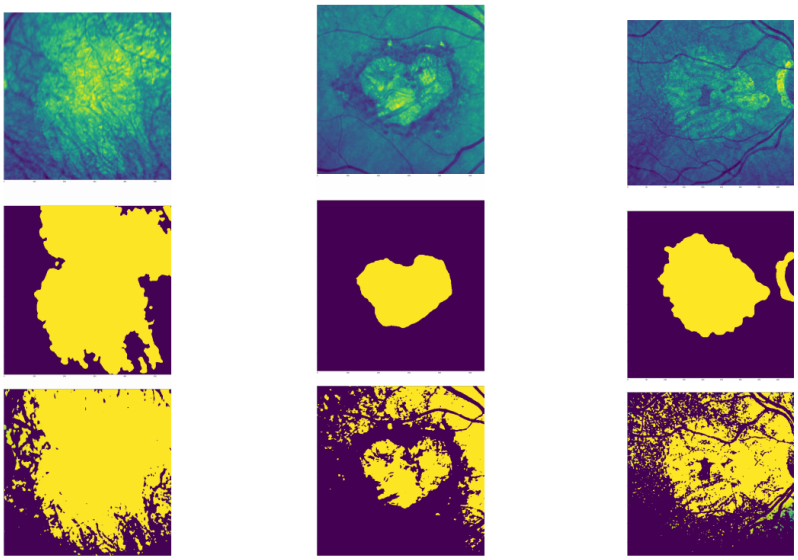


Figure 2.28: Row 1: original image in false colors; Row 2: ground truth; Row 3: W-Net result

In Figure 2.27, we show an example of some of the results achieved by our proposed W-Net. In this Figure, we show mostly examples where the result was satisfactory compared with the lesion

segmentation suggested by the ophthalmologists.

On the other hand, in Figure 2.28, we show some cases where W-Nets results are a lot less good. Several explanations are possible: Too difficult textures resulting in a failed segmentation process is the first that comes to mind. Another one can be a too big variance of textures thus making it difficult to have a pure cluster with only the lesion. Our team has actually investigated this second possibility since the number of cluster is a well-known problem in clustering: increasing the number of cluster helped a lot with achieving better results, but at the cost of having to add some supervision to decide which clusters to merge in order to find the healthy and sick areas.

On average, we found that W-Nets had a precision around 90% and a recall around 85% on average when applied to all images of all series. Theses results are very encouraging for an unsupervised method and the scores were probably slightly lowered by difficulties to segment the first images of the series that often have one or both eyes with no lesion, or nothing easy to spot.

2.3.4 Analyzing the lesion progression using joint-autoencoders

In this subsection, we propose to apply the change detection algorithm presented in Section 2.2.2.3 to the case of ARMD images with the goal of detecting the evolution of the lesions between two medical check-up exams [167]. Thus in this section, we consider pairs of images.

2.3.4.1 Algorithm

Let us consider a series of M images representing the progression of ARMD in a patient's eye. After the pre-processing and once the images have been aligned and cropped, all images from the same series have the same number of N useful patches. From there, to pre-train or network, we sample $\left\lfloor \frac{N}{M} \right\rfloor$ of the patches for every image hence, regardless of the size of the series, we use a total of N patches. This allows us to build a unique autoencoder AE that works for all pairs in the series, and to prevent overfitting.

As an example, for a series of 16 images and 600×600 useful patches per image, we would randomly sample $\frac{1}{16}$ of all possible patches for each image of the series (22,500 patches per image), and use a total of 360,000 patches to pre-train our network.

When processing the patches, our network applies a Gaussian filter in order to weight the pixels by giving more importance to the center of the patch in the RE calculus.

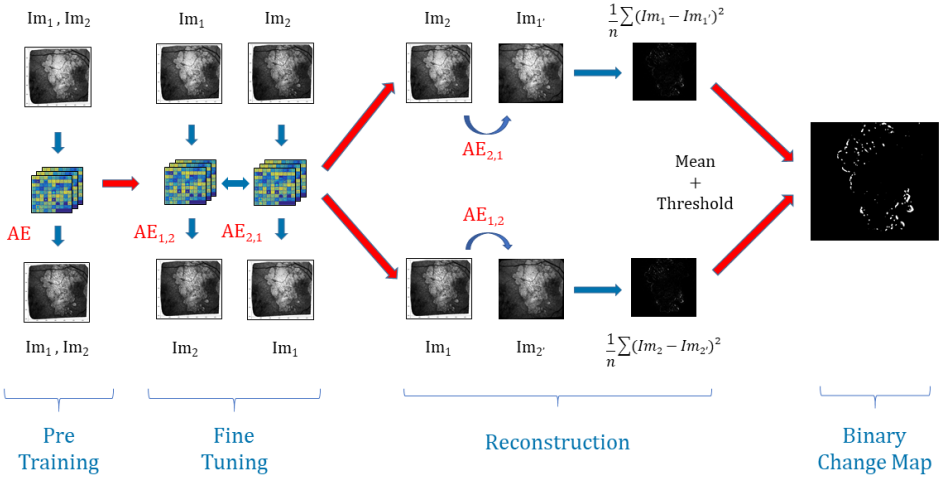


Figure 2.29: Joint Autoencoder architecture for ARMD

From there, we use the same procedure with a joint-autoencoder as with remote sensing images [99]: We first pre-train the network with a single autoencoder. We then duplicate this autoencoder to create the joint-autoencoder that we fine-tune in both temporal direction. Finally, we compute the average reconstruction error in both direction, and we apply Otsu thresholding [120]. With that, our hope is that this architecture shown in Figure 2.29 will find the lesion growth between the two images highlighted by the reconstruction error.

The architecture for the joint-autoencoder model is shown in Table 2.7 and is a fully convolutional autoencoder model. We used kernels of size 3, and a padding and stride of 1. As one can see, the network is noticeably smaller than the ones we used in Table 2.1; this is due to the simpler nature of the images, their smaller size and also a larger patch size due to difference in the resolution compared with remote sensing images.

	Fully-Conv AE for ARMD
encoder	Conv(B,16)+ReLU
	Conv(16,16)+ReLU
	Conv(16,32)+ℓ_2-$norm$
decoder	Conv(32,16)+ReLU
	Conv(16,16)+ReLU
	Conv(16,B)+ReLU

Table 2.7: Joint-autoencoder architecture for lesion evolution analysis in ARMD patients.

2.3.4.2 Results

The detailed results are available in the journal version of this work [168], but this section gives a few key ideas of what we observed.

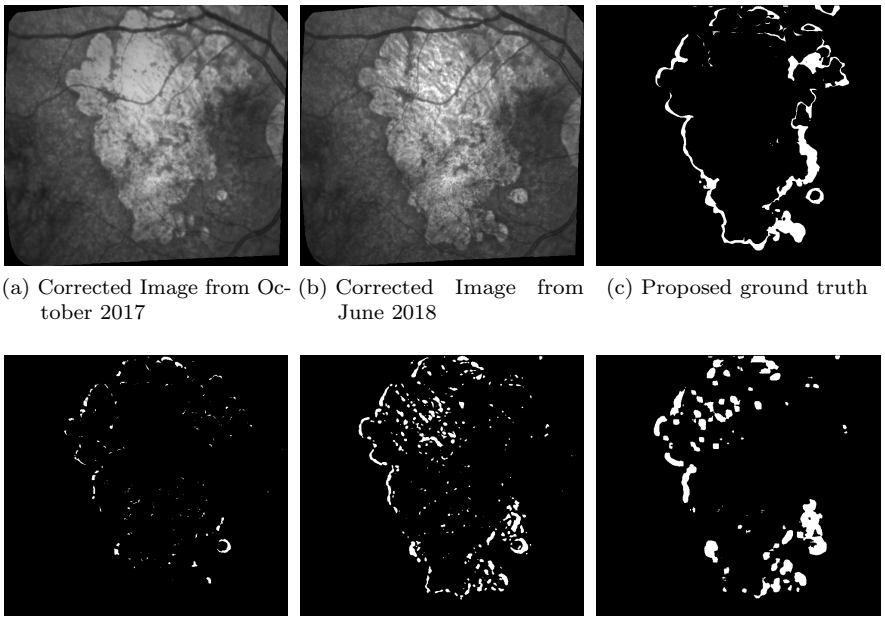
First, when working with pairs of images instead of individual images, it was quite obvious that the ground truths proposed by expert ophthalmologists had flaws: When we tried to build the change maps using the difference between the proposed expert segmentations we noticed quite a few inconsistencies that went unnoticed with individual images such as shrinking lesions (which is

not possible), holes in the middle of the lesions, inconsistent growth and so forth. In particular, the area around the optic nerve was particularly plagued with the issues, so much that we decided to discard this area for our evaluation (see Figure 2.31(d) where it is clearly visible).

Then, while we knew that the ground-truth had many issues, it is fair to say that the dice indexes were significantly lower than for the W-Net algorithm. With an average precision of 0.32, an average recall of 0.35 and an average F1 Score of 0.31, it is fair to say that our results were not great. It is difficult to this day to say if the problem came from reliability issues with the ground-truth, if it was the algorithm, or both.

Finally, the images below show some visual results so that the reader can make its own mind on the results. In our opinion, these results are still quite satisfactory, and as one can see they are better than these achieved by concurrent algorithms proposed by Celik et al. [162] -that are still quite good-, or by Kanezaki et al. [161] which are extremely noisy compared to the results of the two other methods.

As mentioned before, it can be seen from some of these figures that the ground-truth built by subtracting the doctor's segmentations are not always reliable: This is for instance the case in Figure 2.32, where both Celik approach and our method seem to achieve something closer to the truth.



(a) Corrected Image from October 2017

(b) Corrected Image from June 2018

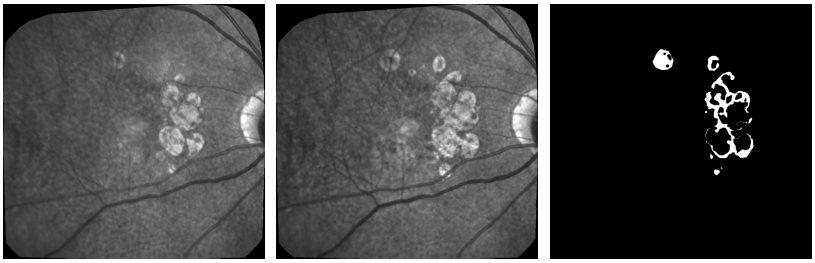
(c) Proposed ground truth

(d) Asako Kanezaki's approach, F1 score = 0.15

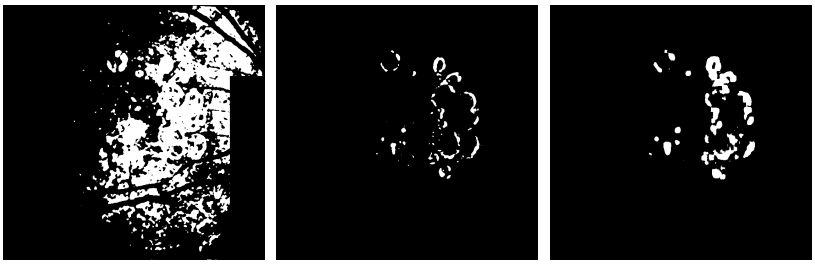
(e) Turgay Celik's approach, F1 score = 0.35

(f) Our Fully Convolutional AE, F1 score = 0.4

Figure 2.30: Comparison example of the three methods on patient 005.

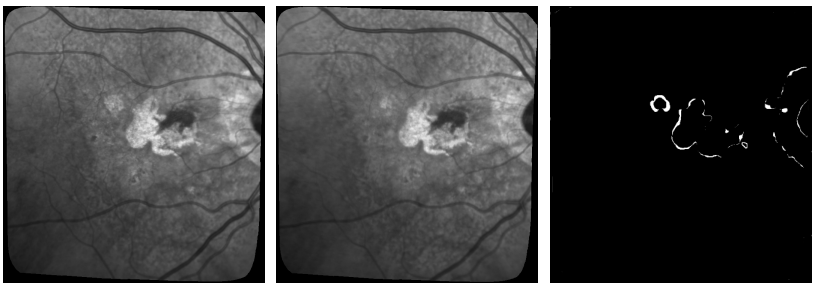


(a) Corrected Image from April 2017 (b) Corrected Image from October 2017 (c) Proposed ground truth

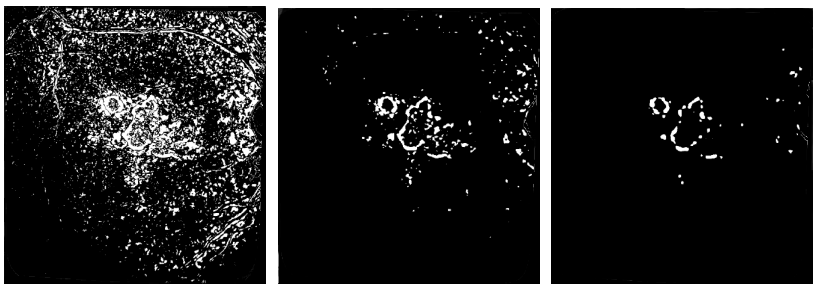


(d) Asano Kanezaki's approach, F1 score = 0.17 (e) Turgay Celik's approach, F1 score = 0.43 (f) Our Fully convolutional AE, F1 score = 0.43

Figure 2.31: Comparison example of the three methods on patient 001.



(a) Corrected Image from November 2017 (b) Corrected Image from May 2018 (c) Proposed ground truth



(d) Otsu thresholding, F1 score = 0.05 (e) Turgay Celik's approach, F1 score = 0.253 (f) Our Fully convolutional AE, F1 score = 0.38

Figure 2.32: Comparison example of the three methods on patient 010.

The results presented here have been obtained with parameters $p = 13$ (patch size) and $\sigma = 12$ (Gaussian weights) for patients with large lesions, and $p = 7$ and $\sigma = 5$ for patients with smaller

lesions. Our experiments indeed showed that different parameters gave the best results depending on the size of the lesions.

2.4 Chapter Conclusion

In this chapter, I have presented several works and applications that involved the use of deep learning methods in unsupervised contexts. In particular, the problem of change detection and time series analysis have been analyzed for applications in remote sensing and medical imaging.

My goal was to study the possibility of using strong deep learning methods for applications where no labeled data were available. With that in mind, it is my opinion that this chapter has shown that it is indeed possible to use deep learning methods for unsupervised learning tasks, albeit with some limitations that I will discuss now.

This chapter has two main components present in all the neural networks I have presented: convolutional layers which is not really a surprise for applications in image processing, and the autoencoder principle which remains key for the learning process of unsupervised networks. As for the main contribution of this chapter, it is the joint-autoencoder for change detection, which is also a core component of our GRU-based time series analysis framework.

Strangely enough, this "joint-autoencoder for change detection" contribution is an exploit of the autoencoder weakness and inability to accurately predict an image at time $t+1$ based on image t (because it can only properly map the recurrent textures of majority classes). While I am quite proud of this contribution, I can't help but find it disturbing that the core idea of this method relies solely on a neural network weakness rather than its strengths. One may wonder what guarantees we have that all interesting changes are within the minority classes of textures, but also what the risks are that the network will sometimes make better than expected predictions of the future (if we buff it enough for instance) and will thus miss the changes. Indeed, our work offers no answers to these questions.

The second important contribution of this chapter is the unsupervised image time series analysis framework. We have shown that even for complex time series like remote sensing images, it is possible to efficiently combine neural networks with clustering and graph approaches (graph synopsis) to produce very satisfactory results. However, we have also seen that the architecture to do so is quite complex -and somewhat heavy- as it uses 3 different autoencoders: The joint-autoencoder for binary change detection, another autoencoder for feature extraction, and a third one combined with GRU for the clustering. While these autoencoders share common convolutional structures, it remains extremely heavy in term of training (procedure and training time), and there is a high risk of error accumulation after each step.

This chapter has also put an accent on applications with the Tohoku tsunami case study and the eye fundus images for ARMD lesions. From these, we have seen that the models we have proposed -but also deep learning models in general- can sometimes struggle with real applications. As we have mentioned in the introduction of Chapter 1, when it is applied to real problems we usually expect unsupervised learning to find clusters that match the real classes: damaged building and flooded areas for the tsunami, healthy areas and lesions for ARMD. However, unsupervised algorithms have no notions of these classes, and in particular deep learning algorithms based on autoencoders relies only on patterns and textures. As a result, the results can sometimes be less good than expected. We

can also mention the difficulty to know in advance what will work (or not) for a given application: For instance the change detection joint-autoencoder had truly amazing results for complex remote sensing time series, but was just above average when applied to ARMD images that seemed simpler. And on the other hand, W-Nets were absolutely unpractical when we tried them on our satellite images time series, but worked very well for ARMD.

We can say after this chapter that there are solutions to use deep learning methods in unsupervised environments, and that these methods will still do better that the non deep-learning based approaches that are often still state-of-the-art. They also have the advantage that they don't need the huge amount of labeled data that is necessary to supervised neural networks to attain their top performances: a few images without labels are fairly enough for the unsupervised neural networks we have presented. However, the lack of supervision has a cost in term of performances. It also induces the risks of finding clusters that have no practical use for real applications. And finally, recycling an architecture from one application to another might be even less reliable than with supervised methods. This also opens the question of using weakly supervised method, we will discuss this point some more in Chapter 3 as this is part of some promising future works.

Ultimately, we can conclude that deep learning solutions exist for unsupervised contexts, that they work well at providing preliminary results that can be exploited. But that it is also likely that tuning these networks to achieve "good enough" results will take more time than with their supervised counterparts, and also that caution is advised when re-using an algorithm that was designed for another problem. Last but not least, it is my opinion that if labeled data are available, supervised neural networks should be preferred as they will almost always give better results.

Chapter 3

Retrospective thoughts and research perspectives

"The cake is a lie"

Portal (2007)

Contents

3.1 The road ahead for clustering related projects

In Chapter 1, I presented some of my contributions related to multi-view clustering. To be fairly honest, it is difficult to deny that this axis on multi-view clustering will probably have a lower priority than the deep learning one in the years to come. There are multiple reasons for this: I have been working on this subject since the beginning of my PhD thesis and I'd rather move on to something else, the funds and students to work on this topic are harder to find, and the lack of concrete applications makes it less enjoyable. Nevertheless, they are a few things that I can see as closing perspectives to this work and that I will briefly sum up in this section.

