

People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
8 Mai 1945 Guelma University



Faculty of Mathematics, Computer Science and Material Sciences  
Department of Computer Science  
Laboratory of Information and Communication Sciences and Technologies

**THESIS**  
**IN VIEW OF OBTAINING**  
**THE DOCTORATE DEGREE IN 3<sup>rd</sup> CYCLE**

**Domain:** Mathematics and Computer Science. **Field:** Computer Science  
**Specialty:** Computer Science

**Presented by:**

**Nebili Wafa**

*Entitled*

**Recognition and interpretation of activities in videos**

Defended: .....

In front of the jury composed of:

<b>Mr. Nemissi Mohamed</b>	<b>Professor</b>	Univ. of 8 Mai 1945, Guelma	President
<b>Mr. Seridi Hamid</b>	<b>Professor</b>	Univ. of 8 Mai 1945, Guelma	Supervisor
<b>Mr. Farou Brahim</b>	<b>MCA</b>	Univ. of 8 Mai 1945, Guelma	Co-supervisor
<b>Mr. Benmohammed Mohamed</b>	<b>Professor</b>	Univ. of Abdelhamid Mehri, Constantine 2	Examiner
<b>Mr. Farah Nadir</b>	<b>Professor</b>	Univ. of Badji Mokhtar, Annaba	Examiner
<b>Mr. Bouhadada Tahar</b>	<b>Professor</b>	Univ. of Badji Mokhtar, Annaba	Examiner
<b>Mr. Kouahla Mohamed Nadjib</b>	<b>MCA</b>	Univ. of 8 Mai 1945, Guelma	Invited member

2020-2021

# DEDICATIONS

*To My Parents,*

*To My Dear Sisters: Meriem and Alir,*

*To My Dear Brother and the Great Programmer: Charefeddine  
Tchi,*

*To My Husband, Bendjedid Bilal,*

*To My Darling Nephew Djoud,*

*To My Aunt Sabrina and their Husband Madjid,*

*To My Husband's Mother, Braïka,*

*To My Uncles: Radwane, Wahel, Mouhamed Fouri and their  
Families,*

*To All My Friends: Amel, Manel, Rokaii, Nada, Imen, Houda,  
Khawla and Samia,*

*To All My Family,*

*To All My Friends*

# THANKS

First of all, I would like to thank God (**Allah**), who has given me the willingness, the patience and the power to realize this thesis.

I sincerely thank my thesis director Professor **Seridi Hamid** for his invaluable advice, his help throughout the period of this work.

I would like to express all my gratitude to Dr. **Farou Brahim**, my co-supervisor, who allowed me to discover a very interesting field "video surveillance".

Thank you Pr.**Seridi Hamid** and Dr.**Farou Brahim** for encouraging me in the most difficult moments. I will never forget your kindness, your availability and your precious help in my research work.

I would like to express my deep gratitude to Dr.**Bendjabbar Safia**, Dr.**Bendjaballah Malek**, Dr.**Kouahla Zineddine** and Mrs **Gabriela Kouahla** for their precious help especially in the technical level.

I would like also to thank all the teachers of Computer Science department of 8 Mai 1945 Guelma university for the quality of their formation.

A big thanks to the LabSTIC laboratory team who contributed, each in their own way, and i do not forget the laboratory engineer Miss **Kharoubi Madiha**.

My sincere thanks: to my parents, my sisters, my brother and my husband who supported me a lot during the difficult times.

Finally, my thanks to my friends and colleagues, especially the 2017-2018 promotion.

## ملخص

طرح الخلفية تصنف من بين الأنشطة الصعبة في مجال الرؤية بالكمبيوتر. حيث أن طبيعة البيئة قد تنطوي على تغييرات معينة بسبب تأثير الضوء ، الخلفية الديناميكية ، الظل ، تأثير التمويه ، .. إلخ. هناك العديد من الطرق التي تم اقتراحها لاكتشاف الأجسام المتحركة ، لكن معظمها فشل في تمثيل الوسائط المتعددة للمشاهد. بناءً على نظام المناعة المصطنع ، تم اقتراح طرق فعالة لطرح الخلفية. كاقترح أول ، تم دمج طريقة غوسي واحد مع نظام المناعة المصطنع لتقديم أفضل تمثيل لتغيرات البكسل في المشاهد التي تحتوي على خلفية ديناميكية. يتم استخدام نظام المناعة الاصطناعي كأداة تصنيف تفصل بين المستضدات الممثلة بوحدات البكسل الأمامية من الأجسام المضادة التي تمثل وحدات البكسل الخلفية. تم تصميم كل بكسل من هذا الاقتراح باستخدام شعاع ميزات يحتوي على سمات غوسية. كاقترح ثان ، استخدمنا نظام المناعة المصطنع لتحديد عدد الغوسيات في نموذج خليط الغوسيات بشكل ديناميكي بدلاً من تحديدهم من قبل المستخدم. في هذا الاقتراح، تم إنشاء مجموعة من الغوسيات الجدد باستخدام استراتيجيتين مختلفتين: الأولى (التوليد العشوائي) تستخدم نظام المناعة المصطنع لتحسين قرار النظام، بينما في الاستراتيجية الثانية (التوليد الموجه) ، يتم استخدام نظام المناعة المصطنع لتوجيه إنتاج الغوسيات في نموذج خليط الغوسيات. لتقليل تأثير الإضاءة ، يتم تحويل كل صورة في مقطع الفيديو من فضاء الألوان RGB إلى HSV. نظام المناعة الاصطناعي استفاد أيضاً من بعض التعديلات لتقليل اعباء البحث ومنع حدوث تضخم البيانات في مجموعة خلايا الذاكرة. من أجل تقليل تكلفة البحث على خلية الذاكرة الأكثر تمثيلاً للمستضد، قمنا بإعادة هيكلة بنية مجموعة خلايا الذاكرة على شكل شجرة ثنائية (kd-tree). بالإضافة إلى ذلك ، أضفنا آليتين أساسيتين لنظام المناعة المصطنع لتجنب حدوث تضخم في البيانات في مجموعة خلايا الذاكرة. قمنا بتنفيذ واختبار الطرق المقترحة على مجموعات بيانات معروفة. النتائج التي تم الحصول عليها مرضية إلى حد كبير مقارنة بطرق أخرى.

**الكلمات المفتاحية:** تعدين البيانات ، طرح الخلفية ، الأجسام المتحركة ، بكسل الخلفية ، بكسل امامي، نظام المناعة الاصطناعي، الكشف ، المراقبة بالفيديو ، النموذج الغاوسي.

# Résumé

La soustraction de l'arrière-plan parmi les activités difficiles dans la vision par ordinateur, puisque la nature de l'environnement pouvant impliquer certains changements dus à l'effet de la lumière, l'arrière-plan dynamique, l'ombre, l'effet de camouflage, ..etc. Plusieurs méthodes de détection des objets en mouvements sont proposées, mais la majorité d'entre eux non pas réussi à gérer la multimodalité des scènes. On se basant sur le système immunitaire artificielle de reconnaissance, des méthodes efficaces de soustraction de l'arrière-plan sont proposées. Dans la première proposition, la méthode Single Gaussienne est combinée avec le système immunitaire artificielle de reconnaissance pour mieux représenter les variations de pixels dans les scènes qui contiennent un arrière-plan dynamique. Le système immunitaire artificielle de reconnaissance est utilisé comme un outil de classification qui sépare les antigènes représentés par les pixels de premier plan des anticorps qui modélisent les pixels d'arrière-plan. Chaque pixel de cette proposition est modélisé avec un vecteur de caractéristique contenant des attributs de Gaussien. Comme deuxième contribution, nous avons utilisé le système immunitaire artificielle de reconnaissance pour gérer dynamiquement le nombre des Gaussiens dans le modèle mélange des gaussiens au lieu de fixer de l'avance par l'utilisateur. Dans cette contribution, un ensemble des nouveaux Gaussiens sont générés en utilisant deux stratégies différentes: la première (génération aléatoire) utilise le système immunitaire artificielle de reconnaissance pour améliorer la décision du système, tandis que dans la deuxième stratégie (génération dirigée) améliore la production des meilleurs modèles représentants de l'arrière-plan. Pour réduire l'effet de luminosité, chaque image de la séquence vidéo est transformée de l'espace de couleur RVB à l'espace HSV. Le système immunitaire artificielle de reconnaissance a aussi bénéficié de certaines modifications pour réduire le coût de recherche sur un modèle plus représentant de l'antigène actuel et éviter l'explosion de données dans l'ensemble de cellules mémoire. Afin de réduire le coût de recherche sur la cellule mémoire la plus représentantes à l'antigène, la structure de l'ensemble des cellules mémoire est redéfini comme une arbre binaire (kd-tree). De plus, nous avons ajouté deux mécanismes au système immunitaire artificielle de reconnaissance de base pour éviter l'explosion des données dans l'ensemble des cellules mémoire. Les méthodes proposées sont implémentées et testées sur des bases des données publiques. Les résultats obtenus sont largement satisfaisants par rapport aux autres méthodes cités dans l'état de l'art.

**Mots-Clés:** Data Mining, soustraction d'arrière-plan, objets en mouvement, pixel arrière-plan, pixel avant plan, système immunitaire artificielle de reconnaissance, détection, vidéo-surveillance, modèle gaussien.

# Abstract

Background subtraction is among the difficult activities in computer vision. Since the nature of the environment may involve some changes due to light effect, dynamic background, shadow, camouflage effect, etc. Many moving object detection methods are proposed, but the majority of them fails to handle the multi-modality of scenes. Based on the Artificial Immune Recognition System, efficient background subtraction methods are proposed. In the first method, the Single Gaussian model is combined with the Artificial Immune Recognition System to better represent pixel variations in the scenes that contain dynamic background. Artificial Immune Recognition System is used as a classification tool that separates antigens represented by the foreground pixels from antibodies that modelled the background pixels. Each pixel in this proposition is modelled with a feature vector contains Gaussian attributes. As a second contribution, we have used the Artificial Immune Recognition System for managing the number of Gaussians dynamically in Gaussian Mixture Model instead to fix them a priori by the user. In this contribution, a set of new Gaussians is generated using two different strategies: the first one (Random generation) uses the Artificial Immune Recognition System for improving the system decision, while in the second strategy (Directed generation), the Artificial Immune Recognition System is used to improve the production of background models. To reduce the effect of brightness, each frame in the video sequence is transformed from the RGB to HSV color space. Artificial Immune Recognition System has also benefited from some modifications to reduce research cost and to avoid the explosion of data in the memory cells set. For reducing the research cost on the most representative memory cell to the current antigen, the structure of the memory cells set is redefined as a binary tree (kd-tree). Furthermore, we have added two mechanisms to the basic Artificial Immune Recognition System to avoid the explosion of data in the memory cells set. The proposed methods are implemented and tested on public datasets. The obtained results are largely satisfactory compared to other state-of-the-art methods.

**Key-words:** Data Mining, Background Subtraction, Moving Objects, Background Pixel, Foreground Pixel, Artificial Immune Recognition System, detection, Video Surveillance, Gaussian Model.

# Contents

ملخص	i
Résumé	ii
Abstract	iii
List of Figures	vii
List of Tables	viii
List of Acronyms	ix
General introduction	1
<b>I State of the art</b>	<b>4</b>
<b>1 Data Mining</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 Definition of Data Mining . . . . .	5
1.3 Data Mining applications . . . . .	6
1.4 The objective of Data Mining . . . . .	7
1.5 Data Mining techniques . . . . .	8
1.5.1 Principal Component Analysis (PCA) . . . . .	8
1.5.2 Apriori Algorithm . . . . .	9
1.5.3 K-means . . . . .	10
1.5.4 Fuzzy C-Means (FCM) . . . . .	12
1.5.5 Hierarchical Classification . . . . .	14
1.5.6 Artificial Neural Network (ANN) . . . . .	16
1.5.7 Kohonen Self-Organized Maps (KSOM) . . . . .	17
1.5.8 Artificial Immune Systems (AIS) . . . . .	19
1.5.9 Genetic Algorithms (GA) . . . . .	20
1.5.10 Particle Swarm Optimization (PSO) . . . . .	22
1.5.11 Grey Wolf Optimizer (GWO) . . . . .	23
1.5.12 Ant colonies . . . . .	25
1.5.13 Bee colonies . . . . .	26
1.5.14 Discriminant Analysis (DA) . . . . .	28
1.5.15 Regression Analysis . . . . .	29
1.5.16 Naïve Bayes (NB) . . . . .	30
1.5.17 Support Vector Machine (SVM) . . . . .	31

1.5.18	K Nearest Neighbors (KNN)	32
1.5.19	Adaptative Boosting (Adaboost)	33
1.5.20	Decision Tree	35
1.6	Proposed taxonomy	36
1.7	Conclusion	38
<b>2</b>	<b>Background Subtraction for video surveillance</b>	<b>40</b>
2.1	Introduction	40
2.2	Video surveillance	41
2.2.1	Definition	41
2.2.2	Video properties	41
2.2.2.1	Number of frames by second	41
2.2.2.2	Rate	41
2.2.2.3	Variable Bit Rate (VBR)	41
2.2.2.4	Interlacing	41
2.2.2.5	Display resolution	42
2.2.2.6	Proportion	42
2.2.3	Video surveillance applications	42
2.3	Background subtraction	44
2.3.1	Definition	44
2.3.2	Background subtraction methods	45
2.3.2.1	Single Gaussian model (SG)	45
2.3.2.2	Gaussian Mixture Model (GMM)	47
2.3.2.3	Generalized Single Gaussian model (GSG)	50
2.3.2.4	Generalized Gaussian Mixture Model (GGMM)	51
2.3.2.5	Kernel Density Estimation (KDE)	52
2.3.2.6	Support Vector Machine (SVM)	54
2.3.2.7	Support Vector Regression (SVR)	54
2.3.2.8	Support Vector Data Description (SVDD)	55
2.3.2.9	Recursive learning	57
2.3.2.10	Wiener filter	58
2.3.2.11	Kalman filter	59
2.3.2.12	Frame Difference Method (FDM)	60
2.3.2.13	Codebook	61
2.3.2.14	Sub-spatial learning based on PCA	62
2.3.2.15	Neural network methods	64
2.4	Proposed taxonomy	64
2.5	Conclusion	66
<b>II</b>	<b>Proposed systems</b>	<b>67</b>
<b>3</b>	<b>An Enhanced Artificial Immune Recognition System (EAIRS)</b>	<b>68</b>
3.1	Introduction	68
3.2	Concepts	69
3.3	AIRS stages and proposed modifications	70
3.3.1	Initialization	70
3.3.2	Memory cell identification and ARB generation	73



3.3.3	Competition for resources and development of a candidate memory cell . . . . .	79
3.3.4	Memory cell introduction . . . . .	82
3.3.5	Classification . . . . .	83
3.4	Tests and results . . . . .	84
3.4.1	Datasets . . . . .	84
3.4.2	Results . . . . .	87
3.4.3	Discussions . . . . .	89
3.5	Conclusion . . . . .	90
<b>4</b>	<b>Background subtraction using Artificial Immune Recognition System</b>	<b>91</b>
4.1	Introduction . . . . .	91
4.2	Background subtraction using Artificial Immune Recognition System and Single Gaussian (AIRS-SG) . . . . .	92
4.2.1	Why AIRS and SG . . . . .	92
4.2.2	The global architecture . . . . .	92
4.2.2.1	Pre-processing . . . . .	93
4.2.2.2	AIRS-SG model . . . . .	94
4.2.2.3	Post processing . . . . .	96
4.2.3	Experiments and results . . . . .	99
4.2.3.1	Settings . . . . .	99
4.2.3.2	Datasets . . . . .	99
4.2.3.3	Performance evaluation . . . . .	101
4.3	Using Resources Competition and Memory Cell Development to Select the Best Gaussian Mixture Model (GMM) for Background Subtraction . . . . .	107
4.3.1	Proposition One (Random Generation) . . . . .	109
4.3.2	Proposition Two (Directed Generation) . . . . .	111
4.3.3	Experimental validation and discussion . . . . .	111
4.3.3.1	Qualitative results . . . . .	112
4.3.3.2	Quantitative results . . . . .	112
4.3.3.3	Discussions . . . . .	114
4.4	Conclusion . . . . .	115
	<b>General conclusion</b>	<b>117</b>
	<b>Bibliography</b>	<b>147</b>
	<b>Author's publications</b>	<b>148</b>

# List of Figures

1.1	Data Mining applications . . . . .	7
1.2	Principal Component Analysis steps . . . . .	8
1.3	Steps of apriori algorithm . . . . .	9
1.4	K-means steps . . . . .	11
1.5	Fuzzy C-Means steps . . . . .	13
1.6	Descending hierarchical classification . . . . .	15
1.7	Ascending Hierarchical Classification . . . . .	15
1.8	Kohonen Self-Organized Maps steps . . . . .	18
1.9	Genetic Algorithms steps . . . . .	20
1.10	Particle Swarm Optimization algorithm . . . . .	22
1.11	Grey Wolf Optimizer algorithm . . . . .	24
1.12	Ant system algorithm . . . . .	25
1.13	Artificial bee colony algorithm . . . . .	27
1.14	Regression Analysis steps . . . . .	29
1.15	K Nearest Neighbors steps . . . . .	32
1.16	Adaptative Boosting algorithm . . . . .	34
1.17	Proposed taxonomy for Data Mining techniques . . . . .	37
2.1	Video surveillance applications . . . . .	43
2.2	Steps of background subtraction methods . . . . .	44
2.3	Flowchart of Single Gaussian model . . . . .	46
2.4	Flowchart of Gaussian Mixture Model . . . . .	49
2.5	Proposed taxonomy for background subtraction methods . . . . .	66
3.1	Flowchart of the proposed initialization of MC and ARB sets . . . . .	72
3.2	Histogram of the classification accuracy of AIRS, AIRS2 and EAIRS on the datasets: Iris, Pima-Diabetes, Inosphere and Sonar . . . . .	88
4.1	The global architecture of AIRS-SG . . . . .	93
4.2	AIRS-SG flowchart . . . . .	97
4.3	Flowchart of GMM-AIRS . . . . .	109

# List of Tables

3.1	Description of the attributes used in Pima-Diabetes dataset . . . . .	85
3.2	Description of Hepatitis attributes . . . . .	86
3.3	Description of Wisconsin-Breast Cancer attributes . . . . .	86
3.4	Description of the used datasets . . . . .	87
3.5	Our obtained results on the datasets: Iris, Pima-Diabetes, Inosphere, Sonar, Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer . . . . .	87
3.6	Classification accuracy of EAIRS, AIRS, AIRS2 on the datasets: Iris, Pima-Diabetes, Inosphere and Sonar . . . . .	87
3.7	Comparison of EAIRS results with other methods on: Iris, Pima-Diabetes, Inosphere and Sonar datasets . . . . .	88
3.8	Comparison of the obtained results of EAIRS with other methods on: Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer datasets . . . . .	89
4.1	Description of Dynamic Background category of CDnet 2014 dataset . . .	100
4.2	Some images of our results on the Dynamic Background category of the dataset CDnet 2014 . . . . .	102
4.3	Some results of AIRS-SG on the datasets Wallflower (WavingTrees), WaterSurface, Fountain and UCSD . . . . .	102
4.4	Re, Sp, FPR, FNR, PWC, Precision, F-Measure of AIRS-SG on CDnet 2014 Dynamic Background category . . . . .	103
4.5	Comparison of quantitative results with well-known background subtraction methods on Fountain01 video . . . . .	104
4.6	Comparison of quantitative results with well-known background subtraction methods on Fountain02 video . . . . .	104
4.7	Comparison of quantitative results with well-known background subtraction methods on Canoe video . . . . .	105
4.8	Comparison of quantitative results with well-known background subtraction methods on Boats video . . . . .	105
4.9	Comparison of quantitative results with well-known background subtraction methods on Overpass video . . . . .	106
4.10	Comparison of quantitative results with well-known background subtraction methods on Fall video . . . . .	106
4.11	Comparison of quantitative results with well-known background subtraction methods on Wallflower dataset . . . . .	112
4.12	Qualitative results on Wallflower dataset . . . . .	113
4.13	Qualitative results on UCSD dataset . . . . .	114

# List of Acronyms

<b>AIRS</b>	Artificial Immune Recognition System
<b>ANN</b>	Artificial Neural Network
<b>DA</b>	Discriminant Analysis
<b>FCM</b>	Fuzzy C-Means
<b>GA</b>	Genetic Algorithms
<b>GMM</b>	Gaussian Mixture Model
<b>GGMM</b>	Generalized Gaussian Mixture Model
<b>GWO</b>	Grey Wolf Optimizer
<b>HSV</b>	Hue Saturation Value
<b>ICA</b>	Independent Component Analysis
<b>KDE</b>	Kernel Density Estimation
<b>KNN</b>	K-Nearest Neighbors
<b>KSOM</b>	Kohonen Self-Organized Maps
<b>MoG</b>	Mixtures of Gaussians
<b>NB</b>	Naïve Bayes
<b>NTSC</b>	National Television System Committee
<b>PCA</b>	Principal Component Analysis
<b>RGB</b>	Red Green Blue
<b>SG</b>	Single Gaussian
<b>SGG</b>	Single General Gaussian
<b>SL-PCA</b>	Sequential Logistic Principal Component Analysis
<b>SVDD</b>	Support Vector Data Description
<b>SVM</b>	Support Vector Machines
<b>SVR</b>	Support Vector Regression

# General introduction

To provide security in public and private places, officials have set up an infrastructure that contains hundreds of surveillance cameras in major cities. These cameras offer only the possibility of storing and broadcasting the captured videos in order to treat them manually by agents specialized in the detection of anomalies and suspicious behaviours. Faced to this number of camera, it is very difficult, if not impossible, to place a human operator behind each camera. Furthermore, human nature tends to get bored with repetitive tasks, which decrease the performance of the monitoring task. With the development of the technology, it appeared intelligent systems called video surveillance systems. These systems have substituted the human operator in the monitoring task. By installing surveillance cameras in a specific location, these systems can keep track of people and detect suspicious scenes, which facilitates making a quick decision based on the analysis of these scenes.

Background subtraction (also called motion detection or moving objects segmentation) is a crucial step in any video processing system since it allows the identification of moving objects from the background (static objects). Background subtraction process, taken a sequence of images coded in any color space (RGB, HSV, etc) and produced black and white images. Black pixels represent the background, while white pixels constitute the foreground. To subtract the background, you should create a background model using a set of frames devoid from the moving objects, then using an on-line or an off-line process, you can identify the foreground and the background pixels.

The one-line process allows updating the background model during the whole execution to pick up changes of the background during the video sequences, while in the off-line process, the background model is not changed.

An efficient method for detecting the moving objects should separate the background from the foreground as well as possible while minimizing execution time and memory space.

In the literature, several methods have been proposed to improve the quality of background subtraction methods while minimizing false detections. Among methods that have achieved great success are Single Gaussian (SG) and Gaussian Mixture Model (GMM) since they are simple, very fast and not expensive in the calculation.

However, these methods impose many problems. For example, SG method is sensitive to

the fast variation of the dynamic background, indeed one Gaussian cannot memorize all states of the pixel.

However, these methods impose many problems. For example, the SG method is sensitive to the fast variation of the dynamic background. Indeed one Gaussian cannot memorize all states of the pixel.

The number of Gaussians represents a major drawback in GMM which can influence the results quality. Indeed, a small number of Gaussians can reduce the history of the pixel, which causes a problem if the background is dynamic. On the other hand, if we use a very large number of Gaussians without updating the number of Gaussians, we cannot correctly classify the stationary objects that become moving objects.

Recently, scientific research has exploited Data mining techniques to reduce the quantity of data and preserve from them only the most important (knowledge). Data mining techniques contain several methods grouped into three main families: descriptive techniques, bio-inspired techniques and predictive techniques.

Bio-inspired techniques are widely used to solve complex optimization and machine learning problems. These techniques try to model the natural systems or to describe the biological adaptations of organisms in their environment.

Artificial Immune Recognition System (AIRS) is among the bio-inspired approaches that have achieved considerable success. This approach exploited different aspects of the natural immune system, such as the concept of memorization, recognition, cloning, mutation, etc, to solve complex problems. Despite the great success of the AIRS, the latter has some drawbacks that influence their performance in terms of data explosion, results quality and calculation cost.

We will present in this thesis new systems for background subtraction using the AIRS approach. SG is combined with AIRS to better represent pixel variations in the scenes that contain dynamic backgrounds. AIRS is used as a classification tool that separates the antigens represented by the foreground pixels from antibodies that modelled the background pixels. Each pixel in this proposition is modelled with a feature vector that contains Gaussian attributes. In the same context, AIRS is used to create two methods (Random generation, Directed generation) for managing the number of Gaussians dynamically in the GMM instead to fix them a priori by the user. This proposition allows to generate new Gaussians using two different strategies: in the first one (Random generation), we use the AIRS to improve the system decision, while in the second strategy (Directed generation), the AIRS is used to improve the production of background models. To reduce the effect of brightness, we have firstly transformed each frame from the RGB to HSV color space. The original version of AIRS has also benefited from some modifications to overcome the problems cited above. For reducing the research cost, the structure of the memory cells set (MC) set is redefined as a binary tree (kd-tree). We have also introduced two new mechanisms to the basic AIRS to avoid the explosion of data in MC set.

To cover all proposals, this thesis is organized into four chapters described as follow:

- **Chapter 1:** presents an in-depth study on the different Data Mining techniques, their drawbacks and their recent improvements.
- **Chapter 2:** addresses a state of the art of the recent background subtraction methods.
- **Chapter 3:** gives an overview of the Artificial Immune Recognition System (AIRS): it discusses the different stages of the AIRS, their drawbacks and the modifications which we have introduced on the original version of the AIRS to overcome these drawbacks.
- **Chapter 4:** describes the global architecture of our proposed systems for subtracting the background either in scenes that have a dynamic background or to trait other problems.

We will finish our thesis with a conclusion, and we will give some perspectives concerning our propositions in the background subtraction domain.

# Part I

## State of the art



# 1

## Data Mining

### 1.1 Introduction

Recently, due to the advancement of technology, data quantity has greatly increased. In fact, a large part of this data is not useful, the most important is the knowledge. Data Mining has appeared as a new strategy that analyzed large quantities of data using a set of techniques to identify patterns or relationships that represent better the data.

In this chapter, we will give an overview of the techniques of Data Mining and the recently proposed invariants. At the end of this study, we will give new taxonomy that organizes these techniques into categories.

### 1.2 Definition of Data Mining

Data Mining represents a new methodology to extract implicit information (knowledge) from a large amount of data. It is also defined as a process for finding hidden information from a huge database.

Data Mining used a set of techniques to create representative patterns that convert raw data into information understandable from the users [UM17], [SSSS14].

## 1.3 Data Mining applications

Nowadays, Data Mining has attracted the attention of users due to its high capacity to convert a huge amount of data into useful information [SSSS14] [Win21a] [Win21d]. Users have exploited this capacity in several applications such as:

- **Marketing:** customer profiling, retention, identification of potential customer, market segmentation, shopping model, etc.
- **Fraud detection:** identification of credit card fraud and intrusion detection.
- **Text and web mining:** Data Mining can be used for searching text or information on the web or for giving raw data.
- **Bioinformatics and Genome:** among the applications of Data Mining in the Bioinformatics field we find:
  - Structural pattern discovery.
  - ADN Mining and identification of genes.
  - Analysis of genetic networks and protein pathways.
- **Medicine and pharmacy:** Data Mining is used in the diagnosis to discover diseases from symptoms of the patient's or to choose the most appropriate medication to cure a given disease.
- **Telecommunication:** Data Mining plays an important role in mobile telecommunications, mobile computing, and information services. It helps to compare information, such as user group behavior, profits, data traffic, system overload, etc.
- **Pattern recognition and Video surveillance:** such as recognition of numbers and letters, recognition of objects in images, interpretation of activities in videos, background subtraction, etc.
- **E-learning:** Data Mining can help institutions to predict students that would enroll in a particular program, who would require additional assistance to graduate, refine the overall management of registrations.
- **Energy Industry:** techniques of Data Mining are more used in the industry. They help to provide feasible solutions for decision-making and management. Furthermore, we can with the Data Mining predict risks, power outputs, and the clearing price of electricity.
- **Geography:** many Geographic Information Systems (GIS) and navigation applications use Data mining techniques. Data Mining is used in the extraction of geographical, environmental, and astronomical data.

- **Criminal Investigation:** Data Mining is used in the study of crime characteristics. At first, text-based crime reports are converted into word processing files. After that, the identification and crime-machining process are done by discovering patterns in massive stores of data.
- **Manufacturing engineering:** we can exploit Data Mining techniques as extraction tools of the relationships between portfolios and product architectures.

And there are many other applications that used Data Mining such as chemical engineering, fluid dynamics, climate, and any system that contains a huge dataset.

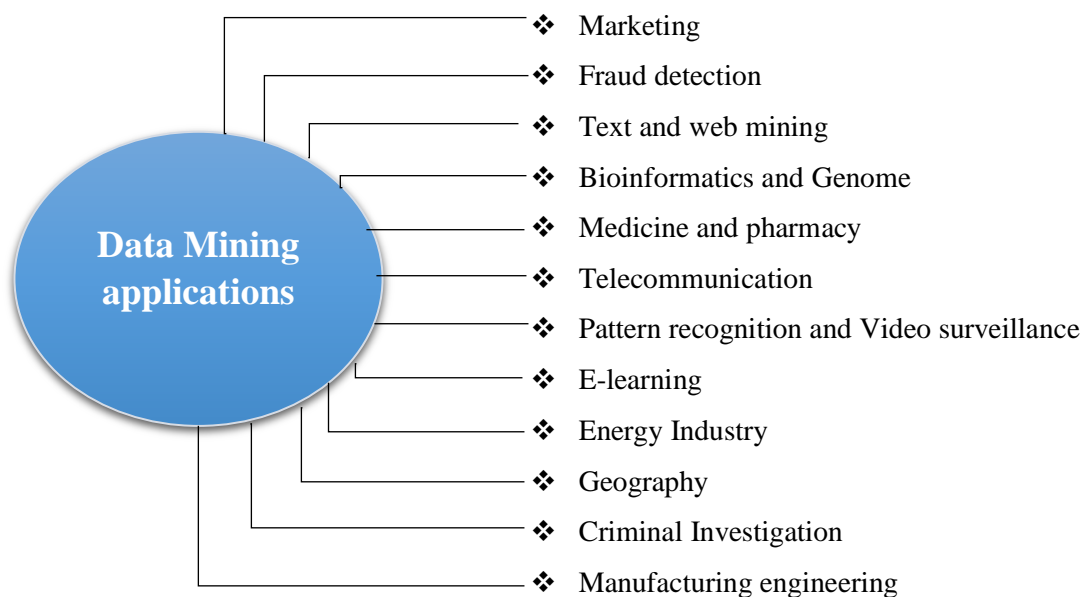


Figure 1.1: Data Mining applications

## 1.4 The objective of Data Mining

The main objective of Data Mining is to learn from data, which means to obtain useful information from a large amount of data [BIR04] [NHW<sup>+</sup>11].

Nowadays, and with the development of technology, the demand for the data industry is increased which led to increasing demands for data analysts and data scientists [Win21c]. Data Mining can help analysis to recognize significant facts, describe relationships, or discover patterns and trends that represent better the data and can identify exceptions and anomalies that might otherwise go unnoticed [Win21b].

With the help of Data Mining techniques, we can make a prediction, which makes it easier to take better decisions in a short time [Win21c].

Data mining helps to develop smart and scalable systems for the exploration of:

- Large and multidimensional database.

- Distributed data [Win21c].

With the Data Mining we can:

- Limit user intervention.
- Represent in a simple format the knowledge and visualize it in usable form.

## 1.5 Data Mining techniques

### 1.5.1 Principal Component Analysis (PCA)

Principal Component Analysis is one of the most popular techniques of factorial analysis. It uses to find a more faithful representation for the large data by projecting them in a reduced space called factorial space. Indeed, PCA transforms  $n$  variables of dimension  $p$  to uncorrelated variables called the principal components following a set of steps (see Figure 1.2).

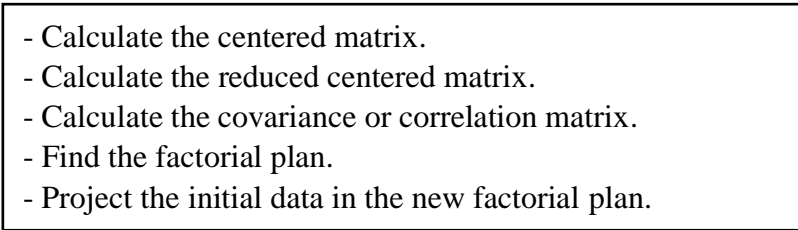
- 
- Calculate the centered matrix.
  - Calculate the reduced centered matrix.
  - Calculate the covariance or correlation matrix.
  - Find the factorial plan.
  - Project the initial data in the new factorial plan.

Figure 1.2: Principal Component Analysis steps

Some research has introduced modifications in the PCA to achieve a better representation of the data. Authors of [FMT<sup>+</sup>18] assumed that the linear relationship between variables caused a loss in the quantity of information. And in order to solve this problem, they introduced a non-linear function (kernel function (KPCA)) which allows searching the principal components in a dimensional space larger than the original space of the variables. The main idea consists to project the space of the input variables into a larger space called  $H$  via a non-linear function  $\Phi$ , then the linear PCA is applied in  $H$ . Another type of principal component analysis with a random projection of data is proposed by [Pal18]. This type replaced the correlation/covariance matrix in the old PCA with an asymmetric random square matrix of normal distribution. This proposition can be considered as a PCA with relaxed constraints depending on the correlation/covariance matrix. From the obtained results, which are similar to those of the classical PCA, we can deduce that the robustness of the PCA is not in the shape of the correlation and covariance matrix but in the symmetry. To achieve faster convergence of the PCA, authors of [PK17] introduced two statistical models based on Huber functions to replace the centered matrix in the classical PCA. These models are tested only on small samples with Gaussian distributions and

they did not exist research on the validity of this improvement for other distributions. In [LZZ20] the principal component analysis is improved using a logarithmic transformation method to reduce the size of the non-linear data. After the definition of the principal factors, the logistic regression model applied to study the relationship between the principal factors and the projected data. But this proposition is costly in the execution time. To obtain a good classification of risk scenarios, authors of [HM20] proposed an improvement in the data projection phase of the PCA (PCA-I). After the definition of the principal components, the new coordinates gave from the weighted average of the data projected in the new factorial plan to better adjust the risk scenarios. However, this modification does not allow to correctly classify the difficult accident scenarios, for this purpose authors have introduced the notion of fuzziness in the PCA-I. The fuzziness factor is used to define the membership level (more or less) of an accident scene to a class. But the fuzziness factor in this proposition is fixed manually after several empirical experiences.

## 1.5.2 Apriori Algorithm

The extraction of the relevant association rules is a major operation in huge databases. This operation requires an efficient process that filtered a large quantity of the data so that only interesting models and relations are saved. Works were done in [HPY00] show that the apriori algorithm is among the most popular algorithm used in the extraction of association rules.

Apriori algorithm was firstly proposed in 1994 by Agrawal and Sikrant [ISN16] for learning association rules to recognize frequent properties and predict their categories. The algorithm executes in three main steps, the first one consists to generate all the frequent itemsets, then the apriori algorithm deletes the itemsets that have a lower frequency than the minimum support, at the end a set of association rules is generated from the remaining itemsets. These three steps are repeated until the frequent itemsets are empty.

- Repeat these steps until the frequent itemsets is empty.
  - Generate a set of k frequent itemsets.
  - Delete itemsets that have a frequency lower than the minimum support (minsupp).
  - Generate association rules from the remaining itemsets (frequent).

Figure 1.3: Steps of apriori algorithm

During the execution, the length of the frequent itemsets generated at each iteration becomes long, and the number of accesses to the database increases. To overcome these problems, authors in [Sha16] proposed two solutions, the first solution used a bucket hash technique for dividing the database into two buckets (parts) containing frequent itemsets. Then the apriori algorithm is applied simultaneously on these two buckets. In the second

solution, the itemsets are divided at each iteration into two subsets, the first one contains the items whose frequency exceeds the minimum support  $S$ , while the other subset contains the items which have a frequency lower than  $S$ . Then at each iteration, the algorithm removes items that have a frequency lower than  $S$ . The produced itemsets compared with the database at the end of each iteration to determine the set of frequent itemsets.

This improvement needs more comparison operations at each iteration, which progress the complexity of the algorithm.

To reduce the number of accesses to the database Patil et al. [PDK16] used an intermediate database with dynamic size. At each level, transactions that contain  $k$  frequent items copied in the intermediate database. These  $k$  items are then deleted from the original database. During the execution, the apriori algorithm uses the intermediate database, which allows avoiding the analysis of the entire original database at each iteration and reducing the number of accesses to the database. It is really that in this proposition, the number of accesses to the database is reduced but the storage space has increased.

In the same context, Patel et al.[PRM13] introduced checkpoints to reduce the execution time and the storage space of the candidates during the database analysis. The basic idea is to define two checkpoints in the database. Then at each iteration, the algorithm estimated the support value for all items based on the checkpoints and the minimum support value to remove infrequent items. In this proposition, the execution time and the storage space are decreased, but the number of the estimated parameters is increased.

Sun [Sun20] proposed an improved apriori algorithm to reduce the number of database scans. This improvement used weighted items and weighted transactions to identify from them the most important in the database. Firstly, a matrix of 0-1 transactions is created by analyzing the transaction database. Based on this matrix, the algorithm assigned weights to items and to transactions. At this step, the weighted support of each item is calculated. This support uses with the minimum support to find the frequent itemsets.

Another improvement of the apriori algorithm based on a boolean matrix and index sorting rules has been proposed in [Du20]. After analyzing the database, a binary boolean matrix is obtained. This matrix has a pre-processing phase that removes unnecessary transactions and items. The remaining items are combined according to the index sorting rules for obtaining new items that can improve the frequent item extraction and reduce the storage space. It is true that these improvements have reduced storage space, but they have not reduced the number of memory accesses.

### 1.5.3 K-means

It is an unsupervised learning method proposed by MacQueen et al. [M<sup>+</sup>67] in 1967 as a solution to the clustering problem. This method aims the classification of  $n$  data through

$K$  clusters fixed a priori while minimizing a quadratic error function called the objective function (see the equation below).

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i - C_j\|^2 \quad (1.1)$$

Where

$\|x_i - C_j\|$  is the distance between a data point  $x_i$  and the cluster center  $c_j$ .