3.1.1 Theoretical perspectives for multi-view clustering

The work presented in Section 1.5 shows some of our preliminary results about what could become a theory of stability for multi-view clustering. This work is novel and yields a real potential to sort out the properties of the many algorithms that have been proposed for multi-view clustering, collaborative clustering, distributed clustering and unsupervised ensemble learning. Indeed, it is currently impossible to know how these methods behave by other means than empirical observations.

What we have done so far was a translation of the formalism proposed by Ben David et al. [42] from regular clustering to multi-view clustering, and exploring what basic theorems and properties we could figure out from this basis. What we should have done is to clearly separate the different cases between multi-view, collaborative, distributed and ensemble applications. The mistake that slowed us down was to try to come out with a global theory while it is almost certain that each case is different. Our future works shall take the inverse approach of starting from a clear specific case and try to generalize from it. With time, I am confident that we could come to results and properties that are more impactful and could even help guiding the way future algorithms should be designed.

3.1.2 Applications of multi-view clustering and unsupervised ensemble learning

The second part of my future works regarding multi-view clustering and unsupervised ensemble learning concerns applications for the analysis of text corpuses, with two specific applications in mind: Unsupervised recommender systems and the fusion of multi-view representations of text data [29, 69]. This will be a continuity of my collaboration with Associate Professor Juan Zamora from the Pontifical Catholic University of Valparaiso in Chile and could be done through the CONICYT-FONDECYT funds that we acquired recently (see Section 6.3). Our goal would be to greatly reduce the complexity of state of the art unsupervised partition merging algorithms that nowadays rely on the use of the graph-cut algorithm through multiple views [45, 80], which is extremely slow with large datasets and multiple views. It is our hope that by using Kolmogorov complexity [72] and the method presented in Section 1.4.3 we could reduce the complexity of the graph by first merging the clusters and views that present the lowest number of conflicts.

These projects are also subject to the evolution of the covid-19 pandemia and -to a certain extent- me getting the habilitation for which I write this manuscript: the CONICYT-FONDECYT funds we secured are restricted to scholar exchanges and joint Master or PhD students.

3.2 From time series analysis to time series prediction

In Chapter 2, I presented several contributions related to image time series analysis and change detection. It seems to me that the obvious follow-up to these works would be to move towards time series predictions. And it is part of my research perspectives in the coming years, in particular with medical images for the study of different pathologies. While there are also possibilities with remote sensing images (deforestation, ice caps melting, urbanization, etc.), it seems to me that the lower number of classes in medical images makes them easier to study from a prediction outlook, and it is also in my opinion easier to propose statistical growth models for the classes of interest in medical images. Furthermore, many patterns in landcover evolution may be a lot more random and influenced by human interventions that are impossible to predict. On the other hand, disease progression tends to follow specific pathways with less of randomness involved. Furthermore, medical doctors tend to agree that ARMD is a disease that should be easy to follow.

Within this context, one objective of Clément Royer's PhD thesis -that I co-advise with Florence Rossant and Michel Pâques- is to study the growth patterns of Age Related Macular Degeneration (ARMD), with the goal of being able to predict how the lesions will evolve after a certain number of months, and eventually to propose statistical growths models. These predictions could in turn be used to assess the efficiency of experimental treatments and drugs. Obviously I mention ARMD because it is one of my ongoing project, but the same types of predictive models could be used on other pathologies such as cancer tumors for instance.

3.2.1 Generative adversarial networks for ARMD time series predictions

Since I work mostly in unsupervised environments, some leads that I plan to explore for predictive models include Generative Adversarial Networks (GAN) [169], a recent family of neural networks specialized in generating output distributions as close as possible from their input, and that have turned out to be great at creating fake images [170][171]. The principle of GANs is to have two neural networks contesting against one another: the generator which from a known training set or distribution, tries to create new data resembling the original ones or following the distribution; and the discriminator which tries to discriminate fake generated data from original ones or those following a known distribution. From there, the two networks improve theirs models by competing against each other, and the "science of using GANs" relies on choosing the proper task that the generator should try to mimic depending on the intended final application.

Examples of GANs being used for time series predictions of future images or future videos frames already exist in the literature [172][173], and have even been applied to medical images [174] for tasks such as brain tumor growth predictions [175]. However, while GANs had originally being designed for unsupervised applications, most of the current successful prediction implementations are supervised and required labeled data.

It means that there is a room for the development of unsupervised algorithms using the principle of GANs for time series predictions. Using the power of GANs to generate realistic images, it is our team goal to be able to generate images of what a lesion and its evolution could look like after a given number of months. Promising couplings that could do this include: GANs with LSTMs [136] (with existing project about this coupling in supervised learning[1]) and autoencoders [63] with

[1] https://www.researchgate.net/project/S-LSTM-GAN-Shared-recurrent-neural-networks-with-adversarial-training

GANs.

Finally, it is worth mentioning that in case we could not find a clever idea to achieve a fully unsupervised GAN based framework for time series prediction, there is the possibility to use a semi-supervision process involving the results of the W-Nets segmentations presented in Section 2.3.3 and some doctor's manual segmentations as training data.

To conclude on this idea of using GANs for time series predictions: These networks seems to be a necessary tool for their ability to generate realistic images, which is something we need to make predictions on how the lesions may evolve. The first things that we would have to test is the ability a simple GAN to produce realistic images that look like ARMD lesions, and to see if ophthalmologists could be fooled by these. Then, we also want to take advantage of the fact that GANs are more than just neural networks, they are also a deep learning framework that can be applied to existing neural networks. It means that there are many ways to apply them to other neural networks with the goal of enhancing their abilities.

3.2.1.1 Generative Adversarial Networks as a powerful enhancement tool for existing networks

As we have just mentioned, GANs can be used as networks of their own that generate realistic images. Beyond what we already discussed, they have other useful applications in image processing or pre-processing: Image denoising [176], image upscaling to a better resolution [177], or even image transformation from one format (or image modality) to a different one [178].

But more importantly, GANs are also a learning framework that binds well with other networks and can improve the way they learn, and ultimately lead to better results. This way of using GANs is certainly something that I want to explore in the future:

- It has been shown that using an autoencoder as a generator within a GAN framework achieves better results than the regular auto-encoding process [170][179][180]. Instead of being simply trained from its reconstruction error, the autoencoder is also trained against a discriminator, which makes the results more realistic, especially when it comes to complex images. This type of architecture is interesting in the sense that it is a GAN but with the possibility of using latent features at the bottle-neck of the autoencoder used as generator.

- In the same vein, the training W-Nets within a GAN framework have also shown to greatly improve the results [181, 182]. This can be explained by W-Nets being complex autoencoders has shown in Figure 2.26. As such, adding a discriminator helps to reconstruct better images that are never really considered by the user as W-Nets usually only return the segmentation and not the reconstructed images. By having better a reconstruction, the network has better latent features, and therefore a better segmentation.

- Similar improvements [183] have been detected when combining CycleGANs [184] with U-Nets [103].

As one can see, this offers quite a few possibilities to improve several of the methods proposed in this manuscript both for medical and remote sensing images, with the goal of making them more robust. But more importantly, there are a lot of possibilities to develop new and original deep neural unsupervised frameworks based on GANs that may reach performances that are more acceptable for end-user applications.

3.2.2 Proposing mathematical growth models for ARMD

Despite being the leading cause of vision loss of those over the age of 65 in the industrialized world, and while some models and underlying mechanisms have been studied in animals [185], the mechanisms and growth patterns for humans are still poorly understood. What is known is that a first lesion will appear and grow from the center area of the macula, and then other lesions will appear in other areas (apparently at random) around the initial lesion and around the optical nerve area. The lesions will grow and merge as they start entering in contact with one another.

In my opinion it would be interesting to propose statistical models for these growth patterns, perhaps starting with simple gaussian mixture models. This would be useful in complementing the neural network approaches for several reasons: First neural networks tend to be black boxes and most likely won't help explain the underlying mechanisms for the disease even if we manage to get accurate predictions. Second, having a statistical model for the lesions could be beneficial to propose better neural generative models. Finally, we can take the problem the other way around and imagine that we could use the images or segmentations produced by neural networks (GAN, W-Net or otherwise) to figure out the model and tune its parameters.

In turn, having accurate predictions and growth models could help with finding the underlying biological causes by indicating the physicians what structures to look at and when.

3.3 Why fully unsupervised learning might be an illusion, and why we should be okay with it

When we think about machine learning, the first thing that comes to mind is often artificial intelligence. Somehow, it is a common opinion that data scientists and scholars that work on Machine Learning will produce the intelligent machines of tomorrow. This led many people from the field -myself included-, to wonder how "intelligent" are the algorithms that other people and myself are working on. Afterall, neural networks were designed with the idea of mimic the human brain, and while they are certainly a leap forward in artificial intelligence, they have so far failed to achieve the higher level of cognition that have been predicted since the end of the 90s. This raises several questions that many people in the field are interested in: Are we doing it wrong when we develop these so called AI ? What does "understanding" mean for current AI algorithms [186] ? Are these algorithms really intelligent, or just particularly good at certain tasks ? How do they compare to humans ?

In this section, I share some of my personal thoughts on the matter, and in particular I develop some ideas to study the specific case of unsupervised learning when it comes to assessing the so called intelligence of machine learning and AI algorithms.

Before focusing on the case of unsupervised learning, let us take a look at the algorithms that could be considered state of the art in artificial intelligence. When we talk about achievements in artificial intelligence, the first program that comes to the mind of data scientists and common people alike is generally AlphaGo, the computer program developed by DeepMind to play Go and that beat the Grand Champion Lee Sedol in 2016 [2], becoming the first AI to manage such a feat for the game of Go. This was an extraordinary achievement at the time because unlike chess which

[2] `https://www.bbc.com/news/technology-35785875`

has "only" 10^{50} state spaces and would require a tree of size 10^{123} to learn it by brute force, the game of Go features 10^{172} state spaces and would require a game tree of size 10^{360} to do the same. As such, it was long considered that the game of Go could not be "brute forced" by an AI, thus making AlphaGo a true achievement and an example of the power and intelligence of deep neural networks. If we now look at how AlphaGo was trained, it was mostly done by feeding it a large amount of human games to use as examples. It is therefore a supervised algorithm. A subsequent and better version of AlphaGo called AlphaGo Zero [187] was on the other hand what we call a "self-trained" algorithm: Instead of feeding it humans games or human data to learn from, it learned by playing millions of games against itself. DeepMind, that was later acquired by Google, detailed how AlphaGo and its subsequent improvements worked and it can roughly be summed up as follows: A "policy network" selects the next move to play based on a combination of smart and dynamic game tree exploration and move immediate gain evaluation. A second network called the "Value network" assesses who the current winner might be and attempts to learn how the opponent plays. In other words, AlphaGo is still a tree exploration algorithm that chooses his moves based on stastistics, but it does it in a very smart way by exploring the game tree on the fly, and also by considering the opponent style to guess what he may or may not play. Explained like that it is unclear if AlphaGo itself was any smart at all, or if all the credits should be given to its creators intelligence. And to be fairly honest, I don't think it is possible to see anything about the intelligence of an AI based on board games. Especially with a game as complex as Go, due to the depths of the game tree and the nature of the game, you can't tell if any move by the AI is genius, normal, sub-optimal, or just nonsensical. And it is the ability to detect these small quirks and mistakes that can really help us to determine how smart an AI really is.

Luckily, there is a lesser known cousin of AlphaGo developed by the same company DeepMind, and which plays a game for which such quirks, mistakes and weird moves are a lot easier to detect. AlphaStar [3] is an AI that plays the real time strategy game Starcraft II by Blizzard Entertainment, thus raising the difficulty even higher than with the game of Go with elements such as real time strategy analysis, economy management, exploration tasks and fog of war, unit composition choices, as well as unit micro-management during battle. As a Starcraft player since 1999 (reaching my peak level in the top 10% European ELO ranking in late 2011) and scholar working in the field of Machine Learning, I could not pass this possibility of analyzing how smart an AI really is. Very much like AlphaGo, AlphaStar is a self-trained algorithm [188] that uses several agents. And also like AlphaGo, it did beat the grand champions of Starcraft 2 (albeit after the equivalent of 200 years of training against itself). But, and this is what is interesting about AlphaStar, it sometimes looses and when watching many of its games it is possible to spot some of these quirks I mentionned earlier. I won't enter into technical details that won't speak to people that don't know Starcraft II, but in many occasions it is possible to see it when AlphaStar is thrown off (or taken off guard) by something that takes it out of what it knows: it starts to act weirdly and make decisions and moves that make absolutely no sense. In some occasions, some players have even spotted AlphaStar doing what is colloquially known as "derping", meaning that it moments it was stuck in loops of contradictory instructions, or sequences of instructions that were very reminiscent of what could be observed with old AI following simple decision trees. Furthermore, despite hundreds of years of learning, some of the basic defense mechanics used by even average human players have either

[3] https://deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning

never been learned or have been discarded as not optimal enough by AlphaStar.

In my opinion, AlphaStar is certainly the most advanced AI these day, and it is the one that does the most complicated tasks. I am extremely impressed by what it is capable of, and by the perfect execution of whatever it decides to do. And yet, from what I could observe, it cannot be qualified as "smart", nor is it capable of creativity since experiences have proven that it cannot solve simple situations (that an average human player could easily solve), if it has never encountered a similar enough situation before. Furthermore, its AI nature is also often displayed by inexplicable quirks, even if said quirks don't often result in the AI losing. And so, we can probably conclude that 2 of the smartest and well-known AI these days, AlphaGo and AlphaStar are not actually intelligent. And there is even more disturbing: both of these are supervised neural networks ! Indeed, even if they are self-trained against themselves, the game moves are still being labeled as winning or losing moves. And so, we see that these very advanced AI are in fact advanced Bayesian decisions trees that are very deep indeed due to the humongous training process they went through (and which is far beyond human capability), but that ultimately are not that intelligent. We can therefore wonder what we could say about the unsupervised algorithms that are not even given a specific thing to learn, and are left to find things on their own without even a reward system when they do great.

Of course, my research is much less advanced than DeepMind: I do multi-view clustering and image analysis. However the questions of what the algorithms I am working on actually learn, and how smart they are, is no less valid. And once again, in unsupervised learning, you don't even tell your algorithm what it is supposed to learn. In the next two subsections, I will present some of my ideas related to the (lack of) intelligence of unsupervised learning, how smart it really is, the impossible things we expect from it, and why it sometimes works.

3.3.1 Measuring how smart unsupervised deep learning algorithms really are

To have an idea of how smart unsupervised deep networks could be, let us consider 3 examples from this manuscript:

- The deep cooperative reconstruction system from Section 1.3.
- The joint-autoencoders for non-trivial change detection proposed in Section 2.2.2.
- The W-Net from Section 2.3.3 that we used for the automated segmentation of ARMD lesions.

The deep reconstruction system mostly relies on the principle of information compression and encoding coupled with a weighting mask system to optimize the reconstruction. There is no intelligence here, just a purely mathematical optimization process using gradient descent. However, if we take a look at Figure 1.6 that shows some failed reconstructions on the MNIST-like dataset dataset and if we consider that a random forest algorithm was able to correctly identify what numbers these were supposed to be, we can safely assume that in this experiment both the deep reconstruction network and the random forest algorithm clearly had a different intrepretation than us of what numbers should look like. Not only it is unclear what these algorithms actually learned about number representation, but it is clear that whatever it was wasn't intelligent at all.

Moving on to the joint-autoencoders, we clearly explained that the key principle of this method to detect non-trivial changes was to look for areas of the reconstructed images where the algorithm was making mistakes. By construction, this neural network is not intelligent. The principle is smart and it works relatively well, but the network itself has no intelligence in it.

Finally, we end up with the most interesting case: The W-Net that we used for the automatic segmentation of ARMD lesions. This is an interesting case in the sense that unlike the two other networks, this one is typically what we expect of a neural network in an unsupervised learning context: We expect it to find something very specific (a lesion, a tumor, cats, cars, etc.) in images but without us telling it what it is supposed to find, simply because we did not have the time or money to have someone feeding it thousands of examples of whatever it is supposed to find should look like. Given that we are talking about medical applications, and even if ARMD is not a life threatening condition that should be operated, I came up with the following question: Is asking a W-Net where the lesions are in a medical image the equivalent of asking a 5 years old kid what he or she finds interesting in the same image ? Let us consider this comparison for a moment. The 5 years old kid and the W-Net have many things in common when it comes to detecting ARMD lesions (or any kind of lesions for that matter):

- Both of them have a good visual abilities to analyze shapes and textures (granted that the convolutional layers of the W-Net have been well configured).
- Both of them can draw stuffs -with perhaps the W-Net being slightly better- and could inpaint whatever it is they find interesting in the image.
- Neither of them has any idea about what an ARMD lesion is, or should look like.

And so after discussing this problem with colleagues that actually work with children to study their cognitive abilities to solve algorithmic problems, we thought that it could actually be an interesting experiment to do. The goal would be to assess the mental age of the W-Net algorithm compared with human children by assessing from what age the cognitive bias and external knowledge (which is a form of supervision) of human beings would make them better than W-Nets at figuring out what could be interesting in these eye fundus images. The protocol would be relatively easy to design to have a fair comparison; different age groups would be given the same images than the W-Net during its training process. And after letting them look at all of them, we would simply ask them the same thing than to the W-Net: "paint the elements of the images that you find remarkable, and regroup similar elements within the same color". I am actually genuinely curious to see what the result of such experiment would be, and if some of the misidentifications by younger children would be the same as the ones made by W-Nets. It is certainly a kind of experiment I look forward to do in the next years and that could shed some light on the "intelligence" of unsupervised deep learning methods for this type of task.

3.3.2 Is there a massive reinforcement learning bias in all successful unsupervised learning applications ?

From what has been discussed so far in this section, it could appear that unsupervised learning is doomed to failure when it comes to solve real problems. And yet, there are many works (not only from this manuscript) that show that it can achieve fair to good results for many real tasks. So, one may wonder what it is that makes deep learning work fine in unsupervised settings when it seems that these algorithms are not only hardly intelligent, but are also seemingly left totally unsupervised to figure things on their own. But are they really left on their own and unsupervised ?

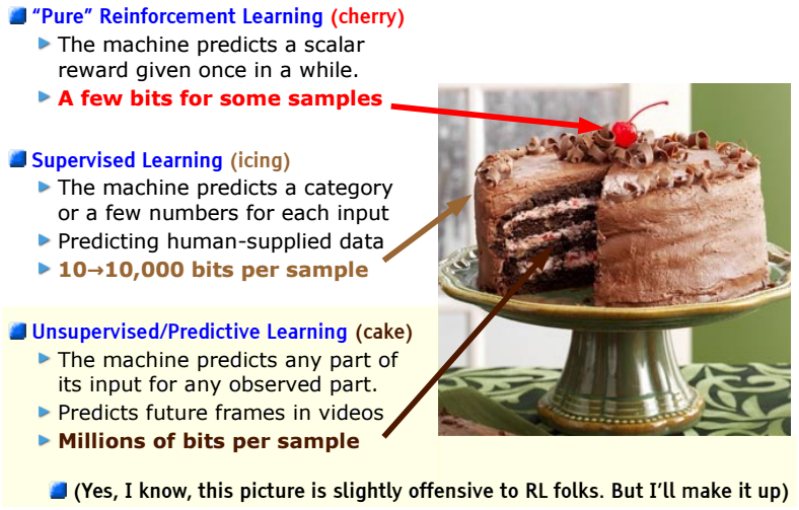


Figure 3.1: The cake analogy, by Yann Le Cun

In his original cake analogy (see Figure 3.1), Professor Yann Le Cun explained that unsupervised learning was most of the machine learning cake, that supervised learning was the icing, and pure reinforcement learning the cherry on top of the cake. If this analogy is true, given what we currently see with unsupervised learning publication, then I think that this particular cake is a black forest cake. If you are not familiar with it, it has cherries on top of it, but also inside of it.

I believe that everyone working with unsupervised learning algorithms (deep or not) already had this experience where we, or one of our student, spent days or weeks tuning the parameters of an algorithm until it gave good results for the applications of our choice. And perhaps some of us wondered why the results were improving slowly but surely as new parameters, objective functions or configurations were tested (See Figure 3.2 for an illustration).

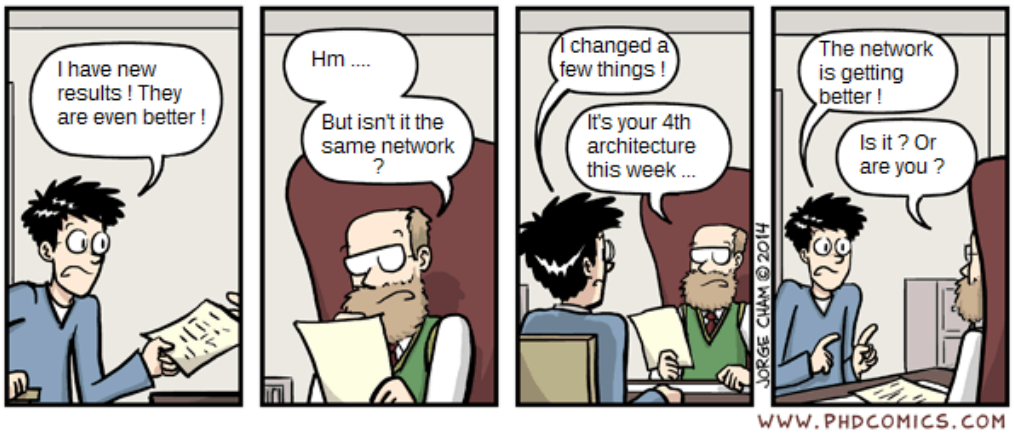


Figure 3.2: Is the network getting better ? Is Clément getting better ? Or is Clément unknowingly doing reinforcement learning on the network architecture and parameters ?

I choose to call this a "hidden reinforcement learning bias", but you could say it is just manual evolutionary optimization of the parameters, or human guidance of the algorithm. In my opinion it cannot be qualified as evolutionary optimization of the parameters since it is impossible to define

an objective function without labeled data. The reward here comes from the user (or algorithm architect), so it has to be reinforcement learning. Whatever the case, it is my strong belief that the secret of many successful unsupervised algorithms is to have a human in the loop that more or less conscientiously guide the algorithm: successful pieces of architectures and parameter settings are rewarded and kept for ulterior attempts, and unsuccessful ones are discarded. And we keep improving the architecture through trials and errors based on the user impression of the algorithm result's quality. I call it a "hidden reinforcement learning bias" not only because it looks like manual reinforcement learning, but also because the moment we have an external user giving a feedback on whether a result is good or not and changing the architecture to improve things, then we have introduced a bias and we are not in a fully unsupervised setting anymore. The algorithm is unsupervised, but the setting is not since many hyper-parameters are guided via reinforcement learning.

Since I have observed this type of algorithm guidance to achieve better results with my own algorithms and with these of my students (See Figure 3.2 again), I took the time to document it for the ARMD lesion segmentation algorithm: I asked one of my interns working on W-Nets to keep track of all the changes he made to his network architecture during his 5 months internship and to keep a record of the dice indexes as the unsupervised network was seemingly getting better and better. The results are shown in Figure 3.3 and clearly demonstrate the phenomenon of manual guidance of the algorithm through external supervision, as well as trials and errors to modify the architecture. While we can probably ignore the very first point which was a preliminary result, there is a 11 points gain on the F1-score between the second set of parameters tried and the one that was finally kept.

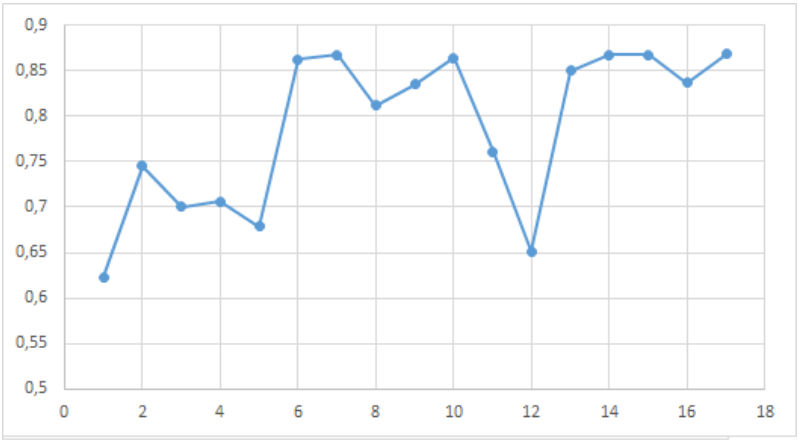


Figure 3.3: Example of manual optimization – Evolution of the F1-score through 17 iterations of the W-Net architecture: Layer drop-out was added for model 6, models 11 and 12 were an attempt at doing more epochs and adding extra clusters (to be merged later), and models 16 and 17 had modified pooling layers.

It is worth mentioning that while such a bias exist for both deep and regular unsupervised learning algorithm, the problem is probably worst in the case of Deep Learning where the number of parameters, types of layers, filters, types of activation functions and many others, leave a lot of room for improvement between the results of an architecture that was taken right off the shelf, and one that was carefully tuned towards providing good results for the "unsupervised" target

application. This raises a certain number of questions :

- Is this type of supervised manual reinforcement learning cheating ? I believe that everyone does it, in particular when it comes to publishing and because of the pressure to beat the other methods. The real conundrum is to decide whether or not it is fair to do it only for the method we want to promote, or if it should also be done for the methods with which we compare our algorithm. One the one hand, the comparison seems unfair if we don't do it. On the other hand, if we change the architecture of competing method, then we are not making a comparison with the exact network from the algorithm we cited.
- Can we still call our method "unsupervised" when it is obvious that we introduce some supervision in the form of algorithm guidance ? And does a fully unsupervised application really exist ?
- Can we quantify the trade-off between spending time doing this type of architecture optimization, or deciding to spend time labeling the data and go for a supervised algorithm ?

To the first question, I have the beginning of an answer: in many cases when you try to publish a new unsupervised algorithm for real applications, you have to do an unfair comparison with a plethora of supervised methods, and you have to use supervised indexes which your algorithm was not designed to optimize in the first place (and quite often with ground-truths that you don't have). Otherwise no-one actually believes that your algorithm works. It is therefore only fair that we manually tune our proposed methods to be more competitive. As for comparison between unsupervised methods, well, I believe the right thing to do would be to try to tune all of them for fairer comparisons.

I have no clear answer at this stage for the other two questions, but I think that these should be studied seriously, and that because we are having these questions, we should consider the importance of keeping at least a bit of supervision in unsupervised environments.

3.4 Introducing supervision in unsupervised environments

3.4.1 Humans in the loop

As we have seen before, it is difficult for fully automated (unsupervised) methods to provide fully satisfactory results in term of accuracy for real applications. Indeed, applications such as client segmentation require a high accuracy to be satisfactory, land cover analysis in remote sensing is also a high precision task, and when it comes to medical image segmentation it is also extremely important to have very high quality results. For all of these applications, a fine quality control through visual analysis of human operators is required in order to assess the genuine performance of the methods. Furthermore, we have mentioned many times the issue of the lack of ground-truth, and reliability issues when this ground truth exists. As a consequence, we may question the quality of these methods, and it is obvious that human intervention remains necessary at some point.

The benefit of human experts has already been assessed in various domains. Nevertheless, involving a large amount of human analysis should be avoided due to reproducibility and fairness issues that we have mentioned in the previous section. It therefore seems that the main research

challenge is to find the minimal conceivable amount of human involvement. There is actually a research field called crowd-sourcing which revolves around -given a set of human experts- estimating or weighting their individual contribution and relevance [189] to take the final decision [190], or to refine accuracy assessment results [191].

Several levels of human intervention can be considered and are detailed thereafter.

3.4.1.1 Human intervention before the unsupervised learning process

Human intervention before the learning can take two forms: data pre-processing and quality control; or the introduction of a few labeled data in a process known as semi-supervised learning.

Data pre-processing and quality control is extremely important before any unsupervised learning task since we know in advance that these algorithms tend to be less effective than their supervised counterpart. As such, a pre-assessment of the data is an important step to rule out any quality issue that could negatively affect the results, but also to detect if the data are a good fit or not for the algorithms that we have chosen. Indeed, the introduction of Chapter 1 about clustering clearly explained how these algorithms have models that match with specific distributions only. As for deep learning algorithms applied to images, it should be obvious to anyone that a human intervention is necessary to set up the patch size and eventually modify the first convolutional layers depending on the image resolution and quality.

Semi-supervised learning [192] is another possible human intervention that could happen before running the algorithm and would involve labeling a few data and feeding them to the algorithm. While this idea has proved useful to improve clustering results [193], in practice it requires to have algorithms that are compatible with this type of learning. This is not the case for most clustering and unsupervised deep learning algorithms. In the same way that supervised learning algorithms have been adapted to be able to benefit from a few extra unlabeled data, it would be interesting to proposed way of adapting unsupervised algorithms to do the same with a few labeled data.

Finally, self-supervision is another process that can be useful as a pre-training step to guide an unsupervised algorithm. The principle of self-supervision is to have a supervised algorithm learning to predict a subset of the data (or features), using the rest of the data (or features). It can therefore be seen as a specific case of unsupervised learning where the goal is to train the network to recognize what the final user really cares about inside the data. It is for instance widely used in language processing to predict the next word in a sentence without any supervised learning. Within this framework of learning, the human intervention would consist in choosing which part of the data to withhold and how to best guide the self-supervision process to learn what is most relevant. While they have encounter a great success [194][195][196], these methods are so far also limited to enhancing supervised learning algorithms. Very much like semi-supervision, rethinking unsupervised algorithms -and even unsupervised frameworks- could be useful to benefit from the advantages of these techniques.

3.4.1.2 Human intervention during the learning process

This is obviously the most interesting research perspective, and one that we have already mentioned in section 3.3.2 where we have shown how human intervention helps a lot at finding the best architecture and parameters.

A more clever solution that could be automated would be to use the principle of Active Learning (AL) [197], another type of semi-supervised learning, where an oracle (a human here) guides the algorithm through the learning process. Unlike in passive learning (and in the process we discussed in section 3.3.2), it is the learner (the algorithm) that actively seeks answers from the oracle, thus making the process a lot more effective.

Once again, such active learning techniques would blur some more the line between unsupervised and supervised learning. But since most practical applications seems to involve very much supervised expectations, it may be a necessary evil. Furthermore, these techniques have proved effective for supervised learning [198], and have been shown to reduce the number of data needed to train deep learning networks [199].

3.4.1.3 Human intervention after the learning process

As we have been discussing since the beginning of this manuscript, when it comes to unsupervised learning and real applications there is one human interaction that can't be avoided regardless of the quality of the results: mapping the clusters to classes of interest.

This task is relatively easy when the number of clusters is small, identical to the number of classes and if they match well. It is a whole other story when the number of cluster increases and when we have mismatches as indexes such as cluster purity and entropy tend to become less reliable to make enlighten decisions [200]. It is therefore my opinion that while this post-learning human intervention is necessary and can even be useful to rate the quality of the result, it is too late to improve anything at this stage. Useful human intervention to improve the quality of the results should therefore come earlier in the process.

It is however important to mention that while post-learning human intervention is too late to improve an unsupervised learning process, it is still common to have human intervening after for a process called "pseudo-labeling" [201, 202]. In this case, unsupervised learning can be seen as a pre-processing step requiring human intervention before the use of a supervised algorithm. The idea is simple: The clusters or segments produced by the unsupervised algorithm are processed by a human user that samples the most reliably annotated data that will then be fed to a supervised algorithm [203]. Combined with transfer learning from other labeled datasets [204], this type of technique is very common but still suffer from many difficulties: the manual sampling of enough reliable pseudo-labeled data is a difficult task and mistakes may result in error amplification during the classification process. As for the transfer learning task, while progresses have been made, it still remains difficult and it is an active research area, especially for deep learning algorithms.

3.4.2 One shot learning

Aside from putting humans on the loop, there is another lead that could be exploited to improve upon unsupervised learning algorithms. One of the human brain greatest ability is that it can learn about a new object or concept even if it sees it only once. The equivalent of this ability in Machine Learning is a field called *One shot learning* [205]. Like for the human brain, the idea is that each class of object could be learned only based on a single labeled example which should give the best and the most generalized description of the class it represents.

Very much like semi-supervised learning or active learning, using such techniques would require significant modifications of the existing algorithms, or even to redesign them entirely. It is however

undeniable that adding even low quality one shot learning abilities to an unsupervised algorithm would help greatly to achieve clustering results that are more stable, and also allow for an automatic cluster to class mapping without human intervention. It is my opinion that if one shot learning can improve results quality at a low cost (one labeled example per class is a cheap price), if it solves the problem of clustering stability, and if it reduces the amount of required human intervention, then it is the most promising research perspective to get closer to efficient self-supervised algorithms.

Part II

Curriculum Vitae

Chapter 4

Employement and education

4.1 Civil Status

M. Jérémie SUBLIME Born on May 29th, 1989
French Nationality
Home Address :
3 Mail Jean Zay
La Plaine St Denis
93210 Saint Denis

Associate Professor at ISEP
Work Address :
ISEP, Office L303
10 rue de Vanves,
92130 Issy Les Moulineaux.
Tel.: (33)-1 49 54 52 19
email: jeremie.sublime@isep.fr

4.2 Employement

09/2016–Now	:	**Associate Professor at ISEP** Data Science Department, Member of LISITE, Team DaSSIP
01/2018–Now	:	**Researcher** at *Laboratoire d'Informatique de Paris Nord* Member of team A3, Research group ADA, LIPN - CNRS UMR 7030
11/2016–12/2017	:	**Associate Researcher** at *Laboratoire d'Informatique de Paris Nord* Member of team A3, LIPN - CNRS UMR 7030
09/2015–08/2016	:	**Teaching Assistant at University Paris 13 – Institut Galilée** Associate PhD student,Member of team A3, LIPN - CNRS UMR 7030
10/2013–08/2016	:	**PhD student at AgroParisTech/INRA** Member of team LINK, INRA - UMR MIA 518
04/2011–08/2011	:	**R&D Intern at Astrium EADS**

4.3 Education

11/2016	:	**PhD in applied Computer Science** **AgroParisTech** (Université Paris-Saclay) - France
10/2013	:	**Engineer's degree in Software Engineering, HCI Major** **EISTI** Cergy - France
08/2013	:	**Master of Computer Science and Information Engineering** **Inha University** - South Korea

4.3.1 PhD thesis

My PhD thesis subject was "*Contributions to collaborative clustering and its potential applications on very high resolution satellite images*". I defended it on November 9th 2016, in front of the following jury:

Dr. Michael Aupetit, HDR	QCRI – Hamad Bin Khalifa University	Reviewer
Pr. Younès Bennani	Université Paris 13	Thesis co-Director
Pr. Antoine Cornuéjols	AgroParisTech	Thesis Director
Pr. Pascale Kuntz	Polytech'Nantes, Université de Nantes	Reviewer
Pr. François Yvon	Université Paris-Saclay	Examiner, Jury President

4.3.2 Master's thesis

My Master's thesis subject was "*A Genetic Algorithm with Constraint Sastisfaction Problems for multi-objective Optimization in Workflow Scheduling*". I defended it in Spring 2013, in front of the following examination committee:

Pr. Sang-Chul Lee	Inha University	Examiner
Pr. Geun-Sik Jo	Inha University	Academic Advisor
Pr. Min-Seok Song	Inha University	Examiner

Chapter 5

Teaching activities

5.1 Teachings

	Course	Level	Project	TP	TD	CM
2020–2021						
ISEP	Analyse de données*	ING2	-	2×21	-	2×21
ISEP	BDD & Technos web	ING1	51	-	-	4
IPP	Datastream Processing*	M2	-	8	-	4
2019–2020						
ISEP	Analyse de données*	ING2	-	21	-	21
ISEP	BDD & Technos web	ING1	40	-	-	4
IPP	Datastream Processing*	M2	-	8	-	4
2018–2019						
ISEP	Analyse de données*	ING2	-	21	-	21
ISEP	Programmation web*	ING2	40	-	-	6
ISEP	Machine Learning*	ING3	-	-	21	-
UPSaclay	Datastream Processing*	M2	-	6	-	3
2017–2018						
ISEP	Analyse de données*	ING2	-	21	-	21
ISEP	BDD & Technos web*	ING1	51	-	-	4
ISEP	Java & Algorithmique	ING1	-	-	-	2×13.5
ISEP	Machine Learning*	ING3	-	-	21	-
UPSaclay	Datastream Processing*	M2	-	6	-	3
ISEP	Introduction to Data Science*	B3	-	9	-	6
2016–2017						
ISEP	Analyse de données*	ING2	-	21	-	21
ISEP	Java & Algorithmique	ING1	-	2×35	-	2×12
ISEP	BDD & Technos web	ING1	50	-	-	-
2015–2016						
UP13	Bases de données avancées	M1	-	27	-	3
UP13	Python et robotique	L1	-	20	15	-
2014–2015						
UP13	Applications Web	M2	-	36	-	15
UP13	Génie Logiciel	L2	-	-	9	-
Total :			232h	316h	66h	233h

This table contains all the courses I taught since I started as a Teaching Assistant and then as an Associate Professor at ISEP. Courses marked with an "$*$" were taught in English, and all others in French.