The main idea of the algorithm is very simple, it consists to randomly define  $k$  centroids, each one represents a cluster. Then the data points are assigned to the nearest centroids. At each iteration, the algorithm recalculates the new centroids of  $K$  clusters and it redistributes data points according to its new centroids. The execution ends when the centroids stop moving. The steps of k-means can be summarized in the following Figure: This method is easy and simple to implement, but the number of clusters and

- Choose randomly  $k$  points as the initial centroids of the clusters.
- Assign each data point to the nearest centroid.
- Recalculate the new centroids.
- Repeat the two previous steps until the centroids stop moving.

Figure 1.4: K-means steps

the choice of the initial centroids influences the results quality and the time convergence of the algorithm.

Several works proposed to develop efficient methods to make the selection of initial centroids, and the number of clusters in K-means automatic, among these works: Song et al. [SLL20] that proposed to initialize centroids by the mean of data, and they suggested clustering data by radius  $R$  rather than setting  $K$  clusters. In the beginning, the average of initial data is used as the first center value, then the distance ( $D$ ) between data and this center is calculated. If  $D$  is greater than  $R$  then a new cluster will be created. Otherwise, the algorithm contained to distribute data on the centroids. At this time, if the number of data in the cluster exceeds or equals maximum clustering constraint  $C_{max}$ , a new cluster will be created. On the other hand, if the number of data in the cluster is less than a minimum clustering constraint  $C_{min}$ , this cluster will be deleted and their data will be distributed to the nearest clusters. The proposed approach gives a better accuracy rate compared to the old version of K-means. However, it uses new parameters which need an estimation at the beginning (radius  $R$ , maximum clustering constraint, and minimum clustering constraint).

Hybridization between the k-means algorithm and the genetic algorithm has been proposed by Krishna and Murty [KM99] to correctly determine the initial centers of the clus-

ters of the k-means algorithm. In the same context, Murugesan and Murugesan [MM20] proposed to initialize centroid values with data that have the largest mean distance value. At first, a distance matrix for each pair of data is created. After that, the average distance by column is calculated. The values of this matrix are used to identify the first centroid. To select the other centers, the authors proposed to calculate the mean distance between each data point and the previous centroid. The data point that has a maximum distance will be selected as a new center. In [Hou19] authors used a hash function to calculate the initial values of the centroids. The hash function allows to map large data into a small space and to visualize data grouping relationships. After that, a search process on the  $k$  densest spaces proceeds. The average of each space represents a centroid value.

In order to achieve rapid convergence of the k-means algorithm Nanda et al. [NSC<sup>+</sup>17] used the central limit theorem to initialize the  $k$  centroid values and they estimated the number of the  $k$  clusters with the self-tuning algorithm. Another proposition of [MB12] used Kohonen self-organizing maps to visualize the number of clusters of k-means. Concerning the initialization of centroids, authors have proposed to use genetic algorithms. A new method for initializing k-means centroids based on Kruskal algorithm has been proposed by Zhang [Zha19]. This method represented data with a graph. The vertices are the data points and arcs are weighted according to the distance between two points. The first  $k$  vertices produced by the minimum cost covering tree of the Kruskal algorithm is used as the initial  $k$  centroids of the k-means algorithm. This algorithm has achieved good results, however, the calculation complexity is higher.

Authors of [ZZZ18] used the Canopy Density Algorithm as a pre-processing procedure of k-means. Canopy Density Algorithm is an unsupervised learning algorithm that used two distance thresholds to classify data to the nearest cluster. The number of clusters and centroids produced from this algorithm are used as input parameters for the k-means algorithm.

Generally, if we use other algorithms to initialize K-means parameters, we can produce good results, but the execution time will be increased.

#### 1.5.4 Fuzzy C-Means (FCM)

FCM was firstly proposed by Dunn [Dun73] in 1973 as a method for grouping data that belong to two or more groups (fuzzy membership). The algorithm aims to minimize an objective function of the form:

$$J(U, C) = \sum_{i=1}^n \sum_{j=1}^k (u_{ij})^m \|x_i - C_j\|^2 \quad (1.2)$$

With:  $\sum_{j=1}^k u_{ij} = 1, \forall i$  et  $u_{ij} \in [0, 1], \forall i, j$  et  $0 < \sum_{i=1}^n u_{ij} < n, \forall n$



such that:

$n$ : is the number of data.

$m$  : it's the fuzzy index  $\in [1, \infty[$  and it's usually set at 2.

$k$  : number of clusters.

$u_{ij}$  : is the membership degree of a data point  $x_i$  to the cluster  $C_j$ .

$\|x_i - C_j\|$ : is the distance between a data point  $x_i$  and the cluster center  $C_j$ .

$U = (u_{ij})_{n \times k}$ : represents the fuzzy membership matrix and  $C$  is the set of cluster centers.

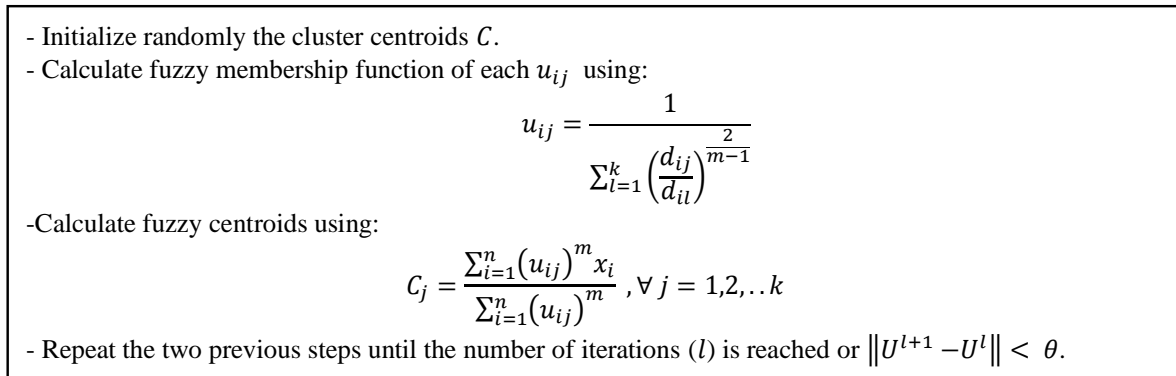


Figure 1.5: Fuzzy C-Means steps

Where:

$$d_{ij} = \|x_i - C_j\| \quad (1.3)$$

$l$  and  $\theta$  are respectively the number of iterations and the tolerance threshold, with  $\theta \in [0, 1]$ .

The initial value of parameters is one of the impact factors that influence algorithm performance. For example, in the fuzzy c-means algorithm initial values of the centroids can accelerate the convergence of the algorithm towards the global optimum or can make the algorithm skate in the local minimum. In addition, the quality of the results of this letter depends on the number of clusters chosen at the beginning.

Recently, several extensions of fuzzy c-means have been proposed to set their parameters and improve the quality of their results, from these extensions we find: Cardone et al. [CDM20] that proposed a weighted fuzzy c-means algorithm to fix the initial centroid values. In this algorithm, weights assigned to the data according to Shannon's fuzzy entropy function. The centroids obtained from weighted fuzzy c-means used as initial centers in the fuzzy c-means algorithm. In the same context, Particle Swarm Optimization (PSO) algorithm is used in [SJ19] to initialize the centroids of the fuzzy c-means clusters. However, the quality of the results of PSO depends on the initial values of their parameters. Yang et al. [YN17] proposed a robust fuzzy c-means without any a priori initialization of parameters and free of the fuzzy index. This version allows defining automatically the

optimal number of clusters. In the beginning, each data is considered as a cluster. Each cluster has a mixing proportion. The mixing proportion is calculated at each step of execution. The algorithm rejects clusters that have a mixing proportion lower than the inverse of the total number of data. At the end of each iteration, the number of clusters is updated, and members of the rejected cluster will be distributed at the nearest cluster. In this work, authors have substituted fuzzy index with the entropy in the membership function. However, this proposition is sensitive to the number of data, an increase in the number of data influence its performance.

Schwämmle and Jensen [SJ10] proposed a formula based on the number and the size of data for estimating the fuzzy index, and they used the minimum centroid distance to fix the number of clusters of fuzzy *c*-means algorithm. Dembele and Kastner [DK03] proposed an empirical method for estimating the fuzzy index of microstructure data. This method is based on the distribution of distances between genes in a data set. In addition, the authors have suggested estimating the number of clusters with the CLICK algorithm. However, the quality of the results of this proposition remains valid only for microstructure data. To improve the quality of the results of fuzzy *c*-means to noisy data, authors of [YCL19] proposed a fuzzy *c*-means with cross-entropy. Cross-entropy is used as a distance measure to calculate the membership degree of data.

In the same context, an hybridization between fuzzy *c*-means and Cohort Intelligence Optimization Algorithm (CIO) has been proposed by [KKS20] as a learning tool of FCM to increase the accuracy of the obtained results.

### 1.5.5 Hierarchical Classification

Hierarchical classification among the unsupervised techniques that aims either to partition a population of individuals into set of groups (clusters) or to merge individuals into homogeneous groups. From this principle, two types of hierarchies are distinguished: Descending and Ascending Hierarchical Classification.

In Descending Hierarchical Classification we start with a general solution (cluster) that contains all the observations, then at each iteration each cluster splits into two other clusters (children). Clusters separation is done so that the clusters are dissimilar as possible. This operation is carried out for each child until we obtain only one observation in each cluster.

For the Ascending Hierarchical Classification we start with  $n$  specific solutions each one is considered as a cluster, then at each step, we merge the two most similar clusters into one cluster, the process stops when all the observations (clusters) are grouped into one cluster.

The hierarchical classification produced a tree of successive partitions called a dendrogram, this tree must be cut at one moment to determine partitions of each cluster.

- Consider all individuals as one cluster.
- Split the cluster into two other clusters dissimilar as possible.
- Repeat the previous step until we obtain singletons that contain only one individual.

Figure 1.6: Descending hierarchical classification

- Consider each individual as a cluster.
- Calculate the similarity between all pairs of clusters.
- Merge the two most similar clusters into one cluster.
- Repeat the two previous steps until all individuals are merged in one cluster.

Figure 1.7: Ascending Hierarchical Classification

One of the problems that have resided when we use the hierarchical classification is how to automatically determine the correct cutoff level of the tree to obtain an optimal number of clusters. To resolve this problem authors of [ASS14] proposed to use the standard depth search (DFS) algorithm for defining the right dendrogram cutoff point. The algorithm traversed the dendrogram from the root to the leaves and at each time it compared the similarity of the root node with its children, if the root value exceeds the average of children, the current node is defined as a cutoff point otherwise the algorithm descends to the low level.

Authors of [ZXL16] proposed an automatic method for determining the optimal number of clusters in the Ascending Hierarchical Classification. This method creates  $k$  clusters at each iteration, these clusters are then analyzed via CSP index. CSP index calculated the average between the degree of compactness and the degree of the separation intra-cluster. The optimal number of clusters is the number of clusters that have the maximum value of the mean CSP.

Some research focused on the effect of distance on the hierarchical classification. Nasibov and Kandemir-Cavas [NKC11] integrated the ordered weighted average operator (OWA) as a measure of distance between clusters. And after creating each cluster, the root mean square (RMS), the standard deviation, and R-squared validity index (RS) are calculated to evaluate the homogeneity intra-and inter-cluster for obtaining good clustering results. However, OWA needs a robust method to regulate the weights. Fan [Fan19] proposed a hierarchical clustering algorithm based on a probabilistic optimal estimation (OPE). The main idea of this algorithm consists to use the probability of data distribution to create new clusters rather than using traditional distance. Fan [Fan19] considered that if two clusters have a high probability they belong to the same cluster. The algorithm starts with a sample of individuals that have a higher probability distribution, then at each

iteration, the most homogeneous individuals are combined together.

Traditional hierarchical algorithms are not extensible for the large database. If we use a large amount of data, the calculation cost became higher. In order to overcome this problem, Wang et al. [WWW20] proposed a new hierarchical method based on boundary point detection and the K-nearest neighbor algorithm. Firstly, the K-nearest neighbor algorithm is applied with a boundary point detection method to select a subset of small-scale original data, then traditional clustering algorithms are used to group this subset into clusters. At the end, the K-nearest neighbor algorithm is applied again to assign the remaining data points to the clusters created. This methodology is valid for large datasets, but the choice of  $K$  in the K-Nearest Neighbors algorithm influences the quality of the obtained clusters.

### 1.5.6 Artificial Neural Network (ANN)

Artificial neural networks are among the bio-inspired techniques that serve to reproduce the capacities of the human brain. In the neural system, the neuron receives electrical signals from its dendrites. If the sum of these signals exceeds an activation threshold, this neuron will transmit an electronic signal to the neighboring neurons.

From this principle, McCulloch and Pitts [MP43] proposed the first mathematical model of an artificial neuron. Indeed, each neuron is represented with a weighted sum of signals. The response is triggered when the sum of the signals exceeds a threshold. However, this model is defined without a learning rule. In 1949, Hebb [Heb49] proposed a learning rule which allows modifying the synaptic weights according to the activity of neighboring neurons.

Usually, a neural network is composed of a set of layers, each layer has a number of neurons that have connected with the neurons of the previous layer with weighted links, except the input layer. The output of each node is calculated according to an activation function that received the weighted sum of inputs nodes and bias. ANN model is initialized with random weights, then during the learning process, weights are modified until the model can generate outputs close to the desired outputs. Once the training is finished a classification model is created to classify new examples.

Results quality in ANN depends on its parameters initialization (weights, number of neurons, and number of hidden layers). Several works have been proposed for estimating the parameters of neural networks. Authors in [JWW<sup>+</sup>17] proposed a deep neural network model with weights regulation (rDNN). rDNN used the degree of correlation between the hidden layer and the following output layer as initial weights. In the same context, Yu et al. [YHY<sup>+</sup>10] proposed to estimate the initial weights of neural network layers using genetic algorithms. The learning step ( $\eta$ ) is a very important parameter in the neural network, it is defined as a positive constant that intervened to change weights. If the

learning step is initialized with a small value, the training process takes a lot of time. While a higher learning step can accelerate the training speed but the local optima may be exceeded. Yang and Xu [YX09] proposed a mathematical formula based on the initial learning step  $\eta_0$ , the learning error and a positive factor to change automatically the learning step at each iteration.

Recently, Back-Propagation (BP) neural network model has achieved considerable success in the development of expert systems due to its performance and adaptability. However, this model has some drawbacks such as getting trapped in local minima and premature convergence. To overcome these problems, Wang et al. [WLYT20] proposed a Back-Propagation neural network model (BP) based on HPSO and Cellular Automata (CA). HPSO is combined with CA to better balance local exploitation and global exploration and to prevent particles from getting trapped in the local optima. HPSO-CA then is used in the Back-Propagation neural network model to prevent the model from falling into local minima. Experimental results on 15 benchmark complex and real-world datasets show the performance of this algorithm, however, HPSO needing a lot of parameters to estimate for achieving a better solution.

Mirjalili [Mir15] has trained Multi-Layer Perceptron (MLP) using Gray Wolf Optimizer (GWO) algorithm. During the execution, GWO tries to find the optimal parameters of MLP (weights and bias) that increase the accuracy.

Essa et al. [EAEE20] used the Harris Hawks Optimizer (OHH) algorithm to estimate the initial weights of an artificial neural network. The algorithm takes as input a set of  $N$  networks then it provided the optimal set of weights. These weights are used to build a robust classification model. Authors in [LCAG20] introduced  $K$  Nearest Neighbors (KNN) classifier as the output layer of the deep neural network to better classify outputs. Generally, the complexity of neural networks increased exponentially and it is proportional to the size of the training data. The use of other algorithms to estimate ANN parameters will also increase the complexity and computation time of ANN.

### 1.5.7 Kohonen Self-Organized Maps (KSOM)

It is an unsupervised classification technique inspired by the human brain's neuronal. This method creates a network (map) of neurons connected (nodes) according to a topology. Each neuron in the map is connected to the input vector with weights. Weights are firstly initialized randomly, then adjusted during the training process. For each input, its activation is calculated with all neurons (nodes) of the map for determining the winning node. The activation is measured with the distance between neurons of the map and the input vector. Node, which has a low activation, is called the winning node. At each iteration an input is selected randomly, after determining the winner node, the weight of the winner node and the neighboring nodes is updated. This process is repeated until the

stopping criteria are reached [Koh82].

1-Initialize randomly the weight vectors for the neurons of the map.

2- Choose randomly an input vector from the data set.

3- Identify the winning node (neuron) using:

$$a(j) = \sqrt{\sum_{i=1}^n (e(i) - w_{ij})^2}$$

4- Determine the neighborhood of the winning node.

5- Update the weight of winning node and the neighboring nodes.

6- If the number of iterations is not completed, repeat the following steps: 2 -5.

Figure 1.8: Kohonen Self-Organized Maps steps

Such that:

$e$ : is the input vector.

$a(j)$  is the activation function for the neuron  $j$ .

$W_{ij}$  : is the weight vector.

Recently, many research has been modified KSOM and proposed to combine it with other intelligent techniques to increase its performance. Fernandes et al. [FMM<sup>+</sup>08] proposed a hybridization between KSOM algorithm and ant algorithm. In this proposition, data samples are considered as artificial ants that passed in a 2D toroidal grid and changed the environment by approximating similar vectors to create homogenous clusters. Furthermore, through the evaporation technique ants can select the next nearest cell in the topological map, which allows identifying automatically the topology of the neighborhood in the KSOM algorithm.

Authors of [AYIR17] proposed two modifications of KSOM algorithm: the first one consists to use the pheromone density (PDM) of the Ant Clustering Algorithm (ACA) as a measure of distance in the activation function. While the second modification has introduced two functions: picked up and dropped down of the ACA algorithm to adjust weights of KSOM algorithm. In the same context, [LNDA12] exploited the capabilities of genetic algorithms to initialize KSOM weights. Genetic algorithms selected from a population of random weights a set of weights that will produce a minimum error rate. But Genetic algorithms require a lot of time to converge.

Convergence time is among the drawbacks of KSOM. This problem is related to the learning rate since this latter controlled the volume of knowledge acquired by the network. In order to trait, this problem Galutira et al. [GFM19] changed the decreasing function of the learning rate with an exponential function which has an average variation rate.

This function increases the convergence of KSOM and improves their results. Authors of [FA20] proposed a Self-Organizing Map with Local Adaptive Receptive Field (LARSSON). In LARSSON the structure of the network changed dynamically. When new information is acquired, some nodes and connections are added to the network. Furthermore, the authors added a regulator to the weight update process for mitigating the interference with the acquired knowledge, and they used a learning rate varied with the number of winner nodes. This proposition gives a good precession rate compared to basic KSOM, however, it used many parameters which needed an estimation.

### 1.5.8 Artificial Immune Systems (AIS)

The natural immune system can protect itself from the strong elements. It can recognize self-cells from non-self-cells. Inspired by this intelligent mechanism, computer science has developed systems that can serve as a metaphor for some aspects of the natural immune system.

The first model of the immune network was proposed in 1974 by the biologist Jerne [Jer74]. This model represents the dynamic behavior of the immune system even in the absence of the antigen. Indeed, the immune network is built from the idio-type-paratope relationships between B cells-antigens and between antibodies of B cells [THH<sup>+</sup>08]. The interconnection of the cells makes the cognitive capacity of IS similar to that of single neural network [Nan09].

Negative selection algorithm [FPAC94] is another type of artificial immune system based on the principle of self-tolerance in the production of mature  $T$  cells. The main objective of this algorithm is to produce a set of T-cells that can relate only to non-self cells. The algorithm in general is defined in three main steps: the first one consists to define a set of self-cells, while in the second step we generate a set of detectors from the set of self-cells that can recognize the non-self cells. The final step represents a control or a test of the efficiency of the generated detectors.

Authors of [DCVZ02] proposed an algorithm of clonal selection which described the interaction of the immune system with antigens. The algorithm is based on the concept of only B cells which recognized the antigen are cloned and mutated to be introduced later as new memory cells. The algorithm starts with an antibody initialization process, then at each iteration it selected  $n$  antibodies ( $AC$ ) that have a high affinity with the input antigen. These  $AC$  are cloned and mutated to produce new cells. The best cells produced are selected as new memory cells ( $mc$ ) to replace the worst memory cells in the  $MC$  set. In the danger theory, the immune system interacted with the self-cell infected instead of non-self cells. The infected cells send an alarm signal, which allows the immune system to distinguish between dangerous and non-dangerous cells. Authors of [Mat02] exploited this idea to develop an algorithm that characterized this theory. The danger in this algorithm

measure following the gravity of the damage on the infected cells.

In the context of supervised learning, Watkins and Boggess [WB02] proposed an Artificial Immune Recognition System (AIRS). The principle of the algorithm inspires by the production process of memory cells. During the learning stage, AIRS used the antigen-antibody representation to build a robust classification model.

The relation between antigen-antibody is measured with the degree of similarity (stimulation) between the training data (antigens) and the classification models (memory cells set). The memory cell the most stimulated with the antigen is mutated to generate a set of new clones. The new clones are filtered through the process of resource competition to select a candidate memory cell ( $m_{C_{candidate}}$ ). The candidate memory cell has an opportunity to introduce later as a new memory cell. Authors of [WTB04] discussed the different steps of AIRS and proposed some modifications in some processes to increase data reduction capacity.

From the research that exploited the capacity of the immune system in the context of unsupervised learning we find: [TNH00, DCVZ00, TNH99].

### 1.5.9 Genetic Algorithms (GA)

This type of algorithm is inspired by the evolution of biological cells in the body. Genetic algorithms are first proposed by John Holland and his colleagues and students at the University of the Netherlands in 1975. These algorithms are developed as an approximate solution for optimization problems, which do have not an exact solution.

The main objective of genetic algorithms consists to search a set of representative models from a population. In the beginning, GA selected randomly an initial population from the data set. Then, this population is evaluated to select a sub-population closest to the optimal solution. After that, the crossover and mutation processes are applied to create new individuals. These steps are repeated for the new individuals from the evaluation until the optimal solution or the number of iteration is achieved.

- 1-Choose randomly a population from the initial data set.
- 2-Evaluate this population.
- 3-If the stop criterion is reached then stop, otherwise go to step 4.
- 4-Select a sub-population closest to the optimal solution.
- 5-Apply crossing and mutation operations to create new individuals.
- 6- Accept the new offspring and added it to the population, then go to the step 2.

Figure 1.9: Genetic Algorithms steps

The random nature of many processing steps of genetic algorithms decreases their performance and their accuracy. For this purpose, many extensions of GA are proposed in the



literature to enhance the quality of their results.

During the evaluation process, authors of [CW04] proposed to divide the population into two subpopulations: the first one is called the nuclei and it has a large size, while the second population has smaller individuals with rapid genetic evolution and its individuals called colonies. The nuclei traversed a search space in which the probability of obtaining an optimal solution is high. In the same time, colonies search in areas not searched from the nuclei then they sent migrants to the nuclei. The migrants are samples that can contain optimal solutions. Furthermore, this system has a mechanism to know the search areas of the nuclei to avoid that colonies search in the same areas.

To approximate the standard genetic algorithm to the biological basis, Jafari-Marandi and Smith [JMS17] proposed a fluid genetic algorithm (FGA). In the crossover process of FGA two random chromosomes are combined to create an initial new chromosome. Then a learning rate will be added or decreased from the value of this chromosome, which allows creating a final new chromosome. Furthermore, in FGA the mutation operation is canceled, only the crossing operation is applied. However, the mutation function can be produced more diverse individuals that may be an optimal solution.

Wang et al. [WLS20] choose the best neighboring chromosome to perform the crossover operation. At the first, a chromosome  $x_i$  is selected randomly, then the best neighbor chromosome of  $x_i$  is selected for the crossover operation. The neighborhood of a chromosome is supervised according to a radius  $R$  chosen by the user. Also, to improve the convergence of the algorithm and the search strategy of the optimal solution, the theory of the good point is used to generate the initial population rather than to choose them randomly. In addition to that, Wang et al. [WLS20] have integrated the chaotic search operator in the GA to avoid that the algorithm fell into the local optimum. These improvements have realized a considerable accuracy, but this proposition needed a robust method to estimate  $R$  and to change it according to the size of the population.

Datta and Dasgupta [DD20] modified the GA to increase their accuracy. This modification consists to apply the wheel selection operation before the crossover operation. Chromosomes that appeared less than 50% during the wheel run are allowed to perform the crossover operation. But, chromosomes that appear more than 50% times during the wheel run are passed directly to the mutation operation without performing the crossover operation. The wheel is turned 16 times in this proposition to produce good accuracy compared to the traditional GA. But this number is chosen empirically, and it does not exist a method to fix them automatically.

Hemanth and Anitha [HA19] introduced different binary logic operations in the process of crossover and mutation process. These binary operations are used and designed with specific constraints to generate offspring unlike the conventional binary operations of GA which are used randomly. The modified GA is used as a feature selection process which improved the accuracy of the obtained results from 4 to 6% compared to conventional

genetic algorithms. Here, the used constraints are validated only for the selected feature in images and they changed from one problem to another.

### 1.5.10 Particle Swarm Optimization (PSO)

Particle Swarm Optimizer or PSO is one of the most popular optimization algorithms. It was proposed firstly by Kennedy and Eberhart in 1995 [KE95]. The main objective of PSO is to search an optimal solution or the nearest solutions to the optimal solution in a well-defined space of D-dimension. PSO has  $N$  particles, each particle in the swarm has a position that represents a point in the space of search and a velocity.

Initially, position values are chosen randomly, but the velocities values are initialized with zero. Positions are evaluated using a fitness function. At each iteration and after adjusted velocities, particles change their position according to their best previous positions (local position) and the best global position of the swarm (see Equations 1.4 and 1.5).

$$x_{t+1}^i = x_t^i + v_{t+1}^i \quad (1.4)$$

$$v_{t+1}^i = wv_t^i + C_1r_1(p_t^i - x_t^i) + C_2r_2(G - x_t^i) \quad (1.5)$$

Where:

$w$ : is the inertia weight.

$C_1, C_2$ : are the acceleration coefficients. Called respectively cognitive and social parameters.  $C_1, C_2 \in [0, 2]$ .

$r_1, r_2$ : are two random numbers in the range  $[0, 1]$ .

```

While( $t < Max_{iter}$ ) do
  For each particle  $x^i$  do
    Calculate their fitness function ( $F(x^i)$ ).
    If ( $F(x^i) < F(p^i)$ ) then
       $p^i \leftarrow x^i$ 
    End if
    If ( $F(x^i) < F(G)$ ) then
       $G \leftarrow x^i$ 
    End if
    Adjust the velocity using equation (1.4) and update position with equation (1.3).
  End for
   $t \leftarrow t + 1$ 
End while

```

Figure 1.10: Particle Swarm Optimization algorithm

Various modifications have been proposed recently to enhance the PSO model, from these: Yue-lin and Yu-hong [YIYh07] that proposed to use an inertia weight that randomly

changes according to the simulated annealing method and they introduced an evolution strategy that depends on the variance of the fitness function of particles to improve the process of global research of PSO.

In the same context, authors of [LSGX09] have proposed two models that can operate in parallel to adjust the inertia weights of two sub-swarms of PSO. The first model is a non-linear mathematical equation, while the second model is defined by chaotic dynamic maps that are based on logistic equations. Each of these models is used under conditions depending on the value of the fitness function.

Some research work is interested to change the research topology structure of PSO. Wang and Feng [WF20] divided the swarm of particles into a set of sub-swarms. During the execution, the center of each sub-swarm is considered as an optimal solution. The position and the velocity of the particles are adjusted according to this center. This modification allows exploring a wide research space, but it requires an efficient process to control and detect collision between particles.

### 1.5.11 Grey Wolf Optimizer (GWO)

Grey wolf optimizer is one of the recent bio-inspired method [MML14], which addresses social leadership and hunting behavior of grey wolves in nature. In the swarm of wolves, there exist four types of wolves: Alpha is the chief of the swarm who makes the decision, Beta wolf help Alpha in decision-making, Delta is the wolf which submitted the information to Alphas and Betas. The last type is Omega the scapegoat which is responsible for hunting.

From this principle, authors of [MML14] divided also the initial population of wolves into four types (Alpha, Beta, Delta, and omega). The three first types of wolves are considered as the three best solutions that guided the other wolves (Omega) toward promising areas of the search space. During the optimization process, search agents try to change their position around Alpha, Beta, and Delta to find the optimal solution. The new position of search agents is re-adjusted as follow:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (1.6)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (1.7)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (1.8)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1(\vec{D}_\alpha) \quad (1.9)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2(\vec{D}_\beta) \quad (1.10)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3(\vec{D}_\delta) \quad (1.11)$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (1.12)$$

Where:

$$\vec{A} = 2\vec{a} \cdot r_1 - \vec{a} \quad (1.13)$$

$$\vec{C} = 2r_2 \quad (1.14)$$

$r_1, r_2$  are two random vectors in  $[0, 1]$ , and  $\vec{a}$  is linearly decreased with the iteration steps from 2 to 0.

From the Equations 1.13, 1.14, we can observed that  $\vec{A}$  and  $\vec{C}$  are two vectors generated randomly. These vectors are used for exploitation and exploration of the research space. When  $|A|$  is between -1 and 1 exploration occurs. The vector  $C$  promotes an exploration if it is greater than 1. Otherwise exploitation is reached.

```

Initialize the population of grey wolf optimizer
Initialize a, A and C
Calculate the fitness function of all wolves.
Choose the three first best solutions as Alpha, Beta and Delta.
While (t < Maxiter) do
    For each search agent do
        Update the position of the current search agent using the equations (1.11).
    End For
    Update a, A and C
    Calculate the fitness function of all search agents
    Update Alpha, beta and Delta
    t ← t + 1
End while
Return alpha position as the best solution

```

Figure 1.11: Grey Wolf Optimizer algorithm

To enhance the global exploration of the grey wolf optimizer, Long et al. [LCJ<sup>+</sup>20] proposed a hybridization of grey wolf optimizer and cuckoo search (GWOCS). This co-operation can achieve an efficient balance between the exploitation and the exploration process. Also, authors of [LCJ<sup>+</sup>20] used Opposition-Based Learning (OBL) strategy as a process for deciding the layer individuals ( $\alpha, \beta$  and  $\delta$ ). The introduction of OBL aims to improve the diversity of the GWO population.

Saremi et al. [SMM15] proposed to use Evolutionary Population Dynamics (EPD) in the Grey Wolf Optimizer. EPD removes at each iteration half of the worst search agent and reinitialize their position around the three best solutions. Therefore, this modification is interested to improve the position of the worst wolves for finding the best solution.

Authors of [ZLL20] suggested an improved grey wolf optimizer algorithm based on the introduction of covariance matrix adaptation-evolution strategy (CMA-ES).

This algorithm is executed in two stages to improve the convergence speed of the GWO. In the first stage, GWO is executed as a global search tool to supervise the locations of the optimal solution region. Alpha, Beta, and Delta wolves obtained in the previous stage are used in the second stage as starting points for the three CMA-ES instances. CMA-ES instances are generated to perform parallel local exploitation.

### 1.5.12 Ant colonies

During the foraging process, ants traced an odor trail with the pheromone to mark their trajectory. Ants may know the minimal trajectory to the food without any a priori overview on the road. Depending on the quantity of the pheromone dropped, ants can supervise the minimal trajectory to the nest. Ants can also mark their trajectory in the back and forth, which increased the evaporation of the trajectory with the pheromone. Researchers have exploited this principle to solve many combinatorial optimization problems. Dorigo et al. [DG96] proposed an Ant System to solve the traveling salesman problem (TSP). TSP aims to find the minimal trajectory closed for visiting each city. During the execution, a starting city is chosen for each ant, and then a state transition rule is applied to supervise the next city visited. Once ants finished their tour, the length tour for each ant is calculated. After that, the pheromone value of the construction trails is updated. This stage is considered as local update process.

```

While ( $t < Max_{iter}$ ) do
  For each ant  $k$  do
    Choose a starting city
    For each city  $i$  not visited do
      Choose a city  $j$  from the list of remaining cities  $J_i^k$  using

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il})^\alpha (\eta_{il})^\beta} & \text{if } j \in J_i^k \\ 0 & \text{otherwise} \end{cases}$$

      End for
    Calculate the length  $L_k$  of the tour generated by ant  $k$ 
  Update the pheromone value in all the trail levels  $\tau_{ij}$  using

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}$$

  Where:  $\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$ 
  And:  $\Delta \tau_{ij}^k \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ travels on the edge } (i, j) \\ 0 & \text{otherwise} \end{cases}$ 

   $t \leftarrow t + 1$ 
End while

```

Figure 1.12: Ant system algorithm

Where:

$\tau_{ij}$ : the intensity of pheromone trail between cities  $i$  and  $j$ .

$\eta_{ij}$ : is the visibility of city  $j$  from city  $i$ .

$\alpha$ : parameter to regulate the influence of  $\tau_{ij}$ .

$\beta$ : parameter to regulate the influence of  $\eta_{ij}$ .

$\rho$ : is a parameter between 0 and 1 for regulating the reduction of  $\tau_{ij}$ .

$\Delta\tau_{ij}$ : allows to increase of trail level on edge  $(i, j)$  caused by the ant  $k$ .

$m$ : number of ants.

$Q$ : quantity of pheromone laid by an ant.

$L_k$ :tour length of the ant  $k$ .

Authors of [DG97] kept the same principle of [DG96] and they proposed to update the pheromone value of the best solution (shorter tour). This modification can consider as a global update process.

Stützle and Hoos [SH00] proposed two modifications in the Ant System. The first one is to allow only the ants, which have achieved the shorter tours to change the pheromone value. While the second, consists in limiting the pheromone value on an interval  $[\tau_{min}, \tau_{max}]$  and to initialize it with  $\tau_{max}$ . These modifications allow to achieve strong exploitation of the search space and to avoid premature convergence of the search.

Bullnheimer et al. [BHS97] proposed a Rank-based Ant System (ASRank) which introduced an ant ranking notion in the pheromone update process. After all, ants finished their tour, they are sorted in descending order according to their tour length. These ranks are used as a weight for ants. Pheromone values of the  $\sigma - 1$  best trails are updated according to the rank of ants, but the trail of the best solution benefit from an additional quantity of pheromone. Indeed, in ASRank only the best ants can update the pheromone values dropped in the trails.

In 1999 authors of [DDC99] defined a general standard of the Ant System called the Ant Colony Optimization algorithm (ACO). ACO is defined in three main functions AntSolutionsConstruct, PheromoneUpdate, and DeamonActions. The first two functions are respectively responsible for trail construction, updating pheromone, and evaporation. DeamonActions is a function, which allows adjusting the phenomenon towards the optimal trail.

### 1.5.13 Bee colonies

Bee colonies include a set of algorithms inspired by the behavior of bees. The basic idea incorporates in these algorithms consists in creating a colony of artificial bees capable to solve difficult optimization problems successfully. In the literature, three types of bees are distinguished: scout bees, which flew randomly in a search space without orientation, employee bees (EBs), which are placed firstly in a random position, and then they exploited

the neighborhood to find the best position. The last type is the onlooker bees (OBs), this type used the solutions obtained by the population in order to choose the better starting solution, and then it exploited the neighborhood to search an optimal solution.

From this principle, several models of bee colonies have been proposed. The artificial bee colony algorithm (ABC) is the first model proposed by Karaboga [Kar05] in 2005 as a tool for solving multidimensional and multimodal optimization problems. Initially, the employed bees (EB) are randomly placed in the search space in order to find feeding areas with high nectar quality (the optimal solution). At each iteration, the EBs try to improve their position according to a local search step. If an EB has not been able to improve their position within a period of time, she will abandon this position. After all, EBs change their positions. Each OB chooses a position of the EB by the standard wheel selection and it attempts to improve its position following a local search step.

```

Place each employed bee on a random position in the search space
While stopping criterion not met do
  For all EBs do
    If steps on the same position = limit then
      Choose random position in search space
    Else
      Try to improve position using
           $\theta_i^* = \theta_i + rand. (\theta_i - \theta_k)$ 
      If better position found then
        Change position
        Reset steps on same position
      End if
    End if
  End for
  For all OBs do
    Choose position of employed bee using
           $P_i = \frac{F(\theta_i)}{\sum_{k=1}^{n_e} F(\theta_k)}$ 
    Try to improve position of employed bee using
           $\theta_i^* = \theta_i + rand. (\theta_i - \theta_k)$ 
  End for
End while

```

Figure 1.13: Artificial bee colony algorithm

Where:

*rand*: is a random number between -1 and 1.

$\theta_k$ : is the position of another randomly chosen *EB* with index *k*.

$\theta_i$ : is the position of the current bee.

$n_e = \frac{n}{2}$  and *n* is the number of bees.

In order to improve the convergence rate and to avoid that ABC stuck into local optima, Famila et al. [FJSS19] proposed to enhance the ABC algorithm with the merits of Grenade Explosion Method (GEM) and Cauchy Operator. GEM and Cauchy operator are integrated into onlooker bee and scout bee phases to facilitate the dynamic moving from one region to another, which improve exploitation and exploration process.

In the same context, authors of [ZPH<sup>+</sup>19] used a statistical learning mechanism of the search path to improve the exploitive strategy and to accelerate the convergence of ABC. However, all these propositions are used only for solving a single objective problem. Bee Colony Optimization [TD05] is another method based on two alternating stages: forward pass and backward pass. The forward pass allows to create and improve solutions, while the backward pass allows bees to share and exchange information on the solutions found.

### 1.5.14 Discriminant Analysis (DA)

Discriminant Analysis is one of the statistical models designed for supervised learning. The aim of discriminant analysis is to build a model that will give a better prediction of data. Indeed DA predicts one or multiple qualitative variables with one or more quantitative variables. Classification by discriminant analysis consists of using the labeled information to learn a discriminant projection so that the homogeneity intraclass be large and the homogeneity inter classes be small.

According to the relationship between variables, two types of discriminant analysis are distinguished: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). In the first type, variables follow a multi-normal distribution with the same covariance matrixes in each group. Point clouds of this type have the same shape in the representation space. While in the second type, variables density follows a multi-normal distribution with different covariance matrices [EKC<sup>+</sup>17].

Recently, several improved Linear Discriminant Analysis (LDA) dedicated to data size reduction are proposed. Su et al. [SQH<sup>+</sup>18] suggested a linear discriminate analysis with a regularized projection matrix (R-MDA). During the execution, projection vectors are reshaped as projection matrices that are stacked together according to well-defined constraints. The projection of matrix rather than vectors allows LDA to study the correlation between rows and columns of projected matrices, which is crucial for the extraction of relevant information to create a robust data reduction model. However, the matrix projection operation as vectors is more complicated and during the execution, it requires a lot of memory space and a large computing time especially in large databases.