The naming conventions for courses and student levels are the following: L for *licence* (French equivalent of the bachelor's degree in 3 years), M for *Master*, ING for Engineer students (year 3 to 5 of higher education in France), and B for Bachelor students (exchange students from foreign universities in most cases). Any of these letters will be followed by a number indicating which year

of study is considered: For instance “M1” is the first year of Master’s degree.

5.2 Administrative responsibilities

5.2.1 Module responsibilities

Table 5.1 below shows the module I was in charge of. While it does not necessarily involve actually teaching in the module, this type of responsibility includes:

- Creating or updating the program of the course.
- Creating or validating the exams and projects for the course.
- Managing and often recruiting the different professors, teachers and teaching assistants.
- Being the interface between the students, the administrative staff and the heads of department for this module.

	Module Name	Level	Students	ECTS	Period
ISEP	Analyse de données	ING2	30	5	Spring 2021
IPP	Datastream Processing	M2	35	2.5	Spring 2021
ISEP	Analyse de données	ING2	45	5	Fall 2020
ISEP	Analyse de données	ING2	30	5	Spring 2020
IPP	Datastream Processing	M2	39	2.5	Spring 2020
ISEP	Analyse de données	ING2	59	5	Fall 2019
ISEP	Analyse de données	ING2	74	5	Spring 2019
UPSaclay	Datastream Processing	M2	45	2.5	Spring 2019
ISEP	Analyse de données	ING2	71	5	Fall 2018
ISEP	Analyse de données	ING2	71	5	Fall 2018
UPSaclay	Datastream Processing	M2	42	2.5	Spring 2018
ISEP	Java & Algorithmique	ING1	184	6	Spring 2018
ISEP	Analyse de données	ING2	61	5	Fall 2017
ISEP	Java & Algorithmique	ING1	127	6	Spring 2017
ISEP	Analyse de données	ING2	59	5	Fall 2016
UP13	Applications Web	M2	41	4	Fall 2014

Table 5.1: Module responsibilities

5.2.2 Specialties, Majors and double degrees responsibilities

- 09/2019–Now - Head of the Data Intelligence Major (responsable de parcours) at ISEP.
- 09/2017–10/2020 - Head of the Business Intelligence Major (responsable de parcours) at ISEP.
- 10/2017–Now - ISEP coordinator for the Data Science Master of Paris Saclay University and then from Paris Polytechnic Institute (IPP).

In September 2017, I took over Pr. Raja Chiky as the Head of the Business Intelligence Major at ISEP. In engineering schools, major correspond to specialties that students choose for their last 2 years before graduation. Heads of majors are generally tasked with articulating and updating

mandatory and optional courses that are given within their majors, and with defining eventual prerequisites to join in their major. Other responsibilities include the validation of internships, semesters abroad and "*alternance* programs" (co-op programs) followed by students of the major, as well as individual career counseling. In September 2019, the Business Intelligence Major became the Data Intelligence Major with an update of the program that among other things included a stronger focus on Artificial Intelligence and Machine Learning, and the possibility for *alternance* students (co-op students) to join in this major.

My role as ISEP coordinator for the Data Science Master of Paris Saclay and Paris Polytechnic Institute revolves around participating in the selection process, attending various meetings, taking charge of one module (Data Stream Processing, see Table 5.1), participating in internship defenses for 5 students every year, and being the interface between ISEP and the Master for credits and diploma validations.

5.3 Students follow-up

The list below gives an account of the engineer students I followed each year that were in a part-time cursus with 50% of their time spent in companies (co-op students). This type of follow-up includes the reading of quarterly reports about the work done in their companies, and at least 2 meetings per year with them and their corporate tutor to discuss progresses, and assess the adequation between the student professional project, the courses they follow, and the tasks assigned to them in their company.

Miss Salma Mrassi	Alternance	SNCF	09/2019–Now
M. Timothée Pionnier	Alternance	ORANGE	09/2019–Now
Miss Nelly Lahmar	Alternance	SFR - Altice	09/2018–09/2020
M. Nikola Milojic	Contrat Pro.	BNP Paribas	09/2018–09/2019
M. Ilyas Bentayeb	Alternance	SFR	09/2016–09/2019
M. Nikola Milojic	Alternance	SFR	09/2016–08/2018

5.4 Teachings in thematic schools

- July 2019 - Invited Professor at the Transilvania University of Brasov (Roumania) for the *MLASS 2019* summer school on Deep Learning.
- December 2016 - Invited Professor at the Sidi Mohamed Ben Abdellah University of Fès (Morocco) for the *ETA'16* Fall school on Deep Learning and Data Science.
- June 2015 - Tutorial on collaborative clustering applications given at the *FOCOLISE* summer school organized by the ICube laboratory in Strasbourg.

Chapter 6

Research related activities

6.1 Students supervision

6.1.1 PhD students

12/2020 – Now : Miss Nan Ding

- Subjet : *Segmentation of the Eye DuraMater for 3D Modeling of Axons and Optic Nerves*
- Thesis Directors : Pr. Florence Rossant (30%) & Pr. Michel Pâques (10%)
- Thesis Advisors : Dr. Hélène Urien (50%) & Dr. Jérémie Sublime (10%)
- Fundings : IHU FOReSIGHT scholarship (50%) and ISEP scholarship (50%)

11/2020 – Now : M. Clément Royer

- Subject : *ARMD progression analysis using deep learning methods*
- Thesis Directors : Pr. Florence Rossant (40%) & Pr. Michel Pâques (10%)
- Thesis Advisor : Dr. Jérémie Sublime (50%)
- Fundings : Sorbonne University doctoral contract

10/2017 – 09/2020 : Miss Ekaterina Kalinicheva [graduated]

- Subject : *Unsupervised Satellite Image Time Series Analysis using Deep Learning Techniques*
- Thesis Director : Pr. Maria Trocan (40%)
- Thesis Advisor : Dr. Jérémie Sublime (60%)
- Fundings : Sorbonne University doctoral contract
- Remarks : 9 co-publications

01/2016 – 12/2018 : M. Denis Maurel [graduated]

- Subject : *Contributions to inter-views communications applied to collaborative learning*
- Thesis Director : Pr. Raja Chiky (20%)
- Thesis Advisors (after december 2016): Dr. Jérémie Sublime (50%) et Dr. Sylvain Lefebvre (30%)

- Fundings : ISEP scholarship
- Remarks : 4 co-publications

6.1.2 Interns

04/2020 – 09/2020 : M. Clément Royer (M2, Sorbonne University)

- Sujet : *ARMD evolution analysis using unsupervised deep learning methods*
- Advisors : Pr. Florence Rossant & Dr. Jérémie Sublime
- Fundings : XV-XX Hospital research funds
- Remarks : Integrated as a PhD student

09/2019 – 02/2020 : M. Guillaume Dupont (ING2, ISEP)

- Subject : *Deep learning techniques applied to the study of ARMD medical time series*
- Advisors : Pr. Florence Rossant & Dr. Jérémie Sublime
- Fundings : ISEP Internship scholarship
- Remarks : 2 co-publications

07/2019 – 12/2019 : M. Matthieu Pombet (ING2, ISEP)

- Subject : *Machine Learning and Data Mining to identify programming beginners? strategies when solving programming exercises*
- Advisors : Dr. Patrick Wang, Dr. Ilaria Renna & Dr. Jérémie Sublime
- Fundings : ISEP Internship scholarship

6.1.3 Master's and Bachelor's thesis

03/2019 – 07/2019 : M. Ken Chen (B4, Nanjing University of Aeronautics and Astronautics)

- Subject : *Kolmogorov complexity based collaborative clustering*
- Advisor : Dr. Jérémie Sublime
- Context : Exchange student, Bachelor's thesis

6.1.4 End of study projects and other research projects

10/2020 – 02/2021 : M. Corentin Le Guevel, M. Clarence Lacombe, M. Sunny Raj Mangu, M. Jérémie Mear & M. Thibaut Eschoua (ING3, ISEP)

- Subject : *Assessing the percentage of vegetation in French cities using Deep Learning on remote sensing images*
- Advisors : Dr. Jérémie Sublime
- Context : End-of-study project

10/2020 – 02/2021 : M. Baudouin Naline, M. Gégoire Fessard, M. Mingyang Li & M. Zheqi He (ING3, ISEP)

- Subject : *ID picture analysis using Neural Networks*
- Advisors : Dr. Jérémie Sublime
- Context : End-of-study project

04/2019 – 06/2019 : M. Evander Deocariza-Nee (B4, Stanford University)

- Subject : *N-grams Naive Bayes Classifier for Block-Based Programming Exercises Analysis in Educational Data Mining*
- Advisors : Dr. Patrick Wang, Dr. Ilaria Renna & Dr. Jérémie Sublime
- Context : Stanford University Overseas Study Program in Paris, Research project

10/2018 – 02/2019 : M. Alexandre Gay, M. Mathieu Hinh, M. François Robard & Miss Yue Zhao (ING3, ISEP)

- Subject : *Breaking reCAPTCHAv2 using Convolutional Neural Networks*
- Advisors : Dr. Jérémie Sublime
- Context : End-of-study project

10/2018 – 02/2019 : M. Renaud Saggio, M. Loann Barraud, M. Ke Fang, Miss Syrine Radhouane & M. Quentin Lucas (ING3, ISEP)

- Subject : *Study of various methods for image segmentation*
- Advisors : Dr. Jérémie Sublime
- Context : End-of-study project

09/2018 – 12/2018 : M. Yizhi Li (B3, Beijing University of Posts and Telecommunications)

- Subject : *Machine learning for satellite imagery analysis*
- Advisors : Dr. Jérémie Sublime & Miss Ekaterina Kalinicheva
- Context : Exchange student, Research project

6.2 International collaborations

I have an ongoing collaboration with Associate Professor Juan Zamora Osorio from the Pontifical Catholic University of Valparaiso in Chile. Our joint work is concerned with application of information theory based multi-view clustering to text corpuses. Such applications include text corpuses multi-view segmentation, text recommander systems, and more theoretical work on unsupervised ensemble learning. Professor Juan Zamora visited me at ISEP in January 2019 (1 week) and

November 2019 (2 weeks), and our collaboration resulted in a conference paper being published and a grant from the CONICYT-FONDECYT program that should allow me to visit if the Covid-19 pandemic situation improves.

I am also working on a regular basis with Assistant Professor Pierre-Alexandre Murena from Aalto University (Helsinki, Finland). This is a joint research collaboration with Basarab Matei from Sorbonne Paris North University and it resulted in several of the results presented in this manuscript about clustering stability and other theoretical aspects of multi-view clustering.

6.3 Projects and fundings

Project "*Institut Hospitalo-Universitaire FOReSIGHT*" (2021–2023): ISEP and the 15-20 Hospital Clinical Imaging center participated in this call and proposed a project to study various eye pathologies such as glaucoma with Machine Learning methods. Through this project, we funded half a PhD scholarship for Miss Nan Ding.

Chilean grant CONICYT-FONDECYT, project 11200826, (2021–2023): This project carrier is the Pontifical Catholic University of Valparaiso in Chile, and I am officially involved in it as a foreign expert invited on the project. The grants accounts for 62 millions Pesos (66k€) that can be spent in travel expanses for invited professors, or for interns and PhD students. The main thematic of the project revolves around text recommendation systems and is an application of my work on multi-view clustering with Associate Professor Juan Zamora Osorio.

ANR project COCLICO, ANR-12-MONU-0001, (2012–2016): "*COllaboration, CLassification, Incrémentalité et COnnaissances*". This project funded my PhD thesis and some of my early post-graduation works. The goal of the project was to develop collaborative and incremental machine learning methods that could be applied to remote sensing images. The partners involved were: the ICube and LIVE laboratory (Strasbourg University), the LIPN (University Paris 13), AgroParisTech and the UMR ESPACE DEV (University Montpellier 2).

6.4 Scientific animation

6.4.1 Scientific societies

- 08/2019 – Now : Member of the European Neural Network Society (ENNS).
- 09/2017 – Now : Member of the "*Société Savante Francophone en Apprentissage Machine*" (SSFAM)
- 02/2017 – 2019 : Member of the CES (expert committee) "*Détection de changements dans les images à haute fréquence*" from Theia
- 07/2015 – Now : Member of the International Neural Network Society (INNS).
- 09/2014 – 09/2016 : Member of the "*Société Francophone de Classification*" (SFC)

6.4.2 Program committee memberships

- IJCNN 2021 : Program Committee member
- ICONIP 2020 : Adjunct Program Committee member
- FSDM 2019 : Session Chair for the Data Mining session
- IJCNN 2019 : Program Committee member
- ICONIP 2018 : Adjunct Program Committee member
- WCCI 2018 : Program Committee member and Session Chair
- ICONIP 2017 : Adjunct Program Committee member
- IJCNN 2017 : Adjunct Program Committee member

6.4.3 Workshops and special sessions organization

- Journées dl2t "*Deep Learning, Teledetection, Temps*" (THEIA - CES Détection/ CNRS - PEPS) – November 2017: Host and co-organizer.
- Websys 2017 : PC member for the special session on "Intelligent Processing of Multimedia in Web Systems".
- IJCNN 2017 : Organizing member of the "Autonomous Learning in Machine Learning" Workshop.
- ICONIP 2016 : Organizing member of the "Topological and Graph Based Clustering Methods" special session.
- SoCPaR 2015 : Organizing member of the "Incremental Machine Learning" special session.
- IJCNN 2015 : Organizing member of the "Learning from multiple learners" Workshop.

6.4.4 Editorial work and reviewing activities

I am a topic editor and member of the reviewer board for the MDPI Journal of Imaging.

Most of my reviewing activities can be verified through my Publon profile. The table below summarizes them for the period 2016–2020. Please note that only journal reviews are indicated in this table as well as in Figure 6.1. My reviewing activities for conferences are not listed simply because I don't keep track of all conferences for which I have made reviews. Nevertheless, I can probably mention at least ICONIP, IJCNN and ICANN for which I have been doing reviews every year for a while now.

Journal	Editor	Reviews	Impact Factor	SJR quartile
Knowledge and Information Systems	Springer	21	2.936	Q2
Remote Sensing	MDPI	19	4.509	Q1
Sensors	MDPI	6	3.275	Q2
Signal, Image and Video Processing	Springer	5	1.794	Q2
Pattern Recognition	Elsevier	3	7.196	Q1
Applied Sciences	MDPI	3	2.474	Q3
Transactions on Computational Social Systems	IEEE	2	3.29	Q2
PLoS ONE	PLoS	2	2.74	Q1
Transactions on Geoscience and Remote Sensing	IEEE	1	5.855	Q1
Biomedical Optics Express	OSA	1	3.921	Q1
Computational Intelligence	Wiley	1	1.196	Q3
Engineering Applications of Artificial Intelligence	Elsevier	1	4.201	Q1
Future Generation Computer Systems	Elsevier	1	6.125	Q1
ISPRS International Journal of Geo-Information	MDPI	1	2.239	Q3
Knowledge-Based Systems	Elsevier	1	5.921	Q1

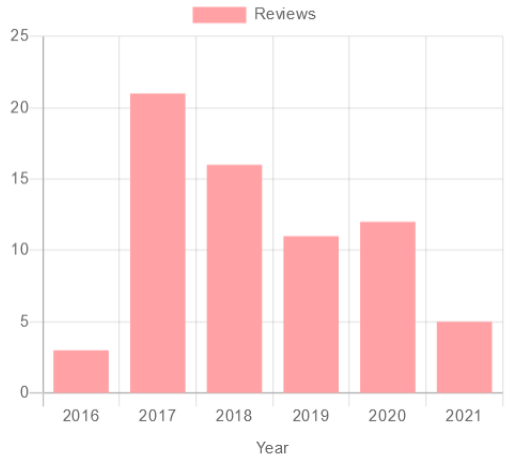


Figure 6.1: My yearly journal reviews between 2016 and 2021

Other administrative activities

6.5 Elected positions

Elected Secretary at the "*Comité Social et Économique*" (Economic and Social Council) of ISEP (12/2018 – Now): This is a 4-year elected position in a committee at the interface between ISEP employees, ISEP direction board and the administration board. This committee is made of 5 elected employees and 2 members of the direction board. It deals with various issues such as working conditions, health and security issues, the school social policy, employees wellbeing, as well as the organization of various social events. Due to the covid-19 pandemic and administrative changes in collective agreements at ISEP, the two first years have been both busy and very enriching in term of skill acquisition.

Elected Professor (non-voting) at ISEP Administration Board (12/2019 – Now): This is a 2-year position that allows two professors and an administrative staff member to attend the meeting of

the Administration Board, without voting rights. For me, it is an interface position between the professors and the administrative board, and which is complementary with my other mandate at the Economic and Social Council.

Elected PhD student at the "*Conseil Scientifique et Pédagogique*" (Scientific and Pedagogic Council) of Doctoral School ABIES (01/2015 – 12/2016)

Elected PhD student at the Laboratory board of AgroParisTech UMR MIA 518 (01/2015 – 12/2015)

6.6 Work groups

Member of the AI committee at ISEP (09/2017–now): This committee was tasked with reforming ISEP courses related to Artificial Intelligence and Machine Learning, and in particular to propose an introductory class to Artificial Intelligence available to all engineering students without any prerequisites. Other tasks for this groups included a better coordination between the different statistics, data analysis and machine learning classes, as well as a reflexion on equipping ISEP with Deep Learning ready solutions (in the form of GPU-equipped computers, but also external computing platforms) for both students and research staff. My implication in this work gorup is due both to my research activities in Deep learning, but also my position as head of a major specialized in Data Science. To this day, I am still in charge of defining and updating the configuration for all ISEP computers equipped with high speed GPUs, in close collaboration with the IT department.

Member of Doctoral School ABIES workgroup on good practices for advising PhD students (01/2016–06/2017): At the end of my PhD studies, I join this grouped composed of PhD students, young new associate professors and experienced full professors (I switched from the PhD group to the associate professor group when I graduated). The goal of this workgroup was to tackle various difficulties that one may encounter when advising PhD students. There was a particular focus on new Associate Professors that were co-advising or advising solo for the first time, but also on conflict management between advisors, as well as between PhD students and advisors. Another point of interest was on the different "advisory styles" and their compatibility with students personality and also sometimes with other advisors using different styles. We came up with some idea, guidelines and good practices that we hope can make the experience more enriching for the PhD students, and also more enjoyable for everyone.

Chapter 7

Scientific production and citation metrics

This chapter regroups all of my publications since the beginning of my graduate studies. Most of them can be authentified and found on my dblp profile. The PDF files for nearly all of them are available from my webpage.

When available, the ranking CORE 2020 for the conferences, and the impact factor (IF) or the scimagojr quartile (SJR) for the journal papers, are mentioned next to each publication. Furthermore, my name is in bold in all authors lists, and the students that participated to any scientific production while under my supervision (PhDs or interns) are in italic.

7.1 Journal papers

1. **Jérémie Sublime**: The 2011 Tohoku Tsunami from the Sky: A review on the evolution of Artificial Intelligence Methods for Damage Assessment. In MDPI - Geosciences 11 (3) 133, 2021. (SJR : Q2)

2. *Guillaume Dupont*, *Ekaterina Kalinicheva*, **Jérémie Sublime**, Florence Rossant and Michel Pâques: Analyzing Age-Related Macular Degeneration Progression in Patients with Geographic Atrophy Using Joint Autoencoders for Unsupervised Change Detection. In MDPI - Journal of Imaging 6(7) 57, 2020. (SJR : Q2)

3. *Ekaterina Kalinicheva*, **Jérémie Sublime** and Maria Trocan: Unsupervised Satellite Image Time Series Clustering Using Object-Based Approaches and 3D Convolutional Autoencoder. In Remote Sensing 12(11), 2020. (IF : 4.509, SJR : Q1)

4. *Ekaterina Kalinicheva*, Dino Ienco, **Jérémie Sublime** and Maria Trocan: Unsupervised Change Detection Analysis in Satellite Image Time Series using Deep Learning Combined with Graph-Based Approaches. In IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (IEEE JSTARS), 13: 1450-1466 (2020). (IF : 3.827, SJR : Q1)

5. **Jérémie Sublime**, Guénaël Cabanes and Basarab Matei: Study on the Influence of Diversity and Quality in Entropy Based Collaborative Clustering. Entropy 21(10): 951 (2019). (IF : 2.494, SJR : Q2)

6. **Jérémie Sublime** and *Ekaterina Kalinicheva*: Automatic Post-Disaster Damage Mapping Using Deep-Learning Techniques for Change Detection: Case Study of the Tohoku Tsunami. Remote Sensing 11(9): 1123 (2019). (IF : 4.509, SJR : Q1)

7. **Jérémie Sublime**, Basarab Matei, Guénaël Cabanes, Nistor Grozavu, Younès Bennani and Antoine Cornuéjols : Entropy Based Probabilistic Clustering, Pattern Recognition 72: 144-157,

2017. (IF : 7.196, SJR : Q1)

8. **Jérémie Sublime**, Andrès Troya-Galvis, and Anne Puissant: Multi-scale analysis of very high resolution satellite images using unsupervised techniques. Remote Sensing, Volume 9(5):495, 2017. (IF : 4.509, SJR : Q1)

9. **Jérémie Sublime**, Nistor Grozavu, Guénaël Cabanès, Younès Bennani, and Antoine Cornuéjols: From Horizontal to Vertical Collaborative Clustering using Generative Topographic Maps. International Journal of Hybrid Intelligent Systems, Volume 12-4, 2016.

10. Sonia Yassa, **Jérémie Sublime**, Rachid Chelouah, Hubert Kadima, Geun-Sik Jo and Bertrand Granado: A Genetic Algorithm for Multi-Objective Optimization in Workflow Scheduling with Hard Constraints. International Journal of Metaheuristics, 2013.

7.2 Peer-reviewed international conference papers with proceedings and indexing

1. *Denis Maurel*, Sylvain Lefebvre and **Jérémie Sublime**: Deep Cooperative Reconstruction with Security Constraints in multi-view environments. In 20th IEEE International Conference on Data Mining (ICDMW'2020), MDSM Workshop, 2020.

2. *Guillaume Dupont*, *Ekaterina Kalinicheva*, **Jérémie Sublime**, Florence Rossant and Michel Pâques: Unsupervised Change Detection using Joint Autoencoders for Age-Related Macular Degeneration Progression. In: The 29th International Conference on Artificial Neural Networks, ICANN (2) 2020. (Core : B)

3. Juan Zamora and **Jérémie Sublime**: A New Information Theory Based Clustering Fusion Method for Multi-view Representations of Text Documents. In: The 22nd HCI International Conference, HCII 2020 (14) 2020: 156-167.

4. *Ekaterina Kalinicheva*, **Jérémie Sublime** and Maria Trocan: Change Detection in Satellite Images Using Reconstruction Errors of Joint Autoencoders. In: The 28th International Conference on Artificial Neural Networks, ICANN (3) 2019: 637-648. (Core : B)

5. **Jérémie Sublime**: Incremental Collaborative Clustering using Information Theory and Information Compression. In: Fuzzy Systems and Data Mining (FSDM 2019), Kitakyushu, Japan, 2019.

6. *Ekaterina Kalinicheva*, **Jérémie Sublime** and Maria Trocan: Object-Based Change Detection in Satellite Images Combined with Neural Network Autoencoder Feature Extraction. In: IEEE International Conference on Image Processing Theory, Tools and Applications IPTA 2019, Istanbul, Turkey, 2019.

7. *Ekaterina Kalinicheva*, **Jérémie Sublime** and Maria Trocan: Neural Autoencoder for Change Detection in Satellite Image Time Series. In: The 25th IEEE International Conference on Electronics, Circuits and SystemsIEEE ICECS 2018, Bordeaux, France.

8. Pierre-Alexandre Murena, **Jérémie Sublime**, Basarab Matei and Antoine Cornuéjols: *An Information Theory based Approach to Multisource Clustering.* In IJCAI-ECAI 2018 :2581-2587, Stockholm, Sweden. (Core : A*)

9. **Jérémie Sublime** and Sylvain Lefebvre: Collaborative Clustering through Constrained Networks using Bandit Optimization. In IEEE International Joint Conference on Neural Networks, IJCNN 2018. (Core : A)

10. **Jérémie Sublime**, *Denis Maurel*, Nistor Grozavu, Basarab Matei and Younès Bennani: Optimizing exchange confidence during collaborative clustering. In IEEE International Joint Conference on Neural Networks, IJCNN 2018. (Core : A)

11. *Denis Maurel*, **Jérémie Sublime** and Sylvain Lefebvre: Incremental Self-Organizing Maps for Collaborative Clustering. In: The 24th International Conference on Neural Information Processing (ICONIP 2017), Guangzhou, China, 2017. (Core : A)

12. **Jérémie Sublime**, Basarab Matei and Pierre-Alexandre Murena: Analysis of the influence of diversity in collaborative and multi-view clustering. In IEEE International Joint Conference on Neural Networks, IEEE IJCNN'17), Anchorage, Alaska, USA, 2017.(Core : A)

13. **Jérémie Sublime**, Younès Bennani and Antoine Cornuéjols: Collaborative-based multi-scale clustering in very high resolution satellite Images. The 23rd International Conference on Neural Information Processing (ICONIP 2016), Kyoto, Japan: in Lecture Notes in Computer Science, LNCS Springer, Proc. of ICONIP'16, 2016. (Core : A)

14. **Jérémie Sublime**, Nistor Grozavu, Younès Bennani and Antoine Cornuéjols: Vertical Collaborative Clustering using Generative Topographic Maps. In: The 7th International Conference on Soft Computing and Pattern Recognition (SoCPaR'15), Fukuoka, Japan, 2015.

15. **Jérémie Sublime**, Younès Bennani and Antoine Cornuéjols: Collaborative Clustering with Heterogeneous Algorithms. IEEE International Joint Conference on Neural Network, (IEEE IJCNN'15), Killarney, Ireland, 2015. (Core : A)

16. Jérémie Sublime, Andrès Troya-Galvis, Younès Bennani, Pierre Gançarski and Antoine Cornuéjols: Semantic Rich ICM Algorithm for VHR Satellite Images Segmentation. International Association for Pattern Recognition (IAPR), International Conference on Machine Vision Applications (MVA'15), Tokyo, Japan, 2015.

17. **Jérémie Sublime**, Younès Bennani and Antoine Cornuéjols: A New Energy Model for the Hidden Markov Random Fields. The 21th International Conference on Neural Information Processing, Kuching, Sarawak, Malaysia, in Lecture Notes in Computer Science, LNCS Springer, Proc of ICONIP'14, 2014. (Core : A)

7.3 Other conferences

1. *Ekaterina Kalinicheva* , **Jérémie Sublime** and Maria Trocan: Analysis of Objects Evolution in Satellite Image Time Series Transformed with Neural Network Autoencoders. In the First

International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI), Barcelona, Spain, 2019.

2. *Denis Maurel*, **Jérémie Sublime** and Sylvain Lefebvre: Cartes Auto-Organisatrices Incrémentales appliquées au Clustering Collaboratif. In Extraction et Gestion des Connaissances, EGC 2018, St Denis, France, 2018

3. **Jérémie Sublime**: Smart view selection in Multi-view Clustering. In SIS 2017 - Statistics and data Science, Firenze, Italia

4. Pierre-Alexandre Murena, **Jérémie Sublime**, Basarab Matei and Antoine Cornuéjols: Collaborative clustering based on Algorithmic Information Theory. In Conférence sur l'Apprentissage Automatique (CAp'17), Grenoble, France, 2017

5. **Jérémie Sublime**, Nistor Grozavu, Guénaël Cabanes, Younès Bennani and Anntoine Cornuéjols: Collaborative learning using topographic maps. In AAFD&SFC'16, at Marrakech, Morocco, 2016.

6. **Jérémie Sublime**, Younès Bennani and Antoine Cornuéjols: A Compactness-based Iterated Conditional Modes Algorithm For Very High Resolution Satellite Images Segmentation, Extraction et Gestion des Connaissances 2015 (EGC'15), Luxembourg, 2015.

7. **Jérémie Sublime**, Younès Bennani and Antoine Cornuéjols: Un nouveau modèle d'énergie pour les champs aléatoires de Markov cachés. In SFC'14, Société Francophone de Classification, Rabat, Morocco, 2014.

8. **Jérémie Sublime**, Sonia Yassa and Geun-Sik Jo: A genetic algorithm with the concept of viral infections to solve hard constraints in workflow scheduling. Conference of the Korean Intelligent Information Society (KIIS'12), Seoul, 12/2012

7.4 Miscellany

7.4.1 Oral talks

1. Seminary on "Unsupervised analysis of Satellite images time series using deep learning techniques", L2TI, Sorbonne Paris North University, May 2019.

2. Invited talk on "Travaux en IA à l'ISEP sur l'imagerie satellite (ongoing research at ISEP on satellite images)", ISEP, March 2019.

3. *Denis Maurel* and **Jérémie Sublime**: Seminary on "Deep Cooperative Reconstruction with Security Constraints", LIPN, Sorbonne Paris North University, December 2018.

4. Seminary on "Unsupervised learning for multi-source applications and satellite image processing", LINK, AgroParisTech, March 2018.

5. Seminary on "Unsupervised learning for multi-source applications and satellite image processing", LRI, Université Paris Sud, February 2018.

6. Seminary on "Collaborative clustering and its applications", LTCI, Telecom ParisTech, April 2017.

7. Invited talk on "Collaborative-based multi-scale clustering in very high resolution satellite Images". In: Collaboration, classification, connaissances et données de l'environnement Workshop, SAGEO'16, Nice, December 2016.

7.4.2 Thesis Manuscripts

1. **Jérémie Sublime**: Contributions to collaborative clustering and its potential applications on very high resolution satellite images. (Contributions au clustering collaboratif et à ses potentielles applications en imagerie à très haute résolution). PhD Thesis, University of Paris-Saclay, France, 2016.

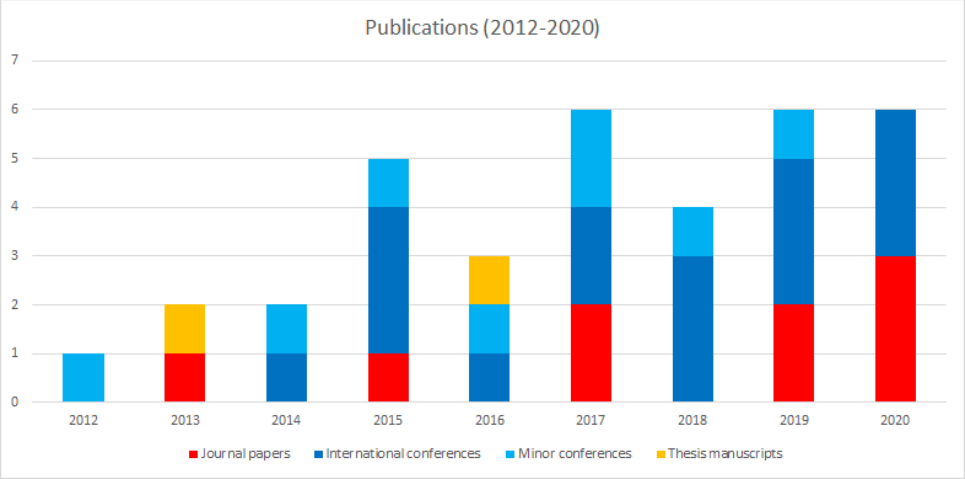
2. **Jérémie Sublime**: A Genetic Algorithm with Constraint Sastisfaction Problems for multi-objective Optimization in Workflow Scheduling. Master's Thesis, Inha University, 2013.

7.5 Publications by categories and citation metrics

The table below sorts my publications by category and by research axis:

	Multi-view clustering	Deep Learning & Imaging	Others
Q1 journals	1	4	-
Other journals	2	2	1
A and A* conferences	6	2	-
B conferences	-	2	-
Other intnl. conferences	4	3	-
Minor conferences	4	3	1

The figure below shows the evolution of my publications between 2012 and 2020.



The table below was build from my Google Scholar web page based on my data from April 6th 2021. It is a summary of common metrics that can be linked to a scholar profile to assess the impact of his publications. I took 2012 as the year for my first publication and 2014 as the year for my first cite.