In the same context, Wen et al. [WFC<sup>+</sup>18] proposed a robust sparse linear discriminant analysis (RSLDA) to solve the sensitivity of LDA into the number of projection factors, the noisy data and to ensure correct interpretation of the features. RSLDA used the  $l_{2,1}$  norm to construct the discriminant projection matrix. A sparse matrix and an orthogonal



matrix are also introduced simultaneously in the RSLDA model to build a noise-resistant model and to allow the transformed data to preserve the main discriminating information. Using the maximum likelihood estimates in LDA and QDA for samples of comparable size produces a high error rate. To overcome this problem, authors of [Fri89] proposed a Regularized Discriminant Analysis model. This model replaced the covariance matrices of the samples by their average when the number of individuals is greater than their dimension. In the case where the size of samples is smaller than their dimension, the covariance matrix is shrined towards the identity matrix multiplied by its average eigenvalue. These improvements have increased the power of the discriminant analysis when the samples size is smaller than their dimension, but these improvements are not valid in the QDA if the samples size is greater than their dimension. In the LDA, classes have similar centers. However, the short distance between class centers makes the separation between them difficult. Yu et al. [YGW<sup>+</sup>20] proposed an improved linear discriminant analysis that adopts a decreasing function of the distances between any two different classes. This function added weights to the classes. Edge classes have small weights, while non-edge classes have large weights. The suggested improvement reduced effectively the dominant influence caused by edge classes on the between-class scatter matrix, which allows a good separation of classes. Zheng et al. [ZHWC20] used a spatial similarity measure of large data as weights in the between-class scatter matrix. The integration of the similarity measure function ensured a better spatial separation of the data and provided an optimal transformation matrix for the data dimension reduction process.

### 1.5.15 Regression Analysis

Regression analysis is one of the statistical methods used to study relationships between data to create a prediction model. Regression based on the idea of determining a dependent variable by one or more independent variables. It assumed that there is a causal relationship between these two types of variables, indeed independent variable value affects dependent variable value. For example, to see the effect of speed on road accidents, regression analysis is used to show the relationship between speed and the number of accidents. To create a regression model, we distinguish four main steps (see Figure 1.15).

- 1- Prepare the data determine variables and the type of the relationship used.
- 2- Search a regression model that reflects the relationship between variables.
- 3- Once the regression model is developed statistical tests of validation are applied.
- 4- If the model is adequately described, it can be used to predict the new variables.

Figure 1.14: Regression Analysis steps

According to the nature of relationships between variables, four types of regression are

distinguished: simple regression, multiple regression, linear regression, and non-linear regression. The first type studies the relationship between one dependent variable and several independent variables, while the second type expresses the relationship between several depend variables and one or more independent variables. In linear regression, it exists a linear relationship between the independent variables and the dependent variables, but in the non-linear regression type, there is no linear relationship between the independent variables and the dependent variables.

The presence of outliers in the linear regression decreased the accuracy of this latter since the support of the outliers is not known beforehand. In order to solve this problem, Liu et al. [LCR18] added a regulator of  $l_0$  norm to the regression function for modeling better the outliers. In the same context, authors of [SCJ17] proposed a greedy method to estimate the regression vector from linear outliers. This method has achieved a good accuracy rate, but it required that the sampling matrix satisfied certain conditions and the noise be rare.

To build a robust regression model, She and Owen [SO11] imposed a non-convex scarcity criterion on the mean shift parameter estimator of the regression model. The achieved results are better when the number of outliers is high, but this method is sensitive to noisy data.

Gorji and Aminghafari [GA19] proposed a robust nonparametric regression method. This method extracts the regression function (linear or non-linear) from the decomposition of a trajectory matrix. First, inputs are transformed into a trajectory matrix. Then the linear regression model of Kurz-Kim et al. [KKRS04] is used to decompose the trajectory matrix into triples pseudo-singular. With this pseudo-singular, the regression function (linear or non-linear) is estimated. The results obtained by simulation and on real examples show the efficiency of this method, but it needs to estimate the regression function for each new observation, which is costly in terms of time execution and memory space.

Authors of [CLL20] used the concept of shrinkage to define robust locations and dispersions estimators of linear regression parameters. After the elaboration of the regression model, the authors used a weighted Mahalanobis distance for classifying the data to the right class.

### 1.5.16 Naïve Bayes (NB)

Naïve Bayes is a probabilistic supervised learning method that used a Bayes rule with a simplifying hypothesis called naïve. This hypothesis assumed that there is conditional independence between all pairs of variables. Indeed, NB assumed that the existence of a characteristic for a class is strongly independent of the existence of other characteristics. For each input, a priori probabilities are calculated. These probabilities are used from the Bayes formula to determine the posterior probabilities.

The prediction of the membership class is done with the posterior maximum rule (MAP) [DP97].

$$MAP = \operatorname{argmax}_{c_k \in C} (P(c_k/x_{new})) \quad (1.15)$$

Where:

$C$ : is the existing classes and  $c_k$  is the belonging class.

Recently, some modifications have been introduced on the Naïve Bayes model. Borgelt and Kruse [BK02] proposed a Bayes formula valid for both continuous and discrete variables. Indeed, continuous and discrete variables are treated by the same Bayes formula.

In the same context, Geng et al. [GMB<sup>+</sup>19] proposed a bayesian classifier with free model. This classifier does not require independence between attributes, also it treated the continuous and discrete variables using the same Byes formula without assuming their a priori probability distributions.

However, these models are sensitive to the missing attribute values.

Despite the great success of Naïve Bayes in various fields, this classifier requires many labeled data from each class to train it well. To overcome this problem Chen et al. [CDD<sup>+</sup>19] introduced the correlation between classes in the NB classifier. The correlation allows giving more information about data and classes, which increases the accuracy of NB even if small learning data are used.

To improve the precision rate of NB, authors of [JHL<sup>+</sup>19] proposed to estimate the conditional probabilities of NB by the kernel density estimator (KDE). Sarker et al. [SKCH17] estimated the conditional probabilities of NB with the Laplacian estimator to decrease false detections produced from the missing attribute values.

All the proposed improvements which are cited above are valid in well-defined constraints.

### 1.5.17 Support Vector Machine (SVM)

SVM is one of the supervised learning algorithms dedicated to binary classification problems. It was first proposed with Vladimir Vapnik [Vap63] in 1963. SVM searches from the learning set the hyperplane that has the maximum margin.

In the literature several problems cannot be separated with a hyperplane, in this case, SVM transforms the input data space into a larger dimensional space according to a kernel function, to find a linear separator.

In the Support Vector Machines number of support, vectors are proportional to the complexity and the number of the training data. Authors in [PP18] used the minimum spanning tree to reduce the number of support vectors of the SVM. The global idea of this proposition consists in applying the minimum spanning tree on the training set to identify the boundary points and to remove the data points that are located further from the boundary points.

This model determines well the optimal support vectors of SVM. But, the calculation complexity of this model is higher.

Yao et al. [YLY<sup>+</sup>13] proposed a hybridization between the k-means algorithm and the SVM. k-means algorithm is used to identify data points that are located at the cluster boundary. Using these data points an SVM model is established. However, k-means needs a robust parameters estimation process to obtain clusters well separated.

Another proposition to improve the performance of SVM with Ordered Weighted Average Operator (OWA) was presented from [MMM18]. Indeed, the loss function in the conventional SVM is reformulated as a weighted sum using the OWA operator. This proposition is executed into two main steps: in the first step, conventional SVM is applied to sort training data according to their distance from the hyperplane, after that the SVM is trained again using the OWA operator to build an efficient SVM model.

However, this proposition consumes a lot of time since it will train the SVM twice.

Hybridization between the genetic algorithms (GA) and the particle swarm optimization (PSO) algorithm has been combined with the SVM in [WJM<sup>+</sup>20]. This method exploited the sharing mechanism and the speedy convergence of PSO algorithm and the capability of the genetic algorithms for optimizing the support vectors and the kernel function of SVM. However, the quality of the results of PSO and GA depend on the method used for estimating their parameters.

In the same context, Xu et al. [XRC<sup>+</sup>20] applied partial least squares regression on the training data for optimizing SVM support vectors.

### 1.5.18 K Nearest Neighbors (KNN)

$K$  Nearest Neighbors or KNN is a supervised classification method that aims to classify a target point (to an unknown class) based on a similarity measure (generally a distance function).

The target point is labeled by the majority class of the  $K$  nearest neighbors [CH67].

1- Define the number of the  $k$  nearest neighbors.

2- Calculate the distance between the new data and all the classification models using the Euclidean distance:

$$D(x, y) = \sqrt{\sum_{i=1}^{|x|} (x_i - y_i)^2}$$

3- Sort the classification models according to their distance from the smallest to the largest.

4- Select from the sorted classification models the  $k$  closest neighbors.

5-  $x_c = \operatorname{argmax}_{c \in C} (y_{j_c}), j = 1, 2, \dots, k$

Figure 1.15: K Nearest Neighbors steps

In the literature, several extensions have been proposed to improve the KNN model. Among the most critical parameters in the KNN model is the number of nearest neighbors ( $K$ ). Authors in [ET17] used an adaptive dependence region (DR) limited with a radius  $r$  to define the nearest neighbors. All individuals located in this region are the closest neighbors. Using an adaptive dependence region we can increase the accuracy of KNN, but the radius  $r$  requires an efficient method to estimate.

Nam Bui et al. [BJN<sup>+</sup>19] adopted the Particle Swarm Optimizer (PSO) algorithm to identify the optimal value of  $K$  and to choose the best similarity measure from a set of distances ( $d$ ). PSO particles performed a global search for the best parameter values ( $K$  and  $d$ ) that create a robust KNN model to the false detection. However, PSO needs an estimation of its parameters to achieve good results quality.

Zeng et al. [ZYZ09] changed the majority voting rule with the Nearest Pseudo-Neighbor (PNN). PNN assigns new data to the class of nearest neighbors that have the smallest sum of the weighted distances. Concerning the parameter  $K$  authors have estimated by the cross-validation method which is cited in [MH06]. PNN achieved better results than the traditional K-Nearest Neighbor in the large training sample size. But in the small training sample size, the accuracy of PNN is not good.

Taneja et al. [TGAJ15] proposed an improved fuzzy KNN algorithm using clustering technique to improve precision quality and membership speed of KNN. This algorithm is defined in two main stages: the first one is considered as a pre-processing step and it consists in identifying the fuzzy areas with the FCM algorithm. The fuzzy membership matrix produced by FCM is used from the fuzzy KNN in the second stage to classify new data. In this proposition, the fuzzy index and the initial centroids must be estimated by a robust method to obtain good results.

In order to increase the accuracy rate of KNN Li et al. [LLZX18] proposed an entropy weight local-hyperplane K-Nearest-Neighbor (EWHK). This algorithm used the entropy information of samples to calculate the weight distance between new data and the classification models for identifying the  $K$  nearest neighbors. The  $K$  nearest neighbors are used to create a set of hyperplanes. The new data is labeled with the class of the nearest hyperplane. EWHK is tested only on the breast cancer dataset. Further tests based on other datasets are desired to demonstrate the effectiveness of this proposition.

### 1.5.19 Adaptative Boosting (Adaboost )

Adaptative Boosting also called AdaBoost was proposed by Freund and Robert [FS95] as a supervised learning algorithm that used the principle of boosting to improve the performance of any learning algorithm.

The fundamental principle of Adaboost consists in training a set of samples using well-defined algorithms. Initially, examples are weighted with identical weight, then at each

iteration, a sample of examples is chosen to train it with an algorithm ( $A$ ). After that, a weighted process is applied to change the weights of sample individuals that are trained. Misclassified individuals have high weights compared to the individuals which are well classified. At the end of each iteration, Adaboost generated a weighted classification rule. These rules are used later for classifying new examples.

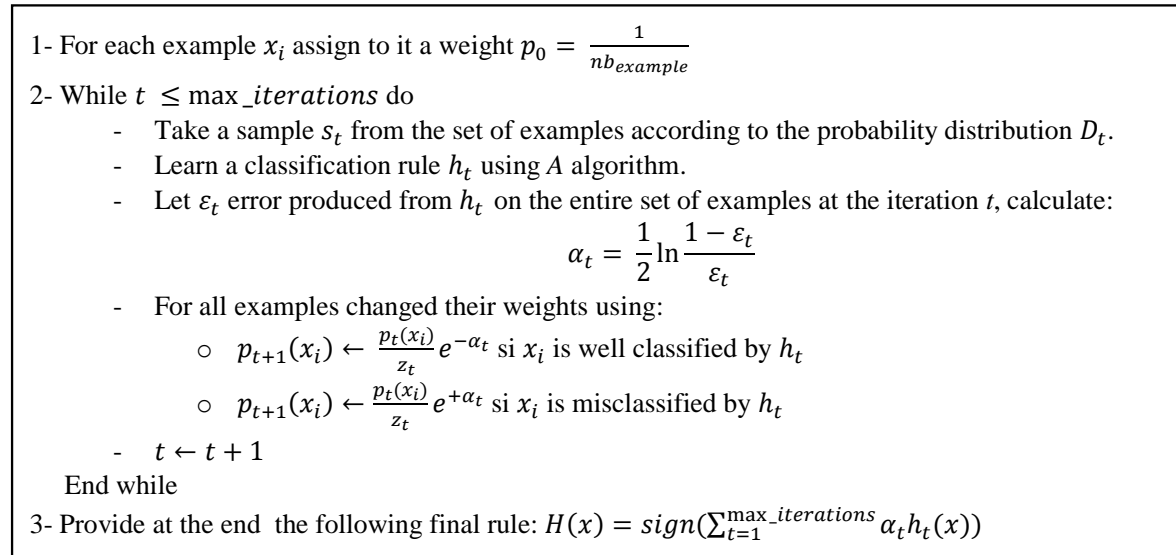


Figure 1.16: Adaptive Boosting algorithm

Such that  $z_t$  is a normalization value used to ensure that the sum of the weights always remains equal to 1.

In the conventional AdaBoost algorithm, weak classifiers are not modified once they are formed. ISABOOST [QTYH13] is a new version of AdaBoost that has a mechanism for adjusting the internal structure of the weak classifiers formed before assigning weights. This algorithm gives another opportunity for the weak classifiers to be stronger at the classification of hard examples.

Li and Zhang [LZ04] used a backward mechanism at each iteration of AdaBoost learning to minimize the error rate. Furthermore, the authors proposed a stepwise approximation of posterior probability as a new statistical model for forming weak classifiers. These modifications provided lower error rates during the learning and the test stages compared to the conventional AdaBoost.

AdaBoost.RT is an AdaBoost that has a relative error threshold to limit correct and incorrect predictions. In the majority of works, the error threshold is fixed manually by the user. Tian et al. [TM09] proposed an auto-adaptive method that automatically changed the error threshold value  $\phi$ . Initially,  $\phi$  is fixed randomly between zero and one, then at each iteration, it is modified automatically according to the error rate value achieved during the execution.

In the learning phase, AdaBoost increased the weights of misclassified individuals. These

individuals can be misclassified during several iterations, which caused an imbalanced weight of individuals. To overcome this problem, Tang et al. [TTD<sup>+</sup>20] proposed a modification in the weight update process of AdaBoost. This modification consists in updating update the weights of the current iteration than adjusting them according to the weights of the previous iteration. With this mechanism, the difference between weights of iterations  $t - 1$  and  $t$  is reduced which allows attenuating the imbalance in the weight distribution of individuals.

In the same context, authors of [WW20] proposed two strategies for adjusting individual weights. The first strategy introduced a positive operator in the weight update function of sample individuals that have been well classified in the previous iteration, but they are misclassified in the current iteration. While the second strategy, updated weights of individuals that have misclassified in the iterations  $t - 1$  and  $t$  according to the error rate and the error number threshold.

Li et al. [LZZ<sup>+</sup>19] proposed a new extension of AdaBoost based on the area under curve index (AdaBoost-A) to improve the error calculation process. AdaBoost-A introduced the index of the area under the curve into the formula that calculated the error of the weak classifier. This improvement reduced the weight of the duplicate examples to avoid wasting the capacity and execution time of AdaBoost.

However, all the proposed extensions are tested for an AdaBoost that contains special classifiers. Indeed, all these propositions are valid for specific algorithms.

### 1.5.20 Decision Tree

The decision tree is a supervised classification method that allows structuring a set of choices in a graphic format (tree). The decision tree is composed of a set of nodes called attributes connected with branches. Branches have a set of values for predictive variables. At the end of the tree, there are leaves, which give the possible decisions.

Among the most popular decision tree models that are based on the Precise Probability Models, ID3 [Qui86]. This model created a decision tree from a set of given attributes using an iterative process. At each iteration, ID3 searches the relevant attribute using Shannon's entropy then places it as a parent node in the tree. The attribute, which has a higher entropy gain, is chosen as a parent node. Despite the simplicity of ID3, but this model treats only the discrete variables and is sensitive to missing values.

In order to overcome problems of ID3, [Qui93] proposed C4.5 model. C4.5 placed the attribute which has a maximum ratio of information gain to information loss as a parent node rather than taking the attribute which has a maximum information gain as indicated in ID3. C4.5 has been also extended to be able for treating continuous variables and missing values. Despite, C4.5 has solved the problems of ID3, but it suffers from other problems that lie in the speed and memory space consumption.

C5.0 [Doe18] has the same principle as C4.5 but it contained some mechanisms that offer more speed, more precision, and consume less memory space. Wang and Kong [WK19] proposed an improvement of the C4.5 decision tree to increase its accuracy and to reduce its computational complexity. Before the entropy calculation, the authors performed a data discretization. After that, the attribute which has a high rate of weighted information gain is selected as a partition node. In this improvement, initial data are weighted to distinguish the attributes that have a great gain of information. Furthermore, Shannon's entropy has benefited from a small improvement, the log operation is substituted with Taylor Expansion for reducing calculation complexity.

Breiman [Bre01] used boosting techniques and a decision tree structure to construct a robust model called Random Forest (RF). During the execution, many binary decision tree models are created from a set of samples randomly generated by bootstrap. These models are filtered according to a majority voting method to select the best one that used in the classification. The classification model needs the construction of combination rules for well classifying new examples, which is generally complicated.

In 1996 it appeared an imprecise probability model (Imprecise Dirichlet Model (IDM)) [Wal96] which consists in constructing probability intervals for variable values instead of assigning them a specific probability. According to this principle, [AM03] Abellán and Moral used a total imprecise uncertainty measure on convex sets of probability distributions to create a credits decision tree (CDT). Xue et al. [XWC<sup>+</sup>19] proposed a fuzzy ID3 model (FDT) that used the fuzzy entropy to calculate the information gain of the decision attributes. The attribute which has the highest gain of fuzzy information is selected as a decision node, its branches are configured according to the values that can take this attribute. Once the FDT tree is constructed, the prediction of new examples is done according to a fuzzy matching method.

In the same context, Li et al. [LXD19] used Deng's entropy as a measure for splitting attributes to find the best decision attributes. These attributes are used to build a decision tree for classifying fuzzy data. The produced decision tree is applied directly as a classification tool rather than constructing combination rules, which reduces the complexity of the algorithm.

## 1.6 Proposed taxonomy

According to the in-depth study that we have performed, we can divide the techniques of Data Mining into three main categories: descriptive techniques, bio-inspired techniques, and predictive techniques (See Figure 1.17).



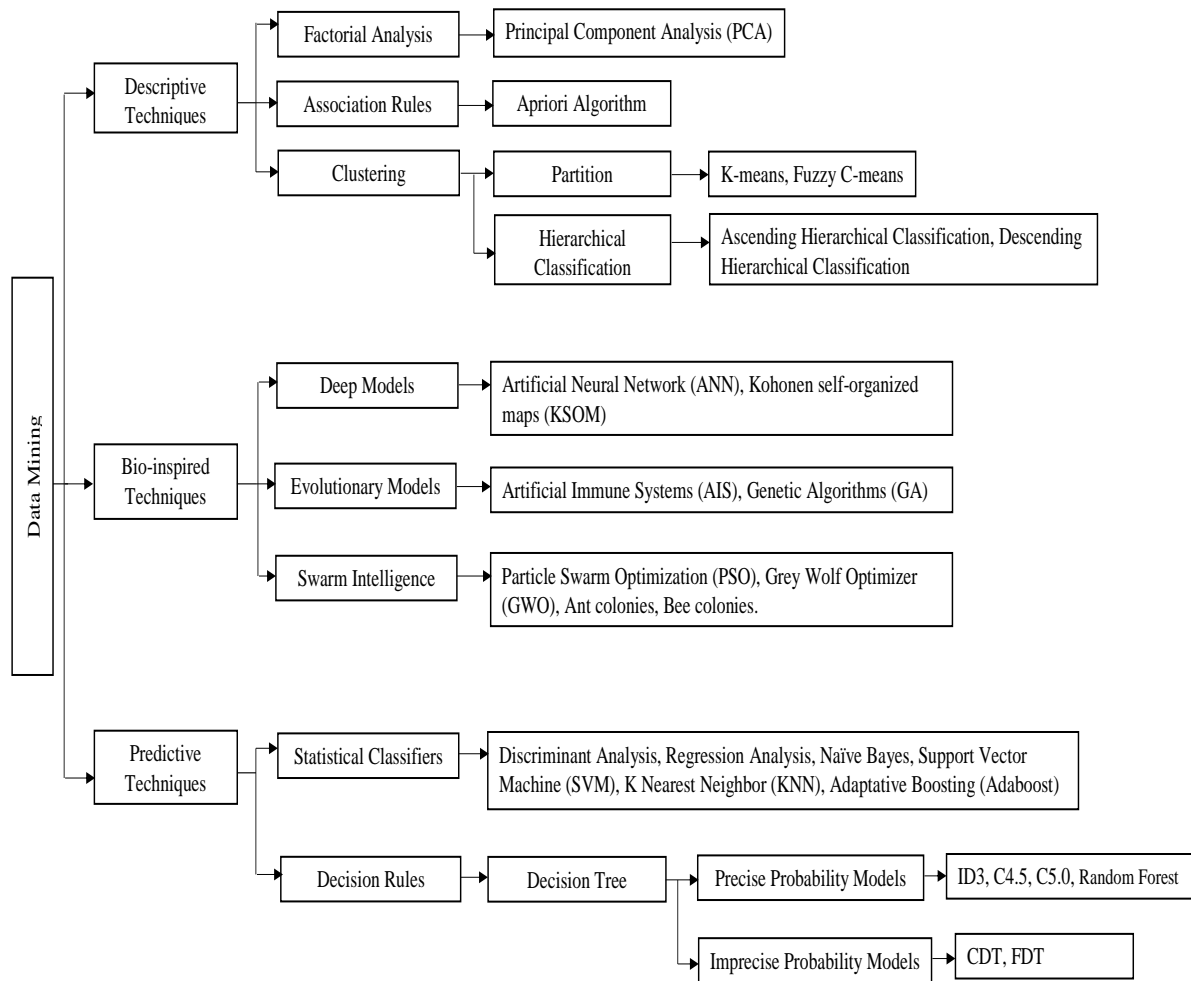


Figure 1.17: Proposed taxonomy for Data Mining techniques

### • Descriptive techniques

This category included a set of techniques for organizing data according to criteria without any prior supervisor. This category has also been divided into three other sub-categories:

- **Factorial Analysis:** consists in searching the most correlated axes that best represent the cloud of large points in small space.
- **Association Rules:** in the Association Rules we try to find different combinations or relevant relationships that explain better a large quantity of data.
- **Clustering:** the last type of these sub-categories is used generally to group data into groups so that heterogeneous data are separated into different groups and similar data are grouped in the same group.

- **Bio-inspired techniques**

These techniques consist in modelling natural systems or to describe biological adaptation of organisms in their environment. They are used to solve complex problems of optimization, machine learning, etc.

According to the operating mechanism and the nature of the behavior treated, we can divide these techniques into three sub-categories:

- **Deep Models:** models of this category inspired from the functioning principle of the human brain. It is based on the construction of a robust mathematical model, which is used later in the classification of new individuals.
- **Evolutionary Models:** they contain a set of methods that reflect evolutionary science. They are based on the concept of mutation to create new solutions.
- **Swarm Intelligence:** Swarm Intelligence also called distributed intelligence, is a concept that relies on the coordination between a massive number of individual intelligence to perform complex tasks.

- **Predictive techniques**

Techniques of this category explain new phenomena based on similar experiences already existed. It is composed of:

- **Statistical Classifiers:** in this sub-category mathematical theories and expressions are defined for describing the prediction model.
- **Decision Rules:** in the Decision Rules we try to construct rules that help in the decision, each rule has one or more premises that lead to a specific result.

## 1.7 Conclusion

In this chapter, we have presented a detailed overview of Data Mining. We have described precisely the applications of Data Mining, the objective, and the different techniques of Data Mining. We have also discussed the advantages and the drawbacks of the techniques of Data Mining and the recent propositions to overcome the drawbacks. Generally, the proposed improvements allow obtaining good results valid within limited constraints like using additional parameters that are fixed empirically, limited the number of the experimental examples, etc.

We have finished our study with new taxonomy that divided the techniques of Data Mining into three main categories: descriptive techniques, bio-inspired techniques, and predictive techniques, each one has been divided into other sub-categories.

As is mentioned above, Data Mining has many fields, in our study we have interested in the video surveillance domain, especially, we try to create new background subtraction

methods using Data Mining techniques. The following chapter gives a state of art on the recent methods used for subtracting the background.

# 2

## Background Subtraction for video surveillance

### 2.1 Introduction

Video surveillance is a new technology that allows automating the remote monitoring of public and private places. By installing surveillance cameras, we can keep track of people and detect suspicious scenes, which facilitates making a quick decision based on the analysis of these scenes.

Background subtraction is an essential step in any video processing. It allows classifying pixels into background and foreground. The background subtraction process takes as input an encoded image in any color space and it produces at the end a black and white image.

The robustness of the background subtraction systems depends on the exploitation of an efficient method that ensures good separation between the background and the foreground pixels.

Several methods have been proposed to improve the detection of moving objects while minimizing false detections.

In this chapter, we will present the most popular methods used for subtracting the background and the recent modifications that have been proposed to improve these methods. After that, we finish our study with new taxonomy that divided the background subtraction methods into four main categories.

## 2.2 Video surveillance

### 2.2.1 Definition

Video surveillance is a telemonitoring operation that allows controlling the public or the private places at distance. It is designed precisely for the security task.

By installing surveillance cameras in a specific location, we can keep track of people and detect suspicious scenes, which facilitates making a quick decision based on the analysis of these scenes [ff121].

The setting up of a video surveillance system requires reception equipment, management equipment, and visualization equipment.

### 2.2.2 Video properties

#### 2.2.2.1 Number of frames by second

Represents the number of still images in a unit of video period. For old cameras, this number extends from six or eight frames by the second (frame/s), but in professional cameras, the number of frames by second is around 120 frames or more. For example, NTSC (the United States, Canada, Japan, etc.) used 29.97 frame/s, while Pal (Europe, Asia, Australia, etc.) and SECAM (France, Russia, regions of Africa, etc.) use only 25 frame/s.

#### 2.2.2.2 Rate

Bit Rate is a measure of the information content in a video stream. Usually, we used bit by the second (bit/s or bps) or megabit by the second (Mbit/s). A higher bit rate produced a better video quality [FAR16].

#### 2.2.2.3 Variable Bit Rate (VBR)

VBR is a strategy to maximize the visual quality of the video while minimizing the bit rate. This type of rate is more suitable for scenes that have quick movements [FAR16].

#### 2.2.2.4 Interlacing

Interlacing defines how to achieve the good visual quality of the video with limited bandwidth. Horizontal scan lines of each interlaced frame are enumerated consecutively and divided into even and odd lines. For example, PAL format is usually indicated by  $576i50$ . Where 576 is the horizontal resolution,  $i$  indicates interlacing, and 50 represents 50 frames by second.

In progressive scanning, each frame contains all the lines of the scan. That means, the image will be displayed completely one time, but in interlacing, the odd lines will be displayed first, then the even lines [FAR16].

### 2.2.2.5 Display resolution

The size of a visual image in the digital video is measured by the number of pixels, while in an analog video the size of the images is measured by the vertical and horizontal scan lines of the resolution.

If we used a digital video, the resolution of images is described as follows:

720/704/640  $\times$  480/60 to NTSC resolution and 768/720  $\times$  576/50 to PAL or SECAM resolution.

For analog video, the number of visible scan lines is constant ( $486_{NTSC} = 576_{PAL}$ ), while the horizontal measurement is changed according to the signal quality [FAR16].

### 2.2.2.6 Proportion

It describes the dimensions of the video screens and the image elements of the video. In general, popular video formats are rectangular, described by width and height. Conventional televisions have a proportion of 4:3, or about 1.33:1. High-quality televisions have a proportion of 16:9, or about 1.78:1.

Pixels on a computer screen have a square shape. But, digital videos have different proportions. For example, an NTSC DV image of 720/480 pixels will be displayed with a proportion of 4:3 for the small pixels. But they displayed with a proportion of 16:9 for the big pixels [FAR16].

## 2.2.3 Video surveillance applications

Video surveillance system provides high-level information to the surveillance officer, which facilitates the decision-making process in a very short time.

Several fields use intelligent video surveillance systems, among them we find [FAR16], [ff21]:

- **Public and commercial security:**

- Monitoring of banks, airports, administrations, etc.
- Surveillance of public places such as public gardens, libraries, museums, stations, movie theaters, and the monitoring of forests for fire detection.
- Surveillance of private places such as houses, stores, car parks, warehouses, etc. And we can use video surveillance for monitoring patients in hospitals.

- **Road safety**

Video surveillance applications provide real-time access to measure the congestion of pedestrians and cars in the busiest areas such as highways. This information is used by private companies to inform drivers equipped with GPS to avoid crowds.

Video surveillance can be used also to:

- Measure speeding of vehicles on the highways.
- Detect red light violations and passing the continuous line.

- **Military security**

Video surveillance is used to:

- Guard the borders from the enemies.
- Guard military equipment.
- Detection of arms and Recognition the quality of the latter.
- Measuring the flow of refugees.
- Determine where the bombs are located.

- **Industrial Security**

Production sites and factories are equipped with surveillance cameras to increase the number of real-time monitoring points, the status of equipment installations, and the progress of processes.

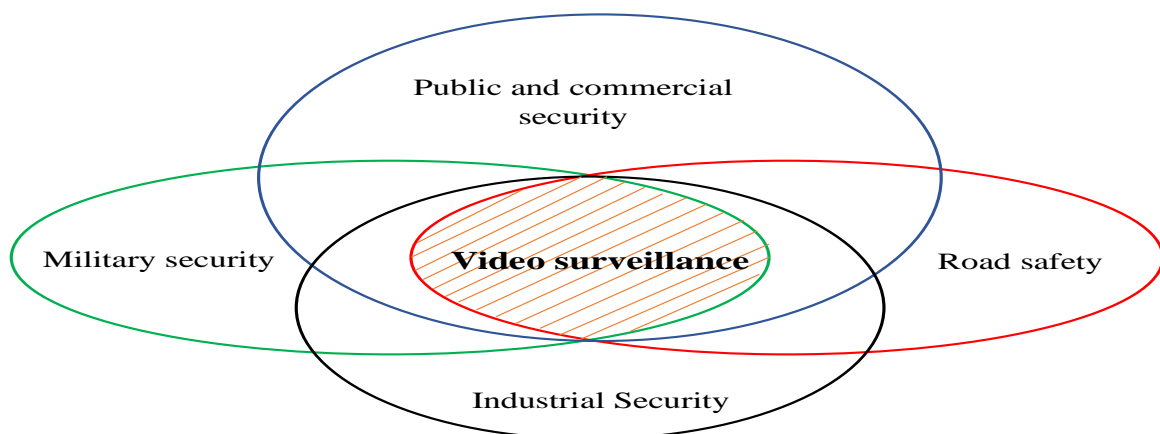


Figure 2.1: Video surveillance applications

All these applications and any other video surveillance application require to start with the background subtraction phase. In the following, we will define some basic notions of background subtraction and the most popular methods used for subtracting the background.

## 2.3 Background subtraction

### 2.3.1 Definition

It is a binary operation that classifies pixels into the background and foreground pixels. Such that the background pixels take the value 0 and the foreground pixels are labeled with 255. Background pixels describe the static objects (Background), while the foreground pixels are the pixels that constitute the moving objects.

The background subtraction process takes an image coded in any color space and it produced a black and white image. According to Bouwmans [Bou12], all the background subtraction methods shared the following steps:

1. **Background Initialization**

During this step, we create a primary background model and we try to train it using a set of background frames (images devoid from the moving objects). Several ways are distinct to design the primary background model (statistical, fuzzy,neuro-inspired, etc.).

2. **Foreground Detection**

Once the primary background model is elaborated, we compare each new frame with this model to detect the moving objects.

3. **Background Maintenance**

In this step, the settings of the background model are updated to pick up any changes in the background within the video sequences.

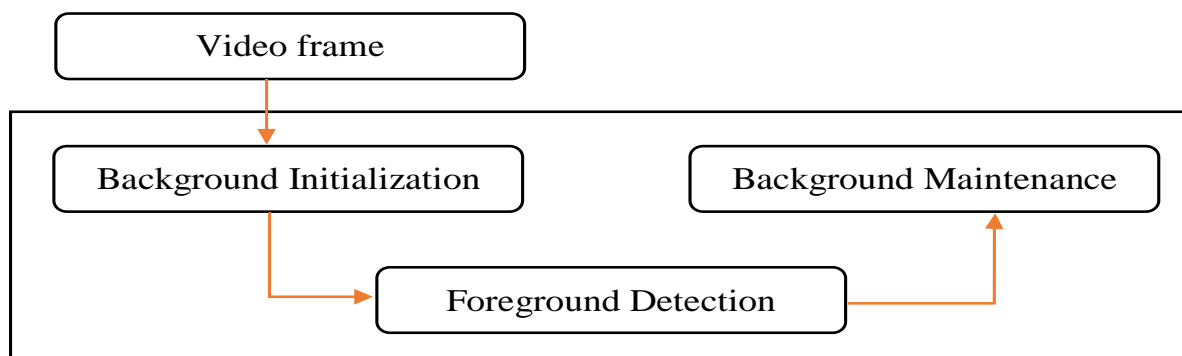


Figure 2.2: Steps of background subtraction methods



## 2.3.2 Background subtraction methods

There exist many background subtraction methods, in this sub-section, we try to give an overview of the most recent background subtraction methods and their proposed improvements.

### 2.3.2.1 Single Gaussian model (SG)

Single Gaussian (SG) was firstly proposed by Wren et al. [WADP97b] in 1997 as a background subtraction model. This model described the  $n$  last variations of each pixel with a probability density function (see Equation 2.1).

$$P(P_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(P_t-u)^2}{2\sigma^2}} \quad (2.1)$$

SG assumed that pixels are independent and it modeled the variations of each pixel separately with a single Gaussian. Each Gaussian is characterized with a mean  $u$  and a variance  $\sigma$ . The mean describes the dominant color of the current pixel  $P_t$ , while the variance represents the viability of the pixel around the mean.

For each pixel  $P$  at the time  $t$  we can determinate its nature using Equation 2.2. If the Equation 2.2 is satisfied  $P_t$  is a background pixel, otherwise  $P_t$  represents the foreground pixels.

$$\frac{|P_t - u_t|}{\sigma_t} < 2.5 \quad (2.2)$$

To adapt the Gaussian model with the different changes of scenes (brightness, dust, etc), a recursive update process of the Gaussian parameters is applied at each new frame (see Equation 2.3 and Equation 2.4).

$$u_t = (1 - \alpha)u_{t-1} + \alpha P_t \quad (2.3)$$

$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha(P_t - u_t)(P_t - u_t)^T \quad (2.4)$$

Where:

$\alpha$ : is the learning rate that determines the speed of adaptation.

To better understand how SG works see the following flowchart.

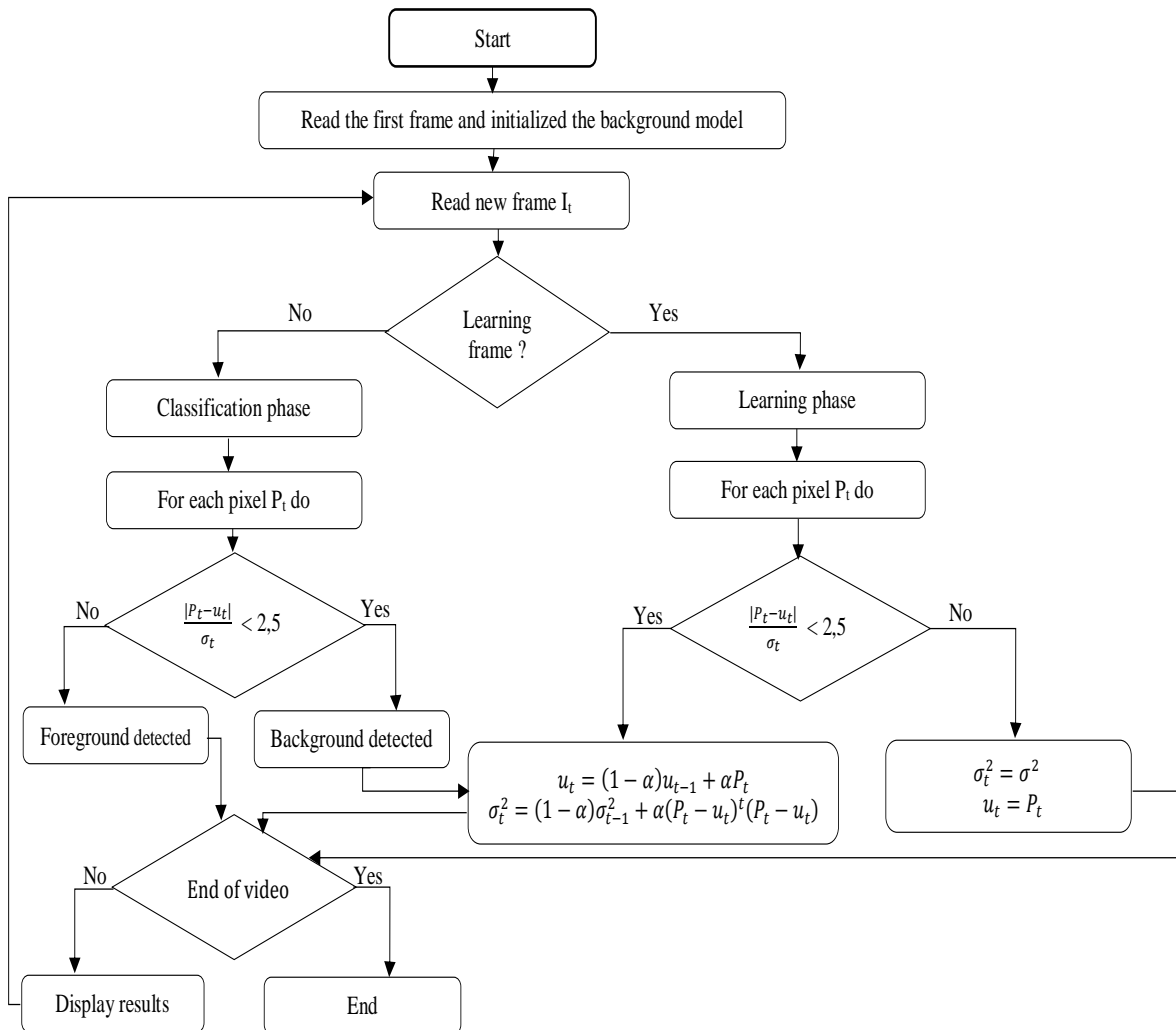


Figure 2.3: Flowchart of Single Gaussian model

This model is simple, fast, and easy to implement. But, it is sensitive to fast pixel variations. Indeed, one Gaussian cannot memorize all the old states of pixels.