Publications	37
Cites	186
h-index [206]	8
g-index [207]	12
m-index (since 2012)	0.89
i-10	8
Cites/Year (since 2014)	22.38
Cites/Paper	5.03
Authors/Paper	3.19
Cites/Author	51.0
Papers/Author	15.37
Kardashian Index	0.28

Bibliography

[1] Rui Xu and D. Wunsch, II, "Survey of Clustering Algorithms," *Trans. Neur. Netw.*, vol. 16, no. 3, pp. 645–678, May 2005. 12

[2] Dongkuan Xu and Yingjie Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 8 2015. 12

[3] Victoria Hodge and Jim Austin, "A Survey of Outlier Detection Methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004. 12

[4] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, Inc., New York, NY, USA, 1987. 12

[5] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664 – 681, 2017. 13

[6] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, 1998. 13

[7] Daoying Ma and Aidong Zhang, "An Adaptive Density-Based Clustering Algorithm for Spatial Database with Noise," *Data Mining, IEEE International Conference on*, vol. 0, pp. 467–470, 2004. 13

[8] Jiawei Han, Micheline Kamber, and Jian Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. 13

[9] M. Ester, H.-P. Kriegel, J. Sander, and X Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *In International Conference on Knowledge Discovery and Information Retrieval*, 1996, pp. 226–231. 13, 32

[10] P. Viswanath and Rajwala Pinkesh, "l-DBSCAN: A Fast Hybrid Density Based Clustering Method," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, Washington, DC, USA, 2006, pp. 912–915, IEEE Computer Society. 13

[11] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," in *In ACM SIGMOD international conference on Management of data*. 1999, pp. 49–60, ACM Press. 13, 32

[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sept. 1999. 13

[13] Joe H. Ward Jr., "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963. 13, 67, 71

[14] Youbao Tang, Xiangqian Wu, and Wei Bu, "Saliency Detection Based on Graph-Structural Agglomerative Clustering," in *Proceedings of the 23rd ACM International Conference on Multimedia*, New York, NY, USA, 2015, MM '15, pp. 1083–1086, ACM. 13

[15] Wei Zhang, Deli Zhao, and Xiaogang Wang, "Agglomerative clustering via maximum incremental path integral," *Pattern Recognition*, vol. 46, no. 11, pp. 3056–3065, 2013. 13

[16] William H. Day and Herbert Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7–24, December 1984. 13

[17] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," *SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, jun 1998. 13

[18] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297. 13, 21, 31, 50, 61, 77

[19] Hugo Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Polon. Sci. Cl. III. 4*, pp. 801–804, 1956. 13

[20] Chieh-Yuan Tsai and Chuang-Cheng Chiu, "Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm," *Comput. Stat. Data Anal.*, vol. 52, no. 10, pp. 4658–4672, 2008. 13

[21] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981. 13

[22] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, July 2002. 13

[23] Juan Antonio Cuesta-Albertos and Ricardo Fraiman, "Impartial trimmed k-means for functional data," *Comput. Stat. Data Anal.*, vol. 51, no. 10, pp. 4864–4877, 2007. 13

[24] Shu-Chuan Chu, John Roddick, and Jeng-Shyang Pan, "Improved search strategies and extensions to k-medoids based clustering algorithms," *Int. J. Bus. Intell. Data Min.*, vol. 3, no. 2, pp. 212–231, 2008. 13

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977. 13

[26] Jianbo Shi and Jitendra Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. 13

[27] Marina Meila and Jianbo Shi, "Learning Segmentation by Random Walks," in *In Advances in Neural Information Processing Systems*. 2001, pp. 873–879, MIT Press. 13

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119. 13

[29] Maha Fraj, Mohamed Aymen Ben HajKacem, and Nadia Essoussi, "Ensemble method for multi-view text clustering," in *Computational Collective Intelligence - 11th International Conference, ICCCI 2019, Hendaye, France, September 4-6, 2019, Proceedings, Part I*, 2019, pp. 219–231. 13, 90

[30] Arthur Zimek and Jilles Vreeken, "The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives," *Machine Learning*, vol. 98, no. 1-2, pp. 121–155, 2015. 14

[31] Steffen Bickel and Tobias Scheffer, "Multi-view clustering," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*. 2004, pp. 19–26, IEEE Computer Society. 14

[32] N. Karthikeyani Visalakshi and K. Thangavel, *Distributed Data Clustering: A Comparative Analysis*, pp. 371–397, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 14

[33] Antoine Cornuéjols, Cédric Wemmert, Pierre Gançarski, and Younès Bennani, "Collaborative clustering: Why, when, what and how," *Information Fusion*, vol. 39, pp. 81–95, 2018. 14, 34

[34] Witold Pedrycz, "Collaborative fuzzy clustering," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1675–1686, 2002. 14

[35] Nistor Grozavu and Younès Bennani, "Topological collaborative clustering," *Australian Journal of Intelligent Information Processing Systems*, vol. 12, no. 3, 2010. 14, 15, 35

[36] Alexander Strehl, Joydeep Ghosh, and Claire Cardie, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002. 14

[37] Stefan T. Hadjitodorov, Ludmila I. Kuncheva, and Ludmila P. Todorova, "Moderate diversity for better cluster ensembles," *Inf. Fusion*, vol. 7, no. 3, pp. 264–275, Sept. 2006. 14

[38] Tossapon Boongoen and Natthakan Iam-on, "Cluster ensembles: A survey of approaches with recent extensions and applications," *Comput. Sci. Rev.*, vol. 28, pp. 1–25, 2018. 14

[39] Ludmila I. Kuncheva and Christopher J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, May 2003. 14

[40] Cedric Wemmert and Pierre Gancarski, "A multi-view voting method to combine unsupervised classifications," *Artificial Intelligence and Applications, Malaga, Spain,*, pp. 447 – 452, 2002. 14, 37

[41] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997. 14

[42] Shai Ben-David, Ulrike Von Luxburg, and Dávid Pál, "A sober look at clustering stability," in *International Conference on Computational Learning Theory*. Springer, 2006, pp. 5–19. 15, 41, 90

[43] Nistor Grozavu, Mohamad Ghassany, and Younès Bennani, "Learning confidence exchange in collaborative clustering," in *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011*. 2011, pp. 872–879, IEEE. 15

[44] Denis Maurel, *Contributions aux communications inter-vues pour l'apprentissage collaboratif. (Contributions to inter-views communications applied to collaborative learning)*, Ph.D. thesis, Sorbonne University, France, 2018. 15, 23

[45] Zhao Kang, Xinjia Zhao, Chong Peng, Hongyuan Zhu, Joey Tianyi Zhou, Xi Peng, Wenyu Chen, and Zenglin Xu, "Partition level multiview subspace clustering," *Neural Networks*, vol. 122, pp. 279–288, 2020. 15, 41, 90

[46] Parisa Rastin, Guénaël Cabanes, Nistor Grozavu, and Younès Bennani, "Collaborative clustering: How to select the optimal collaborators?," in *IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015*. 2015, pp. 787–794, IEEE. 15, 18

[47] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of 2nd Berkeley Symposium*, Berkeley University of California Press, Ed., 1951, pp. 481–492. 15, 47

[48] Jérémie Sublime, Basarab Matei, and Pierre-Alexandre Murena, "Analysis of the influence of diversity in collaborative and multi-view clustering," in *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*. 2017, pp. 4126–4133, IEEE. 15, 18

[49] Jérémie Sublime, Basarab Matei, Guénaël Cabanes, Nistor Grozavu, Younès Bennani, and Antoine Cornuéjols, "Entropy based probabilistic collaborative clustering," *Pattern Recognit.*, vol. 72, pp. 144–157, 2017. 15, 35, 50

[50] Teuvo Kohonen, Ed., *Self-organizing Maps*, Springer-Verlag, Berlin, Heidelberg, 1997. 15, 31, 77

[51] Mohamad Ghassany, Nistor Grozavu, and Younès Bennani, "Collaborative clustering using prototype-based techniques," *Int. J. Comput. Intell. Appl.*, vol. 11, no. 3, 2012. 15

[52] Jérémie Sublime, Denis Maurel, Nistor Grozavu, Basarab Matei, and Younès Bennani, "Optimizing exchange confidence during collaborative clustering," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, 2018, pp. 1–8. 15

[53] Jérémie Sublime, Guénaël Cabanes, and Basarab Matei, "Study on the influence of diversity and quality in entropy based collaborative clustering," *Entropy*, vol. 21, no. 10, pp. 951, 2019. 18, 47

[54] Parisa Rastin, Basarab Matei, Guénaël Cabanes, Nistor Grozavu, and Younès Bennani, "Impact of learners' quality and diversity in collaborative clustering," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 2, pp. 149–165, 2019. 18

[55] Jérémie Sublime and Sylvain Lefebvre, "Collaborative clustering through constrained networks using bandit optimization," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, 2018, pp. 1–8. 18

[56] Omar Besbes, Yonatan Gur, and Assaf Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, Eds., pp. 199–207. Curran Associates, Inc., 2014. 19

[57] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2003. 19, 47

[58] Xiao-Ning Wang, Jin-Mao Wei, Han Jin, Gang Yu, and Hai-Wei Zhang, "Probabilistic confusion entropy for evaluating classifiers," *Entropy*, vol. 15, no. 11, pp. 4969–4992, 2013. 21

[59] Jin-Mao Wei, Xiao-Jie Yuan, Qing-Hua Hu, and Shu-Qin Wang, "A novel measure for evaluating classifiers.," *Expert System Applications*, vol. 37, no. 5, pp. 3799–3809, 2010. 21

[60] Denis Maurel, Sylvain Lefebvre, and Jeremie Sublime, "Deep Cooperative Reconstruction with Security Constraints in multi-view environments," in *ICDMW 2020*, Online, Italy, Nov. 2020. 23, 27, 47

[61] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A. Bilmes, "On deep multi-view representation learning: Objectives and optimization," *CoRR*, vol. abs/1602.01024, 2016. 23

[62] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 2011, pp. 689–696. 23

[63] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 24, 25, 50, 56, 91

[64] Zheng Fang, Sen Zhou, and Jing Li, "Multi-view autoencoder for image feature learning with structured nonnegative low rank," in *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*, 2018, pp. 4033–4037. 24

[65] Meng Li, Liangzhen Lai, Naveen Suda, Vikas Chandra, and David Z Pan, "Privynet: A flexible framework for privacy-preserving deep neural network training with a fine-grained privacy control," *arXiv preprint arXiv:1709.06161*, 2017. 25

[66] Seyed Ali Osia, Ali Shahin Shamsabadi, Ali Taheri, Kleomenis Katevas, Sina Sajadmanesh, Hamid R Rabiee, Nicholas D Lane, and Hamed Haddadi, "A hybrid deep learning architecture for privacy-preserving mobile analytics," *arXiv preprint arXiv:1703.02952*, 2017. 25

[67] Tin Kam Ho, "Random decision forests," in *Third International Conference on Document Analysis and Recognition, ICDAR 1995, August 14 - 15, 1995, Montreal, Canada. Volume I*, 1995, pp. 278–282. 27, 77

[68] Pierre-Alexandre Murena, Jérémie Sublime, Basarab Matei, and Antoine Cornuéjols, "An information theory based approach to multisource clustering," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2018, pp. 2581–2587. 29, 31, 47

[69] Juan Zamora and Jérémie Sublime, "A new information theory based clustering fusion method for multi-view representations of text documents," in *Social Computing and Social Media. Design, Ethics, User Behavior, and Social Network Analysis - 12th International Conference, SCSM 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19-24, 2020, Proceedings, Part I*, 2020, pp. 156–167. 29, 38, 90

[70] C. S. Wallace and D. M. Boulton, "An information measure for classification," *The Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968. 29

[71] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465 – 471, 1978. 29

[72] Ming Li and Paul M.B. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer Publishing Company, Incorporated, 3 edition, 2008. 30, 36, 90

[73] Ray J Solomonoff, "A formal theory of inductive inference. part i," *Information and control*, vol. 7, no. 1, pp. 1–22, 1964. 30

[74] M. Hutter, "A theory of universal artificial intelligence based on algorithmic complexity," Tech. Rep., Apr. 2000. 30

[75] L. Kaufman and P.J Rousseeuw, "Clustering by means of medoids. in statistical data analysis based on the l1 norm and related methods," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1987. 31

[76] Christopher M. Bishop, Markus Svensen, and Christopher K. I. Williams, "Gtm: The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215–234, 1998. 31

[77] Pierre-Alexandre Murena, *Minimum complexity principle for knowledge transfer in artificial learning. (Principe de minimum de complexité pour le transfert de connaissances en apprentissage artificiel)*, Ph.D. thesis, University of Paris-Saclay, France, 2018. 31, 32, 41, 45

[78] Pierre Gançarski and Cédric Wemmert, "Collaborative multi-step mono-level multi-strategy classification," *Multim. Tools Appl.*, vol. 35, no. 1, pp. 1–27, 2007. 37

[79] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang, "Large-scale multi-view spectral clustering via bipartite graph.," in *AAAI*, Blai Bonet and Sven Koenig, Eds. 2015, pp. 2750–2756, AAAI Press. 41

[80] Zhao Kang, Zipeng Guo, Shudong Huang, Siying Wang, Wenyu Chen, Yuanzhang Su, and Zenglin Xu, "Multiple partitions aligned clustering," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus, Ed. 2019, pp. 2701–2707, ijcai.org. 41, 90

[81] Ulrike von Luxburg, "Clustering stability: An overview," *Found. Trends Mach. Learn.*, vol. 2, no. 3, pp. 235–274, 2009. 41

[82] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville, *Deep Learning*, Adaptive computation and machine learning. MIT Press, 2016. 50

[83] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. 50

[84] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1–9. 50

[85] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. 50

[86] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017, pp. 4278–4284. 50

[87] Syed Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Khurram Khan, "Medical Image Analysis using Convolutional Neural Networks: A Review," *Journal of Medical Systems*, vol. 42, pp. 226, 10 2018. 50

[88] Mei Wang and Weihong Deng, "Deep Face Recognition: A Survey," *CoRR*, vol. abs/1804.06655, 2018. 50

[89] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1013–1020. 50

[90] V. John, A. Boyali, S. Mita, M. Imanishi, and N. Sanma, "Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016, pp. 1–8. 50

[91] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926. 50

[92] Jérémie Sublime, Andres Troya-Galvis, Younès Bennani, Antoine Cornuéjols, and Pierre Gançarski, "Semantic rich ICM algorithm for VHR satellite images segmentation," in *14th IAPR International Conference on Machine Vision Applications, MVA 2015, Miraikan, Tokyo, Japan, 18-22 May, 2015*, 2015, pp. 45–48. 50

[93] Jérémie Sublime, Andrés Troya-Galvis, and Anne Puissant, "Multi-scale analysis of very high resolution satellite images using unsupervised techniques," *Remote. Sens.*, vol. 9, no. 5, pp. 495, 2017. 50, 53

[94] Jérémie Sublime, Antoine Cornuéjols, and Younès Bennani, "Collaborative-based multi-scale clustering in very high resolution satellite images," in *Neural Information Processing - 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16-21, 2016, Proceedings, Part III*, 2016, pp. 148–155. 50

[95] Serge Beucher and Christian Lantuéjoul, "Use of watersheds in contour detection," 01 1979, vol. 132. 50

[96] Xide Xia and Brian Kulis, "W-net: A deep model for fully unsupervised image segmentation," *CoRR*, vol. abs/1711.08506, 2017. 50, 81

[97] Ekaterina Kalinicheva, *Analyse Non-supervisée de Séries d'Images Satellites avec des Méthodes d'Apprentissage Profond. (Unsupervised Satellite Image Time Series Analysis using Deep Learning Techniques)*, Ph.D. thesis, Sorbonne University, France, 2020. 51

[98] Rodrigo Caye Daudt, Adrien Chan-Hon-Tong, Bertrand Le Saux, and Alexandre Boulch, "Learning to understand earth observation images with weak and unreliable ground truth," in *2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019*, 2019, pp. 5602–5605. 54

[99] Ekaterina Kalinicheva, Jérémie Sublime, and Maria Trocan, "Change detection in satellite images using reconstruction errors of joint autoencoders," in *Artificial Neural Networks and Machine Learning - ICANN 2019: Image Processing - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part III*, 2019, pp. 637–648. 54, 58, 78, 83

[100] J. S. Deng, K. Wang, Y. H. Deng, and G. J. Qi, "PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data," *International Journal of Remote Sensing*, vol. 29, no. 16, pp. 4823–4838, 2008. 55

[101] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," in *2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018*, 2018, pp. 2115–2118. 55

[102] T. Lei, Q. Zhang, D. Xue, T. Chen, H. Meng, and A. K. Nandi, "End-to-end Change Detection Using a Symmetric Fully Convolutional Network for Landslide Mapping," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3027–3031. 55

[103] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 10 2015, vol. 9351, pp. 234–241. 55, 56, 77, 80, 92

[104] Qing Wang, Xiaodong Zhang, Guanzhou Chen, Fan Dai, Yuanfu Gong, and Kun Zhu, "Change detection based on faster r-cnn for high-resolution remote sensing images," *Remote Sensing Letters*, vol. 9, no. 10, pp. 923–932, 2018. 55

[105] P. Du, S. Liu, P. Gamba, K. Tan, and J. Xia, "Fusion of Difference Images for Change Detection Over Urban Areas," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1076–1086, Aug 2012. 55

[106] K. Tan, X. Jin, A. Plaza, X. Wang, L. Xiao, and P. Du, "Automatic Change Detection in High-Resolution Remote Sensing Images by Using a Multiple Classifier System and Spectral-Spatial Features," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 8, pp. 3439–3451, Aug 2016. 55

[107] Guo Cao, Bisheng Wang, Haro-Carrión Xavier, Di Yang, and Jane Southworth, "A new difference image creation method based on deep neural networks for change detection in remote-sensing images," *International Journal of Remote Sensing*, vol. 38, no. 23, pp. 7161–7175, 2017. 55

[108] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, USA, 2010, ICML'10, pp. 807–814, Omnipress. 55

[109] A. A. Nielsen, "The Regularized Iteratively Reweighted MAD Method for Change Detection in Multi- and Hyperspectral Data," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 463–478, Feb 2007. 55

[110] Yuan Xu, Shiming Xiang, Chunlei Huo, and Chunhong Pan, "Change Detection Based on Auto-encoder Model for VHR Images," *Proc SPIE*, vol. 8919, pp. 02–, 10 2013. 55, 56, 63

[111] Lynda Khiali, Mamoudou Ndiath, Samuel Alleaume, Dino Ienco, Kenji Ose, and Maguelonne Teisseire, "Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach," *International Journal of Applied Earth Observation and Geoinformation*, vol. 74, pp. 103 – 119, 2019. 55, 66, 68, 70

[112] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017. 56

[113] A. Sento, "Image compression with auto-encoder algorithm using deep neural network (DNN)," pp. MIT–99–MIT–103, Oct 2016. 56

[114] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang, "Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2016, NIPS'16, pp. 2810–2818, Curran Associates Inc. 56

[115] Jonathan Masci, Ueli Meier, Dan Ciresan, and Jürgen Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," 06 2011, pp. 52–59. 56

[116] Chen Xing, Li Ma, and Xiaoquan Yang, "Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images," *Journal of Sensors*, vol. 2016, pp. 1–10, 01 2016. 56

[117] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin, "Improved Deep Embedded Clustering with Local Structure Preservation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1753–1759. 56

[118] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin, "Deep Clustering with Convolutional Autoencoders," in *Neural Information Processing*, Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy, Eds. 2017, pp. 373–382, Springer International Publishing. 56, 61

[119] Ekaterina Kalinicheva, Jérémie Sublime, and Maria Trocan, "Neural network autoencoder for change detection in satellite image time series," in *25th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2018, Bordeaux, France, December 9-12, 2018*, 2018, pp. 641–642. 57

[120] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979. 57, 83

[121] Ekaterina Kalinicheva, Jérémie Sublime, and Maria Trocan, "Object-based change detection in satellite images combined with neural network autoencoder feature extraction," in *Ninth International Conference on Image Processing Theory, Tools and Applications, IPTA 2019, Istanbul, Turkey, November 6-9, 2019*, 2019, pp. 1–6. 58

[122] Jérémie Sublime and Ekaterina Kalinicheva, "Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the tohoku tsunami," *Remote. Sens.*, vol. 11, no. 9, pp. 1123, 2019. 58, 59, 65, 66, 78

[123] S. Karimzadeh, S. Samsonov, and M. Matsuoka, "Block-based damage assessment of the 2012 ahar-varzaghan, iran, earthquake through sar remote senisng data," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2017, pp. 1546–1549. 59

[124] Pietro Milillo, Giorgia Giardina, Matthew J. DeJong, Daniele Perissin, and Giovanni Milillo, "Multi-temporal insar structural damage assessment: The london crossrail case study," *Remote Sensing*, vol. 10, no. 2, pp. 287, 2018. 59