Dong et al. [DHYD18] proposed two modifications in the SG model, the first one consists in using an adaptive learning rate that changed according to the video's frame number rather than using a fixed learning rate. While in the second modification, a new strategy for updating the background model is proposed.

This strategy changed Gaussian parameters according to the frame counter for each pixel to record the motion state of the pixel which is judged as foreground. The proposed model achieved good accuracy compared to that of the conventional SG, but the execution speed of the algorithm is lower than that of the SG.

Yin et al. [YKG<sup>+</sup>14] proposed a combination between SG and mean shift algorithm to detect moving objects. The mean shift algorithm in this contribution is used to correct the background model generated with SG. This method has achieved good accuracy in different scenes. But, it is sensitive to the complex background and the noise.

Authors in [ZFX09] proposed an adaptive SG with an integer learning rate and a selective update process to create a robust background model into the different scenes changes. However, the use of a selective update process requires consulting at each time a method that verified the change of states in the video sequence, which makes this proposition slow in the execution.

Many other extensions are presented in [HW03], [KPLB05], [ACZC06], [CMM07], [PKY<sup>+</sup>08], [RKW09], [LBC11] to improve the detection rate of moving objects, but all these propositions are valid within constraints imposed by the authors on the environment to ensure a proper functioning of these letter.

### 2.3.2.2 Gaussian Mixture Model (GMM)

In complex scenes, modeling the background variation using a single Gaussian method is not enough. Indeed one Gaussian cannot memorize all states of the pixel. To overcome this problem, a multimodel approach called Gaussian Mixture Model (GMM) has been developed. The initial version of GMM was proposed by Friedman and Russel [LBC11] in 1997 for road traffic monitoring.

This version represented each pixel with a mixture of three Gaussians and it estimated GMM parameters using the Expectation Maximization (EM) algorithm. In 1999 Stauffer and Grimson [SG99] proposed a standardization of GMM with efficient update equations. GMM assumed that the pixel state is in one of the  $K$  Gaussian distributions. During the execution, GMM compared the value of each pixel with its  $k$  Gaussian probability distributions. A high probability indicated that this pixel is a background pixel.

The probability for observing the value of a pixel  $P$  in time  $t$  is given by:

$$P(P_t) = \sum_{i=1}^K w_{i,t} \eta(\mu_{i,k}, \Sigma_{i,t}, P_t) \quad (2.5)$$

Such that:

$P_t$ : pixel value at time  $t$ .

$K$ : gaussians number.

$w_{i,t}$ : the weight calculated for the  $i$ th Gaussian at time  $t$ .

$\mu_{i,k}$ : the mean of the  $i$ th Gaussian at time  $t$ .

$\Sigma_{i,k}$ : the covariance matrix of the  $i$ th Gaussian at time  $t$ .

$\eta(\mu_{i,k}, \Sigma_{i,t}, P_t)$ : is a probability density function.

After initializing Gaussian parameters ( $w_t, \mu_t, \sigma_t$ ), each pixel value is compared with its existing  $K$  distributions using:

$$\frac{|P_t - u_t|}{\sigma_t} < 2.5 \quad (2.6)$$

If the Equation 2.6 is verified, parameters of the matched Gaussians will be updated with:

$$w_{i,t} = (1 - \alpha)w_{i,t-1} + \alpha \quad (2.7)$$

$$\mu_{i,t} = (1 - \phi_i)\mu_{i,t-1} + \phi_i P_t \quad (2.8)$$

$$\sigma_{i,t}^2 = (1 - \phi_i)\sigma_{i,t-1}^2 + \phi_i(P_t - \mu_{i,t})^T(P_t - \mu_{i,t}) \quad (2.9)$$

$$\phi_i = \alpha\eta(P_t/\mu_i, \sigma_t) \quad (2.10)$$

For the other unmatched distributions, only the weight is updated using:

$$w_{i,t} = (1 - \alpha)w_{i,t-1} \quad (2.11)$$

After this step, all weights are normalized to ensure that their sum is always equal to 1. If all the Gaussian distributions ( $K$ ) are not satisfied the Equation 2.6, the pixel is classified as foreground and the least probable Gaussian will be replaced by the following distribution:

$$\sigma_i^2 = \sigma_t \quad (2.12)$$

$$w_i = \phi_t \quad (2.13)$$

$$\mu_i = P_t \quad (2.14)$$

Once the background model is created, a classification of new pixels is performed. To classify pixels into background and foreground, Gaussian distributions are sorted according the value of:  $w_{i,t}/\sigma_{i,t}$ .

The first  $\beta$  distributions that verified the following equation are selected to represent the background.

$$\beta = \underset{i=1}{\operatorname{argmin}} \left( \sum_{i=1}^K w_{i,t} > B \right) \quad (2.15)$$

Where  $B$  is the minimum part of the data corresponding to the background. For more details see the Flowchart below.

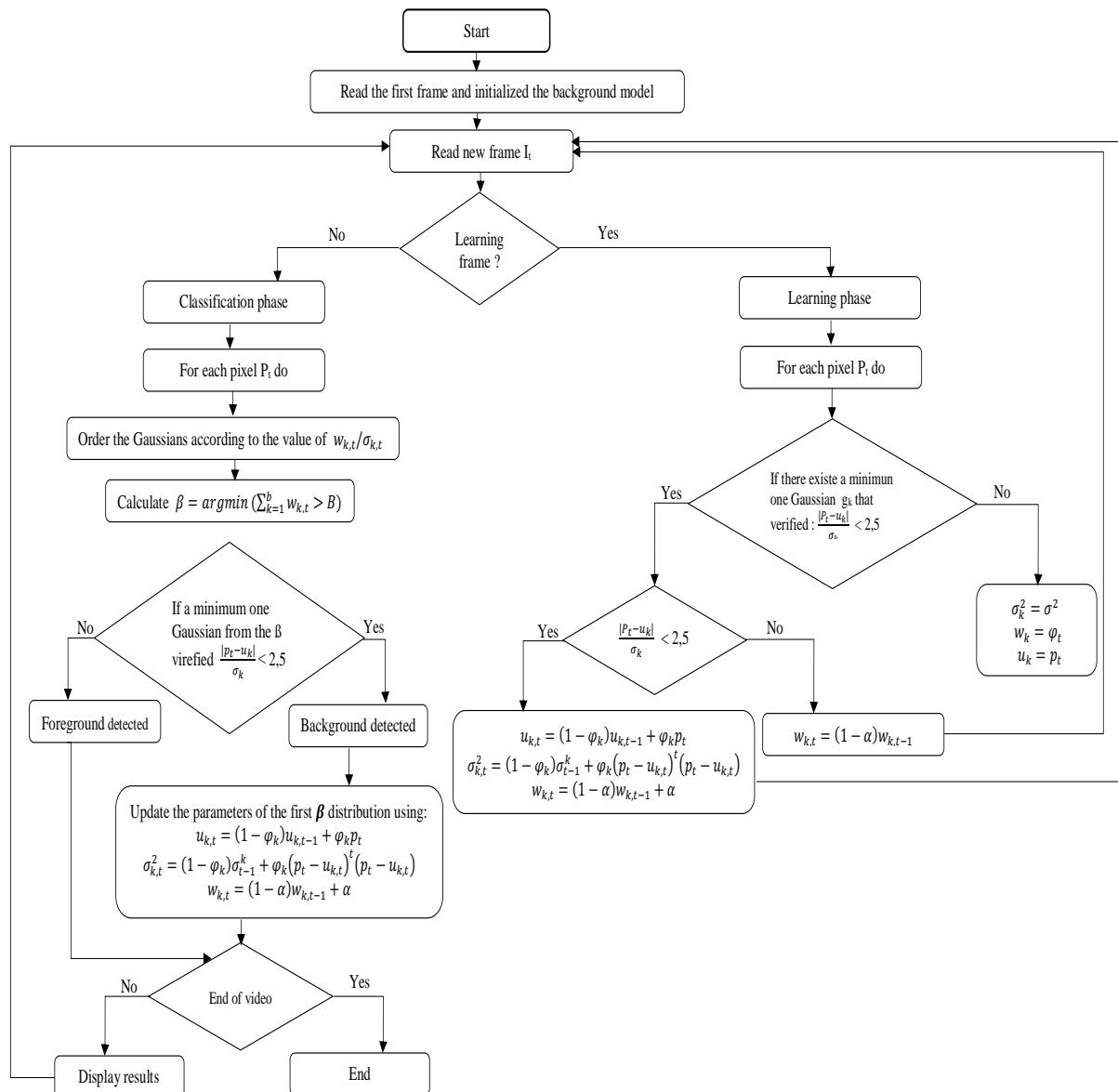


Figure 2.4: Flowchart of Gaussian Mixture Model

Despite the success achieved from GMM, this method remains sensitive to light changes and hidden areas. Many researches are proposed to improve results quality of GMM.

Some works have combined GMM with other methods to create a robust background subtraction model, from these methods we find: GMM and k-means [CSN10], GMM and PSO [WS07], MoG and fuzzy logic [EBBV09] and Markov Random Fields [SW06]. However, the results quality of these methods depends on the value of the initial parameters that need a robust method to estimate it.

Other contributions used heterogeneous features with GMM, from these contributions we have: GMM with Spatio-temporal distribution [XSH16] and boosted Gaussian Mixture Model [PMR10]. boosted Gaussian Mixture Model used chromatic and hysteresis features with GMM for better classifying pixels. Heterogeneous features can improve GMM

precision, but they affect the execution speed of GMM.

Farou et al. [FKSA17] divided each frame into blocks and assigned to each block a spotter. If the spotter attracts a change within blocks of the frame  $n$  and the frame  $n + 1$ , GMM is applied to detect foreground pixels of this block. Experimental results on some datasets show that this proposition has reduced the false detection and it increased GMM speed. In the same context, authors of [DHJ<sup>+</sup>19] introduced some improvements to GMM method. After initializing the background model, each frame is divided into blocks of  $16 \times 12$  pixel. Then the Kanade-Lucas-Tomasi Feature Tracker (KLT) method is used during the learning stage for compensating camera motion. Once the background model is elaborated, the neighborhood of pixels is used to well classify the pixels. To set the learning rate adaptively, a variable age value is proposed for each pixel. These improvements have created a robust and speed GMM method.

To accelerate background modeling speed, reduce the computational complexity and improve the detection rate, Zuo et al. [ZJYK19] proposed a GMM based on the average of image blocks. During the learning, stage frames are divided into blocks, then using the image block averaging method the background model is reconstructed. The background model produced is used with GMM to detect moving objects.

However, in all these propositions the number of blocks in each frame affects the obtained results, sometimes the use of large blocks increased the false detection, for this purpose many empirical tests are required before fixing the number of blocks.

Soeleman et al. [SYN<sup>+</sup>20] have refined the Gaussian Mixture Model using the Region Growing method. This improvement has achieved good results in the camouflage areas, but it is sensitive in the dynamic background.

### 2.3.2.3 Generalized Single Gaussian model (GSG)

The generalized Single Gaussian model was firstly proposed in 2008 by Kim et al. [KSK<sup>+</sup>08]. In this model, pixel variations are described by Laplace's distribution (see Equation 2.16). Laplace's distribution taken into consideration only the variance to represent states of pixels.

Laplace's distribution is closer to a Single Gaussian distribution when the background is quasi-static, but it is sensitive to the background that contains various changes.

$$P(P_t) = \frac{\rho\gamma}{2\Gamma(\frac{1}{\rho})} e^{-(\gamma\rho|P_t - \mu|^\rho)} \quad (2.16)$$

Where:

$$\gamma = \frac{1}{\sigma} \left( \frac{\Gamma(\frac{1}{\rho})}{\Gamma(\frac{1}{\rho})} \right) \quad (2.17)$$

$\Gamma(\cdot)$ : Gamma function.

$\sigma$ : the variance of the distribution.

From the Equation 2.16, we can deduced that if  $\rho = 1$  a Laplace distribution is achieved, while if we fixed the value of  $\rho$  to 2 we obtained Gaussian distribution.

To distinguish the foreground pixels from the background pixels, excess kurtosis of the first  $m$  frames is calculated. Excess kurtosis measured the distribution of data compared to the normal distribution. Excess kurtosis is fixed to 3 for Laplace distribution and 0 for the Gaussian distribution.

The optimal parameters of the models are estimated using maximizing likelihood method of an observed value (see Equation 2.18).

$$g_2 = \frac{N \sum_{i=1}^N (x_i - \mu)^4}{\left( \sum_{i=1}^N (x_i - \mu)^2 \right)^2} \quad (2.18)$$

Where  $\mu$  is the mean and  $N$  is the number of samples.

In this model we distinguish three pixel states as follow:

$$BD(P) = |L_1(P) - L_B(P)| \quad (2.19)$$

$$\begin{cases} 0 & \text{if } BP(P) < k_1 b(P) \\ 1 & \text{if } k_2 n(P) \leq BD(P) \\ \text{suspicious} & \text{if } k_1 b(P) \leq BD(P) \leq k_2 b(P) \end{cases}$$

Such that:

$L_1$  and  $L_B$ : are the luminance components of the pixel  $P$  at the current time and at the background model estimation.

$b$ : is a background model scale setting.

The third case in Equation 2.19 will be refined with the hue component to treat shadow areas. After that, the pixel is classified as either a background or foreground pixel. Indeed, this method allows treating well problems related to changing of brightness.

#### 2.3.2.4 Generalized Gaussian Mixture Model (GGMM)

To enhance the precision of the Gaussian mixture model, authors of [ABZ07] proposed a new model called Generalized Gaussian Mixture Model (GGMM). This model is more robust in dark areas and is more flexible to data shape.

In GGMM, each pixel is represented with its intensity in RGB color space and the probability for observing the current pixel value in RGB color space is given by:

$$P(P_i) = \sum_{i=1}^K w_{i,t} \eta(P_t, \mu_{i,k}, \Sigma_{i,t}, \lambda_i) \quad (2.20)$$

With:

$$\eta(P, \mu, \Sigma, \lambda) = \prod_{j=1}^d A(\lambda_j) \exp\left(-B(\lambda_j) \left|\frac{P - \mu_j}{\sigma_j}\right|^{\lambda_j}\right) \quad (2.21)$$

Where:

$$A(\lambda) = \frac{\left(\frac{\Gamma(\frac{3}{\lambda})}{\Gamma(\frac{1}{\lambda})}\right)^{\frac{1}{\lambda}}}{2\sigma\Gamma\left(\frac{1}{\lambda}\right)} \text{ and } B(\lambda) = \frac{\Gamma\left(\frac{3}{\lambda}\right)}{\Gamma\left(\frac{1}{\lambda}\right)} \quad (2.22)$$

Such that:

$K$  and  $d$ : are the number of distributions of each pixel.

$w_{i,t}$ : is the weight of the  $i$ th Gaussian at the time  $t$ .

$u_{i,t}$  and  $\Sigma_{i,t}$ : are respectively the average vector and the matrix of variance of the  $i$ th distribution.

$\eta$ : is the probability density function.

$\lambda$ : is a parameter which indicated whether distribution used Laplace or Gaussian, if  $\lambda = 0$  Gaussian distribution is used, if  $\lambda = 3$  a Laplace distribution is used.

Parameters update process remains the same as the conventional GMM. This model can identify the optimal number of Gaussians at each frame using the minimization of the minimum message length criterion (LMM).

### 2.3.2.5 Kernel Density Estimation (KDE)

Kernel Density Estimation (KDE) has been proposed as a technique for background subtraction from a set of samples without any a priori information on their form of distribution.

To build a background subtraction model robust to dynamic changes Elgammal et al. [EHD00] proposed to estimate the probability density function of  $N$  recent samples of each pixel's intensity values by a kernel estimator (see Equation 2.23). These samples are taken at consecutive time intervals ( $I$ ).

$$P(P_t) = \frac{1}{N} \sum_{i=1}^N K(P_t - P_i) \quad (2.23)$$

Where  $K$  represents a kernel function taken as a normal Gaussian function  $N(\sigma, 0)$  (Equation 2.24)

$$K(P_t - P_i) = \frac{1}{2\Pi^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(P_t - P_i)^T \Sigma^{-1} (P_t - P_i)} \quad (2.24)$$

With:

$d$ : is the color vector dimension.

Each kernel function is characterized with a kernel bandwidth  $\Sigma$ . Generally,  $\Sigma$  is estimated by the median value of the difference between consecutive images.



Elgammal et al. [EHD00] assumed that color channels are independent and they assumed that each color channel has a different kernel bandwidth. In this case, the bandwidth of the kernel function is defined as follows:

$$\begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \quad (2.25)$$

The probability density function for this case is defined as follows:

$$P(P_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(P_{t,j}-P_{i,j})^2}{\sigma_j^2}} \quad (2.26)$$

Once the probability density function is estimated, the foreground pixels are identified according to the following equation:

$$P(P_t) < T \quad (2.27)$$

With  $T$  is an empirically fixed threshold.

If the Equation 2.27 is not satisfied the pixel  $P_t$  is classified as background.

For adapting this method to the different changes of the background, an updated model is proposed by Elgammal et al. [EHD00] according to two ways: the first is to update the samples that constitute the background model using a very fast adaptation following selective maintenance based on the classification of moving objects. This way is used for sensitive detections. The second way undergoes non-selective maintenance that adapts the background model slowly for the most stable representations.

Despite the success of KDE, this method has some major drawbacks depend on the storage space requires to save the  $N$  samples used in the decision and the update process. In addition, the complexity of the algorithm is very high in computing time which limits the use of this algorithm only for applications that are not in real-time.

Several improvements have been proposed to overcome these problems. Some propositions are focused on changing the bandwidth like: [Tav05], [WZX+20]. Authors of [WZX+20] proposed an adaptive KDE with variable bandwidth. In this proposition, the bandwidth is determined for each pixel according to the uncertainty of the sample measurement  $x_i$  and the estimated measurement  $x$ . This proposition has improved the detection results, but its computation complexity is higher.

Other works are interested in the modification of the kernel function, from these works we cite: Tanaka et al. [TSAT07] that replaced the Gaussian function with a rectangular function. Cvetkovic et al. [CBSDW06] proposed to change the Gaussian function by a polynomial function. Some works are focused on: the reduction of the number of samples

by determining the right buffer size [IGTP05], the use of diversified sampling [MS04] or the use of a sequential sampling based on the Monte Carlo distribution [TGL07].

Zhang [ZYT<sup>+</sup>18] suggested a Kernel Density Estimation with several thresholds. Each pixel in this improvement has a fixed threshold that changed according to the dynamic nature of the pixel.

To improve the performance of KDE, some algorithms have been combined with the latter, from these algorithms we find: the Markov random field method [SS05], graph theory [Mah06] and PCA [Özg07]. Authors of [CMM19] proposed an improved kernel density estimation with early-break method and LUT to well modeling the background.

### 2.3.2.6 Support Vector Machine (SVM)

Background modeling based on Support Vector Machine (SVM) has been proposed firstly by Lin et al. [LLC02]. Since SVM generated only binary outputs, a probabilistic version of SVM is used for background subtraction. In this version, the output of the conventional SVM is transformed using a sigmoidal function (see Equation 2.28).

$$P(y = \frac{1}{f}) = \frac{1}{1 + \exp(Af+B)} \quad (2.28)$$

Where:

$y$ : class label.

$f$ : the output of the decision function of SVM.

$A$  and  $B$ : are adapted using an estimation of the maximum likelihood of the set  $(f, y)$ , and they are derived by minimizing the negative log likelihood function which is described as follow:

$$\min - \sum_i \frac{y_i + 1}{2} \log \left( \frac{1}{1 + \exp(Af_i+B)} \right) + \left( 1 - \frac{y_i + 1}{2} \right) \log \left( 1 - \frac{1}{1 + \exp(Af_i+B)} \right) \quad (2.29)$$

The optimization of the Equation 2.29 gives a minimization of the two previous parameters ( $A$  and  $B$ ).

Usually, the data set is divided into two portions: 80% for the formation of SVM and 20% used for parameter minimization. For modeling the background, Lin et al. [LLC02] used a training base containing 100 background images of size  $160 \times 160$ . Each image is divided into several blocks of size  $4 \times 4$ . An image block is considered as background if the probability of SVM output is greater than a fixed threshold.

### 2.3.2.7 Support Vector Regression (SVR)

To treat complex problems, an extended version of the Support Vector Machine dedicated for the regression has been developed. Regression adopts a function that explains the

behavior of a variable  $Y$  by other static variables  $x$  with minimizing the error between  $Y$  and  $f(x)$ . In the Support Vector Regression, the optimal SVM hyperplane is used to predict  $Y$ .

In 2009 Wang et al. [WBN<sup>+</sup>09] proposed an SVR background modeling. In this model, the intensity variations of each pixel are described by a separate SVR model.

For determining if the pixel is a background or not, Wang et al. [WBN<sup>+</sup>09] provided the intensity value of the current pixel to the SVR model that it associates, SVR output is compared with a predefined threshold to identify the background and foreground pixels. Each pixel  $P$  is represented with a set of data  $D$  structured as  $N$  pair of: input data vector  $x_i$  and labels  $y_i$ .

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}, i \in \{1, \dots, N\}, x_i \in R^P \text{ and } y_i \in R \quad (2.30)$$

Each  $x_i$  represents the intensity of the pixel in the frame  $i$ , and  $y_i$  represents the degree of the confidence of this pixel to be a background pixel. The degree of confidence is given by the following linear regression function:

$$f(x_i) = \sum_{j=1}^N (\alpha_i - \alpha'_j) k(x_i, x_j) + \epsilon \quad (2.31)$$

Where:

$k(x_i, x_j)$ : a kernel function.

$\alpha_i, \alpha_j, \epsilon$ : are lagrange multipliers obtained by solving an optimization problem.

SVR takes the intensity of the pixel as an input and produced a degree of confidence as output. A pixel is classified as a background pixel if its confidence level is between  $S_L$  and  $S_H$ .

$$I_f = \begin{cases} 0 & \text{if } S_L < f(x_i) < S_H \\ 1 & \text{otherwise} \end{cases}$$

Such that:

$S_L$  and  $S_H$  are two thresholds fixed by the user.

### 2.3.2.8 Support Vector Data Description (SVDD)

Support Vector Data Description or SVDD is a separator that allows determining the boundaries around a set of data. This type of separator is more used in handwriting recognition, face detection, recognition, and anomaly detection.

In 2009 Tavakkoli et al. [TNBN08] proposed to use SVDD for background subtraction. This model is applied in scenes that have a quasi-stationary background. The boundaries of a dataset are used as detectors of new data and they classified these letters as background or foreground.

Boundaries are represented with a sphere characterized by a center  $\alpha$  and a radius  $R > 0$ . The reduction of the value of  $R^2$  allows minimizing the sphere volume while keeping the totality of the training data  $x_i$  inside the sphere.

The minimization function is described by the following equation:

$$F(R, \alpha) = R^2 \quad (2.32)$$

Where:

$$\|x_i - \alpha\|^2 \leq R^2, \forall i \quad (2.33)$$

In order to find typical values from the learning set, the distance between each data  $x_i$  and the center of the sphere must be strictly inferior to  $R^2$ . In the case where the distance is large, this distance is penalized (see Equation 2.34).

$$F(R, \alpha) = R^2 + C \sum_i \epsilon_i \quad (2.34)$$

With:

$$\|x_i - \alpha\|^2 \leq R^2 + \epsilon_i, \forall i \quad (2.35)$$

According to Equation 2.34, we can see that the introduction of a negligible variable changes the minimization problem.

The confidence parameter  $C$  is an arbiter between the simplicity of the system and the error committed by the latter. The Equation 2.34 can be transformed as the sum of the Lagrange multipliers (see Equation 2.36)

$$L(R, \alpha, \alpha_i, \gamma_i, \epsilon_i) = R^2 + C + \sum_i \epsilon_i - \sum \gamma_i \epsilon_i - \sum \alpha_i \{R^2 + \epsilon_i - (\|x_i\|^2 - 2\alpha x_i + \|\alpha\|^2)\} \quad (2.36)$$

Such that:

$\alpha_i, \gamma_i$ : are two Lagrange multipliers greater than or equal to zero.

To minimize  $L$  it is enough to respect  $R, \alpha, \epsilon_i$ . If we want to maximize  $L$  we must take into account the parameters  $\alpha_i$  and  $\gamma_i$ . If we suppose that  $0 \leq \alpha_i \leq C$ , the Lagrange multiplier  $\gamma_i$  is easily removed and the solution of the optimization problem was described using:

$$L = \sum_i \alpha_i (x_i \cdot x_i) - \sum_{ij} \alpha_i \alpha_j (x_i \cdot x_j) \quad (2.37)$$

With:

$$0 \leq \alpha_i \leq C, \forall \alpha_i \quad (2.38)$$

If a new data satisfied the Equation 2.34, so its Lagrange multiplier will be expressed as  $\alpha_i > 0$ , otherwise it is set to zero.

After this study, we can deduce that we need just the Lagrange multipliers  $\alpha_i > 0$  for the

description of data.

For each pixel, its own SVDD is built during the learning phase. The SVDD is trained using a set of frames representing the background, the support vector detection, and the values of Lagrange multipliers that maximize the Equation 2.36. During the classification of pixels ( $P_t$ ), the distance to the center of the sphere is calculated using the following equation:

$$\|P_t - \alpha_i\|^2 = (P_t \cdot P_t) - 2 \sum_i \alpha_i (P_t \cdot x_i) + \sum_{ij} \alpha_i \alpha_j (x_i \cdot x_j) \quad (2.39)$$

The pixel  $P_t$  is considered as a background pixel if its distance is smaller or equal to  $R^2$ .

$$\|P_t - \alpha_i\|^2 \leq R^2 \quad (2.40)$$

This method has a set of advantages, such as the precession of the model does not depend on the precession of the estimated probability density functions, moreover, the model used less memory space compared to other non-parametric methods. But this method is expensive in terms of computation time since it is optimized at each time the Lagrange parameters.

### 2.3.2.9 Recursive learning

This method provided a non-parametric estimation of the background model which allows a multimodal distribution using a set of models generated previously. Let  $P_t$  a chromaticity vector of the pixel  $P$  at time  $t$ , the background model is defined with the recursive learning [TNB06] by:

$$M_t^B(P) = [1 - B_t]M_{t-1}^B(P) + \alpha_t H_\Delta(P - P_t) \quad (2.41)$$

Where:

$\sum_P H_\Delta = 1$  and  $\sum_P P \times H_\Delta(P) = 0$ .

$M_t^B(\cdot)$ : is the background model at the time  $t$ .

$H_\Delta(\cdot)$ : is the local kernel function with a bandwidth of length  $\Delta$  used to update the background model.

$\alpha_t$  and  $\beta_t$ : are respectively the learning rate and the forgetfulness rate plan.

Authors of [TNB06] proposed to build a model for the background and the foreground pixels. To ensure fast model convergence, a background model learning technique based on pixel history is proposed. At each iteration, the learning rate is updated using the following equation:

$$\alpha_t = \frac{1 - \alpha_0}{h(t)} + \alpha_0 \quad (2.42)$$

With:

$$\alpha_0 = \frac{1}{256 \times \alpha_M} \text{ and } h(t) = t - t_0 + 1$$

$\alpha_0$ : represented a small target rate and  $\alpha_M$  is the variance of the model.

$h(t)$ : is a monotonically increasing function.

$t_0$ : expresses the instant in which a sudden change is applied to the background.

In the classification, a comparison with the background and the foreground models is performed. At the end the pixel is labeled with the winner model (see Equation 2.43).

$$\ln \left( \frac{M_B(t)}{M_F(t)} \right) \geq k \quad (2.43)$$

Where:

$M_F(t)$ : is the foreground model and  $k$  is a fixed threshold.

If the pixel is classified as a foreground, so the foreground model is updated with:

$$M_t^F(P) = [1 - F_t]M_{t-1}^F(P) + \alpha_t H_\Delta(P - P_t) \quad (2.44)$$

Also, if the pixel is detected as a background, so the background model is updated using the Equation 2.41.

### 2.3.2.10 Wiener filter

Wiener filter was first used in 1999 by Toyama et al. [TKBM99] as a predictive model that describes the background changes. This model allows to give an estimation of the pixel value  $P$  that we want to observe at the time  $t$ . The prediction of the pixel value  $P$  at time  $t$  is defined as follow:

$$P_t = - \sum_{i=1}^L \alpha_i P_{t-i} \quad (2.45)$$

Where:

$P_t$ : is the prediction of the pixel value  $P$  at time  $t$ .

$P_{t-i}$ : the pixel value  $P$  at time  $t - i$ .

$\alpha_i$ : the prediction coefficient.

$L$ : is the most recent number of samples.

A pixel is considered a background if the value of the pixel is closer to its prediction. Before determining the nature of pixels, an error function is calculated (see Equation 2.46).

$$E[e_t^2] = E[P_t^2] + \sum_{i=1}^L \alpha_i E[P_t, P_{t-i}] \quad (2.46)$$

Such that  $\alpha_i$  can be defined by the covariance of the values of  $P_t$ .

The pixel  $P$  is classified as foreground if the following equation is satisfied:

$$\frac{|P - P_t|}{4} < \sqrt{E[e_t^2]} \quad (2.47)$$

This model is efficient when the noise is regular and the background is dynamic.

### 2.3.2.11 Kalman filter

Kalman filter is an algorithm that is designed to solve linear problems. It used a series of measurements taken in a period that contains statistical noise (for example a Gaussian noise) and different inaccuracies, then it produce more correct estimates to the unknown variables using a predictive approach.

Pixel state at each instant  $t$  is described by a vector containing the light intensity value of the pixel and a temporal derivative (see Equation 2.48).

$$P_t = \begin{bmatrix} I_t \\ I'_t \end{bmatrix} \quad (2.48)$$

For each pixel, the vector state is updated according to the following equation:

$$\begin{bmatrix} I_t \\ I'_t \end{bmatrix} = \begin{bmatrix} I_{t-1} \\ I'_{t-1} \end{bmatrix} + k \left( I_t - H.A \begin{bmatrix} I_{t-1} \\ I'_{t-1} \end{bmatrix} \right) \quad (2.49)$$

Such that:

$A$ : the system matrix that describes the evolution of the background.

$H$ : named the measurement matrix and it describes the relationship between the measure and the condition.

$k$ : the Kalman gain matrix that defines the learning rate, where  $k = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}$ .

Kalman filter detected the change in the system using the prediction, it calculated the difference between the value estimated by the prediction and the actual measurement of the pixel. Foreground and background pixels are identified by:

$$m(P_t) = \begin{cases} 1 & \text{if } d(\hat{P}_t, P_t) \geq T \\ 0 & \text{otherwise} \end{cases}$$

Where:

$T$ : is a threshold fixed by the user.

$d(\hat{P}_t, P_t)$ : is the difference between the estimated value and the actual measurement of the pixel  $P_t$ .

Several versions of the Kalman filter have been proposed to improve the precession of background subtraction [RMK95], [GZ01], [Z<sup>+</sup>03], [LZ10], [ASA<sup>+</sup>11]. The difference between them lies in the choice of the features.

Also, an extended Kalman filter that substituted the linear function  $H(\cdot)$  with a nonlinear function is proposed from Kwame et al. [KLLH19].

### 2.3.2.12 Frame Difference Method (FDM)

Is among the simplest method for subtracting the background. This method does not require a background model to identify moving objects. A simple analysis of pixel brightness variation is sufficient to detect moving and static objects.

The principle of the method consists in computing the absolute difference between two successive frames [CLK<sup>+</sup>00]. Given a grey-level image, the difference will be calculated for each pixel  $P_{x,y}$  in the frame  $I_{t-1}$  and their corresponding pixel in the frame  $I_t$  (see Equation 2.50).

$$\Delta t_{x,y} = |I_t(x,y) - I_{t-1}(x,y)| \quad (2.50)$$

If the image is encoded in RGB, the difference can be calculated by different variants, such that: Manhattan distance, Euclidean distance, or Chebyshev distance see respectively Equations 2.51, 2.52 and 2.53)

$$\Delta t_{x,y} = |I_t^R(x,y) - I_{t-1}^R(x,y)| + |I_t^G(x,y) - I_{t-1}^G(x,y)| + |I_t^B(x,y) - I_{t-1}^B(x,y)| \quad (2.51)$$

$$\Delta t_{x,y} = \sqrt{(I_t^R(x,y) - I_{t-1}^R(x,y))^2 + (I_t^G(x,y) - I_{t-1}^G(x,y))^2 + (I_t^B(x,y) - I_{t-1}^B(x,y))^2} \quad (2.52)$$

$$\Delta t_{x,y} = \max \{|I_t^R(x,y) - I_{t-1}^R(x,y)| + |I_t^G(x,y) - I_{t-1}^G(x,y)| + |I_t^B(x,y) - I_{t-1}^B(x,y)|\} \quad (2.53)$$

For determining the nature of the pixel  $P_{x,y}$ , if it is a background pixel or a foreground pixel, a binary mask with a predefined threshold ( $T$ ) is applied according to the result of  $\Delta_{x,y}$ .

$$I(x,y) = \begin{cases} 1 & \text{if } \Delta t_{x,y} > T \\ 0 & \text{otherwise} \end{cases}$$

The threshold value is chosen according to the change of scene, a quick change in the scene sequences requires a high threshold value to attract well all the changes.

To improve the precision of this method, Kameda and Minoh [KM96] proposed a Three-frame Difference method. In this proposition, two frame difference operations are applied. The result is obtained from a combination of the two resulting masks using the *AND* logical operator. This method provided good detection results when objects moved slowly. Authors of [SLD<sup>+</sup>06] applied a Three-frame Difference method for subtracting the background. The frame difference operations are applied between frames:  $I_{t-1}$ ,  $I_t$  and the background frame  $I_B$ . In this proposition, the quality of results depend on the background frames chosen. It is necessary to select or to create a background frame that represents all variations of the background (brightness change, dust, etc). Migliore et al. [MMN06] in their article used an efficient background subtraction method for generating a robust background model to the proposition of [SLD<sup>+</sup>06].

Tan et al. [TLL07] calculated the average of the  $N$  first images devoid from the moving



objects to constitute a reference frame. After that, frame difference operation is applied on each frame using the reference frame to detect moving objects.

To create a robust method for moving objects, authors of [ZCX<sup>+</sup>20] proposed a combination between the Frame Difference method (FDM) and spatial Multi-channel CNN. FDM is used to detect and to transform cloudlike objects from large images into small images, these images are sent later to spatial Multi-channel CNN to well-classifying pixels. Experimental results show the effectiveness of this combination, but CNN requires more time in the learning stage, and it undergoes offline learning.

Xu et al. [XJZ20] proposed an improved Frame Difference method that taken into consideration global illumination change. In this proposition, the ratio of global illumination change ( $R_t$ ) is introduced in the equations of the Frame Difference method to overcome problems related to the illumination changes, dynamic background, noise, and ghosts. This proposition has achieved higher detection accuracy, but it requires a robust method to calculate automatically  $R_t$  and to change it according to the change of scenes.

### 2.3.2.13 Codebook

The codebook was firstly proposed by [HHD00]. In this model, each pixel is represented by a set of three visual words: the minimum intensity, the maximum intensity, and the maximum of the difference in intensity between two consecutive images observed in an interval of time (see Equation 2.54).

$$\begin{bmatrix} \min(P) \\ \max(P) \\ \text{diff}(P) \end{bmatrix} = \begin{bmatrix} \min \{V^j(P)\} \\ \max \{V^j(P)\} \\ \max \{|V^j(P) - V^{j-1}(P)|\} \end{bmatrix} \quad (2.54)$$

Such that  $V^j(P)$  represents the intensity value of pixel  $P$  in the  $j$ th image.

$\min(P)$ ,  $\max(P)$  and  $\text{diff}(P)$  are estimated during the learning phase using background images and they are updated regularly over time.

The updating process can be performed locally at the pixel level, or globally taking into account physical changes in the background.

Foreground pixels and background pixels are defined with:

$$B(P) = \begin{cases} 1 & \text{if } |I^t(P) - \min(P)| > \text{diff}(P) \text{ or } |I^t(P) - \max(P)| > \text{diff}(P) \\ 0 & \text{otherwise} \end{cases}$$

Where:

$I^t(P)$ : the pixel  $P$  of the image  $I^t$ .

Several extensions have been proposed in the literature to improve the Codebook technique. Kim et al. [KCHD04] proposed a Codebook with a cylinder look. This model is enriched by three visual words: word occurrence frequency, the maximum duration in

which the word was not solicited during the learning, and the first and the last access to the visual word.

In the same context, authors of [SCBB15a] proposed a Codebook that adjusted the internal parameters of the model after each pixel classification using a feedback mechanism. Krungkaew and Kusakunniran [KK16] used the *lab* color space as the visual word of the Codebook to detect moving objects when the background is dynamic.

Authors in [AW19] used *CIEDE2000* color difference formula to calculate the color difference between two pixels. To well extract moving objects, the pixels of this proposition are coded in *lab* color space.

This method achieved a good detection rate when the background is dynamic, but it is sensitive to camouflage areas. For implementing a fast motion detection method, a combination between Codebook and edge detection is proposed in [LHZ19].

Shang et al. [SYH<sup>+</sup>19] proposed an improved Codebook based on *YUV* color space. In this proposition, only two visual words are used to describe the background changes. Furthermore, the authors have added an update mechanism in the background modeling stage and the detection process for enhancing the precision of the method in real-time application. This proposition is tested only on one dataset and other tests on other public datasets are required.

#### 2.3.2.14 Sub-spatial learning based on PCA

In 2000 Olivier et al. [ORP00] developed a background subtraction model based on Principal Component Analysis (SL-PCA). The main idea of this model consists in applying PCA on  $N$  images to obtain the eigenvectors. Firstly, the mean  $I_\mu$  and the covariance matrix  $\sigma$  are computed from samples of frames representing the background. These samples are taken from non-consecutive intervals to well captured the diversity of the background.

After this step, a diagonalization is applied on the covariance matrix  $\sigma$  for obtaining matrix of eigenvectors  $V$  and diagonal matrix  $D$  in which its elements are the eigenvalues (see Equation 2.55).

$$M = V.D.V^t \quad (2.55)$$

Only the eigenvectors associated with the  $K$  largest eigenvalues ( $K < N$ ) are taken into consideration to decrease the size of the learning space.

Once the training is finished, current images  $I_t$  are projected in the new space using the mean image  $I_\mu$  and the eigenvectors ( $V$ ) (see Equation 2.56).

$$I'_t = (I_t - I_\mu) V^T \quad (2.56)$$

For identifying the moving objects, just calculate the distance between the current image  $I_t$  and the reconstructed image from its projection in the reduced space  $I'_t$  (see Equation 2.57).

$$|I_t - I_R| > T \quad (2.57)$$

Where:

$$I_R = V.I'_t + I_u \quad (2.58)$$

Where:

$T$ : is a threshold define empirically.

This model contains a set of drawbacks, the first one is that the size of the moving objects must be small. Furthermore, objects in this model will not be stationary in the same location for a long period in the learning phase. In addition to that, the use of this model requires a large calculation time. The last problem is that the SL-PCA does not have a robust background model update process and it does not support multimodal representations.

To solve the problem of the size of the moving objects, Xu et al. [XSG06] [XGS08] proposed a method of error compensation to reduce the impact of moving objects on the background subtraction model. in the same context, Kawabata et al. [KHS06] proposed to estimate the moving objects using a bias of an optimal iterative projection of the dynamic scene.

To speed up the execution time of SL-PCA, Li [Li04] proposed a new background subtraction method based on incremental principal component analysis (SL-IPCA). The notion of weighting has been introduced to SL-IPCA in [SL08]. In the same context authors of [TL08] proposed to substitute the PCA with the Independent Component Analysis (ICA) for subtracting the background.

Another work of [BG07] proposed a decomposition of the video content by an incremental non-negative matrix factorization (NMF). All these propositions have increased the speed of the SL-PCA but their results precision has not surpassed the SL-PCA method.

Dai-Hong et al. [DHLDSY19] proposed a new algorithm that combined principal component analysis and scales invariant feature transform (PCA-SIFT) to construct a robust background model. After constructing the background model, the mean shift is applied to separate the foreground from the background.

Despite, the speed of this proposition, its accuracy, and its precision are not good as that of SIFT.

Authors of [MGN19] proposed a new robust PCA for detecting moving objects with a panoramic camera. This proposition used a weighted total variation regularization and an improved low-rank matrix estimator (OptShrink) to estimate the components of the background and the foreground in the new space.

### 2.3.2.15 Neural network methods

Algorithms of this type build a robust mathematical model to distinguish background pixels from the foreground pixels. During the learning phase, a set of images with their ground truth are used to train the neural network.

The learning is done by modifying the values of the synaptic weights according to the data presented to the network. This process is repeated until the error between the desired and the found output is very small. Once the training is finished, the activation is calculated for each new input. The activation score is compared with a predefined threshold to determine the background and foreground pixels.

Recently, many deep learning models are used to improve the effectiveness of background subtraction systems. Babaei et al. [BDR18] proposed a new method for subtracting the background using the Convolutional Neural Network (CNN). As an extension of this work, authors of [LK18a] used triplet convolutional neural networks to construct a more robust background model.

A semi-automatic CNN has been proposed by Wang et al. [WLJ17]. In This method, authors have used only a few moving objects which are outlined manually for well-classifying pixels. Gregorio and Giordano [DGG17] proposed a weightless neural network called WiS-ARDrp as a background subtraction method.

Auto-Adaptive Parallel SOM Architecture (AAPSA) [RACM16] is a self-organized map that works simultaneously to detect changes in video sequences. Authors of [GJ20] proposed deep learning with an autoencoder network for modeling the background. In this method, a greedy layer-wise pre-training approach is used to optimize weights.

Vijayan and Mohan [VM20] proposed a deep-neural network model for the background subtraction process. The optical-flow information, spatial information and background model information are used to train the network model. This information's help the network model to more capture temporal information, which allow flexible foreground segmentation.

Despite the high precision achieved from the neural network methods, these latter used an offline training mechanism that required a huge number of ground truth examples to create a robust background subtraction method. Indeed, in the real application, these examples are not available. In addition to that, this type of learning needs very powerful computers and takes a lot of time in the training stage. Furthermore, these methods are not adapted to the permanent changes of the background.

## 2.4 Proposed taxonomy

This study allowed us to develop new taxonomy (see Figure 2.5) that divided background subtraction methods into four main categories.

- **Parametric methods**

These methods use an estimation function to describe the  $n$  last variations of each pixel based on several parameters. From these methods, we find a Single Gaussian model (SG), Gaussian Mixture Model (GMM), etc.

- **Non-parametric methods**

Unlike parametric methods, non-parametric methods do not require that specific parameters characterize the population distribution.

Parametric methods category contains methods like Kernel Density Estimation (KDE), Frame Difference Method (FDM), etc and two other sub-categories: Wide Margin Separators and Predictive filters.

- **Wide Margin Separators:** Wide Margin Separators are invented in 1998 by Vapnik et al. [Vap98]. Algorithms of this category can solve problems that are not linearly separable using kernel functions, and they are characterized by: the absence of local minima, the low density of the solution, and they have a control mechanism at the margin.
- **Predictive filters:** Predictive filters allow the creation of predictive models using the history of the old values of the pixel. When the difference between the pixel value and its prediction is significant, the probability that this pixel is a foreground becomes high.

- **Sub-spatial methods**

This type contains a set of methods that taken into consideration spatial information to build a background subtraction model robust to brightness changes. From these methods, we find SL-PCA, SL-IPCA, etc.

- **Neural network methods**

Methods of this category are inspired by human brain function. They build a robust mathematical model to distinguish between background and foreground pixels.

Convolutional Neural Network (CNN) and their extensions are recently among the most used methods of this category due to their success in capturing the spatial and temporal dependencies in an image using relevant filters.

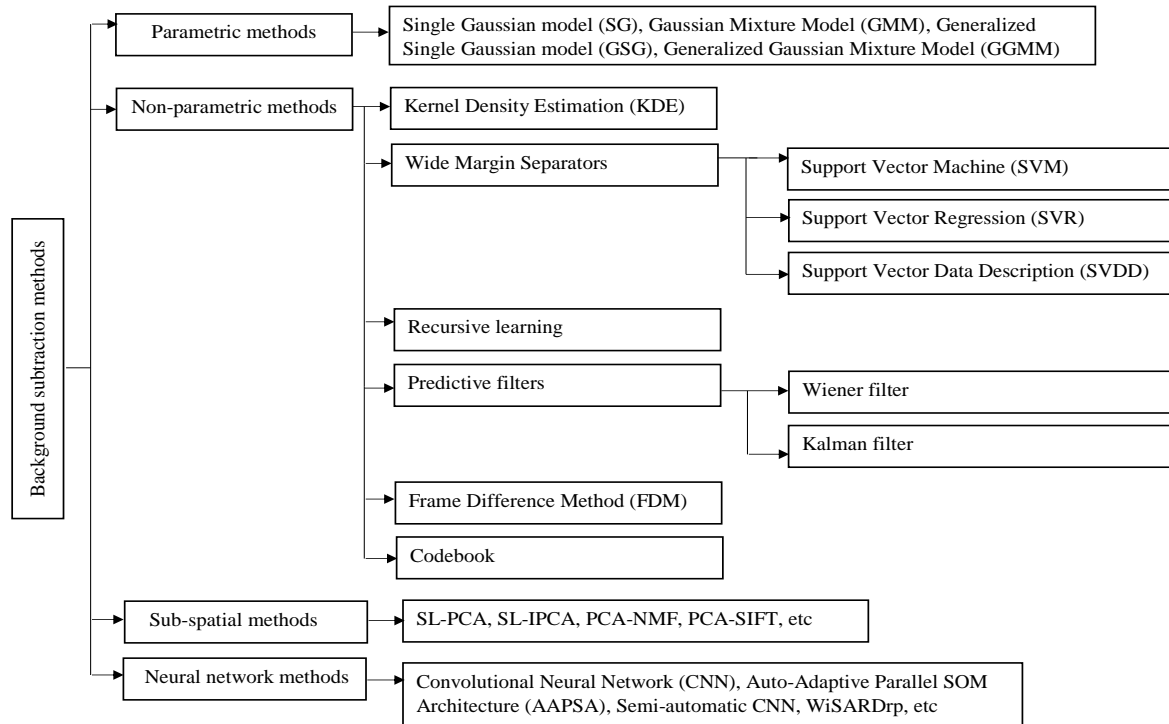


Figure 2.5: Proposed taxonomy for background subtraction methods

## 2.5 Conclusion

In this chapter, we have given an overview of the most popular methods used for subtracting the background. These methods suffer from some problems that influence their results quality. Several modifications have been proposed to overcome the problems of these methods, but most of them remain efficient under certain conditions.

Single Gaussian (SG) and Gaussian Mixture Model (GMM) are among the most popular background subtraction methods since they are simple, fast and use less memory space. In the fourth chapter, we will propose two new background subtraction systems by exploiting: Single Gaussian method, Gaussian Mixture Model and the Artificial Immune Recognition System (AIRS).

Before that, we try to give in the following chapter an overview of the Artificial Immune Recognition System, and we will discuss their steps with some changes which we have introduced.

## Part II

# Proposed systems

# 3

## An Enhanced Artificial Immune Recognition System (EAIRS)

### 3.1 Introduction

In recent years, scientific researchers have been exploited bio-inspired approaches to solve complex optimization and machine learning problems. These approaches try to model the natural systems or to describe the biological adaptations of organisms in their environment.

As mentioned in chapter one, we find many bio-inspired techniques that can serve as metaphors for computer systems such as Genetic Algorithms (GA), Artificial Neural Network (ANN), Particle Swarm Optimization (PSO), Artificial Immune Recognition System (AIRS) and there are many others.

Artificial immune recognition system (AIRS) is among the bio-inspired approaches that have achieved considerable success. This system exploited different aspects of the natural immune system such as the concept of memorization, recognition, cloning, mutation, etc to solve complex problems. Despite the great success of the AIRS, the latter has some drawbacks that influenced their performance in terms of data explosion, results quality and calculation cost.

In this chapter, we will give an overview of the artificial immune recognition system algorithm developed in [WB02], and we will discuss their steps with some modifications which we have proposed to overcome their drawbacks. The effectiveness of these modifications



has been proven by many empirical tests on some public datasets: Iris, Pima-Diabetes, Ionosphere, Sonar, Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer.

## 3.2 Concepts

The natural immune system has an intelligent ability to recognize self and non-self cells. When a foreign element (antigens) enter the body, the immune system performs an antigenic interaction. This interaction is carried out with *ARBs* cells using a degree of correspondence (affinity) between the *ab* antibodies and the *ag* antigen. Once the affinity is determined, the most stimulated *ab* is cloned and mutated to produce new clones. These clones are filtered to organize the survival of individuals within the *AB* population. The remaining *ab* after the filtering process are later selected as memory cells. Inspired by this mechanism, [WB02] proposed a supervised learning algorithm called Artificial Immune Recognition System (*AIRS*) in which the antigens are the training data, and the memory cells constitute the learning model (*MC*). *MC* set is used later for classifying new data using *K* Nearest Neighbor classifier (*KNN*).

Before defining the main steps of the *AIRS*, we will present all concepts and terms used in the *AIRS*.

*nc*: the number of antigen classes.

*MC*: a set of memory cells and *mc* is a member or individual of this set.

*mc.c*: the class of a memory cell *mc*, such that  $mc.c \in 1, 2, \dots, nc$ .

*mc.f*: the feature vector of a memory cell *mc*, and  $mc.f_i$  is the *i*th feature of this vector.

*mc.w*: the weight of an *mc* which indicate the activity degree of the *mc* during their lifetime.

*mc.DL*: a counter that calculates the lifetime of a memory cell *mc*.

*S\_DL*: it is the lifetime threshold for an *mc* in the *MC* set.

*S\_activation*: the minimum threshold for an *mc* to be active.

*AG*: a set of antigens and *ag* is an individual of this set.

*ag.c*: the class of an antigen *ag*, such that  $ag.c \in 1, 2, \dots, nc$ .

*ag.f*: the feature vector of an antigen *ag*, and  $ag.f_i$  is the *i*th feature of this vector.

*ARB*: Artificial Recognition Ball (ARB), also called *B* cell, has in the old version of the *AIRS*: an antibody and a count of the number of resources held. In the version which we have proposed, it has an antibody, a lifetime counter and a weight that measures their activity degree.

*AB*: a set of *ARBs*. It constitute the existing cell population and *ab* (*ARB*) is a member of this population ( $ab \in AB$ ).

*ab.c*: the class of an *ARB*, where  $ab.c \in 1, 2, \dots, nc$ .

*ab.stim*: the stimulation degree of an *ARB* with an antigen *ag*.

*ab.resources*: the number of resources held from an *ARB*.

*ab.w*: the weight of an *ab* which indicates the activity degree of the *ab* during their lifetime.

*TotalNumResources*: the total number of resources allowed in the system.

*TotalWeights*: total weights in the system.

*Affinity*: a measure of similarity between two antibodies or antigen-antibodies, this value is always between 0 and 1, and it is calculated using the Euclidean distance between two feature vectors.

*Stimulation*: a function used to measure the response of an *ARB* to an *ag* or another *ARB*. Noted that the stimulation is the inverse of the affinity.

*Affinity Threshold (AT)*: represents the average affinity of all antigens in the learning set.

*Affinity Threshold Scalar (ATS)*: a value between 0 and 1. This value is involved in the memory cells replacement process.

*Mutation\_rate*: a value between 0 and 1 indicates the probability that a feature of an *ARB* will be mutated.

*Clonal\_rate*: an integer value that determines the number of mutated clones that a given *ARB* can produce.

*Hyper\_mutation\_rate*: an integer value used with the clonal rate to determine the number of the mutated clones that a given *ARB* can produce.

*Stimulation threshold*: a value between 0 and 1 used to indicate if the stop criterion for training an antigen is reached or not.

### 3.3 AIRS stages and proposed modifications

In this subsection, we will give an overview of the Artificial Immune Recognition System. Specially, we try to describe the different stages, equations, and some functions of the AIRS. Once the training model is elaborated, the classification stage is performed using the *K* Nearest Neighbor (*KNN*).

AIRS has four main stages: initialization, memory cell identification and ARB generation, Competition for resources and development of a candidate memory cell, memory cell introduction. In the following, we will describe each stage, and we will try to introduce some modifications to these stages.

#### 3.3.1 Initialization

We can consider it as a pre-processing step because, in this stage, all items of the data set are normalized so that the distance between two feature vectors is always within  $[0, 1]$ . After the normalization step, the affinity threshold (*AT*) is calculated from the training data using the following equation:

$$AT = \frac{\sum_{i=1}^n \sum_{j=i+1}^n Affinity(ag_i, ag_j)}{\frac{n(n-1)}{2}} \quad (3.1)$$

Such that:

$n$ : the number of antigens in the training data.

$Affinity(ag_i, ag_j)$ : defined as the Euclidean distance between two pairs of antibodies or antigens.

The affinity threshold is used later in the memory cell introduction stage. At the end of this stage, the  $MC$  set and the  $ARB$  population are initialized either with a set of  $ag$  chosen randomly, or initialized with  $\emptyset$ .

The elements of  $MC$  set and  $ARBs$  are structured in homogeneous groups ( $MC_{ag,c}$ ,  $AB_{ag,c}$ ).

---

### Algorithm 1 Initialization

---

**Require:**  $AG$

**Ensure:** the initial  $MC$  set and the initial  $ARB$  set

**Variables**

**Begin**

$nbrMC, nbrARB$  : They are respectively the initial number of  $mc$  and  $ARBs$

$minf_i$ : The  $i$ th minimum value in the feature vectors

$maxf_i$ : The  $i$ th maximum value in the feature vectors

**End**

**for each**  $ag \in AG$  **do**

**for each**  $ag.f_i \in ag.f$  **do**

$$ag.f_i \leftarrow \frac{ag.f_i - minf_i}{maxf_i - minf_i}$$

**end for**

**end for**

$$AT \leftarrow \frac{\sum_{i=1}^{|AG|-1} \sum_{j=i+1}^{|AG|} Affinity(ag_i, ag_j)}{\frac{|AG|(|AG|-1)}{2}}$$

$nbrMC \leftarrow value$

**while**  $|MC| \leq nbrMC$  **do**

  Choose randomly an  $ag \in AG$

$$MC_{ag,c} \leftarrow MC_{ag,c} \cup ag$$

**end while**

$nbrARB \leftarrow value$

**while**  $|AB| \leq nbrARB$  **do**

  Choose randomly an  $ag \in AG$

$$AB_{ag,c} \leftarrow AB_{ag,c} \cup ag$$

**end while**

---

$MC$  set is very important since it allows to constitute the learning model. In the old initialization process of the AIRS, if we initialize  $MC$  set with random  $ag$ , we can get bad representatives or find groups of  $mc$  that do not have any representatives.

To avoid these problems, we have suggested a semi-random process that taken the mean vector of  $ag$ , which have the same class as a representative vector of each group ( $ag_m$ ), then we try to assign randomly some  $ag$  to these groups.

This improvement gives a good beginning to the AIRS and increases the probability to generate better  $mc_{match}$ .

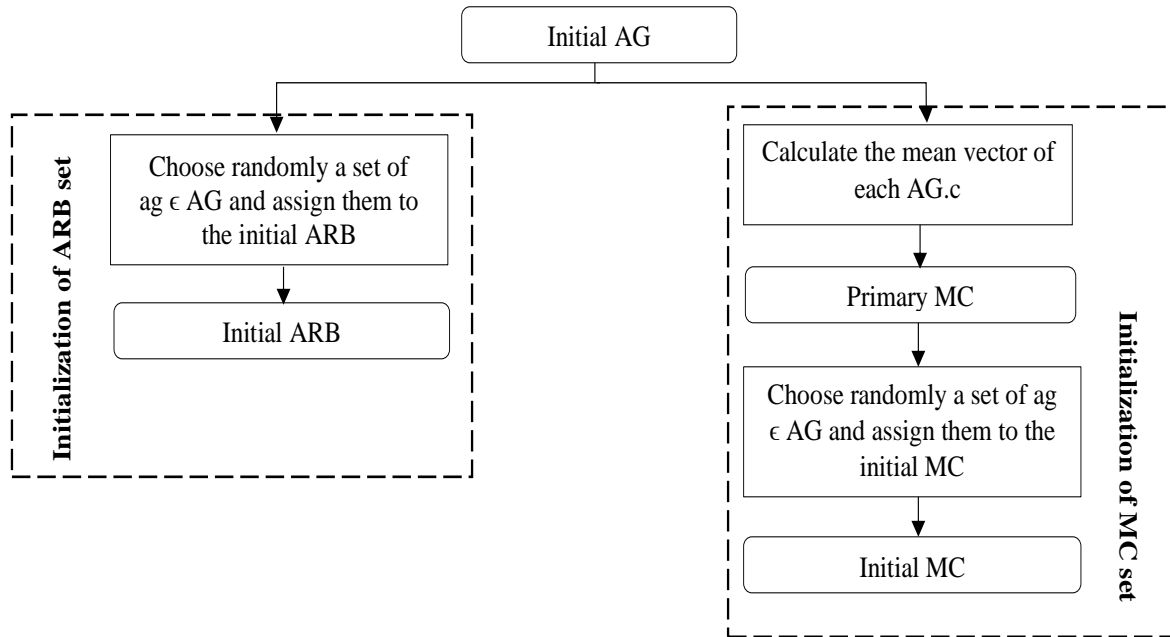


Figure 3.1: Flowchart of the proposed initialization of MC and ARB sets

For each  $mc$  created, we attributed a weight and a lifetime counter.  $mc$  which are assigned randomly have an initial weight equal to 0. Concerning the best representative of  $mc$ , we have given an initial weight different to 0 (see Equation 3.2).

About the lifetime, we have fixed at the beginning to 0 for each created  $mc$ . We have added the weight concept to the  $ARBs$  to substitute the notion of resources in the original version of AIRS.

In the beginning, all the  $ARBs$  weights are initialized with 0. The weight and the lifetime are used later to update the  $MC$  set.

$$mc.w = \frac{1}{nc} \quad (3.2)$$

**Algorithm 2** Proposed initialization**Require:**  $AG$ **Ensure:** the initial  $MC$  set, the initial  $ARB$  set**Variables****Begin** $nbrMC, nbrARB$  : They are respectively the initial number of  $mc$  and  $ARBs$  $minf_i$ : The  $i$ th minimum value in the feature vectors $maxf_i$ : The  $i$ th maximum value in the feature vectors $ag_m$  : The mean vector of the antigens belonging to the same class**End****for each**  $ag \in AG$  **do**  **for each**  $ag.f_i \in ag.f$  **do**

$$ag.f_i \leftarrow \frac{ag.f_i - minf_i}{maxf_i - minf_i}$$

**end for****end for**

$$AT \leftarrow \frac{\sum_{i=1}^{|AG|-1} \sum_{j=i+1}^{|AG|} Affinity(ag_i, ag_j)}{|AG|(|AG|-1)}$$

 $nbrMC \leftarrow value$ **for**  $c = 1$  to  $nc$  **do**   $ag_m \leftarrow mean\_vector(AG_c)$    $mc \leftarrow ag_m$    $mc.w \leftarrow \frac{1}{nc}$    $mc.DL \leftarrow 0$    $MC_c \leftarrow MC_c \cup mc$ **end for****while**  $|MC| \leq nbrMC$  **do**  Choose randomly an  $ag \in AG$    $mc \leftarrow ag$    $mc.w \leftarrow 0$    $mc.DL \leftarrow 0$    $MC_{mc.c} \leftarrow MC_{mc.c} \cup mc$ **end while** $nbrARB \leftarrow value$ **while**  $|AB| \leq nbrARB$  **do**  Choose randomly an  $ag \in AG$    $ab \leftarrow ag$    $ab.w \leftarrow 0$    $AB_{ab.c} \leftarrow AB_{ab.c} \cup ab$ **end while**Create a KD-Tree with the initial  $MC$ 

### 3.3.2 Memory cell identification and ARB generation

In this stage, we distinguish two essential steps: Memory cell identification and ARB generation.

In the first step, the memory cell the most stimulated with the antigen  $ag$  and which has

the same class of the latter is chosen as  $mc_{match}$  (see the Equation 3.3).

$$mc_{match} = \operatorname{argmax}_{mc \in MC_{ag,c}} Stimulation(ag, mc) \quad (3.3)$$

Where:

$$Stimulation(ag, mc) = 1 - Affinity(ag, mc) \quad (3.4)$$

If  $MC_{ag,c}$  is empty in this case  $mc_{match} \leftarrow ag$  and  $MC_{ag,c} \leftarrow MC_{ag,c} \cup ag$ .

---

**Algorithm 3** Memory cell identification
 

---

**Require:**  $ag$

**Ensure:**  $mc_{match}$

**if**  $MC_{ag,c} = \emptyset$  **then**

$mc_{match} \leftarrow ag$

$MC_{ag,c} \leftarrow MC_{ag,c} \cup ag$

**else**

$maxStim \leftarrow -1$

**for**  $j = 1$  to  $|MC_{ag,c}|$  **do**

$valStim \leftarrow stimulation(ag, mc_j)$

**if**  $valStim > maxStim$  **then**

$mc_{match} \leftarrow mc_j$

$maxStim \leftarrow valStim$

**end if**

**end for**

**end if**

---

The structure of  $MC$  set as it proposed in the original version of AIRS requires sequential search at each iteration to find the  $mc_{match}$ , and it requires powerful machines for the enormous datasets since the cost of calculation becomes enormous. To avoid this problem, we propose to reformulate the  $MC$  set as a binary search tree (KD-Tree) where their leaves are bags of  $mc$  the most similar.

The capacity of bags is fixed empirically from the user. If the bag is saturated, it explodes into two new bags, and we choose two  $mc$  the most dissimilar as two representatives for these new bags (left node, right node). The search process on the  $mc_{match}$  is carried out in one direction of the tree, and according to the distance of the  $ag$  from the left node and right node, we can identify the bag that contains the  $mc_{match}$ . Once the bag of  $mc_{match}$  is determined, we proceed on a local search on the  $mc$  the most similar to the current  $ag$  and which have the same class of this  $ag$ . If there is no  $mc$  which has the same class of the  $ag$ , we choose the current  $ag$  as  $mc_{match}$ . This structure allows us to decrease the research complexity on the  $mc_{match}$  from the exponential to logarithmic.

**Notion**

**kd-Tree:** Is a partitioning structure for a known data space that has  $k$  dimensions into sub-regions contains similar elements [ZHWG08]. This structure represents a bi-

nary search tree whose leaves are bags of similar data. Each bag has a capacity; if the bag is full, it splits into two new bags where the data within the bag is similar as possible. The search process took one direction, and it leads according to the distance from the left and right nodes.

---

**Algorithm 4** Proposed Memory cell identification
 

---

**Require:**  $ag, MC$

**Ensure:**  $mc_{match}$

$maxStim \leftarrow -1$

$Find \leftarrow False$

**if**  $MC = null$  **then**

$mc_{match} \leftarrow ag$

$MC.bag \leftarrow MC.bag \cup ag$

**else**

**if**  $(MC.left\_node = null)$  and  $(MC.right\_node = null)$  **then**

$Update\_Weight\_DL(MC.bag, ag)$

$Remove\_mc\_Inactif(MC.bag)$

**for**  $j = 1$  to  $|MC.bag|$  **do**

**if**  $ag.c = mc_j.c$  **then**

$Find \leftarrow True$

$valStim \leftarrow stimulation(ag, mc_j)$

**if**  $valStim > maxStim$  **then**

$mc_{match} \leftarrow mc_j$

$maxStim \leftarrow valStim$

**end if**

**end if**

**end for**

**if**  $Find = False$  **then**

$mc_{match} \leftarrow ag$

$MC.bag \leftarrow MC.bag \cup ag$

**end if**

**else**

**if**  $distance(ag, MC.left\_node) \leq distance(ag, MC.right\_node)$  **then**

$mc_{match} \leftarrow Proposed\ Memory\ cell\ identification(ag, MC.left\_bag)$

**else**

$mc_{match} \leftarrow Proposed\ Memory\ cell\ identification(ag, MC.right\_bag)$

**end if**

**end if**

**end if**

---

In the  $MC$  set, we can find inactive  $mc$ , which have a low participation rate to the antigens  $ag$ . These  $mc$  can influence the quality of the obtained results, and they can cause an explosion if we use large datasets. Inspired from the natural immune system that removes the inactive memory cells, we have introduced two functions **Update\_Weight\_DL** ( $MC.bag, ag$ ) and **Remove\_mc\_Inactive** ( $MC.bag$ ) which respectively allow to up-

date  $mc$  parameters (weight and lifetime) and remove the inactive  $mc$  to avoid the explosion of the  $MC$  set.

Each memory cell has a timer that computes its lifetime  $mc.DL$  during the meeting with an antigen  $ag$ . All the  $mc$  that consume their lifetime and which are not active are removed from the pool of  $MC$  set, while the  $mc$  that consume their lifetime and which is very active has a new lifetime ( $mcDL = 0$ ). In our proposition, the activity rate is measured with the weight. **Update\_Weight\_DL**( $MC.bag, ag$ ) and **Remove\_mc\_Inactive** ( $MC.bag$ ) are applied at each iteration only within the bag that contains  $mc_{match}$ .

---

**Algorithm 5** Remove\_mc\_Inactive
 

---

**Require:**  $MC.bag$

**Ensure:**  $MC.bag$

```

for  $i = 1$  to  $|MC.bag|$  do
  if  $mc_i.DL \geq S\_DL$  then
    if  $mc_i.w \leq S\_activation$  then
       $MC.bag \leftarrow MC.bag - mc_i$ 
    end if
  else
     $mc_i.DL \leftarrow 0$ 
  end if
end for

```

---



---

**Algorithm 6** Update\_Weight\_DL
 

---

**Require:**  $MC.bag, ag$

**Ensure:**  $MC.bag$

```

for  $i = 1$  to  $|MC.bag|$  do
  if  $mc_i.c = ag.c$  then
     $mc_i.DL \leftarrow mc_i.DL + 1$ 
     $mc_i.w \leftarrow mc_i.w + \frac{stimulation(ag,mc_i)}{sum\_stimulation(MC.bag)}$ 
  end if
end for
for  $i = 1$  to  $|MC.bag|$  do
  if  $mc_i.c = ag.c$  then
    Normalized the weight of  $mc_i$  by:
     $mc_i.w \leftarrow \frac{mc_i.w}{sum\_weigh(MC.bag)}$ 
  end if
end for

```

---

In the second step (ARB generation), AIRS generates new  $ARB$  clones using  $mc_{match}$ . Indeed, it creates  $NumClones$  of  $ARB$  by the mutation of  $mc_{match}$  with a rate between 0 and 1. The number of clones is calculated with:

$$NumClones = hyper\_mutation\_rate \times clonal\_rate \times stimulation(ag, mc_{match}) \quad (3.5)$$

Where  $hyper\_mutation\_rate$  is an integer value fixed empirically.



**Algorithm 7** ARB generation**Require:**  $ag, mc_{match}, hyper\_mutation\_rate, clonal\_rate$ **Ensure:**  $AB$ **Variables****Begin** $MU$ : The mutated  $ARB$  clones. $mut$ : A boolean value**End** $MU \leftarrow \emptyset$  $MU \leftarrow MU \cup makeARB(mc_{match})$  $stim \leftarrow stimulation(ag, mc_{match})$  $NumClones \leftarrow hyper\_mutation\_rate \times clonal\_rate \times stim$ **while**  $|MU| < NumClones$  **do** $mut \leftarrow false$  $mc_{clone} \leftarrow mc_{match}$  $mc_{clone}, mut \leftarrow mutation(mc_{clone}, mut)$ **if**  $mut = true$  **then** $MU \leftarrow MU \cup makeARB(mc_{clone})$ **end if****end while** $AB \leftarrow AB \cup MU$ 

The function  $makeARB(x)$  returns an  $ARB$  which  $x$  is the antibody to this  $ARB$ .

The mutation function used is defined as follow:

**Algorithm 8** Mutation function.**Require:**  $x, b, mutation\_rate$ **Ensure:**  $x, b$ **Variables****Begin** $drandom()$ : A function that returns a random number between 0 and 1. $lrandom()$ : A function that returns a random value between 0 and  $nc$ **End****for each**  $x.f_i \in x.f$  **do** $Change \leftarrow drandom()$  $Changeto \leftarrow drandom()$ **if**  $Change < mutation\_rate$  **then** $x.f_i \leftarrow Changeto \times normalization\_value$  $b \leftarrow True$ **end if****end for****if**  $b = True$  **then** $Change \leftarrow drandom()$  $Changeto \leftarrow (lrandom() \bmod nc)$ **if**  $Change < mutation\_rate$  **then** $x.c \leftarrow Changeto$ **end if****end if**

With:

$drandom()$ : a function returns a random number between 0 and 1.

$lrandom()$ : a function returns an integer between 0 and  $nc$ .

We have kept the same principle of the ARB generation process, but we introduce a small modification, we add weight to each new clone generated according to their stimulation with the current antigen.

This modification increases the opportunity for the best clones to be selected as a candidate memory cell.

---

**Algorithm 9** Proposed ARB generation

---

**Require:**  $mc_{match}$ ,  $ag$ ,  $hyper\_mutation\_rate$ ,  $clonal\_rate$

**Ensure:**  $ARB(AB)$

**Variables**

**Begin**

$MU$ : The mutated  $ARB$  clones.

$mut$ : A boolean value

**End**

$MU \leftarrow \emptyset$

$MU \leftarrow MU \cup makeARB(mc_{match})$

$stim \leftarrow stimulation(ag, mc_{match})$

$NumClones \leftarrow hyper\_mutation\_rate \times clonal\_rate \times stim$

**while**  $|MU| < NumClones$  **do**

$mut \leftarrow false$

$mc_{clone} \leftarrow mc_{match}$

$mc_{clone}, mut \leftarrow mutation(mc_{clone}, mut)$

**if**  $mut = true$  **then**

$mc_{clone}.w \leftarrow stimulation(ag, mc_{clone})$

$MU \leftarrow MU \cup makeARB(mc_{clone})$

**end if**

**end while**

$AB \leftarrow AB \cup MU$

---

In the old mutation function, if we attribute a random class to the new clones, we increase the false classifications.

To overcome this problem, we have labeled each new clone with the class of the nearest mean vector generated in the initialization stage.

For more details, the algorithm below explains our proposed mutation function.

**Algorithm 10** Proposed mutation function**Require:**  $x, b, mutation\_rate$ **Ensure:**  $x, b$ **Variables****Begin** $drandom()$ : A function that returns a random number between 0 and 1.**End****for each**  $x.f_i \in x.f$  **do**     $Change \leftarrow drandom()$      $Changeto \leftarrow drandom()$     **if**  $Change < mutation\_rate$  **then**         $x.f_i \leftarrow Changeto \times normalization\_value$          $b \leftarrow True$     **end if****end for****if**  $b = True$  **then**     $Change \leftarrow drandom()$     **if**  $Change < mutation\_rate$  **then**         $x.c \leftarrow argmin_{c_i \in \{1, 2, \dots, nc\}} distance(x, mean(AG.c_i))$     **end if****end if**

All the  $ARBs$  generated during the previous iterations,  $mc_{match}$  and the new clones created with this process are grouped into classes in the  $AB$  set.

### 3.3.3 Competition for resources and development of a candidate memory cell

$AB$  set in this stage will be filtered to keep only the best  $ab$  which are the most successful and correctly in classifying a given antigen. To ensure this objective, authors of [WB02] proposed a mechanism for organizing the survival of individuals within the  $ab$  populations. This mechanism adopts a technique of sharing the cumulative resources according to the nature of the antigen class.  $ab$  which have the same class of the antigen  $ag$  get half of the cumulative resources, while the remaining resources are divided to  $ab$ , which have a class different from the antigen  $ag$ .

Each class has a maximum resource allocation. If the sum of the resources for a class exceeds their maximum allocation, the additional resources will be removed from the least stimulated  $ab$  of this class.  $ab$  that have the number of resources less than or equal to the number of resources removed will be deleted from the pool of  $AB$ .

For more details, see the following algorithm.

**Algorithm 11** Stimulation, resource allocation, and ARB removal**Require:**  $AB$ **Ensure:**  $AB$ **Variables****Begin** $MAX, MIN$ : Two real numbers. $TotalNumResources$ : The number of resources in the system.**End** $minStim \leftarrow MAX$  $maxStim \leftarrow MIN$ **for each**  $ab \in AB$  **do** $stim \leftarrow stimulation(ag, ab)$ **if**  $stim < minStim$  **then** $minStim \leftarrow stim$ **end if****if**  $stim > maxStim$  **then** $maxStim \leftarrow stim$ **end if** $ab.stim \leftarrow stim$ **end for****for each**  $ab \in AB$  **do****if**  $ab.c = ag.c$  **then** $ab.stim \leftarrow \frac{ab.stim - minStim}{maxStim - minStim}$ **else** $ab.stim \leftarrow 1 - \frac{ab.stim - minStim}{maxStim - minStim}$ **end if** $ab.resources \leftarrow ab.stim \times clonal\_rate$ **end for** $i \leftarrow 1$ **while**  $i < nc$  **do** $resAlloc \leftarrow \sum_{j=1}^{|AB_i|} ab_j.resources, ab_j \in AB_i$ **if**  $i = ag.c$  **then** $numResAllowed \leftarrow \frac{TotalNumResources}{2}$ **else** $numResAllowed \leftarrow \frac{TotalNumResources}{2 \times (nc - 1)}$ **end if****while**  $resAlloc > numResAllowed$  **do** $numResRemoved \leftarrow resAlloc - numResAllowed$  $ab_{removed} \leftarrow argmin_{ab \in AB_i} (ab.stim)$ **if**  $ab_{removed}.resources \leq numResRemoved$  **then** $AB_i \leftarrow AB_i - ab_{removed}$  $resAlloc \leftarrow resAlloc - ab_{removed}.resources$ **else** $ab_{removed}.resources \leftarrow ab_{removed}.resources - numResRemoved$  $resAlloc \leftarrow resAlloc - numResRemoved$ **end if****end while** $i \leftarrow i + 1$ **end while**

Once this process finishes, a stop criterion is calculated to check if the stimulation of the  $ab$  with the  $ag$  antigen is sufficient or not.

Let  $\vec{S}$  be a vector of size  $nc$  containing the average stimulation of each  $AB$  class. The stopping criterion is achieved, if and only if  $s_i \geq stimulation\_threshold$  for all elements of  $\vec{S}$ .

$$S_i = \frac{\sum_{j=1}^{|AB_i|} ab_j.stim}{|AB_i|}, ab_j \in AB_i \quad (3.6)$$

During this time, each  $ab \in AB$  has a chance to produce mutated offspring (see Mutation of surviving ARB function) for increasing the diversification of members within the  $AB$  set.

---

**Algorithm 12** Mutation of surviving ARB function
 

---

**Require:**  $AB$

**Ensure:**  $AB$

**Variables**

**Begin**

$MU$ : A set of ARB clones which is mutated.

$mut$ : A boolean value indicates if the mutation operation produces a new element or not.

**End**

$MU \leftarrow \emptyset$

**for do**

$rd \leftarrow drandom()$

**if**  $ab.stim > rd$  **then**

$NumClones \leftarrow ab.stim \times clonal\_rate$

$i \leftarrow 1$

**while**  $i \leq NumClones$  **do**

$mut \leftarrow false$

$ab_{clone} \leftarrow ab$

$ab_{clone} \leftarrow mutation(ab_{clone}, mut)$

**if**  $mut = True$  **then**

$MU \leftarrow MU \cup ab_{clone}$

**end if**

$i \leftarrow i + 1$

**end while**

**end if**

**end for**

$AB \leftarrow AB \cup MU$

---

In the mutation of surviving ARB function, we have used our proposed mutation function, which is described in the subsection (Memory cell identification and  $ARB$  generation), instead to use the traditional mutation function of the basic AIRS.

Once the mutation of the  $ARBs$  finishes, the stop criterion is tested again. If the stop criterion is achieved,  $ab$  the most stimulated with the current antigen is selected as a candidate memory cell ( $mc_{candidate}$ ). When the stop criterion is not achieved, the processes

of this stage will be repeated until the stop criterion is fulfilled. Except in the first passage, the mutation of the  $ARBs$  is not performed when the stopping criterion is reached after the processing of **Stimulation, resource allocation, and ARB removal algorithm** stages.

**Competition for resources and development of a candidate memory cell** process used to avoid the explosion in the  $AB$  set. We have kept the same principle of this process, but we replaced the notion of resources in the **Stimulation, resource allocation, and ARB removal** step with the weights.