[125] Pietro Milillo, Maria Cristina Porcu, Paul Lundgren, Fabio Soccodato, Jacqueline T. Salzer, Eric J. Fielding, Roland Burgmann, Giovanni Milillo, Daniele Perissin, and Filippo Biondi, "The ongoing destabilization of the mosul dam as observed by synthetic aperture radar interferometry," in *2017 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2017, Fort Worth, TX, USA, July 23-28, 2017*, 2017, pp. 6279–6282. 59

[126] Yanbing Bai, Erick Mas, and Shunichi Koshimura, "Towards operational satellite-based damage-mapping using u-net convolutional network: A case study of 2011 tohoku earthquake-tsunami," *Remote Sensing*, vol. 10, no. 10, 2018. 59, 65

[127] Shunichi Koshimura, Luis Moya, Erick Mas, and Yanbing Bai, "Tsunami damage detection with remote sensing: A review," *Geosciences*, vol. 10, no. 5, 2020. 59

[128] Jérémie Sublime, "The 2011 tohoku tsunami from the sky: A review on the evolution of artificial intelligence methods for damage assessment," *Geosciences*, vol. 11, no. 3, 2021. 59

[129] Nobuhito Mori, Tomoyuki Takahashi, Tomohiro Yasuda, and Hideaki Yanagisawa, "Survey of 2011 tohoku earthquake tsunami inundation and run-up," *Geophysical Research Letters*, vol. 38, no. 7, 2011. 59

[130] Ekaterina Kalinicheva, Dino Ienco, Jérémie Sublime, and Maria Trocan, "Unsupervised change detection analysis in satellite image time series using deep learning combined with graph-based approaches," *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 13, pp. 1450–1466, 2020. 66

[131] Fabio Guttler, Dino Ienco, Jordi Nin, Maguelonne Teisseire, and Pascal Poncelet, "A graph-based approach to detect spatiotemporal dynamics in satellite image time series," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 92 – 107, 2017. 66, 68, 69, 148

[132] Markus Goldstein and Seiichi Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," *PloS one*, vol. 11, 04 2016. 67

[133] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep 2004. 68

[134] Lutz Prechelt, *Early Stopping — But When?*, pp. 53–67, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 71

[135] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734, Association for Computational Linguistics. 71, 72

[136] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. 72, 91

[137] Emile Ndikumana, Dinh Ho Tong Minh, Nicolas Baghdadi, Dominique Courault, and Laure Hossard, "Deep Recurrent Neural Network for Agricultural Classification using multitemporal SAR Sentinel-1 for Camargue, France," *Remote Sensing*, vol. 10, no. 8, 2018. 72

[138] Haowen Luo, "Shorten spatial-spectral RNN with parallel-gru for hyperspectral image classification," *CoRR*, vol. abs/1810.12563, 2018. 72

[139] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, July 2017. 72

[140] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," 2014. 72

[141] Ilya Sutskever, Oriol Vinyals, and Quoc Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, vol. 4, 09 2014. 72

[142] Yogesan Kanagasingam, Alauddin Bhuiyan, Michael D. Abràmoff, R. Theodore Smith, Leonard Goldschmidt, and Tien Y. Wong, "Progress on retinal image analysis for age related macular degeneration," *Progress in Retinal and Eye Research*, vol. 38, pp. 20 – 42, 2014. 76

[143] R Priya and P Aruna, "Automated diagnosis of age-related macular degeneration from color retinal fundus images," in *2011 3rd International Conference on Electronics Computer Technology*. IEEE, 2011, vol. 2, pp. 227–230. 76

[144] Cemal Köse, Ugur Sevik, and Okyay Gençalioglu, "Automatic segmentation of age-related macular degeneration in retinal fundus images," *Computers in biology and medicine*, vol. 38 5, pp. 611–9, 2008. 76

[145] David J Ramsey, Janet S Sunness, Poorva Malviya, Carol Applegate, Gregory D Hager, and James T Handa, "Automated image alignment and segmentation to follow progression of geographic atrophy in age-related macular degeneration," *Retina*, vol. 34, no. 7, pp. 1296–1307, 2014. 76, 77, 78

[146] Cemal Köse, Ugur Sevik, Okyay Gençalioglu, Cevat Ikibas, and Temel Kayikiçioglu, "A statistical segmentation method for measuring age-related macular degeneration in retinal fundus images," *Journal of medical systems*, vol. 34, pp. 1–13, 02 2010. 76, 77, 78

[147] Andrés G Marrugo, Maria S Millan, Michal Sorel, and Filip Sroubek, "Retinal image restoration by means of blind deconvolution," *Journal of Biomedical Optics*, vol. 16, no. 11, pp. 116016, 2011. 76, 78

[148] Albert K Feeny, Mongkol Tadarati, David E Freund, Neil M Bressler, and Philippe Burlina, "Automated segmentation of geographic atrophy of the retinal epithelium via random forests in areds color fundus images," *Computers in biology and medicine*, vol. 65, pp. 124–136, 2015. 76, 77

[149] Noah Lee, Andrew F Laine, and R Theodore Smith, "A hybrid segmentation approach for geographic atrophy in fundus auto-fluorescence images for diagnosis of age-related macular degeneration," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 4965–4968. 76

[150] Zhihong Hu, Gerard G Medioni, Matthias Hernandez, and Srinivas R Sadda, "Automated segmentation of geographic atrophy in fundus autofluorescence images using supervised pixel classification," *Journal of Medical Imaging*, vol. 2, no. 1, pp. 014501, 2015. 76, 77

[151] Zhihong Hu, Gerard G Medioni, Matthias Hernandez, Amirhossein Hariri, Xiaodong Wu, and SriniVas R Sadda, "Segmentation of the geographic atrophy in spectral-domain optical coherence tomography and fundus autofluorescence images," *Investigative ophthalmology & visual science*, vol. 54, no. 13, pp. 8375–8383, 2013. 76

[152] Md. Akter Hussain, Arun Govindaiah, Eric Souied, Roland Smith, and Alauddin Bhuiyan, "Automated tracking and change detection for age-related macular degeneration progression using retinal fundus imaging," 06 2018, pp. 394–398. 77, 78, 81

[153] Thanh V. Phan, Lama Seoud, and Farida Cheriet, "Automatic Screening and Grading of Age-Related Macular Degeneration from Texture Analysis of Fundus Images," *Journal of Ophthalmology*, vol. 2016, 3 2016. 77

[154] John Wandeto, Henry Nyongesa, Yves Rémond, and Birgitta Dresp, "Detection of small changes in medical and random-dot images comparing self-organizing map performance to human detection," *Informatics in Medicine Unlocked*, vol. 7, 03 2017. 77

[155] Noah Lee, R Theodore Smith, and Andrew F Laine, "Interactive segmentation for geographic atrophy in retinal fundus images," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE, 2008, pp. 655–658. 77

[156] A Deckert, S Schmitz-Valckenberg, J Jorzik, A Bindewald, FG Holz, and Ulrich Mansmann, "Automated analysis of digital fundus autofluorescence images of geographic atrophy in advanced age-related macular degeneration using confocal scanning laser ophthalmoscopy (cslo)," *BMC ophthalmology*, vol. 5, no. 1, pp. 8, 2005. 77

[157] Giulia Troglio, Marina Alberti, Jon Benediktsson, Gabriele Moser, Sebastiano Serpico, and Einar Stefánsson, "Unsupervised change-detection in retinal images by a multiple-classifier approach," 04 2010, pp. 94–103. 78

[158] Giulia Troglio, A. Nappo, Jon Benediktsson, Gabriele Moser, Sebastiano Serpico, and Einar Stefánsson, *Automatic Change Detection of Retinal Images*, vol. 25, pp. 281–284, 01 2010. 78

[159] P. Burlina, D. E. Freund, N. Joshi, Y. Wolfson, and N. M. Bressler, "Detection of age-related macular degeneration via deep learning," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 184–188. 78

[160] Philippe M. Burlina, Neil Joshi, Michael Pekala, Katia D. Pacheco, David E. Freund, and Neil M. Bressler, "Automated Grading of Age-Related Macular Degeneration From Color Fundus Images Using Deep Convolutional Neural Networks," *JAMA Ophthalmology*, vol. 135, no. 11, pp. 1170–1176, 11 2017. 78

[161] Asako Kanezaki, "Unsupervised image segmentation by backpropagation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018. 78, 84

[162] Turgay Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, pp. 772–776, 2009. 78, 84

[163] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, pp. e453, 2014. 80

[164] Charles Chui and Hrushikesh Mhaskar, "Mra contextual-recovery extension of smooth functions on manifolds," *Applied and Computational Harmonic Analysis*, vol. 28, pp. 104–113, 01 2010. 80

[165] S. B. Damelin and N. S. Hoang, "On surface completion and image inpainting by biharmonic functions: Numerical aspects," *International Journal of Mathematics and Mathematical Sciences*, vol. 2018, pp. 1–8, 2018. 80

[166] Clément Royer, "Master's degree internship report: W-nets applied to ARMD images," Tech. Rep., Sorbonne University, 10 2020. 80

[167] Guillaume Dupont, Ekaterina Kalinicheva, Jérémie Sublime, Florence Rossant, and Michel Pâques, "Unsupervised change detection using joint autoencoders for age-related macular degeneration progression," in *Artificial Neural Networks and Machine Learning - ICANN 2020 - 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15-18, 2020, Proceedings, Part II*, 2020, pp. 813–824. 82

[168] Guillaume Dupont, Ekaterina Kalinicheva, Jérémie Sublime, Florence Rossant, and Michel Pâques, "Analyzing age-related macular degeneration progression in patients with geographic atrophy using joint autoencoders for unsupervised change detection," *J. Imaging*, vol. 6, no. 7, pp. 57, 2020. 83

[169] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, Eds., 2014, pp. 2672–2680. 91

[170] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic, "Alphagan: Generative adversarial networks for natural image matting," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. 2018, p. 259, BMVA Press. 91, 92

[171] Yihao Huang, Felix Juefei-Xu, Run Wang, Xiaofei Xie, Lei Ma, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu, "Fakelocator: Robust localization of gan-based face manipulations via semantic segmentation networks with bells and whistles," *CoRR*, vol. abs/2001.09598, 2020. 91

[172] Yong-Hoon Kwon and Min-Gyu Park, "Predicting future frames using retrospective cycle GAN," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 1811–1820, Computer Vision Foundation / IEEE. 91

[173] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019, pp. 5509–5519. 91

[174] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *CoRR*, vol. abs/1706.02633, 2017. 91

[175] Ahmed Elazab, Changmiao Wang, Syed Jamal Safdar Gardezi, Hongmin Bai, Qingmao Hu, Tianfu Wang, Chunqi Chang, and Baiying Lei, "Gp-gan: Brain tumor growth prediction using stacked 3d generative adversarial networks from longitudinal mr images," *Neural Networks*, vol. 132, pp. 321 – 332, 2020. 91

[176] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang, "Medgan: Medical image translation using gans," *Comput. Medical Imaging Graph.*, vol. 79, pp. 101684, 2020. 92

[177] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár, "Amortised MAP inference for image super-resolution," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017, OpenReview.net. 92

[178] Youbao Tang, Jinzheng Cai, Le Lu, Adam P. Harrison, Ke Yan, Jing Xiao, Lin Yang, and Ronald M. Summers, "CT image enhancement using stacked generative adversarial networks and transfer learning for lesion segmentation improvement," in *Machine Learning in Medical Imaging - 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings*, Yinghuan Shi, Heung-Il Suk, and Mingxia Liu, Eds. 2018, vol. 11046 of *Lecture Notes in Computer Science*, pp. 46–54, Springer. 92

[179] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2016. 92

[180] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed, "Variational approaches for auto-encoding generative adversarial networks," *CoRR*, vol. abs/1706.04987, 2017. 92

[181] Bin Hou, Qingjie Liu, Heng Wang, and Yunhong Wang, "From w-net to CDGAN: bitemporal change detection via deep learning techniques," *IEEE Trans. Geosci. Remote. Sens.*, vol. 58, no. 3, pp. 1790–1802, 2020. 92

[182] Ksenia Bittner, Marco Körner, and Peter Reinartz, "DSM building shape refinement from combined remote sensing images based on WNET-CGANS," in *2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019*, 2019, pp. 783–786. 92

[183] Quan Liu, Isabella M. Gaeta, Bryan A. Millis, Matthew J. Tyska, and Yuankai Huo, "GAN based unsupervised segmentation: Should we match the exact number of objects," *CoRR*, vol. abs/2010.11438, 2020. 92

[184] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2242–2251. 92

[185] Mark E. Pennesi, Martha Neuringer, and Robert J. Courtney, "Animal models of age related macular degeneration," *Molecular aspects of medicine*, vol. 33(4), pp. 487–509, 2012. 93

[186] Jean-Louis Dessalles, "Que veut dire "comprendre" pour une machine?," in *Extraction et Gestion des Connaissances, EGC 2020, Brussels, Belgium, January 27-31, 2020*, 2020, pp. 1–2. 93

[187] Sean D. Holcomb, William K. Porter, Shaun V. Ault, Guifen Mao, and Jin Wang, "Overview on deepmind and its alphago zero AI," in *Proceedings of the 2018 International Conference on Big Data and Education, ICBDE 2018, Honolulu, HI, USA, March 09-11, 2018*, 2018, pp. 67–71. 94

[188] Kai Arulkumaran, Antoine Cully, and Julian Togelius, "Alphastar: an evolutionary computation perspective," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, 2019, pp. 314–315. 94

[189] Peter Welinder and Pietro Perona, "Online crowdsourcing: Rating annotators and obtaining cost-effective labels," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2010, San Francisco, CA, USA, 13-18 June, 2010*, 2010, pp. 25–32. 100

[190] Giorgos Mountrakis, Raymond Watts, Lori Luo, and Jida Wang, "Developing collaborative classifiers using an expert-based model," *Photogrammetric Engineering and Remote Sensing*, vol. 75, no. 7, pp. 831–843, 2009. 100

[191] Giles M. Foody and Doreen S. Boyd, "Using volunteered data in land cover map validation: Mapping west african forests," *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 6, no. 3, pp. 1305–1312, 2013. 100

[192] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, "Introduction to semi-supervised learning," in *Semi-Supervised Learning*, pp. 1–12. The MIT Press, 2006. 100

[193] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J. Mooney, "Probabilistic semi-supervised clustering with constraints," in *Semi-Supervised Learning*, pp. 73–102. The MIT Press, 2006. 100

[194] Y. Yuan and L. Lin, "Self-supervised pre-training of transformers for satellite image time series classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–1, 2020. 100

[195] Carl Doersch and Andrew Zisserman, "Multi-task self-supervised visual learning," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 2070–2079, IEEE Computer Society. 100

[196] Richard Zhang, Phillip Isola, and Alexei A. Efros, "Colorful image colorization," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds. 2016, vol. 9907 of *Lecture Notes in Computer Science*, pp. 649–666, Springer. 100

[197] Yan Xu, Fuming Sun, and Xue Zhang, "Literature survey of active learning in multimedia annotation and retrieval," in *International Conference on Internet Multimedia Computing and Service, ICIMCS '13, Huangshan, China - August 17 - 19, 2013*, 2013, pp. 237–242. 101

[198] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 606–617, 2011. 101

[199] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang, "A survey of deep active learning," *CoRR*, vol. abs/2009.00236, 2020. 101

[200] Jamal Uddin, Rozaida Ghazali, and Mustafa Mat Deris, "Does number of clusters effect the purity and entropy of clustering?," in *Recent Advances on Soft Computing and Data Mining - The Second International Conference on Soft Computing and Data Mining (SCDM-2016), Bandung, Indonesia, August 18-20, 2016, Proceedings*, 2016, pp. 355–365. 101

[201] Dong-Hyun Lee, "Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks," in *Proceedings of the ICML 2013 Workshop: Challenges in Representation Learning*, 2013. 101

[202] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," *CoRR*, vol. abs/1908.02983, 2019. 101

[203] Tao Li, Zhiyuan Liang, Sanyuan Zhao, Jiahao Gong, and Jianbing Shen, "Self-learning with rectification strategy for human parsing," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020, pp. 9260–9269. 101

[204] Yunhao Gao, Feng Gao, Junyu Dong, and Shengke Wang, "Transferred deep learning for sea ice change detection from synthetic-aperture radar images," *IEEE Geosci. Remote. Sens. Lett.*, vol. 16, no. 10, pp. 1655–1659, 2019. 101

[205] Fei-Fei Li, Robert Fergus, and Pietro Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006. 101

[206] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences*, vol. 102, no. 46, pp. 16569–16572, Nov 2005. 122

[207] Leo Egghe, "Theory and practice of the g-index," *Scientometrics*, vol. 69, no. 1, pp. 131–152, 2006. 122

Appendices

8.1 Résumé en français

8.1.1 Résumé

Ce manuscrit d'Habilitation à diriger des recherches est la synthèse de mes travaux réalisés depuis 2016 en tant qu'enseignant-chercheur à l'ISEP et chercheur au LIPN dans l'équipe A3-ADA. J'y présente mes travaux autour de mon thème central de recherche : l'apprentissage non-supervisé. Ces travaux s'articulent autour de 2 grands axes : l'apprentissage non-supervisé dans un contexte multi-vue, et l'apprentissage non-supervisé profond. Ces deux axes découlent directement de mes travaux de thèses sur le clustering collaboratif appliqué aux images satellite à haute résolution.

Mes travaux en apprentissage multi-vue non-supervisé abordent des thématiques telles que la confiance et la pondération des vues dans les environnements non-supervisés, les données manquantes dans un contexte multi-vue, mais aussi des aspects de modélisation du clustering multi-vue afin de théoriser des éléments tels que la stabilité de ce type de méthodes, mais aussi leur capacité à apporter de la nouveauté tout en gardant une cohérence avec les données locales.