The selection of one candidate memory cell may abuse the other  $ab$ , since all the  $ab$  generated after the process of **stimulation, resource allocation, and ARB removal** are also best representatives for the antigen  $ag$ . For this purpose, we have proposed to take the  $AB$  set generated after this process as a set of candidate memory cells ( $MC_{candidates}$ ), instead of choosing only the best  $ab$  like, it indicated in the original version of the AIRS. This improvement increases the probability of introducing good memory cells that give correct classification models to the  $ag$ .

### 3.3.4 Memory cell introduction

This is the last stage in the training of one antigen. The candidate memory cell ( $mc_{candidate}$ ) that has a stimulation with the antigen  $ag$  higher than that of  $mc_{match}$ , will be introduced to the  $MC$  set as a new memory cell. If the affinity between  $mc_{match}$  and  $mc_{candidate}$  is less than  $AT \times ATS$ , the  $mc_{candidate}$  replaces  $mc_{match}$  in  $MC$  set. Once this stage is finished, we proceed to learn another antigen from the training set: we start with the **Memory cell identification and ARB generation stage** to **Memory cell introduction stage**. The training process ends when all antigens of the training set are presented to the system.

---

#### Algorithm 13 Memory cell introduction

---

**Require:**  $mc_{candidate}, ag, AT, ATS, mc_{match}$

**Ensure:**  $MC$

$CandStim \leftarrow stimulation(ag, mc_{candidate})$

$MatchStim \leftarrow stimulation(ag, mc_{match})$

$CellAff \leftarrow affinity(mc_{candidate}, mc_{match})$

**if**  $CandStim > MatchStim$  **then**

**if**  $CellAff < AT \times ATS$  **then**

$MC \leftarrow MC - mc_{match}$

**end if**

$MC \leftarrow MC \cup mc_{candidate}$

**end if**

---

In this stage, we have introduced a small modification concerning the number of memory cells introduced. Indeed, AIRS like it defined, allows to introduce one memory cell; in

our proposition, all the candidate memory cells that have stimulation with the antigen  $ag$  higher than that of  $mc_{match}$  are introduced as new memory cells. This modification gives a high opportunity for the system to produce more representative models.

For each  $mc$  introduced, we create a lifetime counter and we initialize it with 0.

---

**Algorithm 14** Proposed memory cells introduction
 

---

**Require:**  $MC_{candidate}, ag, AT, ATS, mc_{match}$

**Ensure:**  $MC$

**Variables**

**Begin**

*Find*: A boolean value

**End**

*Find*  $\leftarrow$  *False*

*MatchStim*  $\leftarrow$  *stimulation*(*ag*, *mc<sub>match</sub>*)

**for each**  $mc_{candidate} \in MC_{candidate}$  **do**

*CandStim*  $\leftarrow$  *stimulation*(*ag*, *mc<sub>candidate</sub>*)

*CellAff*  $\leftarrow$  *affinity*(*mc<sub>candidate</sub>*, *mc<sub>match</sub>*)

**if** *CandStim* > *MatchStim* **then**

**if** (*CellAff* <  $AT \times ATS$ ) and (*Find* = *False*) **then**

Remove *mc<sub>match</sub>* from the *MC* tree

*Find*  $\leftarrow$  *True*

**end if**

*mc<sub>candidate</sub>.DL*  $\leftarrow$  0

Add *mc<sub>candidate</sub>* to *MC* tree

**end if**

**end for**

---

### 3.3.5 Classification

After the training stage, the *MC* set is elaborated for the classification. The classification is performed with  $K$  Nearest Neighbor classifier (*KNN*) [CH67] according to the majority vote of the  $K$  memory cells that are closest to the input vector.

Once the  $K$  neighbors are identified, the input vector will be labeled by the class of the majority of the  $K$  neighbors.

The search on the  $K$  nearest neighbors requires browsing all the elements of the classification model at each time, which increases the search cost.

The structure of *MC* set that we have proposed (kd-Tree) avoids this type of search since the search on the  $K$  nearest neighbors takes one direction of the tree to perform a local search on the  $K$  nearest neighbors in one bucket of the *MC* tree.

## 3.4 Tests and results

In this section, we will present the results of our proposed AIRS on some public datasets taken from the Irvine automatic learning repository of the University of California (UCI) [Win18].

We have implemented our system in Python environment on an i7 PC with a memory of 8 GB.

The results shown in this chapter represent the average of several cross-validation tests. Cross-validation folders varied from 5 to 13 folders according to the nature of the dataset. As an exceptional case, in the Ionosphere dataset 200 and 151 samples are chosen respectively for the training and the tests in several executions.

The obtained results of our proposed AIRS compare with: the basic AIRS [WB02], an extension proposed by Watkins et al. [WTB04] (AIRS2) and with other public machine learning methods like RBF, C4.5 decision trees, SVM and MLP+BP, etc. To evaluate the efficiency of our improvements, we have used the **Classification Accuracy** as a performance measure.

The parameters of the AIRS: initial number of memory cells and ARBs, stimulation threshold, *mutation\_rate*, *clonal\_rate*, *hyper\_mutation\_rate*, Affinity Threshold Scalar (*ATS*), *S\_DL* and the *K* of the *K* Nearest Neighbors are fixed after several empirically tests respectively to: 1, between 0.8 and 0.9, 0.1, 10, 2, 0.2, 50, between 1 and 7. For *S\_activation* we have chosen between 0.5 and 0.7.

Concerning the size of the bag, we chose to vary it from 2 to 30 according to the size of the dataset used.

### 3.4.1 Datasets

Before presenting the obtained results, we will give a breve description of the used datasets in the following.

- **Iris Dataset:** this dataset contains three classes that indicate the type of iris plant (Iris Setosa, Iris Versicolour and Iris Virginica). Each class has 50 instances and four quantitative attributes of real type (sepal length, sepal width, petal length, petal width).
- **Pima-Diabetes:** Pima-Diabetes dataset describes the medical records for 768 Pima Indians. From certain diagnostic measurements, we can predict whether or not passion has diabetes. Eight diagnostics are taken into consideration (see table 3.1).



Table 3.1: Description of the attributes used in Pima-Diabetes dataset

Attribute	Description
preg	Number of times pregnant
plas	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
pres	Diastolic blood pressure (mm Hg)
skin	Triceps skin fold thickness (mm)
test	2-Hour serum insulin ( $\mu$ U/ml)
mass	Body mass index ( $weightinkg/(heightinm)^2$ )
pedi	Diabetes pedigree function
age	Age (years)

In this dataset, we distinguish two classes (1: tested positive for diabetes, 0: tested negative for diabetes)

- **Ionosphere:** data of this base was collected from a radar system in Goose Bay, Labrador. The main objective of this dataset is to describe if the returns of radar are "good" or "bad".

A "Good" radar returns are those which show the presence of certain types of structure in the ionosphere. Otherwise, the radar returns are bad. Thirty-four attributes are used from this dataset to determine the nature of radar returns.

- **Sonar:** Sonar dataset contains signal values obtained from various aspect angles, spanning 90 degrees for cylinder and 180 degrees for rock. It is composed of 208 patterns in which 60 attributes between 0.0 and 1.0 describe each instance. Each number represents the energy within a particular frequency band, integrated over a certain period.

In this dataset, we distinguish two labels: the letter "R" which is used if the object is rock and the letter "M" which is used if the object is mine (metal cylinder).

- **Heart-Statlog:** it is a binary classification dataset that describes the presence of heart disease for the patients. It has 270 instances and 13 attributes (Age, sex, chest pain type, resting blood pressure, serum cholesterol in mg/dl, fasting blood sugar > 120 mg/dl, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, oldpeak, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy and thal) used to distinguish presence (value 1) from absence (value 2) of heart disease.
- **Hepatitis:** this dataset contains 155 instances and 19 attributes (described in Table 3.2). Their objective consists of detecting hepatitis in people. All the data of this

base were donated by Gail Gong. Noted that this dataset contains a large number of missing values.

Table 3.2: Description of Hepatitis attributes

Attribute	Domain
AGE	10, 20, 30, 40, 50, 60, 70, 80
SEX	male, female
STEROID	no, yes
ANTIVIRALS	no, yes
FATIGUE	no, yes
MALAISE	no, yes
ANOREXIA	no, yes
LIVER BIG	no, yes
LIVER FIRM	no, yes
SPLEEN PALPABLE	no, yes
SPIDERS	no, yes
ASCITES	no, yes
VARICES	no, yes
BILIRUBIN	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
ALK PHOSPHATE	33, 80, 120, 160, 200, 250
SGOT	13, 100, 200, 300, 400, 500
ALBUMIN	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
PROTIME	10, 20, 30, 40, 50, 60, 70, 80, 90
HISTOLOGY	no, yes

- **Wisconsin-Breast Cancer:** this dataset obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. Data of this base are collected periodically as Dr. Wolberg reports his clinical cases. In July 1992, an original Breast Cancer Wisconsin dataset was constructed of 699 instances in a specific interval (see Table 3.3) and ten attributes that describe two classes: benign or malignant. Label 2 is given for the benign classes, while label 4 indicates the malignant class. Noted that, in this dataset, we find 16 missing attribute values.

Table 3.3: Description of Wisconsin-Breast Cancer attributes

Attribute	Domain
Sample code number	id number
Clump Thickness	1-10
Uniformity of Cell Size	1-10
Uniformity of Cell Shape	1-10
Marginal Adhesion	1-10
Single Epithelial Cell Size	1-10
Bare Nuclei	1-10
Bland Chromatin	1-10
Normal Nucleoli	1-10
Mitoses	1-10

Table 3.4 gives a breve description of the used datasets.

Table 3.4: Description of the used datasets

Datasets	Size	# attributs	Attribute Domains	Missing values	# classes
Iris	150	4	Real	No	3
Pima-Diabetes	768	8	Integer, Real	No	2
Ionosphere	351	34	Integer, Real	No	2
Sonar	208	60	Real	No	2
Heart-Statlog	270	13	Categorical, Real	No	2
Hepatitis	155	19	Categorical, Integer, Real	Yes	2
Wisconsin-Breast Cancer	699	10	Integer	Yes	2

### 3.4.2 Results

Tables 3.5, 3.6, , 3.7, 3.8 and Figure 3.2 show the obtained results of our Enhanced Artificial Recognition System (EAIRS) on the datasets: Iris, Pima-Diabetes, Inosphere, Sonar, Heart-Statlog, Hepatitis, Wisconsin-Breast Cancer.

Table 3.5: Our obtained results on the datasets: Iris, Pima-Diabetes, Inosphere, Sonar, Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer

Datasets	Validation methods	Accuracy
Iris	Cross validation 5 folders	96.22%
Pima-Diabetes	Cross validation 10 folders	74.55% (all features)
		75.27% (with selection of features)
Ionosphere	200 : Training, 151 : Test	96.69%
Sonar	Cross validation 13 folders	84.98%
Heart-Statlog	Cross validation 10 folders	83.82%
Hepatitis	Cross validation 10 folders	88.50%
Wisconsin-Breast Cancer	Cross validation 10 folders	97.38%

Table 3.6: Classification accuracy of EAIRS, AIRS [WB02], AIRS2 [WTB04] on the datasets: Iris, Pima-Diabetes, Inosphere and Sonar

Datasets	AIRS [WB02]	AIRS2 Accuracy [WTB04]	EAIRS Accuracy
Iris	96.7%	96.0%	<b>96.22%</b>
Pima-Diabetes	74.1%	74.2%	<b>74.55%</b>
Ionosphere	94.9%	95.6%	<b>96.69%</b>
Sonar	84.0%	84.9%	<b>84.98%</b>

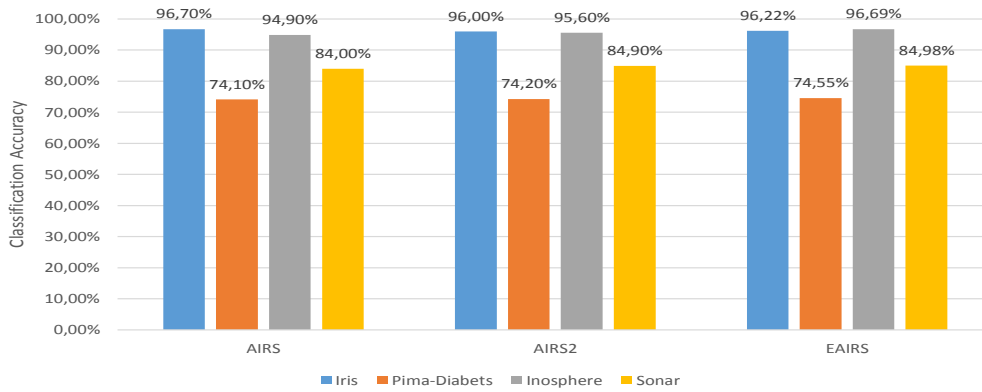


Figure 3.2: Histogram of the classification accuracy of AIRS [WB02], AIRS2 [WTB04] and EAIRS on the datasets: Iris, Pima-Diabetes, Inosphere and Sonar

Table 3.7: Comparison of EAIRS results with other methods on: Iris, Pima-Diabetes, Inosphere and Sonar datasets results are obtained from [Duc00b], [Duc00a]

Iris		Pima-Diabetes		Ionosphere		Sonar	
Grobian (rough)	100%	Logdisc	77.77%	3-NN + simplex	98.7%	TAP MFT Bayesian	92.3%
SSV	98.0%	IncNet	77.6%	3-NN	96.7%	Naive MFT Bayesian	90.4%
C-MLP2LN	98.0%	DIPOL92	77.6%	IB3	96.7%	SVM	90.4%
PVM 2 rules	98.0%	Linear Disc. Anal.	77.5-77.2%	<b>EAIRS</b>	<b>96.69%</b>	Best 2-layer MLP + BP, 12 hidden	90.4%
PVM 1 rule	97.3%	SMART	76.8%	MLP + BP	96.0%	<b>EAIRS</b>	<b>84.98%</b>
FuNe-I	96.7%	GTO DT (5 × CV)	76.8%	AIRS2	95.6%	AIRS2	84.9%
NEFCLASS	96.7%	ASI	76.6%	AIRS1	94.9%	MLP + BP, 12 hidden	84.7%
AIRS1	96.7%	Fischer discr. anal.	76.5%	C4.5	94.9%	MLP + BP, 24 hidden	84.5%
<b>EAIRS</b>	<b>96.22%</b>	MLP + BP	76.4%	RIAC	94.6%	1-NN, Manhattan	84.2%
AIRS2	96.0%	LVQ	75.8%	SVM	93.2%	AIRS1	84.0%
CART	96.0%	LFC	75.8%	FSM + rotation	92.8%	FSM	83.6%
FUNN	95.7%	RBF	75.7%	Non-linear perceptron	92.0%	MLP + BP, 6 hidden	83.5%
		NB	75.5-73.8%	1-NN	92.1%	1-NN Euclidean	82.2%
		KNN, K = 22, Manh	75.5%	DB-CART	91.3%	DB-CART, (10 × CV)	81.8%
		MML	75.5%	Linear perceptron	90.7%	CART, (10 × CV)	67.9%
		SNB	75.4%				
		<b>EAIRS (with selection of features)</b>	<b>75.27%</b>				
		BP	75.2%				
		SSV DT	75% ± 3.6				
		KNN, K = 18, Euclid, raw	74.8% ± 4.8				
		CART DT	74.7% ± 5.4				
		<b>EAIRS (all features)</b>	<b>74.55%</b>				
		DB-CART	74.4%				
		ASR	74.3%				
		AIRS2	74.2%				
		AIRS1	74.1%				
		SSV DT	73.7% ± 4.7				
		C4.5 DT	73.0%				
		Bayes	72.2% ± 6.9				
		CART	72.8%				
		Kohonen	72.7%				
		ID3	71.7% ± 6.6				
		IB3	71.7% ± 5.0				
		IB1	70.4% ± 6.2				
		C4.5 rules	67.0% ± 2.9				
		OCN2	65.1% ± 1.1				
		QDA	59.5%				

Table 3.8: Comparison of the obtained results of EAIRS with other methods on: Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer datasets

Heart-Statlog		Hepatitis		Wisconsin-Breast Cancer	
MLMNB [HA18]	90.6%	Back propagation neural networks [MS15]	94.0%	GA with feature chromosome [ZFJ <sup>+</sup> 11]	99.0%
WLS-TSVM + Polynomial Kernel Function [TSA14]	86.47%	LSSVM with IACA algorithm [HAR <sup>+</sup> 17]	93.7%	PTVPSO [lCYjW <sup>+</sup> 14]	98.44%
Neural network for threshold selection [JK13]	85.19%	Logistic Regression [YJD11]	89.6%	CBFDT [CFD10]	98.4%
LVQ [BBJO <sup>+</sup> 18]	85.07%	<b>EAIRS</b>	<b>88.5%</b>	SVM [CTY <sup>+</sup> 10]	97.6%
Half selection method [JK13]	84.81%	A Radial Basis Function Network (RBFN) [AO17]	88.04%	<b>EAIRS</b>	<b>97.38%</b>
WLS-TSVM + Gaussian Kernel Function [TSA14]	84.62%	SDD,SVD and Multilayer Perceptron [HOOS16]	87.7%	k-means and SVM [ZYL14]	97.38%
Mean selection method [JK13]	84.44%	FT Tree [BAH13]	87.1%	Baseline [KR11]	97.37%
<b>EAIRS</b>	<b>83.82%</b>	WLS-TSVM + Gaussian Kernel Function [TSA14]	86.67%	Neural network for threshold selection [JK13]	97.28%
NB [JLWZ16]	83.74%	WLS-TSVM + Polynomial Kernel Function [TSA14]	85.61%	SVM and evolutionary algorithm [SS13]	97.23%
DSOC GA [BBJO <sup>+</sup> 18]	83.58%	Decision Table [HEOI17]	85.3%	AIRS [GBW02]	97.2%
DFWNB [JLWZ16]	83.33%	NaiveBayes Updatable [KT13]	84.0%	K2MLP [DJM12]	97.0%
AODE [HA18]	83.1%	LMT [BAH13]	83.6%	Half selection method [JK13]	96.71%
NBTree [HA18]	82.5%	KStar [BAH13]	83.4%	Density weighted SVDD [CKB14]	96.2%
RandomForest [JLWZ16]	82.30%	CART Algorithm [Sat11]	83.2%	MI with k-NN [SDLH14]	96.14%
Fuzzy Boosting [ZL10]	82.28%	Randomforest [KT13]	83.0%	Mean selection method [JK13]	95.99%
WAODE [HA18]	82.0%	J48 [KT13]	83.0%	BK with IVFS [LC15]	95.26%
HNB [HA18]	81.9%	Support vector Machine (5×CV) [KKGK14]	83.0%	AFS [LR10]	94.6%
DecisionTable [KT13]	81.78%	IBK [HEOI17]	81.5%	Differential evolution classifier [KLL13]	93.64%
TAN [HA18]	80.5%	C.5 [HEOI17]	74.4%	Tree-based topology-oriented SOM [AO13]	93.32%
AdaBoost-kNN [GG10]	80.0%	C4.5 Algorithm [Sat11]	71.4%	SVM CPBK [LL10]	93.26%
AdaBoost-PCA [HK12]	74.33%	Neural network [BAH13]	70.4%	Naive Bayes [TMA14]	92.42%
AdaBoost-SVM [CXW11]	72.24%	ID3 Algorithm [Sat11]	64.8%		

### 3.4.3 Discussions

From the obtained results, we can see that our improved AIRS has achieved a good rank in most of the datasets cited above. We have surpassed the basic AIRS and AIRS2 in Pima-Diabetes, Ionosphere and Sonar datasets. Compared with the other methods, we have reached good accuracy rate in Ionosphere and Sonar datasets. However, in Pima-Diabetes and Iris datasets, we have acceptable results due to the variability of the attribute values in Pima-Diabetes dataset and the unbalanced distribution of classes in Iris dataset.

For the datasets: Heart-Statlog, Hepatitis and Wisconsin-Breast Cancer, our AIRS has also achieved good results compared with many recent methods. It is always in the 8 top methods.

Noted that our results are the average of many executions, and we have used cross-

validation to split the datasets, unlike many methods that take only the best solution or split the datasets manually and select good features. All these factors contribute to increasing the accuracy rate of some methods.

The use of the cross-validation method is equitable since it allows diversification in the training and the testing sets. Indeed, the cross-validation method divided the dataset into  $k$  folders; after that, it selected at each time one sample as a test set and the other  $k - 1$  samples as the training sets.

The research on  $mc_{match}$  and the  $K$  neighbors of the  $KNN$  algorithm are improved with the new structure of  $MC$  set (it is redefined as kd-tree). Indeed we have reduced the search complexity from the sequential in the basic version of AIRS to the logarithmic using the kd-tree structure.

### 3.5 Conclusion

AIRS has been realized great success to solve optimization and classification problems. However, AIRS suffers from some drawbacks that influence their research cost and results quality, which can cause an explosion of data in the  $MC$  set.

To solve these problems, we have discussed in this chapter the different stages of the AIRS proposed by [WB02], and we have suggested some modifications in their functionalities like the mutation function, memory cell introduction mechanism.

For reducing the research cost on the  $mc_{match}$  we have proposed to change the structure of  $MC$  set as a binary tree (kd-tree). In addition to these modifications, we have introduced two new mechanisms in the original version of AIRS to avoid the explosion of data in  $MC$  set.

The obtained results show the effectiveness of our modifications compared with the original version of AIRS, an extension of AIRS and other recent methods.

In the following chapter, we will adopt the AIRS as a background subtraction method that classifies pixels into background and foreground pixels.

# 4

## Background subtraction using Artificial Immune Recognition System

### 4.1 Introduction

Background subtraction, also called motion detection, is a complicated task that requires a robust method to distinguish moving objects from static objects. Several methods have been proposed to improve the quality of background subtraction methods while minimizing false detections.

Among methods that have achieved a great success: Single Gaussian (SG) and Gaussian Mixture Model (GMM). However, the success of these methods is limited, and they impose many problems that we will in this chapter detail and treat using the Artificial Immune Recognition System (AIRS).

The proposed systems are implemented and tested on some public datasets: CDnet 2014 (Dynamic Background), Wallflower, WaterSurface, UCSD and Fountain and the obtained results are compared with recent methods to improve the effectiveness of these systems.

## 4.2 Background subtraction using Artificial Immune Recognition System and Single Gaussian (AIRS-SG)

The Single Gaussian method is sensitive to scenes that have a dynamic background. Indeed, one Gaussian is not sufficient to model all the variations of the pixel.

To create a robust background method for scenes containing a dynamic background using the SG method, we will adopt a bio-inspired approach to better represent the pixel states in dynamic scenes. The proposed approach benefits from a combination of Artificial Immune Recognition System (AIRS) and Single Gaussian method (SG). AIRS uses to separate the antibodies represented with the background pixels from the antigens that modeled the foreground pixels. The state of each pixel is represented with a Gaussian contains three variables: pixel value, the mean and the variance. The proposed system considered only pixels that represent the background and updated the background model for each background pixel.

### 4.2.1 Why AIRS and SG

The combination between AIRS and SG allows us to benefit from two advantages: the power of AIRS to affect an online update for system parameters and the ability of Gaussians to adapt into scene variations at the pixel level.

The choice of AIRS from many other classifiers is for three main purposes: the first one is related to the principle of the method, which is in accordance with the treated problem since the latter allows to recognize which is belongs to the body (in our case the background) from the foreign elements (in our case moving objects).

The second purpose is the type of video processed. Noted that we are interested in videos that contain a dynamic background. This type of video requires a method that is easily adapted to the frequent changes of the background. AIRS copes with these variations through cloning and mutation operations which allow creating multiple background templates using a small number of frames.

The ability to update online the system parameters is another purpose that gives an advantage to AIRS compared to other classifiers. Indeed, the online update allows the systems to more captured with possible changes of background without going through a re-initialization step which requires system shutdown during a long learning period (for example, growing tree, changing the color of a wall, etc).

### 4.2.2 The global architecture

In this subsection, we will present the global architecture of the proposed system. Our system presented a new background subtraction method of videos captured from a fixed camera and which have a dynamic background. The proposed system was performed from



a combination between AIRS and SG. AIRS uses as a classification tool that recognizes the antigens represented by the foreground pixels (the moving object pixels) from antibodies that model the background pixels. The state of each pixel is represented with a Gaussian. At each iteration, AIRS searches among the existing models the background model which their similarity nearest to the current pixel value. The best background model among the Gaussians is mutated with an empirical rate to predict values that can take this pixel. The created Gaussians are grouped into a set named clones. After that, a filtering process is applied on these clones to save only the best Gaussians representing the background well. The current pixel classifies as a background pixel if we find at the minimum one Gaussian, which represents well the state of this pixel. For more details, Figure 4.1 presents the global architecture of our system.

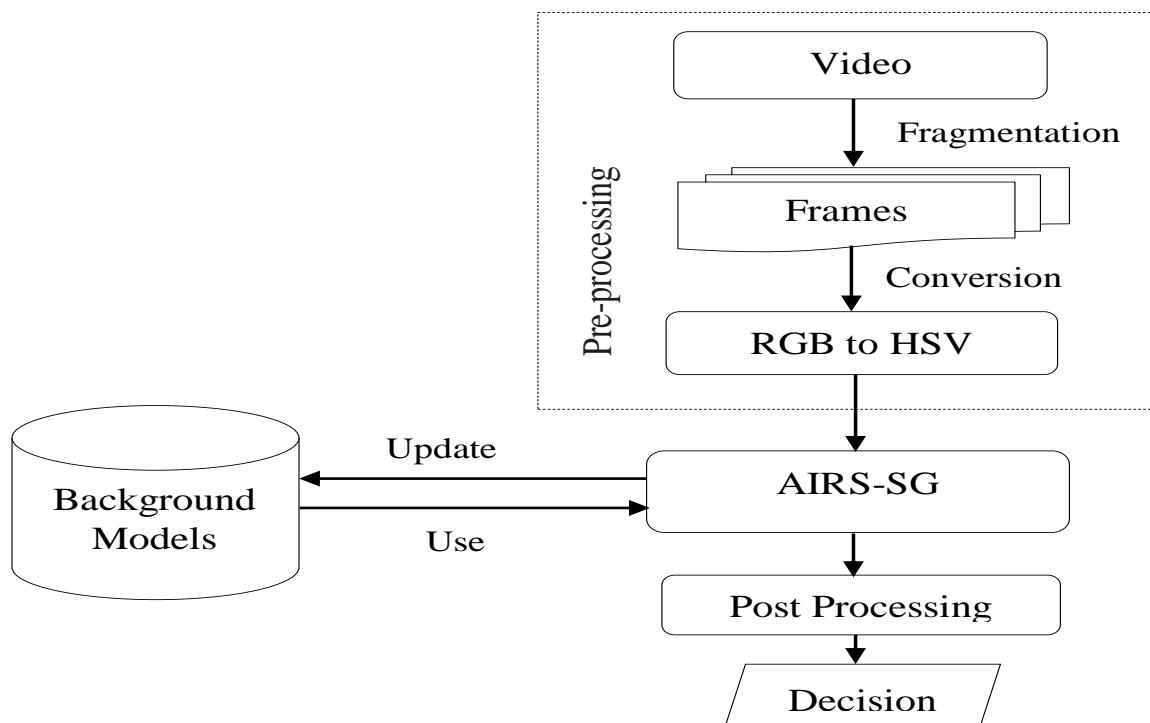


Figure 4.1: The global architecture of AIRS-SG

#### 4.2.2.1 Pre-processing

In this step, the video is split into sequences of 2D images called frames. These frames are the input of our system. Initially, the extracted frames are encoded into RGB mode. However, this mode is sensitive to the light effects [HGM10], and it does not allow the creation of an efficient model.

For this purpose, we have transformed the RGB frames to the HSV space color. The choice of HSV depends on their power to the change in the light, since it separates the luminosity which is represented with  $V$  component from the chromatic properties represented by  $H$  and  $S$  components.

In our proposition, we have chosen to take only the  $H$  component as the pixel value. Indeed, after many experiments, we deduce that the  $S$  component did not add a significant gain in the quality of the results compared to the gain obtained in the processing time by removing the latter. For the  $V$  component, it is ignored in several works to reduce the light affects [WM03], [ZBC02].

#### 4.2.2.2 AIRS-SG model

The fast variations of the dynamic background make the distinction between the background and the foreground pixels very difficult.

SG is among the most popular methods for background subtraction, but SG is sensitive to the dynamic background. Indeed one Gaussian cannot memorize all the states of the pixel.

In order to overcome this problem, we will propose a new approach for dynamic background subtraction called AIRS-SG. This approach exploits the principle of SG, and the robustness of the bio-inspired approach AIRS to create an efficient model to the fast pixel variations. In this model, the value of each pixel ( $H$ ) is considered an antigen, while the corresponding Gaussians are considered the memory cells that are recognized this latter. For more detail, the proposed method is described in four main stages detailed in the following.

##### 1. Initialization and the creation of the first AIRS-SG model

This is a primary step in any background subtraction method. It dedicates to the definition and the identification of parameters. In our case, we have used the SG method on the first frames of the video sequence to initialize the memory cells set ( $MC$ ). This step is important since it ensures a rapid convergence for the recognition system.

Concerning the  $ARB$  set, we preferred to keep empty and build it during the system operation. The affinity measurement as defined in the AIRS algorithm is not adequate in our context. For this reason, we have changed it with the distance between each pixel and their corresponding models (see Equation 4.1).

$$\frac{|P_t - u_t|}{\sigma_t} \quad (4.1)$$

About the affinity threshold, we have proceeded to calculate with the following Equation:

$$AT = \frac{\sum_{i=1}^n \sum_{j=1}^m \frac{P_{i,j} - u_{i,j}}{\sigma_{i,j}}}{n \times m} \quad (4.2)$$

Where,  $n$ ,  $m$  is the number of rows and columns of an image;  $P_{i,j}$ ,  $u_{i,j}$ ,  $\sigma_{i,j}$  are respectively the pixel value, the mean and the variance of the model provided by

SG.

## 2. Memory cell identification and ARB generation

For each pixel, a set of  $MC_{background}$  is created based on the  $mc$  that verified the Equation 4.3. Each element of the  $MC_{background}$  is represented by a Gaussian contains the pixel value, the mean  $\mu$  and the variance  $\sigma$ .

A pixel with an empty  $MC_{background}$  will be considered as a foreground pixel. If the  $MC_{background}$  not empty, a background pixel is detected then the AIRS-SG searches the smallest  $mc$  named  $mc_{match}$  from the  $MC_{background}$  that verifies the Equation 4.4.  $mc_{match}$  will be mutated with an empirically rate to generate new clones. The new clones groupe together with the  $mc_{match}$  in the  $AB$  set.

$$MC_{background} = \left\{ \frac{|P_t - u_{mc}|}{\sigma_{mc}} < 2.5, mc \in MC \right\} \quad (4.3)$$

$$mc_{match} = \operatorname{argmin}_{mc \in MC} (MC_{background}) \quad (4.4)$$

The mutation function is also adapted to take into consideration the feature vector used. The new mutation function is defined as follows:

---

### Algorithm 15 Proposed mutation function

---

**Require:** a Gaussian  $g_t$  (pixel value  $P_t$ , mean  $u_t$ , variance  $\sigma_t$ ),  $b$  : boolean

**Ensure:**  $g_t, b$

```

for each  $g_{t_i} \in g_t, i \in [2, 3]$  do
   $change \leftarrow$  random value between 0 and 1
   $changeto \leftarrow$  random value between 0 and 254
  if  $change < mutation\_rate$  then
     $g_{t_i} \leftarrow changeto \times random\_value + 1$ 
     $b \leftarrow true$ 
  end if
end for
if  $b = True$  then
   $change \leftarrow$  random value between 0 and 1
   $changeto \leftarrow$  random value between 0 and 255
  if  $change < mutation\_rate$  then
     $g_{t_1} \leftarrow changeto$ 
  end if
end if

```

---

## 3. Competition for resources and development of a candidate memory cells

In this stage, all the  $ab$  (Gaussians) generated in the previous step are filtered using the Equation 4.5 for keeping only the Gaussians that represented the background.

$$\frac{P_{ab} - u_{ab}}{\sigma_{ab}} < 2.5 \quad (4.5)$$

$P_{ab}$ ,  $u_{ab}$  and  $\sigma_{ab}$  are respectively the pixel value, the mean and the variance of the Gaussian  $ab$ .

After this step, the remains  $ab$  are grouped in a set called  $MC_{candidate}$ . We have introduced a small modification in this step, we propose to select all the  $ab$  that verified the Equation 4.5 instead of keeping only the best  $ab$  as indicated in the original version of the AIRS. Through this modification, we can manage the multimodality of the background.

**4. Memory cells introduction** This is the last stage of our proposition. During it, each element from the  $MC_{candidate}$  that has an affinity lower than that of the  $mc_{match}$ , is introduced into  $MC$  set as a new memory cell. If the affinity between  $mc_{candidate}$  and  $mc_{match}$  is less than  $AT \times ATS$ , the  $mc_{match}$  will be removed from the pool of  $MC$ .

This stage has also benefited from a small modification in the number of the memory cells introduced into  $MC$  set. Indeed, the basic AIRS allows the introduction of only one memory cell to the  $MC$  set at each iteration. The proposed modification gives the system more opportunity for producing better background models.

To better understand how our system works, see the Figure 4.2 and the Algorithm 16.

### 4.2.2.3 Post processing

Once the detection of moving objects is finished, binary images containing only black and white pixels are produced. Black pixels represent the background, while the white pixels constitute the moving objects.

Any background subtraction method creates some gaps generated by the false detections. These gaps produced isolated pixels or cuts in the moving objects. To solve this problem, we have applied morphological operations on these images.

Dilation operation allows covering the cuts inside the moving objects, while the erosion operation eliminates the isolated pixels generated from the change in brightness, dust, etc. After the morphology operations, we applied a median filter to correct and better supervise the edges of the moving objects.

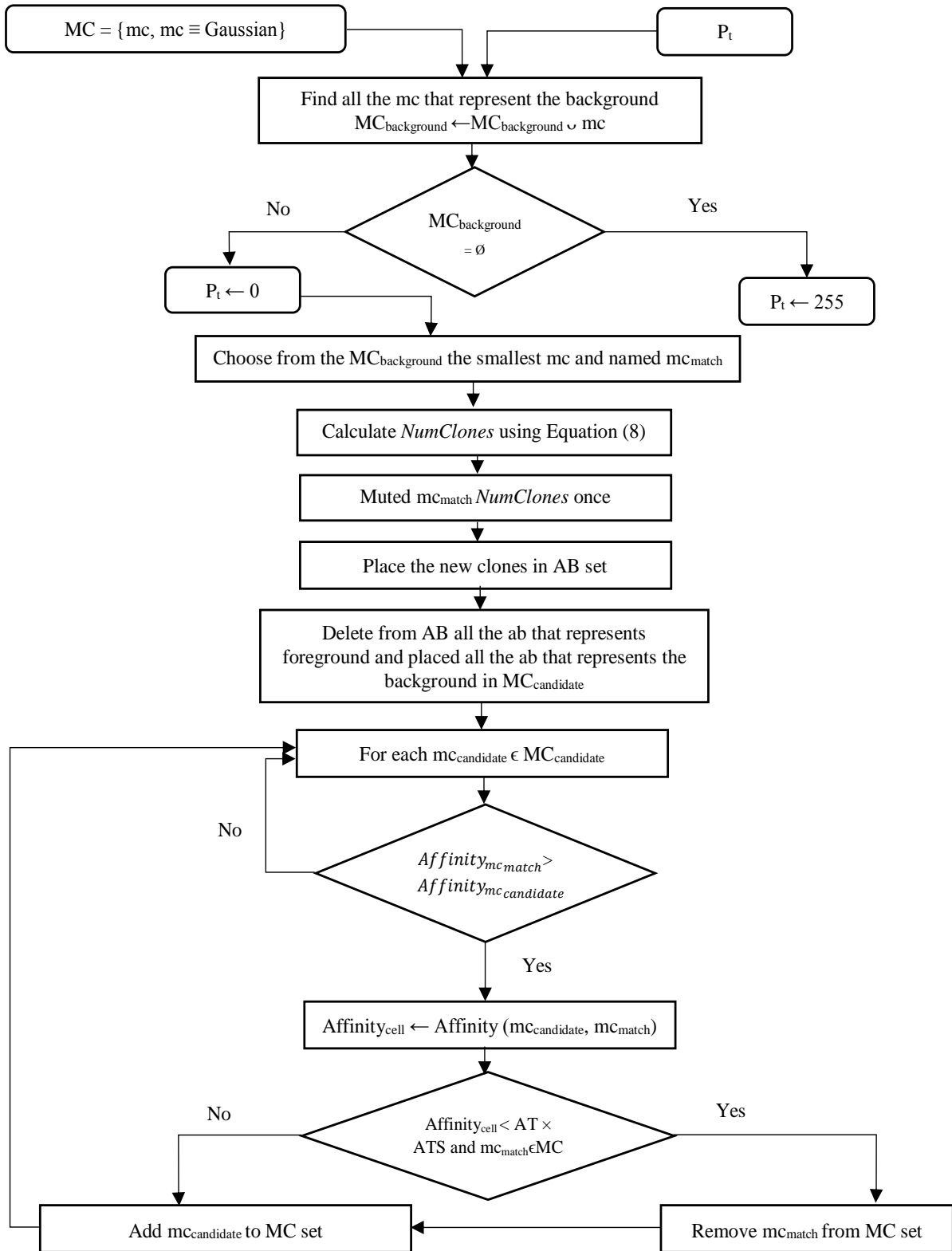


Figure 4.2: AIRS-SG flowchart

**Algorithm 16** AIRS-SG**Require:**  $AG = \{ag_1, ag_2, \dots, ag_n, ag_1, ag_2, \dots, ag_m\}$ ,  $ag_t \equiv P_t$ **Ensure:**  $ag_t$ **Initialization****Begin** $MC$ : A set of memory cells  $mc$ ,  $mc \equiv g_t$ . $AB \leftarrow \emptyset$  $AT \leftarrow \frac{\sum_{i=1}^n \sum_{j=1}^m \frac{P_{i,j} - u_{i,j}}{\sigma_{i,j}}}{n \times m}$ **End****for each**  $ag_t \in AG$  **do** $MC_{background} = \left\{ \frac{ag_t - u_{mc}}{\sigma_{mc}} < 2.5, mc \in MC \right\}$ **if** ( $MC_{background} \equiv \emptyset$ ) **then** $ag_t \leftarrow 255$ **else** $mc_{match} \leftarrow \text{argmin}_{mc \in MC} (MC_{background})$  $NumClones \leftarrow Clonal\_rate \times Hyper\_mutation\_rate \times Affinity(ag_t, mc_{match})$  $MU \leftarrow \emptyset$  $MU \leftarrow MU \cup mc_{match}$ **while**  $|MU| < NumClones$  **do** $mut \leftarrow false$  $mc_{clone} \leftarrow mc_{match}$  $mc_{clone}, mut \leftarrow mutation(mc_{clone}, mutation\_rate, mut)$ **if** ( $mut \equiv true$ ) **then** $MU \leftarrow MU \cup mc_{clone}$ **end if****end while** $AB \leftarrow AB \cup MU$  $MC_{candidate} \leftarrow \emptyset$ **for each**  $ab \in AB$  **do****if**  $\left( \frac{P_{ab} - u_{ab}}{\sigma_{ab}} > 2.5 \right)$  **then** $AB \leftarrow AB - ab$ **else** $MC_{candidate} \leftarrow MC_{candidate} \cup ab$ **end if****end for** $Find \leftarrow false$  $Affinity_{mc_{match}} \leftarrow \frac{ag_t - u_{mc_{match}}}{\sigma_{mc_{match}}}$ **for each**  $mc_{candidate} \in MC_{candidate}$  **do** $Affinity_{mc_{candidate}} \leftarrow \frac{ag_t - u_{mc_{candidate}}}{\sigma_{mc_{candidate}}}$ **if** ( $Affinity_{mc_{candidate}} < Affinity_{mc_{match}}$ ) **then** $Affinity_{cell} \leftarrow (Affinity_{mc_{match}} + Affinity_{mc_{candidate}}) / 2$ **if** ( $(Affinity_{cell} < AT \times ATS) \& (Find \equiv false)$ ) **then** $Find \leftarrow true$  $MC \leftarrow MC - mc_{match}$ **end if** $MC \leftarrow MC \cup mc_{candidate}$ **end if****end for** $ag_t \leftarrow 0$ **end if****end for**

### 4.2.3 Experiments and results

In this section, we will present and discuss the obtained results of our system on some public datasets.

#### 4.2.3.1 Settings

Our system is implemented in Java environment on an i3 PC of fourth generation and 4 GB memory capacity.

The set of parameters used in the test: learning rate ( $\alpha$ ), *mutation\_rate*, *Clonal\_rate*, *Hyper\_mutation\_rate* and Affinity Threshold Scalar (*ATS*) were fixed empirically after several empirical tests respectively to: 0.01, 0.1, 10, 4, 0.01. These values are chosen with respect to their influence on the results quality.

AIRS, like many other classifiers, needs to fix its parameters empirically. Indeed we have not found any work that gives an automatic method for fixing these values correctly. After several empirical tests, we have deduced that the *mutation\_rate* value should be taken very small, or the convergence may be unnecessarily delayed. The *Clonal\_rate* allows controlling the number of the clone generated at each iteration. For our case, we have initialized it at 0.1 due to the small population used in the system. The *Hyper\_mutation\_rate* intervenes as a control process for the number of mutated clones. We have fixed this latter to 4 due to the number of permutations between features (3 possibilities when the mutation is done 2 by 2 and 1 if all features are muted simultaneously). The Affinity Threshold Scalar is set very low to ensure that the dissimilarity between foreground and background is kept since our feature vector contains only 3 characteristics, and the mutation can easily bring the background closer to the foreground.

#### 4.2.3.2 Datasets

To evaluate the performance of our system, we have tested it on some public datasets: CDnet 2014[WJP<sup>+</sup>14], Wallflower (WavingTrees) [TKBM99], WaterSurface [LHGT04], UCSD [SND11] and Fountain [LHGT04]. The following paragraphs detail each dataset.

- **CDnet 2014**

This dataset provides a set of realistic and diverse video sequences captured from a camera without CGI. These video sequences are addressed to cover a wide range of detection challenges in surveillance and intelligent environment scenarios.

CDnet 2014 contains 11 categories of videos: Baseline (BL), Dynamic Background (DB), Camera Jitter (CJ), Intermittent Object Motion (IOM), Shadow (Shd), Thermal (Th), Bad Weather (BW), Low Framerate (LFR), Night Videos (NV), Pan Tilt Zoom (PTZ) and Turbulence (Tb). In which each category has from 4 to 6 video sequences with a spatial resolutions varied from  $320 \times 240$  to  $720 \times 576$ . Each video

in this dataset is accompanied with: a sub-directory named "input" contains all the JPEG frames of the video, a BMP frame that gives the accurate ground-truth segmentation, two files named "ROI.bmp" and "ROI.jpg" showing the spatial region of interest and a "temporalROI.txt" file contains two frame numbers used to calculate the score.

In our work, we have interested in detecting moving objects in videos that have a dynamic background. These videos pose a great problem for the majority of the background subtraction systems. For this purpose, we have used only the Dynamic Background category (DB) from the CDnet 2014 dataset to test our proposition. Dynamic Background category contains six videos that represent the outdoor scenes that have a strong background motion. The table below shows details about video sequences of this category.

Table 4.1: Description of Dynamic Background category of CDnet 2014 dataset

Category	Number of videos	Video name	Total number of frames	Resolution
Dynamic Background (DB)	6	Boats	7999	$320 \times 240$
		Canoe	1189	$320 \times 240$
		Fall	4000	$720 \times 480$
		Fountain01	1184	$432 \times 288$
		Fountain02	1499	$432 \times 288$
		Overpass	3000	$320 \times 240$

- **Wallflower**

This dataset contains 7 video sequences. Frames of each video sequences are provided in BMP format and they have a spatial resolution of  $160 \times 120$ . Each test sequence started after training algorithms with at least 200 background frames, except in the bootstrap sequence.

The number of the evaluate frame and their hand segmented (ground-truth) of this dataset are given in the directory of the video sequence.

Wallflower dataset allows the tracking of people, and it excluded the other objects such as cars, animals, etc, which are in some applications considered as foreground objects. For this purpose, only people are marked as foreground objects. The sequences video are:

**Moved Object (MO):** the objects (chair, phone) of this scene can be moved. But, these objects should be not considered as a part of the foreground. We evaluated the scene 100 frames after the chair moved.

**Time of Day (TD):** the sequences of this video show a dark room become gradually brighter after a few minutes.



**Light Switch (LS):** it describes the sudden changes in illumination and some other impacts that affect the appearance of the background.

**Waving Trees (WT):** a swaying tree is a part of the background and a person walking in front of this tree. A robust model is required to detect this person.

**Camouflage (Ca):** pixel properties of the foreground objects may be subsumed by the pixels that modelled the background.

**Bootstarp (Boo):** a training period without foreground objects in some environments is absent (we distinguish a constant motion in some environments).

**Foreground Aperture (FA):** when a homogeneously colored object moves, a change in the interior pixels cannot be detected. In this case, the entire object cannot appear as foreground.

We have selected only the sequence "Waving Trees" from the dataset Wallflower to test our system since this sequence contains a dynamic background.

- **WaterSurface**

In this dataset, the wavering water surface is a part of the background and a person passing on land is detected as a foreground.

- **UCSD**

This dataset contains 18 video sequences: birds, boats, bottel, chopper, cyclists, folck, freeway, hockey, jump, landing, ocean, peds, rain, skiing, surf, surfers, traffic and zodiac. Frames of each sequence are provided in JPEG format, and they are coded in grayscale. For the ground-truth frames, it is provided in Matlab as a 3D array, where the value 1 indicates the foreground and the value 0 indicates the background. Some ground-truth frames are not available in several video sequences of this dataset. The frames for the training and the test are not specified. Noted that we have used from UCSD dataset only the videos that contain a dynamic background.

- **Fountain**

The background contains a fountain, and some people pass by the latter. These persons should be considered as a foreground.

### 4.2.3.3 Performance evaluation

- **Qualitative results**

Qualitative results show our obtained segmentations on the datasets: CDnet 2014 (Dynamic Background), Wallflower (WavingTrees), WaterSurface, UCSD and Foun-

tain (see Tables 4.2 and 4.3). These results are compared with the ground-truth images.

But, the observable results are not sufficient since they do not ensure a delicate representation of the system’s performance. For this purpose, quantitative results are required to improve and objectively describe the proposed system’s effectiveness.

Table 4.2: Some images of our results on the Dynamic Background category of the dataset CDnet 2014





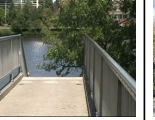


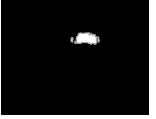








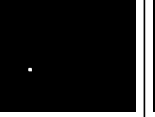































Video name	Fountain01	Fountain02	Canoe	Boats	Overpass	Fall
Test images						
Ground-truth						
Results						

Table 4.3: Some results of AIRS-SG on the datasets Wallflower (WavingTrees), WaterSurface, Fountain and UCSD

Video name	WavingTrees	WaterSurface	Fountain	Bottle	Chopper
Test images					
Ground-truth					
Results					
Video name	Boats	Birds	Freeway	Peds	Rain
Test images					
Ground-truth					
Results					

- **Quantitative results**

The objective of the quantitative results is to prove the efficacy of our system to subtract the background from scenes that contain a dynamic background. Authors of CDnet 2014 have fixed seven measures to evaluate the quality of systems:

1. Recall (Re) :  $\frac{TP}{TP+FN}$
2. Specificity (Sp) :  $\frac{TN}{TN+FP}$
3. False Positive Rate (FPR) :  $\frac{FP}{FP+TN}$
4. False Negative Rate (FNR) :  $\frac{FN}{TP+FN}$
5. Percentage of Wrong Classifications (PWC) :  $100 \times \frac{(FN+FP)}{(TP+FN+FP+TN)}$
6. F\_Measure :  $\frac{2 \times Precision \times Recall}{Precision + Recall}$
7. Precision (Pre) :  $\frac{TP}{TP+FP}$

Such that:

- True positive (TP): the result and the ground-truth are positive (255).
- False positive (FP): the result is positive (255), while the ground-truth is negative (0).
- True negative (TN): the result and the ground-truth are negative (0).
- False negative (FN): the result is negative (0), while the ground-truth is positive (255).

To evaluate our system correctly, the results are compared with hand segmented images called ground-truth. These images are provided by the video sequences of the dataset CDnet 2014 (See Tables 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10).

Table 4.4: Re, Sp, FPR, FNR, PWC, Precision, F-Measure of AIRS-SG on CDnet 2014 Dynamic Background category

Video name	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
Fountain01	0.5476	0.9996	0.0003	0.4523	0.0764	0.5390	0.5433
Fountain02	0.4945	0.9992	0.0007	0.5054	0.1841	0.5850	0.5359
Canoe	0.9773	0.9929	0.0070	0.0226	0.7462	0.7994	0.8794
Boats	0.8460	0.9933	0.0066	0.1539	0.7557	0.4459	0.5840
Overpass	0.3917	0.9949	0.0050	0.6082	1.3122	0.5141	0.4446
Fall	0.4251	0.9930	0.0069	0.5748	1.6119	0.4989	0.4590
The average	0.6137	0.9955	0.0044	0.3862	0.7811	0.5637	0.5744

Table 4.5: Comparison of quantitative results with well-known background subtraction methods on Fountain01 video

Method Name	Video Name	Type of learning	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Fountain01	Offline/Supervised	0.9974	1.0000	0.0000	0.0026	0.0012	0.9887	0.9930
Cascade CNN [WLJ17]		Offline/Supervised	0.9548	0.9999	0.0001	0.0452	0.0163	0.8637	0.9070
IUTIS-5 [BCS17]		Offline/Supervised	0.8663	0.9998	0.0002	0.1337	0.0310	0.7836	0.8229
DeepBS [BDR18]		Offline/Supervised	0.7241	0.9999	0.0001	0.2759	0.0362	0.8186	0.7685
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	0.8774	0.9996	0.0004	0.1226	0.0473	0.6622	0.7547
SuBSENSE [SCBB15b]		Online/Not supervised	0.8771	0.9996	0.0004	0.1229	0.0477	0.6599	0.7531
CwisarDRP [DGG17]		Online/Supervised	0.7181	0.9997	0.0003	0.2819	0.0528	0.6694	0.6929
$M^4CD$ V2.0 [GWG18]		Online/Not supervised	0.8425	0.9935	0.0065	0.1575	0.6621	0.0973	0.1744
SWCD [IOGG18]		Online/Not supervised	0.7122	0.9999	0.0001	0.2878	0.0375	0.8127	0.7591
MBS [SC17]		Online/Not supervised	0.3771	0.9999	0.0001	0.6229	0.0570	0.8557	0.5235
GMM [SG99]		Online/Not supervised	0.7973	0.9841	0.0159	0.2027	1.6017	0.0401	0.0763
BMN-BSN [MGRNO19]		Offline/Supervised	0.5569	0.9560	0.0440	0.4431	4.4336	0.0104	0.0204
KDE [EHD00]		Online/Not supervised	0.7930	0.9890	0.0110	0.2070	1.1162	0.0565	0.1055
AMBER [WD14]		Online/Not supervised	0.7582	0.9998	0.0002	0.2418	0.0378	0.7807	0.7693
FCFNE [YYL19]		Online/Not supervised	0.8817	N/A	0.0004	0.1183	0.0528	0.6963	0.7792
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	0.5402	0.9963	0.0037	0.4598	0.4076	0.1082	0.1802
AAPSA [RACM16]		Online/Not supervised	0.5128	0.9993	0.0007	0.4872	0.1076	0.3880	0.4417
SOBS_CF [MP10]		Online/Not supervised	0.7378	0.9901	0.0099	0.2622	1.0065	0.0586	0.1085
CL-VID [LRMCLBD18]		Online/Not supervised	0.9327	0.9705	0.0295	0.0673	2.9518	0.0256	0.0498
CP3-Online [LHI <sup>+</sup> 15]		95.00%	0.7351	0.9941	0.0059	0.2649	0.6142	0.0934	0.1658
SSOBS [SCM17]	Online/Not supervised	0.6660	0.9612	0.0388	0.3340	3.9081	0.0140	0.0275	
AIRS-SG	Online/Not supervised	0.5476	0.9996	0.0004	0.4524	0.0764	0.5391	0.5433	

Table 4.6: Comparison of quantitative results with well-known background subtraction methods on Fountain02 video

Method Name	Video Name	Type of learning	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Fountain02	Offline/Supervised	0.9978	1.0000	0.0000	0.0022	0.0024	0.9910	0.9944
Cascade CNN [WLJ17]		Offline/Supervised	0.9822	0.9999	0.0001	0.0178	0.0131	0.9580	0.9699
IUTIS-5 [BCS17]		Offline/Supervised	0.9530	0.9999	0.0001	0.0470	0.0192	0.9576	0.9553
DeepBS [BDR18]		Offline/Supervised	0.8784	1.0000	0.0000	0.1216	0.0305	0.9778	0.9254
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	0.9247	0.9999	0.0001	0.0753	0.0230	0.9668	0.9453
SuBSENSE [SCBB15b]		Online/Not supervised	0.9232	0.9999	0.0001	0.0768	0.0235	0.9658	0.9441
CwisarDRP [DGG17]		Online/Supervised	0.9372	0.9998	0.0002	0.0628	0.0358	0.9004	0.9184
$M^4CD$ V2.0 [GWG18]		Online/Not supervised	0.9682	0.9998	0.0002	0.0318	0.0291	0.9035	0.9348
SWCD [IOGG18]		Online/Not supervised	0.9327	0.9998	0.0002	0.0673	0.0299	0.9285	0.9306
MBS [SC17]		Online/Not supervised	0.8818	0.9999	0.0001	0.1182	0.0324	0.9647	0.9214
GMM [SG99]		Online/Not supervised	0.8717	0.9994	0.0006	0.1283	0.0917	0.7451	0.8035
BMN-BSN [MGRNO19]		Offline/Supervised	0.8837	0.9994	0.0006	0.1163	0.0857	0.7579	0.8160
KDE [EHD00]		Online/Not supervised	0.8528	0.9995	0.0005	0.1472	0.0788	0.7955	0.8232
AMBER [WD14]		Online/Not supervised	0.9644	0.9998	0.0002	0.0356	0.0318	0.8959	0.9289
FCFNE [YYL19]		Online/Not supervised	0.9370	N/A	0.0002	0.0630	0.0345	0.9463	0.9416
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	0.8939	0.9999	0.0001	0.1061	0.0349	0.9408	0.9168
AAPSA [RACM16]		Online/Not supervised	0.9222	0.9930	0.0070	0.0778	0.7118	0.2220	0.3579
SOBS_CF [MP10]		Online/Not supervised	0.9257	0.9993	0.0007	0.0743	0.0833	0.7475	0.8271
CL-VID [LRMCLBD18]		Online/Not supervised	0.9671	0.9949	0.0051	0.0329	0.5117	0.2919	0.4484
CP3-Online [LHI <sup>+</sup> 15]		Online/Not supervised	0.6391	0.9992	0.0008	0.3609	0.1576	0.6321	0.6356
SSOBS [SCM17]	Online/Not supervised	0.5993	0.9949	0.0051	0.4007	0.5989	0.2009	0.3009	
AIRS-SG	Online/Not supervised	0.4945	0.9992	0.0008	0.5055	0.1842	0.5850	0.5360	

Table 4.7: Comparison of quantitative results with well-known background subtraction methods on Canoe video

Method Name	Video Name	Type of learning	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Canoe	Offline/Supervised	0.9993	1.0000	0.0000	0.0007	0.0037	0.9996	0.9995
Cascade CNN [WLJ17]		Offline/Supervised	0.9935	0.9993	0.0007	0.0065	0.0930	0.9804	0.9869
IUTIS-5 [BCS17]		Offline/Supervised	<b>0.9052</b>	<b>0.9997</b>	<b>0.0003</b>	<b>0.0948</b>	<b>0.3642</b>	<b>0.9912</b>	<b>0.9462</b>
DeepBS [BDR18]		Offline/Supervised	0.9793	0.9992	0.0008	0.0207	0.1460	0.9795	0.9794
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	0.7677	0.9998	0.0002	<b>0.2323</b>	0.8430	0.9925	<b>0.8658</b>
SuBSENSE [SCBB15b]		Online/Not supervised	0.6590	0.9998	0.0002	0.3410	1.2236	0.9933	0.7923
CwisarDRP [DGG17]		Online/Supervised	0.8296	0.9999	0.0001	0.1704	0.6164	0.9956	0.9051
$M^4CD$ V2.0 [GWG18]		Online/Not supervised	0.4653	0.9996	0.0004	0.5347	0.3702	0.8934	0.6119
SWCD [IOGG18]		Online/Not supervised	0.9402	0.9960	0.0040	0.0598	0.5971	0.8963	0.9177
MBS [SC17]		Online/Not supervised	0.8879	0.9995	0.0005	0.1121	0.4407	0.9863	0.9345
GMM [SG99]		Online/Not supervised	0.8659	0.9964	0.0036	0.1341	0.8225	0.8982	0.8817
BMN-BSN [MGRNO19]		Offline/Supervised	0.9862	0.9847	0.0153	0.0138	1.5268	0.7027	0.8206
KDE [EHD00]		Online/Not supervised	0.8315	0.9980	0.0020	0.1685	0.7860	0.9396	0.8822
AMBER [WD14]		Online/Not supervised	0.9590	0.9960	0.0040	0.0410	0.5286	0.9886	0.9278
FCFNE [YYL19]		Online/Not supervised	0.9114	N/A	0.0017	0.0886	0.5464	0.9589	0.9346
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	0.9394	0.9978	0.0022	0.0606	0.4277	0.9398	0.9396
AAPSA [RACM16]		Online/Not supervised	0.7977	0.9999	0.0001	0.2023	0.7240	0.9973	0.8864
SOBS_CF [MP10]		Online/Not supervised	0.9633	0.9976	0.0024	0.0367	0.3615	0.9364	0.9497
CL-VID [LRMCLBD18]		Online/Not supervised	0.9607	0.9961	0.0039	0.0393	0.5114	0.9014	0.9301
CP3-Online [LHI <sup>+</sup> 15]		Online/Not supervised	0.8969	0.9974	0.0026	0.1031	0.6130	0.9276	0.9120
SSOBS [SCM17]		Online/Not supervised	0.2240	<b>0.8605</b>	<b>0.1395</b>	<b>0.7760</b>	<b>14.3509</b>	<b>0.0100</b>	<b>0.0192</b>
AIRS-SG		Online/Not supervised	0.9773	0.9930	0.0070	0.0227	0.7463	0.7994	0.8795

Table 4.8: Comparison of quantitative results with well-known background subtraction methods on Boats video

Method Name	Video Name	Type of learning	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Boats	Offline/Supervised	0.9989	0.9999	0.0001	0.0011	0.0128	0.9810	0.9899
Cascade CNN [WLJ17]		Offline/Supervised	0.9977	0.9998	0.0002	0.0023	0.0224	0.9676	0.9824
IUTIS-5 [BCS17]		Offline/Supervised	0.6147	0.9999	0.0001	<b>0.3853</b>	<b>0.2527</b>	0.9723	0.7532
DeepBS [BDR18]		Offline/Supervised	0.7098	0.9998	0.0002	0.2902	0.2060	0.9490	0.8121
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	0.5830	0.9997	0.0003	0.4170	0.2943	0.9178	0.7130
SuBSENSE [SCBB15b]		Online/Not supervised	0.5596	0.9997	0.0003	0.4404	0.3107	0.9106	0.6932
CwisarDRP [DGG17]		Online/Supervised	0.8214	0.9992	0.0008	0.1786	0.1946	0.8618	0.8411
$M^4CD$ V2.0 [GWG18]		Online/Not supervised	0.9542	0.9978	0.0022	0.0458	0.3769	0.9403	0.9472
SWCD [IOGG18]		Online/Not supervised	0.9238	0.9984	0.0016	0.0762	0.2040	0.7877	0.8503
MBS [SC17]		Online/Not supervised	0.8528	0.9998	0.0002	0.1472	0.1135	0.9620	0.9041
GMM [SG99]		Online/Not supervised	0.7582	0.9980	0.0020	0.2418	0.3541	0.7014	0.7287
BMN-BSN [MGRNO19]		Offline/Supervised	0.9436	0.9997	<b>3.0642</b>	0.0564	0.0658	0.9511	0.9473
KDE [EHD00]		Online/Not supervised	0.6575	0.9973	0.0027	<b>0.3425</b>	<b>0.4797</b>	0.6089	0.6323
AMBER [WD14]		Online/Not supervised	0.8758	0.9989	0.0011	0.1242	0.1889	0.8319	0.8533
FCFNE [YYL19]		Online/Not supervised	0.8321	N/A	0.0018	0.1679	0.3197	0.7906	0.8108
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	0.7806	0.9992	0.0008	0.2194	0.2174	0.8599	0.8183
AAPSA [RACM16]		Online/Not supervised	0.6284	0.9999	0.0001	0.3716	0.2429	0.9757	0.7644
SOBS_CF [MP10]		Online/Not supervised	0.9122	0.9994	0.0006	0.0878	0.1098	0.9127	0.9124
CL-VID [LRMCLBD18]		Online/Not supervised	0.7236	0.9995	0.0005	0.2764	0.2182	0.9101	0.8062
CP3-Online [LHI <sup>+</sup> 15]		Online/Not supervised	0.8350	0.9922	0.0078	0.1650	0.8748	0.4044	0.5449
SSOBS [SCM17]		Online/Not supervised	0.6894	0.8445	<b>0.1555</b>	0.3106	16.0995	0.1400	0.2327
AIRS-SG		Online/Not supervised	0.8461	0.9934	0.0066	0.1539	0.7557	0.4460	0.5841

Table 4.9: Comparison of quantitative results with well-known background subtraction methods on Overpass video

Method Name	Video Name	Type of learning	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Overpass	Offline/Supervised	0.9970	1.0000	0.0000	0.0030	0.0064	0.9982	0.9976
Cascade CNN [WLJ17]		Offline/Supervised	0.9893	0.9997	0.0003	0.0107	0.0403	0.9808	0.9850
IUTIS-5 [BCS17]		Offline/Supervised	0.8837	0.9997	0.0003	0.1163	0.1858	0.9753	0.9272
DeepBS [BDR18]		Offline/Supervised	0.9158	0.9996	0.0004	0.0842	0.1523	0.9688	0.9416
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	0.8095	0.9994	0.0006	0.1905	0.3160	0.9470	0.8728
SuBSENSE [SCBB15b]		Online/Not supervised	0.7852	0.9994	0.0006	0.2148	0.3506	0.9437	0.8572
CwisarDRP [DGG17]		Online/Supervised	0.8862	0.9994	0.0006	0.1138	0.2095	0.9542	0.9189
$M^4CD$ V2.0 [WGW18]		Online/Not supervised	0.9671	0.9990	0.0010	0.0329	0.1467	0.9266	0.9464
SWCD [IOGG18]		Online/Not supervised	0.8550	0.9978	0.0022	0.1450	0.4078	0.8429	0.8489
MBS [SC17]		Online/Not supervised	0.8427	0.9996	0.0004	0.1573	0.2538	0.9632	0.8990
GMM [SG99]		Online/Not supervised	0.8294	0.9990	0.0010	0.1706	0.3264	0.9191	0.8719
BMN-BSN [MGRNO19]		Offline/Supervised	0.8880	0.9944	0.0056	0.1120	0.7017	0.6832	0.7723
KDE [EHD00]		Online/Not supervised	0.8003	0.9981	0.0019	0.1997	0.4550	0.8512	0.8250
AMBER [WD14]		Online/Not supervised	0.9849	0.9987	0.0013	0.0151	0.1439	0.9143	0.9483
FCFNE [YYL19]		Online/Not supervised	0.9007	N/A	0.0011	0.0993	0.2663	0.9326	0.9164
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	0.9524	0.9994	0.0006	0.0476	0.1255	0.9538	0.9531
AAPSA [RACM16]		Online/Not supervised	0.7046	0.9998	0.0002	0.2954	0.4142	0.9808	0.8201
SOBS CF [MP10]		Online/Not supervised	0.9617	0.9959	0.0041	0.0383	0.4522	0.7627	0.8507
CL-VID [LRMCLBD18]		Online/Not supervised	0.9393	0.9963	0.0037	0.0607	0.4421	0.7772	0.8506
CP3-Online [LHI <sup>+</sup> 15]		Online/Not supervised	0.6724	0.9991	0.0009	0.3276	0.5236	0.9141	0.7749
SSOBS [SCM17]	Online/Not supervised	0.4333	0.9473	0.0527	0.5667	5.9630	0.1004	0.1630	
AIRS-SG	Online/Not supervised	0.3917	0.9950	0.0050	0.6083	1.3123	0.5142	0.4447	

Table 4.10: Comparison of quantitative results with well-known background subtraction methods on Fall video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [LK18b]	Fall	Offline/Supervised	99.79%	0.9949	1.0000	0.0000	0.0051	0.0130	0.9977	0.9963
Cascade CNN [WLJ17]		Offline/Supervised	99.68%	0.9611	0.9994	0.0006	0.0389	0.1282	0.9664	0.9637
IUTIS-5 [BCS17]		Offline/Supervised	99.57%	0.9586	0.9984	0.0016	0.0414	0.2318	0.9147	0.9361
DeepBS [BDR18]		Offline/Supervised	99.14%	0.9182	0.9947	0.0053	0.0818	0.6691	0.7563	0.8294
WisenetMD [LKS <sup>+</sup> 18]		Online/Not supervised	99.36%	0.8752	0.9977	0.0023	0.1248	0.4471	0.8728	0.8740
SuBSENSE [SCBB15b]		Online/Not supervised	99.34%	0.8567	0.9978	0.0022	0.1433	0.4692	0.8758	0.8661
CwisarDRP [DGG17]		Online/Supervised	99.18%	0.7820	0.9976	0.0024	0.2180	0.6260	0.8524	0.8157
$M^4CD$ V2.0 [WGW18]		Online/Not supervised	96.57%	0.9134	0.9686	0.0314	0.0866	3.2409	0.3439	0.4996
SWCD [IOGG18]		Online/Not supervised	99.40%	0.8515	0.9985	0.0015	0.1485	0.4091	0.9117	0.8806
MBS [SC17]		Online/Not supervised	97.80%	0.7421	0.9842	0.0158	0.2579	2.0099	0.4584	0.5668
GMM [SG99]		Online/Not supervised	95.75%	0.8838	0.9608	0.0392	0.1162	4.0532	0.2892	0.4358
BMN-BSN [MGRNO19]		Offline/Supervised	95.65%	0.9473	0.9586	0.0414	0.0574	4.1647	0.2919	0.4463
KDE [EHD00]		Online/Not supervised	92.90%	0.8721	0.9319	0.0681	0.1279	6.9202	0.1875	0.3087
AMBER [WD14]		Online/Not supervised	97.84%	0.9638	0.9806	0.0194	0.0362	1.9713	0.4724	0.6340
FCFNE [YYL19]		Online/Not supervised	N/A	0.9160	N/A	0.0040	0.0840	0.5486	0.8185	0.8645
LCBC [HYK <sup>+</sup> 19]		Online/Not supervised	N/A	0.8270	0.9911	0.0089	0.1730	1.1791	0.6267	0.7131
AAPSA [RACM16]		Online/Not supervised	99.01%	0.6841	0.9976	0.0024	0.3159	0.7946	0.8376	0.7531
SOBS CF [MP10]		Online/Not supervised	90.80%	0.9079	0.9098	0.0902	0.0921	9.0211	0.1537	0.2629
CL-VID [LRMCLBD18]		Online/Not supervised	88.30%	0.9674	0.8832	0.1168	0.0326	11.5261	0.1300	0.2292
CP3-Online [LHI <sup>+</sup> 15]		Online/Not supervised	98.62%	0.5773	0.9956	0.0044	0.4227	1.1844	0.7013	0.6333
SSOBS [SCM17]	Online/Not supervised	93.35%	0.5281	0.9427	0.0573	0.4719	6.4682	0.1425	0.2244	
AIRS-SG	Online/Not supervised	98.21%	0.4251	0.9930	0.0070	0.5749	1.6119	0.4989	0.4591	

- **Discussions**

This section will discuss our results on the dynamic background category of the dataset CDnet 2014.

Our system has achieved good results in Canoe and Boats videos, and acceptable results in Fountain01, Fountain02 and Fall videos. But in Overpass video, we have got low rates due to the similarity of the foreground and background colours, which leads the system to make mistakes and confuse between foreground and background.

The failing of our system in the Overpass video is due to the use of a single component  $H$  of the HSV color space. Indeed our objective is to detect moving objects. Despite the low rate of some evaluation criteria, we have succeeded in our task (Detection even partial with false marginal detection). The qualitative results clearly show our system's effectiveness in detecting the moving objects when the background is dynamic. The partial detection influences our performance measurements due to the quantity of pixels detected compared with the ground truth.

Most of the methods, which have exceeded our system, used in-depth learning without features (deep learning methods). Despite the great success achieved by the deep learning methods, they required many learning data, a lot of time and a super calculator to produce such results. Furthermore, this type of application caused several problems in real environments. Indeed the changes in the structure of the environment (growing trees, changing the color of a wall, new building, etc.) force the system to go through a re-initialization step which requires a system shutdown during a long learning period (several days/months). This problem is shareable for all methods that use offline learning.

Concerning the methods that published their results on CDnet 2014 dataset and which have used supervised learning, they are learned well videos sequences, which justifies their high recognition rate.

### 4.3 Using Resources Competition and Memory Cell Development to Select the Best Gaussian Mixture Model (GMM) for Background Subtraction

Gaussian Mixture Model (GMM) is classified among the most efficient method in background subtraction. Since it combines the simplicity of treatment and the quality of result while benefiting from the execution time and the memory space.

However, the number of Gaussians represents a major drawback in GMM that can influence the results quality.

Indeed, a small number of Gaussians can reduce the history of the pixel, which causes a problem in the dynamic background. On the other hand, if we use many Gaussians without updating the Gaussian numbers, we cannot correctly classify the stationary objects that become moving objects.

To overcome the problem cited above, we have proposed using the Artificial Immune Recognition System (AIRS) to manage the number of Gaussians dynamically instead of fixing them a priori by the user.

Our contribution allows us to generate a set of new Gaussians using the AIRS with two different manners: the first one (Random generation) consists of using the AIRS to improve the system decision, while the second manner (Directed generation) uses the AIRS to improve the production of background models.

In the random generation, we start with the GMM in the learning stage; then, we apply the AIRS in the classification stage to generate several Gaussian models using the Memory cell identification and ARB generation process for all pixels regardless of their nature. The produced models are filtered with the resource competition and memory cell development process of the AIRS to keep only the best models representing the background. Once the AIRS finishes, the GMM method is used again to classify the pixels into the background and foreground pixels.

The directed Generation also begins with a learning stage using the GMM method; after that, we classify pixels into background and foreground using the GMM. Indeed, the GMM is used here to filter the background from the foreground pixels before applying the AIRS steps, which reduces the time consumed in the generation of new models and improves the accuracy since we will produce models only for pixels that represent the background.

Noted that each pixel in these propositions is represented with  $H$  component of the HSV color space. Indeed, this color space is closer to the perception of the human, and it is more robust to the change of brightness than RGB color. In the following sections, we will detail each proposition. Before detailing each proposition, we first present the global architecture of the proposed systems (Figure 4.3).



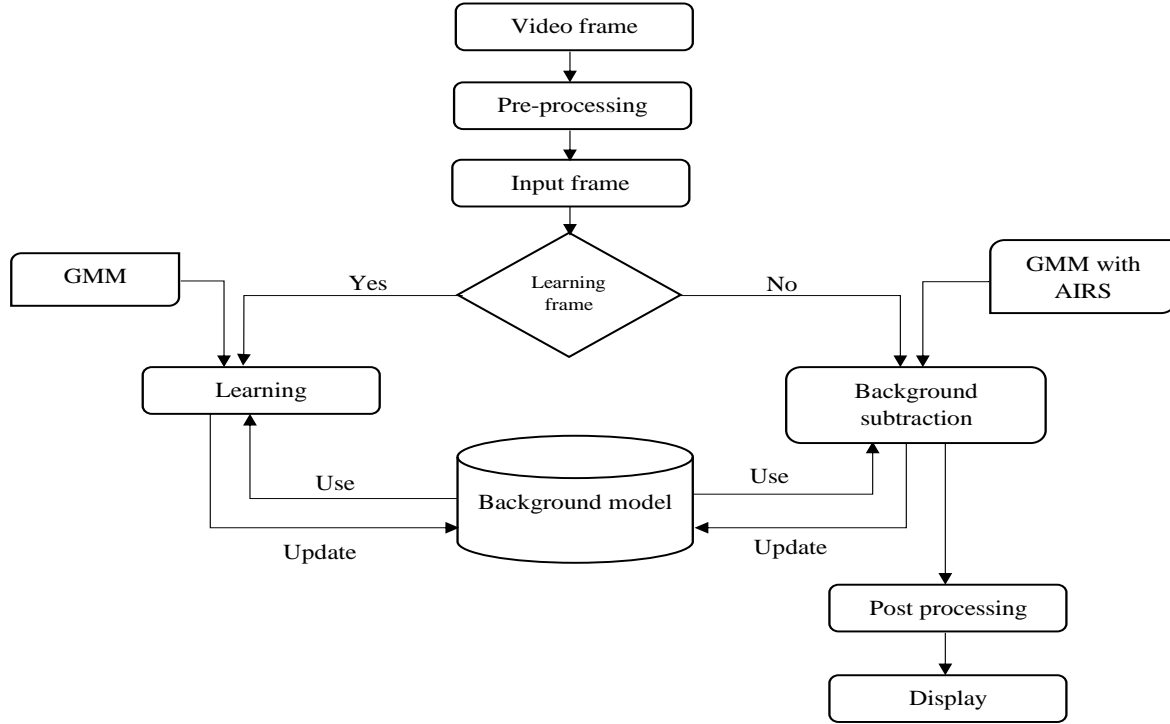


Figure 4.3: Flowchart of GMM-AIRS

### 4.3.1 Proposition One (Random Generation)

In this proposition, several Gaussians are generated for each pixel with the AIRS approach and introduced as new background models. After that, we try to classify this pixel (background or foreground) using the GMM method.

In the beginning, we create a set of Gaussians ( $g_i$ ) that represented the background for the pixel  $P_t$  using the following equation:

$$MC_{background} = \left\{ g_i, \frac{|P_t - \mu_i|}{\sigma_i} < 2.5 \right\} \quad (4.6)$$

Where each Gaussian model  $g_i$  is represented with four variables: the pixel value  $P_i$ , the average  $\mu_i$ , the variance  $\sigma_i$  and the weight  $w_k$ . After that, we select from the  $MC_{background}$  the Gaussian  $mc_{match}$  which has the smallest distance to the value of the current pixel (See the Equation 4.7)

$$mc_{match} = \underset{g_i \in Background}{\operatorname{argmin}} \left( \frac{|P_t - \mu_i|}{\sigma_i} < 2.5 \right) \quad (4.7)$$

$mc_{match}$  is muted in the ARBs generation stage with a mutation function (See Algorithm 15). Once this stage is finished, several new Gaussians (clones) are created.

The number of clones created is calculated by:

$$NumClones = Clonal\_rate \times Hyper\_mutation\_rate \times Affinity(P_t, mc_{match}) \quad (4.8)$$

Where:

*Clonal\_Rate* and *Hyper\_mutation\_rate*: are two integer values chosen empirically.

*Affinity* ( $P_t, mc_{match}$ ): is calculated between the current pixel and the Gaussian  $mc_{match}$  according to the following equation:

$$Affinity(P_t, mc_{match}) = \frac{|P_t - \mu_{mc_{match}}|}{\sigma_{mc_{match}}} \quad (4.9)$$

The *Mutation\_rate* represents a value between 0 and 1.

All the new clones are grouped with the  $mc_{match}$  in the *AB* set. These clones are filtered into **Competition for resources and development of a candidate memory cells** stage, and we keep only the best Gaussians ( $ab$ ) that satisfied:

$$\frac{|P_{ab} - \mu_{ab}|}{\sigma_{ab}} < 2.5 \quad (4.10)$$

$ab$  that remains after this stage are selected as candidate memory cells (see Equation 4.11). Here we have chosen to introduce all the remains  $ab$  as candidate memory cells instead of keeping only the best  $ab$  like indicated in the basic version of AIRS.

$$MC_{candidate} = \left\{ ab, \frac{|P_{ab} - \mu_{ab}|}{\sigma_{ab}} < 2.5 \right\} \quad (4.11)$$

In the last step of this proposition, we try to introduce from the  $MC_{candidate}$  new memory cells. This operation consists of choosing the Gaussians ( $mc_{candidate}$ ) the most representative and adding them as new background models using the following equation:

$$Affinity(P_t, mc_{candidate}) < Affinity(P_t, mc_{match}) \quad (4.12)$$

If the previous condition is satisfied, we compare the mean Affinity of  $mc_{match}$  and  $mc_{candidate}$  with  $AT \times ATS$  (See Equation 4.13).

$$\frac{Affinity(P_t, mc_{candidate}) + Affinity(P_t, mc_{match})}{2} < AT \times ATS \quad (4.13)$$

$AT$  is the average distances between the background pixels and their Gaussians in the training phase (See Equation 4.14).

$$AT = \frac{\sum_{i=1}^n \sum_{j=1}^m Average\_Affinity(P_{i,j}, G_{i,j})}{n \times m} \quad (4.14)$$

With:

$G_{i,j}$ : is a set of Gaussians of the pixel  $P_{i,j}$ .