Mes travaux sur l'apprentissage profond dans un cadre non-supervisé découlent quant à eux du constat que la majorité des méthodes d'apprentissage profond les plus performantes sont supervisées et nécessitent d'importants volumes de données annotées pour leur entraînement. Or, quand on regarde les applications concrètes en imagerie, on s'aperçoit que dans la plupart des cas, ces données annotées ne sont pas disponibles (ou le sont trop tard), sont coûteuses à produire, et que les modèles une fois entraînés sont difficilement transférables. J'aborde donc dans mes travaux des architectures d'apprentissage profond adaptées à des contextes non-supervisés. A travers des applications en imagerie satellite (étude d'urbanisation, cartographie automatique de dégâts de catastrophes naturelles, etc.), ainsi qu'en imagerie médicale (étude de pathologies de l'oeil), mes travaux se sont intéressés à des architectures originales et ont pu en étudier les points forts et les limites.

8.1.2 Synthèse de recherche

Ma thèse ayant porté sur du clustering multi-vue et collaboratif appliqué à des données d'imagerie satellite, mes travaux ont conservé cet ancrage et s'appuient sur deux axes de recherche avec comme point commun l'apprentissage non-supervisé :

- L'apprentissage non-supervisé multi-vue, avec quelques travaux explorant les thématiques connexes du clustering distribué et de l'apprentissage par ensemble non-supervisé.
- L'analyse d'images, et notamment l'utilisation de l'apprentissage profond dans des contextes non-supervisés pour de l'analyse de séries d'images satellites ou médicales.

Sur l'axe clustering multi-vue, après que ma thèse ait été centrée sur la proposition de nouvelles méthodes collaboratives et multi-vue permettant de diversifier les types d'algorithmes pouvant travailler ensemble, je me suis intéressé à la problématique d'évaluer la fiabilité des vues en contexte non-supervisé. Pour celà, j'ai proposé des méthodes d'optimisation mathématique permettant de pondérer le poids des vue, mais aussi de l'optimisation à partir de bandits non-stochastiques pour tenir compte d'une qualité pouvant varier pendant l'entrainement. Enfin, des méthodes basées sur

l'apprentissage profond ont également été proposées pour la reconstruction des données manquantes et l'évaluation automatique de la fiabilité des vues.

Le second volet de mes travaux sur le clustering multi-vue est plus théorique avec notamment la proposition d'utiliser la théorie de l'information et la complexité de Kolmogorov comme base pour des méthodes de clustering multi-vue ou des méthodes d'ensembles non-supervisées. D'autre part, je m'intéresse également -dans le cadre de travaux en cours- au portage des notions de stabilité et d'algorithme risque minimisant proposées par Shai Ben David pour clustering classique, vers le clustering multi-vue où ces propriétés n'ont pas du tout été étudiéees. D'autres notions que la stabilité ont également été étudiées: l'apport de nouveauté dans un contexte multi-vue, et la consistance entre les modèles locaux dans les vues avec un modèle global multi-vue. Des liens ont pu ainsi être découverts liant la stabilité, la nouveauté et la consistance dans un cadre multi-vue.

Enfin, dans le cadre d'une collaboration avec l'Université Pontificale Catholique de Valparaiso, certains de mes travaux en clustering multi-vue ont été repris pour faire de la classification de corpus de textes en combinant des représentations multiples (word2vec, n-grams, tf-idf, etc.).

Sur l'axe apprentissage profond et imagerie, je me suis intéressé à l'utilisation de l'apprentissage profond sur des applications d'imageries pour lesquelles peu d'images annotées sont disponibles ce qui rend impossible le recours aux techniques supervisées. J'ai notamment participé à des travaux sur les séries d'images satellites dans lesquels nous nous sommes intéressés à la détection de changements non-triviaux (constructions, changement de couverture des sols, mais pas les changements saisonniers), en ayant notamment recours à l'utilisation d'auto-encodeurs joints (convolutifs) appliqués à des pairs d'images et à une astuce sur l'erreur de reconstruction des auto-encodeurs qui ne peuvent en principe pas prédire ces changements non-triviaux ce qui les rend donc détectables à la reconstructions. De la détection de changements, nous sommes passés à l'analyse de séries complètes en couplant des modèles de type GRU (Gated recurrent units) avec des auto-encodeurs et des méthodes de synopsis basés sur les graphes pour faire du clustering de la couverture des sols et des changements sur des séries entières, le tout de manière totalement non-supervisée.

Ces travaux ont par la suite été appliqués à plusieurs cas pratiques: détection des suivi des évolutions urbaines sur les villes de Montpellier et Rostov, l'analyse des images du tsunami de Tohoku en 2011, mais ont surtout fait l'objet d'un portage dans le domaine médical. Nous avons en effet ré-appliqué avec succès l'algorithme de détection des changements non-triviaux sur des images de DMLA afin d'analyser l'évolution de la maladie examen après examen. Ce portage en médical ayant permis d'obtenir des financement et de démarrer des thèses, d'autres travaux sont actuellement en cours sur ce domaine, notamment en segmentation d'image non-supervisée en 2D et en 3D.

8.1.3 Projet de recherche à 4 ans

Mon projet de recherche à 4 ans s'articulera toujours autours des 2 axes évoqués précédemment, avec cependant une probabilité forte de sortir de l'apprentissage non-supervisé pur, afin d'aller plus vers des méthodes semi-supervisées, d'apprentissage actif ou de *One-short Learning*.

La première partie de mon projet à 4 ans va s'articuler autours des 2 encadrements de thèse démarrés cette année dans le cadre d'une collaboration entre l'ISEP et le centre Hospitalier des XV-XX :

- La première thèse porte sur l'évolution de la DMLA et est la continuité directe de mes travaux sur les séries d'images. Il s'agira ici de passer de l'analyse simple de séries, à la proposition de modèles prédictifs, toujours dans un contexte non-supervisé faute d'images annotées en nombre et fiabilité suffisante. Les pistes envisagées évoluent notamment autours de l'utilisation de GAN (réseaux adversariaux génératifs) pour générer les images des futures lésions, probablement en les combinant avec des modèles récurrents de type LSTM. En cas de difficultés à avoir des modèles totalement non-supervisés, un guidage à partir de segmentations des lésions acquises via des W-Net est envisagé.
- La seconde thèse porte sur le glaucome et notamment la segmentation des axiomes de la lame criblée de l'oeil à partir d'image OCT en coupe sous plusieurs angles. Ces images présentent des difficultés importantes du fait de la faible qualité des images, de leur non-continuité et la aussi du manque de données annotées. Les modèles probables pour ces données seront très probablement des W-Nets, possiblement combinés avec des réseaux récurrents là aussi, notamment si les non-continuités des coupes ne permettent pas de faire directement de segmentation 3D. Enfin, le recours au GAN n'est pas exclu pour améliorer la qualité de certaines images.

Sur ces deux sujets, une autre thématique importante sera l'interprétabilité des réseaux profonds utilisés, notamment de manière à pouvoir faire le lien avec des mécanismes sous-jacents des deux maladies. En effet, les mécanismes de la DMLA sont notoirement mal connus et avoir un modèle prédictif qu'on pourrait soit expliquer soit accoler à un modèle de croissance mathématique serait un véritable plus. L'apprentissage profond, et encore plus en non-supervisé n'étant pas connu pour son interprétabilité, il y a ici un travail de recherche très important auquel j'espère pouvoir apporter mes contributions.

Sur l'imagerie satellite, les modèles prédictifs sont moins envisageables du fait du nombre beaucoup plus important de classes par rapport aux images médicales, et c'est donc plutôt l'introduction de connaissance humaine extérieur qui va m'occuper ces prochaines années, avec notamment la modification d'algorithmes pour pouvoir faire de l'apprentissage semi-supervisé ou de l'apprentissage actif. En effet, s'il est actuellement possible de donner quelques données non-annotées à des algorithmes d'apprentissage profond supervisés, il n'est à l'inverse pas possible de donner quelques données annotées aux algorithmes non-supervisés. Ce serait pourtant un vrai plus pour guider les clusters vers des classes d'intérêt et augmenter ainsi la qualité des résultats. L'apprentissage en une fois (one shot learning) serait également une bonne solution d'amélioration des algorithmes d'apprentissage profond afin de diminuer leurs besoins en images annotées, tout en ne mettant pas une pression trop forte sur des opérateurs humains.

Concernant l'axe clustering, j'ai déjà évoqué la poursuite de mes travaux sur la stabilité dans un contexte multi-vue. Celà se ferait toujours en collaboration avec l'Université Paris 13 et Aalto University (Helsinki). Ayant obtenu un financement FONDECYT-CONICYT au Chili pour 3 ans, je compte également poursuivre mes travaux d'application de méthodes multi-vue ou d'ensemble pour les systèmes de classification ou de recommandations. Nous travaillons notamment en ce moment sur la proposition d'une méthode d'apprentissage par ensemble non-supervisée reposant sur la théorie de l'information. Il est à noter que les financements obtenus devant concerner des échanges de chercheur, et des co-encadrements de stagiaires et de doctorants, ils sont conditionnés

à l'évolution de l'épidémie de covid-19, et à l'obtention d'une demie-bourse de thèse supplémentaire dans le cas d'un recrutement de doctorant.

Enfin, plusieurs collègues de l'ISEP travaillant sur des thématique de signal m'ont fait part de leur intérêt sur l'utilisation de l'IA pour la future 6G. Des demandes de financement sont en cours. Si elles sont acceptées, elles donneront lieu de mon côté au développement de réseaux de neurones profonds pour de l'analyse et du traitement de signal en temps réel.

List of Figures

RÉPUBLIQUE FRANÇAISE

MINISTÈRE DE L'ÉDUCATION NATIONALE, DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

université PARIS-SACLAY

DOCTORAT

Vu le code de l'éducation, notamment ses articles L.612-7, L. 613-1, D. 613-3 et D. 613-6 ;
Vu le code de la recherche, notamment son article L.412-1 ;
Vu l'arrêté du 25 mai 2016 fixant le cadre national de la formation et les modalités conduisant à la délivrance du diplôme national de doctorat ;
Vu l'arrêté du 10 juillet 2015 portant accréditation de l'école doctorale ;
Vu les pièces justificatives produites par Monsieur Jérémie SUBLIME, né le 29 mai 1989, à Montmorency en vue de son inscription en doctorat ;

Vu le procès-verbal du jury attestant que l'intéressé a soutenu, le **09 novembre 2016**, une thèse portant sur le sujet suivant : **Contributions au clustering collaboratif et à ses potentielles applications en imagerie à très haute résolution** préparée au sein de l'école doctorale **Agriculture, Alimentation, Biologie, Environnement et Santé** devant un jury présidé par M. François YVON et composé de Mme Pascale KUNTZ, M. Michael AUPETIT, M. Antoine CORNUÉJOLS, M. Younès BENNANI ;

Vu la délibération du jury :

Le diplôme de **DOCTORAT** en **informatique appliquée** préparé **à l'Institut des Sciences et Industries du Vivant et de l'Environnement (AgroParisTech)** est délivré à **Monsieur Jérémie SUBLIME**
au titre de l'année universitaire 2016-2017
et confère le **grade de docteur**,
pour en jouir avec les droits et prérogatives qui y sont attachés.

Fait le 15 mai 2017,

Le titulaire

Le président de l'Université Paris Saclay

université PARIS-SACLAY
université-paris-saclay.fr
Espace Technologique
Immeuble Discovery
Route de l'Orme aux Merisiers
91190 SAINT AUBIN / France

Le recteur d'académie, chancelier des universités

ACADÉMIE DE VERSAILLES

Numéro du diplôme : 2016SACLA005

European higher education area - Espace européen de l'enseignement supérieur - European higher education area

RÉPUBLIQUE FRANÇAISE

Ministère de l'Enseignement Supérieur et de la Recherche
École Internationale des Sciences du Traitement de l'Information
Etablissement privé d'enseignement supérieur

DIPLÔME D'INGÉNIEUR
GRADE DE MASTER - MASTER'S DEGREE

Vu le code de l'éducation et notamment ses articles L. 642-1, L642-4 et D222-27
Vu le décret n° 99-747 du 30 août 1999, modifié relatif à la création du grade de master, notamment son article 2, alinéa 2,
Vu l'arrêté d'habilitation n° 2001 - 242 du 22-3-2001 habilitant l'établissement à délivrer le titre d'ingénieur diplômé EISTI
Vu le décret du 20 juin 2008 portant délégation d'attribution aux recteurs d'académie
Vu le procès-verbal du jury attestant que SUBLIME JÉRÉMIE né le 29 mai 1989 à MONTMORENCY (VAL-D'OISE)
a satisfait à l'ensemble des obligations prévues pour la délivrance du diplôme d'ingénieur

Le titre d'ingénieur diplômé de l'Ecole Internationale des Sciences du Traitement de l'Information
Spécialité Génie Informatique
est délivré, au titre de l'année universitaire 2012/2013 à :

SUBLIME JÉRÉMIE

à qui est conféré le grade de master.

Fait à Versailles, le 17 octobre 2013

Le titulaire,

N° d'enregistrement 2633

Le Directeur Général,

Nesim FINTZ

ACADEMIE DE VERSAILLES

Le Recteur de l'Académie

Pierre-Yves DUWOYE

Espace européen de l'enseignement supérieur - European higher education area - Espace européen de l'enseignement supérieur

Inha University

UPON THE RECOMMENDATION OF THE FACULTY OF THE

Graduate School

HAS CONFERRED UPON

Jeremie Sublime

THE DEGREE OF

Master of Engineering(M.E.)

In Computer and Information Engineering

WITH ALL THE RIGHTS AND PRIVILEGES PERTAINING TO THAT DEGREE.

AWARDED AT INCHEON, KOREA

THIS SIXTEENTH DAY OF AUGUST, TWO THOUSAND AND THIRTEEN.

Chong Ho Lee

CHONG-HO LEE, PH.D
DEAN

CB Park

CHOON-BAE PARK, PH.D.
PRESIDENT

28, rue Notre-Dame des Champs
75006 Paris
Tél : 01 49 54 52 00
Fax : 01 49 54 52 01

ATTESTATION TAUX D'ENCADREMENT THESE

Je soussignée, Maria TROCAN, directrice de la thèse de Mlle. Ekaterina KALINICHEVA, certifie que dr. Jérémie SUBLIME a co-encadré avec moi à 60% cette thèse qui porte sur l' « Analyse non-supervisée de séries d'images satellites avec des méthodes d'apprentissage profond ».

Pour faire valoir ce que de droit,
Paris, 02/09/2020

INSTITUT SUPERIEUR D'ELECTRONIQUE DE PARIS
28, rue Notre-Dame-des-Champs
75006 PARIS

10 Rue des Vanves,
92130, Issy-les-Moulineaux

Tél. : 01 49 54 52 00
Fax. : 01 49 54 52 01

isep
École d'ingénieurs du numérique

Paris – Mercredi 9 septembre 2020

Objet : Attestation d'encadrement de thèse

Je soussignée Raja CHIKY, professeur en Informatique au laboratoire LISITE de l'ISEP, et directrice de la thèse de Mr Denis MAUREL intitulée « Contributions aux communications inter-vues pour l'apprentissage collaboratif » et soutenue le 10 décembre 2020, atteste que Mr Jérémie SUBLIME a participé à l'encadrement de cette thèse à hauteur de 50%.
L'encadrement était réparti comme suit :

- Mr Jérémie SUBLIME : 50%
- Mr Sylvain LEFEBVRE : 30%
- Mme Raja CHIKY : 20%

Pour faire valoir ce que de droit,

Prof. Raja CHIKY
Professeur en Informatique
Directrice de l'Innovation
10 rue de Vanves 92130 Issy Les Moulineaux
Tel : 0149545234
Mob : 0684674595
raja.chiky@isep.fr

INSTITUT SUPERIEUR D'ELECTRONIQUE DE PARIS
I.S.E.P.
28, rue N.-D. des Champs
75006 PARIS
E' Privé

INSTITUT SUPERIEUR D'ELECTRONIQUE DE PARIS

ISEP
10, rue de Vanves
92130 Issy-les-Moulineaux

Prof. Florence Rossant
Responsable de l'équipe DaSSIP
florence.rossant@isep.fr
tel : 33 1.49.54.52.62

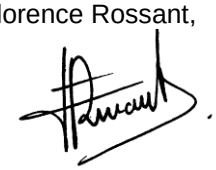
Issy-les-Moulineaux, le 5 Janvier 2021

Attestation d'encadrement de thèse

Je, soussignée Florence Rossant, Professeur à l'ISEP, atteste que Mr Jérémie Sublime co-encadre avec moi deux thèses dont je suis la directrice. Voici le détail de ces encadrements :

- Thèse de Mr Clément Royer, commencée en Octobre 2020 : Jérémie Sublime (50%), Florence Rossant (40%), Michel Pâques (10%)
- Thèse de Mme Nan Ding, commencée en Novembre 2020 : Jérémie Sublime (10%), Hélène Urien (50%), Florence Rossant (30%), Michel Pâques (10%)

Florence Rossant,



École d'ingénieurs du numérique

Paris, le 15 janvier 2021

Je soussigné, Louis-Joseph Brossollet, directeur de l'enseignement de l'Institut Supérieur de l'Electronique de Paris (ISEP) certifie que M. Jérémie Sublime, enseignant-chercheur à l'ISEP depuis septembre 2016, dédie 50% de ses activités à l'enseignement. Sa charge d'enseignement est ainsi de 275 heures équivalent TDs.

Ses thématiques d'enseignement, indifféremment en anglais et en français, à tous niveaux de L1 à M2, portent essentiellement sur la science des données et l'informatique, et leurs applications.

De plus, M Sublime est responsable du parcours « Intelligence des Données » (spécialité des années 2 et 3 du cycle ingénieur) depuis septembre 2017. Ses activités de recherche nourrissent en permanence ses activités d'enseignement par le biais de l'évolution des programmes et de projets associant des étudiants. Il est également le point de contact entre l'ISEP et l'Université Paris Saclay – Institut Polytechnique de Paris sur le double diplôme Ingénieur ISEP – Master « Science des données » ; contribuant notamment à la sélection initiale des élèves de l'ISEP et à leur évaluation finale.

En synthèse, M. Sublime est un enseignant dynamique, apprécié des élèves et des collègues. Il est un acteur important de la qualité, de l'évolution et du rayonnement des programmes d'enseignement à l'ISEP.

Louis-Joseph Brossollet