If the Equation 4.13 is checked, the  $mc_{match}$  will be removed from the pool of  $MC$  set. At the end of this stage, the GMM method is applied on the background models of this pixel for determining their nature (background or a foreground).

### 4.3.2 Proposition Two (Directed Generation)

This proposition follows the same principle of proposition one, but we have guided the production of the Gaussians, which are generated by the AIRS algorithm only for the pixels that represent the background instead to generate for all pixels.

Firstly, we started by classifying each pixel ( $P_t$ ) using GMM method (see GMM method in the Chapter 2). If  $P_t$  is a background pixel, we choose from the first  $\beta$  Gaussians, a memory cell  $mc_{match}$  that satisfied the following equation:

$$mc_{match} = argmin_{g_i \in \beta_{Gaussians}} \left\{ \frac{|P_t - \mu_i|}{\sigma_i} < 2.5 \right\} \quad (4.15)$$

$mc_{match}$  is muted using the mutation function which is described above (see Algorithm 15) to generate  $NumClones$  (see Equation 4.8) new Gaussians.

The affinity measure is calculated also with the pixel value  $P_t$  and the Gaussian  $mc_{match}$  using the Equation 4.9.

After this step, an  $AB$  set is created from  $mc_{match}$  and the new Gaussians generated with the mutation process. In the Competition for resources process, the  $AB$  set is filtered, and we take only the  $ab$  that satisfied the Equation 4.10. After the Competition for resources stage, it remains only  $ab$  that represents the background.

For minimizing the memory space, we have kept the average of Gaussians of  $AB$  set as one candidate memory cell ( $mc_{candidate}$ ).

If the Equation 4.12 is checked, the  $mc_{candidate}$  is introduced into  $MC$  set as a new background model and we compare the mean Affinity of  $mc_{match}$  and  $mc_{candidate}$  with  $AT \times ATS$  (Equation 4.13 and 4.14).

The  $mc_{match}$  is removed from the set of the background models ( $MC$ ) when the Equation 4.13 is satisfied.

### 4.3.3 Experimental validation and discussion

The proposed systems are implemented in Python on a computer with an Intel Core i7 and 8 GB memory capacity. This subsection presented the obtained results of the two

propositions cited above (Random Generation and Directed Generation) on the datasets: Wallflower [TKBM99], and UCSD [SND11]. After several empirically tests, the parameters of the systems, the learning rate, the minimum part of the data corresponding to the background, *Clonal\_rate*, *Hyper\_mutation\_rate*, *mutation\_rate*, *ATS* are set respectively to 0.01, 0.3, 10, 2, 0.1, 0.2.

#### 4.3.3.1 Qualitative results

Table 4.12 and Table 4.13 show the qualitative results of our two propositions on the Wallflower and UCSD datasets. Results of the Wallflower dataset are compared with the ground-truth images and with some public methods. Since there are some missing ground-truth images in the UCSD dataset, we have compared our results with real images acquired from the videos sequences. Qualitative results do not express the effectiveness of systems; for this purpose, quantitative tests are required to bravely demonstrate the robustness of our systems.

#### 4.3.3.2 Quantitative results

To improve the effectiveness of our propositions on Wallflower dataset, three metrics are taken into consideration: Recall, precision and F-measure. These metrics are the most popular and the most used in the pattern recognition and information extraction with binary classification ([MKS<sup>+</sup>99], [Vic17]). The obtained results are calculated using the ground-truth images on the evaluation frames, which are identified in Wallflower dataset.

Table 4.11: Comparison of quantitative results with well-known background subtraction methods on Wallflower dataset

	Error Type	MO	TD	LS	WT	Ca	Boo	FA	Total	Total Errors
SG[WADP97a]	FN	0	949	1857	3110	4101	2215	3464	15696	351339
	FP	0	535	1512	357	2040	92	1290	19437	
MOG [SG99]	FN	0	1008	1633	1323	398	1874	2442	8678	27053
	FP	0	20	14169	341	3098	217	530	18375	
KDE [EHD00]	FN	0	1298	760	170	238	1755	2413	6634	26450
	FP	0	125	14153	589	3392	933	624	19816	
SL-ICA [TL08]	FN	0	1199	1557	3372	3054	2560	2721	14463	15308
	FP	0	0	210	148	43	16	428	845	
SL-INMF [BG07]	FN	0	724	1593	3317	6626	1401	3412	17073	19098
	FP	0	481	303	652	234	190	165	2025	
SL-IRT [LHZZ08]	FN	0	1282	2822	4525	1491	1734	2438	14292	17053
	FP	0	159	1512	7	114	2080	12	2761	
GMM and Block Matching [FKSA17]	FN	0	64	88	367	184	187	89	979	3920
	FP	0	90	605	783	645	457	361	2941	
GMM with MRF [SW06]	FN	0	47	204	15	16	1060	34	1376	3808
	FP	0	402	546	311	467	102	604	2432	
Random Generation	FN	0	348	527	3	18	542	768	2224	37293
	FP	0	15686	15686	2000	460	430	815	35069	
Directed Generation	FN	0	758	1087	60	140	709	227	2981	5180
	FP	0	202	712	173	185	424	503	2199	

Table 4.12: Qualitative results on Wallflower dataset

	MO	TD	LS	WT	Ca	Bo	FA
Tests images							
Ground-Truth							
SG [WADP97a]							
MOG [SG99]							
KDE [EHD00]							
SL-ICA [TL08]							
SL-INMF [BG07]							
SL-IRT [LHZZ08]							
GMM and B-Matching [FKSA17]							
GMM with MRF [SW06]							
Random Generation							
Directed Generation							

Table 4.13: Qualitative results on UCSD dataset

Video Frame	Birds #63	Boats #25	Bottle #11	Chopper #51	Cyclists #23	Flock #171
Test Images						
Random Generation						
Directed Generation						
Video Frame	Freeway #41	Hockey #46	Jump #68	Landing #32	Ocean #151	Peds #100
Test Images						
Random Generation						
Directed Generation						
Video Frame	Rain #192	Skiing #59	Surf #63	Surfers #4	Traffic #102	Zodiac #126
Test Images						
Random Generation						
Directed Generation						

### 4.3.3.3 Discussions

We will discuss in this section the results of our propositions (Random Generation and Directed Generation) on the Wallflower and UCSD datasets.

In Wallflower dataset, proposition one (Random Generation) has achieved good results in the Moved Object, Foreground Aperture, Camouflage, Bootstarp, Waving Trees videos, but it has reported some false detections. Furthermore, this proposition has failed in the scenes, which have a gradual change in brightness like Light Switch and Time of Day videos.

For the UCSD dataset, proposition one has reached generally acceptable results. All the moving objects are detected, but in the scenes, which have progressive pixel variations proposition, one has achieved poor detection quality. This is due to the random generation of Gaussians.

Proposition two has achieved a good detection rate in most videos of the Wallflower dataset. It is ranked in the third position compared to other methods with a total error of 5180.

This proposition (Directed Generation) has reduced the drawbacks of the Random Generation when scenes have a large change in illumination; this is due to the mechanism that generates new Gaussian models only for pixels that represent the background.

In the UCSD dataset, Directed Generation has reached good results. It has a good detection quality compared with proposition one. All the moving objects have been well detected, and the false detections of the Random Generation are reduced in: Birds, Flock and Zodiac videos. But, in Hockey and Jump videos, Directed Generation has some false detections; this is due to the nature of the videos, which does not contain a sufficient number of samples for learning the system.

## 4.4 Conclusion

In this chapter, we have proposed new background subtraction methods based on the hybridization of the single Gaussian (SG) method and Gaussian Mixture Model (GMM) with the Artificial Immune Recognition System (AIRS) approach. This combination allowed us to solve problems related to:

- The detection of the moving objects when the background is dynamic using the SG model.
- The automatic management of Gaussian numbers in the GMM method.

The obtained results on public datasets (CDnet 2014, Wallflower (WavingTrees), Water-Surface, UCSD and Fountain) have proven the effectiveness of these propositions. Noted that these propositions can achieve more efficient results if we use more features. Indeed, we have used only one component ( $H$ ) of the color space HSV.

Despite the fast variations of background pixels are well modelled, and the variations related to light are reduced, but one feature remains insufficient to better detect moving

objects since our objective is to propose new background subtraction methods and not the selection of good discriminator features.



## General conclusion

In our thesis, we have used data mining techniques for subtracting the background in videos. Especially we have used the Artificial Immune Recognition System approach to create intelligent video surveillance systems addressed to the detection of moving objects in scenes captured from a fixed camera. Following this conclusion, we try to give an overview of the problems treated in this thesis and to propose some perspectives. As indicated in the general introduction, our thesis is divided into four chapters:

The first chapter is dedicated to state of art on different techniques of Data Mining. This study allowed us to identify the advantages and drawbacks of each technique. Furthermore, we have taken a view on the recent improvements proposed to overcome the problems of Data Mining techniques. This study has also engendered new taxonomy for Data Mining techniques.

The second chapter provided in-depth research on the recent methods used for subtracting the background in videos. Through this research, we have got an idea about the background subtraction domain, problems related to it and the methods which are proposed to subtract the background.

In the third chapter, we have given an overview of the Artificial Immune Recognition System (AIRS). We have discussed the different stages of the AIRS, their drawbacks and the modifications which we have introduced to it to overcome their drawbacks.

As many Data Mining techniques, AIRS suffer from some disadvantages that influence their performance in term of data explosion, results quality and calculation cost. To reduce the research cost on the most representative memory cell, we have redefined the structure of the memory cells set (MC) as a binary tree (kd-tree). We have also introduced two new mechanisms to the basic AIRS to avoid the explosion of data in MC set.

In the last chapter, we have proposed two new background subtraction systems that can detect moving objects from video captured with a fixed camera using the Artificial Immune Recognition System (AIRS). The single Gaussian method (SG) is combined with AIRS to better represent pixel variations in the scenes that contain dynamic background. AIRS is used as a classification tool that separates the antigens represented by the foreground pixels from antibodies that modelled the background pixels. Each pixel in this proposition is modelled with a feature vector contains Gaussian attributes.

In the second contribution, we have used the AIRS for managing the number of Gaussians dynamically in Gaussian Mixture Model (GMM) instead to fix them a priori by the user. In this contribution, a set of new Gaussians is generated using two different strategies: the first one (Random generation) uses the AIRS for improving the system decision, while in the second strategy (Directed generation), the AIRS is used to improve the production of background models.

To reduce the effect of brightness, we have firstly transformed each frame from the RGB to HSV color space.

The proposed systems are implemented and tested on public datasets. The obtained results are largely satisfactory compared to other state-of-the-art methods.

It is really that the fast variations of the background pixels are well modelled, and the light effect is reduced, but the use of one component ( $H$ ) of the HSV color space remains insufficient to better detect moving objects. Our propositions can achieve more efficient results if we use more features. Noted that we have focused in our work to propose new background subtraction methods, and we have not focused on the selection of good discriminator features.

In future work, we will try to increase the number of features used, and we will exploit other color spaces like HSL or Lab color space to obtain additional discriminative power. We can also extend our methodology and our improved Artificial Immune Recognition System in other multimedia applications, such as image processing.

# Bibliography

- [ABZ07] Mohand Said Allili, Nizar Bouguila, and Djemel Ziou. Finite generalized gaussian mixture modeling and applications to image and video foreground segmentation. In *Fourth Canadian Conference on Computer and Robot Vision (CRV'07)*, pages 183–190. IEEE, 2007.
- [ACZC06] Mohamed F Abdelkader, Rama Chellappa, Qinfen Zheng, and Alex L Chan. Integrated motion detection and tracking for visual surveillance. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 28–28. IEEE, 2006.
- [AM03] Joaquín Abellán and Serafín Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225, 2003.
- [AO13] CéSar A Astudillo and B John Oommen. On achieving semi-supervised pattern recognition by utilizing tree-based soms. *Pattern Recognition*, 46(1):293–304, 2013.
- [AO17] Barakat Saeed Alshamrani and Ahmed Hamza Osman. Investigation of hepatitis disease diagnosis using different types of neural network algorithms. *International Journal of Computer Science and Network Security (IJCSNS)*, 17(2):242, 2017.
- [ASA<sup>+</sup>11] KA Ahmad, Z Saad, Noramalina Abdullah, Z Hussain, and MH Mohd Noor. Moving vehicle segmentation in a dynamic background using self-adaptive kalman background method. In *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, pages 439–442. IEEE, 2011.
- [ASS14] Seza Adjoyan, Abdelhak-Djamel Seriai, and Anas Shatnawi. Service identification based on quality metrics object-oriented legacy system migration towards soa. In *SEKE: Software Engineering and Knowledge Engineering*, pages 1–6. Knowledge Systems Institute Graduate School, 2014.

- [AW19] Su Su Aung and Nu War. Foreground objects segmentation in videos with improved codebook model. In *2019 International Conference on Advanced Information Technologies (ICAIT)*, pages 161–166. IEEE, 2019.
- [AYIR17] Azlin Ahmad, Rubiyah Yusoff, Mohd Najib Ismail, and Nenny Ruthfylydia Rosli. Clustering the imbalanced datasets using modified kohonen self-organizing map (ksom). In *2017 Computing Conference*, pages 751–755. IEEE, 2017.
- [BAH13] Fadl Mutaher Ba-Alwi and Houzifa M Hintaya. Comparative study for analysis the prognostic in hepatitis data: data mining approach. *spinal cord*, 11:12, 2013.
- [BBJO<sup>+</sup>18] André L Brun, Alceu S Britto Jr, Luiz S Oliveira, Fabricio Enembreck, and Robert Sabourin. A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recognition*, 76:175–190, 2018.
- [BCS17] Simone Bianco, Gianluigi Ciocca, and Raimondo Schettini. Combination of video change detection algorithms by genetic programming. *IEEE Transactions on Evolutionary Computation*, 21(6):914–928, 2017.
- [BDR18] Mohammadreza Babaei, Duc Tung Dinh, and Gerhard Rigoll. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, 76:635–649, 2018.
- [BG07] Serhat S Bucak and Bilge Günsel. Video content representation by incremental non-negative matrix factorization. In *2007 IEEE International Conference on Image Processing*, volume 2, pages II–113. IEEE, 2007.
- [BHS97] Bernd Bullnheimer, Richard F Hartl, and Christine Strauss. A new rank based version of the ant system. a computational study. 1997.
- [BIR04] EDITED BY DAVID BIRNBAUM. Application of data mining techniques to healthcare data. *Infection control and hospital epidemiology*, 2004.
- [BJN<sup>+</sup>19] Xuan-Nam Bui, Pirat Jaroonpattanapong, Hoang Nguyen, Quang-Hieu Tran, and Nguyen Quoc Long. A novel hybrid model for predicting blast-induced ground vibration based on k-nearest neighbors and particle swarm optimization. *Scientific reports*, 9(1):1–14, 2019.

- [BK02] Christian Borgelt and Rudolf Kruse. Learning from imprecise data: possibilistic graphical models. *Computational statistics & data analysis*, 38(4):449–463, 2002.
- [Bou12] Thierry Bouwmans. Background subtraction for visual surveillance: A fuzzy approach. *Handbook on soft computing for video surveillance*, 5:103–138, 2012.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [CBSDW06] Sascha Cvetkovic, Peter Bakker, Johan Schirris, and Peter Hn De With. Background estimation and adaptation model with light-change removal for heavily down-sampled video surveillance signals. In *2006 International Conference on Image Processing*, pages 1829–1832. IEEE, 2006.
- [CDD<sup>+</sup>19] Jiangning Chen, Zhibo Dai, Juntao Duan, Heinrich Matzinger, and Ionel Popescu. Improved naive bayes with optimal correlation factor for text classification. *SN Applied Sciences*, 1(9):1129, 2019.
- [CDM20] Barbara Cardone and Ferdinando Di Martino. A novel fuzzy entropy-based method to improve the performance of the fuzzy c-means algorithm. *Electronics*, 9(4):554, 2020.
- [CFD10] Pei-Chann Chang, Chin-Yuan Fan, and Wei-Yuan Dzan. A cbr-based fuzzy decision tree approach for database classification. *Expert Systems with Applications*, 37(1):214–225, 2010.
- [CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- [CKB14] Myungrae Cha, Jun Seok Kim, and Jun-Geol Baek. Density weighted support vector data description. *Expert Systems with Applications*, 41(7):3343–3350, 2014.
- [CLK<sup>+</sup>00] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, 2000:1–68, 2000.
- [CLL20] Elisa Cabana, Rosa E Lillo, and Henry Laniado. Robust regression based on shrinkage with application to living environment deprivation. *Stochastic Environmental Research and Risk Assessment*, pages 1–18, 2020.

- [CMM07] Tao Chen, Julian Morris, and Elaine Martin. Gaussian process regression for multivariate spectroscopic calibration. *Chemometrics and Intelligent Laboratory Systems*, 87(1):59–71, 2007.
- [CMM19] Jianzhao Cao, Ruwei Ma, and Oloro Michael. A fast background model using kernel density estimation and distance transform. *International Journal of Modelling, Identification and Control*, 32(2):135–144, 2019.
- [CSN10] Theekapun Charoenpong, Ajaree Supasuteekul, and Chaiwat Nuthong. Adaptive background modeling from an image sequence by using k-means clustering. In *ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 880–883. IEEE, 2010.
- [CTY<sup>+</sup>10] Hakan Cevikalp, Bill Triggs, Hasan Serhan Yavuz, Yalçın Küçük, Mahide Küçük, and Atalay Barkana. Large margin classifiers based on affine hulls. *Neurocomputing*, 73(16-18):3160–3168, 2010.
- [CW04] Jun Chen and Mark Wineberg. Enhancement of the shifting balance genetic algorithm for highly multimodal problems. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 744–751. IEEE, 2004.
- [CXW11] Xue-Fang Chen, Hong-Jie Xing, and Xi-Zhao Wang. A modified adaboost method for one-class svm and its application to novelty detection. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3506–3511. IEEE, 2011.
- [DCVZ00] L Nunes De Casto and Fernando J Von Zuben. An evolutionary immune network for data clustering. In *Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks*, pages 84–89. IEEE, 2000.
- [DCVZ02] Leandro N De Castro and Fernando J Von Zuben. Learning and optimization using the clonal selection principle. *IEEE transactions on evolutionary computation*, 6(3):239–251, 2002.
- [DD20] Amrita Datta and Mou Dasgupta. On accurate localization of sensor nodes in underwater sensor networks: a doppler shift and modified genetic algorithm based localization technique. *Evolutionary Intelligence*, pages 1–13, 2020.
- [DDC99] Marco Dorigo and Gianni Di Caro. Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, pages 1470–1477. IEEE, 1999.

- [DG96] Marco Dorigo and Luca Maria Gambardella. A study of some properties of ant-q. In *International Conference on Parallel Problem Solving from Nature*, pages 656–665. Springer, 1996.
- [DG97] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997.
- [DGG17] Massimo De Gregorio and Maurizio Giordano. Wisardrp for change detection in video sequences. In *ESANN*, 2017.
- [DHJ<sup>+</sup>19] Enzeng Dong, Bo Han, Hao Jian, Jigang Tong, and Zenghui Wang. Moving target detection based on improved gaussian mixture model considering camera motion. *Multimedia Tools and Applications*, pages 1–16, 2019.
- [DHLDSY19] Jiang Dai-Hong, Dai Lei, Li Dan, and Zhang San-You. Moving-object tracking algorithm based on pca-sift and optimization for underground coal mines. *IEEE Access*, 7:35556–35563, 2019.
- [DHYD18] Enzeng Dong, Bo Han, Xiao Yu, and Shengzhi Du. Moving targets detection based on improved single gaussian background model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1179–1183. IEEE, 2018.
- [DJM12] Wlodzislaw Duch, Norbert Jankowski, and Tomasz Maszczyk. Make it cheap: learning with  $o(nd)$  complexity. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4. IEEE, 2012.
- [DK03] Doulaye Dembélé and Philippe Kastner. Fuzzy c-means method for clustering microarray data. *bioinformatics*, 19(8):973–980, 2003.
- [Doe18] Is see5/c5.0 better than c4.5? <http://rulequest.com/see5-comparison.html>, Last accessed on 14-04-2018.
- [DP97] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.
- [Du20] Ziping Du. Energy analysis of internet of things data mining algorithm for smart green communication networks. *Computer Communications*, 152:223–231, 2020.
- [Duc00a] W Duch. Logical rules extracted from data, 2000.

- [Duc00b] Wlodzislaw Duch. Datasets used for classification: comparison of results. <http://158.75.5.90/kmk/projects/datasets.html>, 2000.
- [Dun73] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.
- [EAEE20] FA Essa, Mohamed Abd Elaziz, and Ammar H Elsheikh. An enhanced productivity prediction model of active solar still using artificial neural network and harris hawks optimizer. *Applied Thermal Engineering*, 170:115020, 2020.
- [EBBV09] Fida El Baf, Thierry Bouwmans, and Bertrand Vachon. Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 60–65. IEEE, 2009.
- [EHD00] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer, 2000.
- [EKC<sup>+</sup>17] Khalil Elkhilil, Abla Kammoun, Romain Couillet, Tareq Y Al-Naffouri, and Mohamed-Slim Alouini. Asymptotic performance of regularized quadratic discriminant analysis based classifiers. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.
- [ET17] Ömer Faruk Ertuğrul and Mehmet Emin Tağluk. A novel version of k nearest neighbor: Dependent nearest neighbor. *Applied Soft Computing*, 55:480–490, 2017.
- [FA20] Paulo HM Ferreira and Aluizio FR Araújo. Growing self-organizing maps for nonlinear time-varying function approximation. *Neural Processing Letters*, pages 1–26, 2020.
- [Fan19] Jiancong Fan. Ope-hca: an optimal probabilistic estimation approach for hierarchical clustering algorithm. *Neural Computing and Applications*, 31(7):2095–2105, 2019.
- [FAR16] Mr Brahim FAROU. *Multimédia mining Reconnaissance des formes dans une vidéo*. PhD thesis, Université Badji Mokhtar-Annaba, 2016.
- [ff121] La vidÉo surveillance. <https://sctisecurite.fr/service/video-surveillance/>, Last accessed on 28-01-2021.



- [ff21] La sécurité dans les lieux publics. <http://www.videosurveillance-home.com/lieux-publics.html>, Last accessed on 28-01-2021.
- [FJSS19] S Famila, A Jawahar, A Sariga, and K Shankar. Improved artificial bee colony optimization based clustering algorithm for smart sensor environments. *Peer-to-Peer Networking and Applications*, pages 1–9, 2019.
- [FKSA17] Brahim Farou, Med Nadjib Kouahla, Hamid Seridi, and Herman Akdag. Efficient local monitoring approach for the task of background subtraction. *Engineering Applications of Artificial Intelligence*, 64:1–12, 2017.
- [FMM<sup>+</sup>08] Carlos Fernandes, Antonio Miguel Mora, Juan Julian Merelo, Vitorino Ramos, and Juan Luis Jiménez Laredo. Kohonants: a self-organizing ant algorithm for clustering and pattern classification. *arXiv preprint arXiv:0803.2695*, 2008.
- [FMT<sup>+</sup>18] Radhia Fezai, Majdi Mansouri, Okba Taouali, Mohamed Faouzi Harkat, and Nasreddine Bouguila. Online reduced kernel principal component analysis for process monitoring. *Journal of Process Control*, 61:1–11, 2018.
- [FPAC94] Stephanie Forrest, Alan S Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-nonsel self discrimination in a computer. In *Proceedings of 1994 IEEE computer society symposium on research in security and privacy*, pages 202–212. Ieee, 1994.
- [Fri89] Jerome H Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.
- [FS95] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [GA19] Ferdos Gorji and Mina Aminghafari. Robust nonparametric regression for heavy-tailed data. *Journal of Agricultural, Biological and Environmental Statistics*, pages 1–15, 2019.
- [GBW02] Donald E Goodman, Lois Boggess, and Andrew Watkins. Artificial immune system classification of multiple-class problems. *Proceedings of the artificial neural networks in engineering ANNIE*, 2:179–183, 2002.
- [GFM19] Edwin F Galutira, Arnel C Fajardo, and Ruji P Medina. A novel kohonen self-organizing maps using exponential decay average rate of change for

- color clustering. In *Intelligent and Interactive Computing*, pages 23–33. Springer, 2019.
- [GG10] Yunlong Gao and Feng Gao. Edited adaboost by weighted knn. *Neuro-computing*, 73(16-18):3079–3088, 2010.
- [GJ20] Jeffin Gracewell and Mala John. Dynamic background modeling using deep learning autoencoder network. *Multimedia Tools and Applications*, 79(7):4639–4659, 2020.
- [GMB<sup>+</sup>19] Zhiqiang Geng, Qingchao Meng, Ju Bai, Jie Chen, Yongming Han, Qin Wei, and Zhi Ouyang. A model-free bayesian classifier. *Information Sciences*, 482:171–188, 2019.
- [GZ01] Dashan Gao and Jie Zhou. Adaptive background estimation for real-time traffic monitoring. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pages 330–333. IEEE, 2001.
- [HA18] Nima Shiri Harzevili and Sasan H Alizadeh. Mixture of latent multinomial naive bayes classifier. *Applied Soft Computing*, 69:516–527, 2018.
- [HA19] D Jude Hemanth and J Anitha. Modified genetic algorithm approaches for classification of abnormal magnetic resonance brain tumour images. *Applied Soft Computing*, 75:21–28, 2019.
- [HAR<sup>+</sup>17] Nursuci Putri Husain, Nursanti Novi Arisa, Putri Nur Rahayu, Agus Zainal Arifin, and Darlis Herumurti. Least squares support vector machines parameter optimization based on improved ant colony algorithm for hepatitis diagnosis. *Jurnal Ilmu Komputer dan Informasi*, 10(1):43–49, 2017.
- [Heb49] DO Hebb. Organization of behavior. new york: Wiley. *J. Clin. Psychol*, 6(3):335–307, 1949.
- [HEOI17] Sara Omer Hussien, Sara Sir Elkhatem, Nisreen Osman, and Ashraf Osman Ibrahim. A review of data mining techniques for diagnosing hepatitis. In *2017 Sudan Conference on Computer Science and Information Technology (SCCSIT)*, pages 1–6. IEEE, 2017.
- [HGM10] Anwaar-ul Haq, Iqbal Gondal, and M Murshed. Automated multi-sensor color video fusion for nighttime video surveillance. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 529–534. IEEE, 2010.

- [HHD00] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W/sup 4: real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, 2000.
- [HK12] Seaung Lok Ham and Nojun Kwak. Boosted-pca for binary classification problems. In *2012 IEEE International Symposium on Circuits and Systems*, pages 1219–1222. IEEE, 2012.
- [HM20] Hadeef Hefaidh and Djebabra Mébarek. Using fuzzy-improved principal component analysis (pca-if) for ranking of major accident scenarios. *Arabian Journal for Science and Engineering*, 45(3):2235–2245, 2020.
- [HOOS16] Intisar Hussien, Sara Omer, Nour E Oweis, and Václav Snášel. Feature selection using semi discrete decomposition and singular value decompositions. In *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16)*, pages 87–97. Springer, 2016.
- [Hou19] Xiangru Hou. An improved k-means clustering algorithm based on hadoop platform. In *The International Conference on Cyber Security Intelligence and Analytics*, pages 1101–1109. Springer, 2019.
- [HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000.
- [HW03] Dongpyo Hong and Woontack Woo. A background subtraction for a vision-based user interface. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 1, pages 263–267. IEEE, 2003.
- [HYK<sup>+</sup>19] Wei He, K Yong, Wan Kim, Hak-Lim Ko, Jianhui Wu, Wujing Li, and Bing Tu. Local compact binary count based nonparametric background modeling for foreground detection in dynamic scenes. *IEEE Access*, 7:92329–92340, 2019.
- [IGTP05] Codrut Ianasi, Vasile Gui, Corneliu I Toma, and Dan Pescaru. A fast algorithm for background tracking in video surveillance, using nonparametric kernel density estimation. *Facta universitatis-series: Electronics and Energetics*, 18(1):127–144, 2005.

- [IÖGG18] Şahin Işık, Kemal Özkan, Serkan Günal, and Ömer Nezih Gerek. Swcd: a sliding window and self-regulated learning-based background updating method for change detection in videos. *Journal of Electronic Imaging*, 27(2):023002, 2018.
- [ISN16] MohammadNoor Injadat, Fadi Salo, and Ali Bou Nassif. Data mining techniques in social media: A survey. *Neurocomputing*, 214:654–670, 2016.
- [Jer74] Niels K Jerne. Towards a network theory of the immune system. *Ann. Immunol.*, 125:373–389, 1974.
- [JHL<sup>+</sup>19] Haijin Ji, Song Huang, Xuewei Lv, Yaning Wu, and Yuntian Feng. Empirical studies of a kernel density estimation based naive bayes method for software defect prediction. *IEICE TRANSACTIONS on Information and Systems*, 102(1):75–84, 2019.
- [JK13] P Jaganathan and R Kuppuchamy. A threshold fuzzy entropy based feature selection for medical database classification. *Computers in Biology and Medicine*, 43(12):2222–2229, 2013.
- [JLWZ16] Liangxiao Jiang, Chaoqun Li, Shasha Wang, and Lungan Zhang. Deep feature weighting for naive bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, 52:26–39, 2016.
- [JMS17] Ruholla Jafari-Marandi and Brian K Smith. Fluid genetic algorithm (fga). *Journal of Computational Design and Engineering*, 4(2):158–167, 2017.
- [JWW<sup>+</sup>17] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):352–364, 2017.
- [Kar05] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Citeseer, 2005.
- [KCHD04] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 5, pages 3061–3064. IEEE, 2004.

- [KE95] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [KHS06] Satoshi Kawabata, Shinsaku Hiura, and Kosuke Sato. Real-time detection of anomalous objects in dynamic scene. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1171–1174. IEEE, 2006.
- [KK16] Rawitas Krungkaew and Worapan Kusakunniran. Foreground segmentation in a video by using a novel dynamic codebook. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016 13th International Conference on*, pages 1–6. IEEE, 2016.
- [KKGK14] C Barath Kumar, M Varun Kumar, T Gayathri, and S Rajesh Kumar. Data analysis and prediction of hepatitis using support vector machine (svm). *IJCSIT*, 5(2):2235–2237, 2014.
- [KKRS04] J-R Kurz-Kim, Svetlozar T Rachev, and Gennady Samorodnitsky. Asymptotic distribution of unbiased linear estimators in the presence of heavy-tailed stochastic regressors and residuals. Technical report, Cornell University Operations Research and Industrial Engineering, 2004.
- [KKS20] Meeta Kumar, Anand J Kulkarni, and Suresh Chandra Satapathy. A hybridized data clustering for breast cancer prognosis and risk exposure using fuzzy c-means and cohort intelligence. In *Optimization in Machine Learning and Applications*, pages 113–126. Springer, 2020.
- [KLL13] David Koloseni, Jouni Lampinen, and Pasi Luukka. Differential evolution based nearest prototype classifier with optimized distance measures for the features in the data sets. *Expert Systems with Applications*, 40(10):4075–4082, 2013.
- [KLLH19] Adobah Benjamin Kwame, Guohai Liu, Hui Liu, and Fida Hussain. Real-time recognition and tracing of moving objects in video images using background subtraction, kalman filter and particle filter. In *2019 25th International Conference on Automation and Computing (ICAC)*, pages 1–5. IEEE, 2019.
- [KM96] Yoshinari Kameda and Michihiko Minoh. A human motion estimation method using 3-successive video frames. In *International conference on virtual systems and multimedia*, pages 135–140, 1996.

- [KM99] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.
- [Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.
- [KPLB05] A Homayoun Kamkar-Parsi, Robert Laganière, and Martin Bouchard. A multi-criteria model for robust foreground extraction. In *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 67–70, 2005.
- [KR11] Seoung Bum Kim and Panaya Rattakorn. Unsupervised feature selection using weighted principal components. *Expert systems with applications*, 38(5):5704–5710, 2011.
- [KSK<sup>+</sup>08] Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama, and Kiyoshi Kogure. Background subtraction using generalised gaussian family model. *Electronics letters*, 44(3):189–190, 2008.
- [KT13] T Karthikeyan and P Thangaraju. Analysis of classification algorithms applied to hepatitis patients. *International Journal of Computer Applications*, 62(15), 2013.
- [LBC11] Miguel López-Benítez and Fernando Casadevall. Versatile, accurate, and analytically tractable approximation for the gaussian q-function. *IEEE Transactions on Communications*, 59(4):917–922, 2011.
- [LC15] Chee Kau Lim and Chee Seng Chan. A weighted inference engine based on interval-valued fuzzy relational theory. *Expert Systems with Applications*, 42(7):3410–3419, 2015.
- [LCAG20] Ritchie Lee, Justin Clarke, Adrian Agogino, and Dimitra Gianakopoulou. Improving trust in deep neural networks with nearest neighbors. In *AIAA Scitech 2020 Forum*, page 2098, 2020.
- [LCJ<sup>+</sup>20] Wen Long, Shaohong Cai, Jianjun Jiao, Ming Xu, and Tiebin Wu. A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. *Energy Conversion and Management*, 203:112243, 2020.
- [LCR18] Jing Liu, Pamela C Cosman, and Bhaskar D Rao. Robust linear regression via  $l_0$  regularization. *IEEE Transactions on Signal Processing*, 66(3):698–713, 2018.

- [lCYjW<sup>+</sup>14] Hui ling Chen, Bo Yang, Su jing Wang, Gang Wang, Huai zhong Li, Wen bin Liu, et al. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy. *Applied Mathematics and Computation*, 239:180–197, 2014.
- [LHGT04] Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.
- [LHI<sup>+</sup>15] Dong Liang, Manabu Hashimoto, Kenji Iwata, Xinyue Zhao, et al. Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. *Pattern Recognition*, 48(4):1374–1390, 2015.
- [LHZ19] Kang LV, Tang-qing HU, and Xu-xiu ZHANG. Improved motion detection algorithm based on hsv codebook model. *Journal of Dalian Minzu University*, (1):17, 2019.
- [LHZZ08] Xi Li, Weiming Hu, Zhongfei Zhang, and Xiaoqin Zhang. Robust foreground segmentation based on two effective background models. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 223–228, 2008.
- [Li04] Yongmin Li. On incremental and robust subspace learning. *Pattern recognition*, 37(7):1509–1518, 2004.
- [LK18a] Long Ang Lim and Hacer Yalim Keles. Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, 112:256–262, 2018.
- [LK18b] Long Ang Lim and Hacer Yalim Keles. Learning multi-scale features for foreground segmentation. *arXiv preprint arXiv:1808.01477*, 2018.
- [LKS<sup>+</sup>18] Sang-Ha Lee, Soon-Chul Kwon, Jin-Wook Shim, Jeong-Eun Lim, and Jisang Yoo. Wisenetmd: Motion detection using dynamic background region analysis. *arXiv preprint arXiv:1805.09277*, 2018.
- [LL10] Der-Chiang Li and Chiao-Wen Liu. A class possibility based kernel to increase classification accuracy for small data sets using support vector machines. *Expert Systems with Applications*, 37(4):3104–3110, 2010.
- [LLC02] Horng-Horng Lin, Tyng-Luh Liu, and Jen-Hui Chuang. A probabilistic svm approach for background scene initialization. In *Proceedings. In-*

- ternational Conference on Image Processing*, volume 3, pages 893–896. IEEE, 2002.
- [LLZX18] Qingbo Li, Wenjie Li, Jialin Zhang, and Zhi Xu. An improved k-nearest neighbour method to diagnose breast cancer. *Analyst*, 143(12):2807–2811, 2018.
- [LNDA12] Laurence A Gan Lim, Raouf NG Naguib, Elmer P Dadios, and Jose Maria C Avila. Implementation of ga-ksom and anfis in the classification of colonic histopathological images. In *TENCON 2012-2012 IEEE Region 10 Conference*, pages 1–5. IEEE, 2012.
- [LR10] Xiaodong Liu and Yan Ren. Novel artificial intelligent techniques via afs theory: Feature selection, concept categorization and characteristic description. *Applied Soft Computing*, 10(3):793–805, 2010.
- [LRMCLBD18] Ezequiel López-Rubio, Miguel A Molina-Cabello, Rafael Marcos Luque-Baena, and Enrique Domínguez. Foreground detection by competitive learning for varying input distributions. *International journal of neural systems*, 28(05):1750056, 2018.
- [LSGX09] Huailiang Liu, Ruijuan Su, Ying Gao, and Ruoning Xu. Improved particle swarm optimization using two novel parallel inertia weights. In *Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on*, volume 1, pages 185–188. IEEE, 2009.
- [LXD19] Mujin Li, Honghui Xu, and Yong Deng. Evidential decision tree based on belief entropy. *Entropy*, 21(9):897, 2019.
- [LZ04] Stan Z Li and ZhenQiu Zhang. Floatboost learning and statistical face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(9):1112–1123, 2004.
- [LZ10] Fei Lei and Xiaoxia Zhao. Adaptive background estimation of underwater using kalman-filtering. In *2010 3rd International Congress on Image and Signal Processing*, volume 1, pages 64–67. IEEE, 2010.
- [LZZ<sup>+</sup>19] Kewen Li, Guangyue Zhou, Jiannan Zhai, Fulai Li, and Mingwen Shao. Improved pso\_adaboost ensemble algorithm for imbalanced data. *Sensors*, 19(6):1476, 2019.
- [LZZ20] Bin Liu, Tianshu Zhao, and Ranxia Zhang. Prediction of private car ownership in china based on the improved pca-logistic model. In *5th*



- International Conference on Social Sciences and Economic Development (ICSSED 2020)*, pages 353–358. Atlantis Press, 2020.
- [M<sup>+</sup>67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [Mah06] Shyjan Mahamud. Comparing belief propagation and graph cuts for novelty detection. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1154–1159. IEEE, 2006.
- [Mat02] Polly Matzinger. The danger model: a renewed sense of self. *science*, 296(5566):301–305, 2002.
- [MB12] Madhusmita Mishra and HS Behera. Kohonen self organizing map with modified k-means clustering for high dimensional data set. *International Journal of Applied Information Systems*, 2(3):34–39, 2012.
- [MGN19] Brian E Moore, Chen Gao, and Raj Rao Nadakuditi. Panoramic robust pca for foreground–background separation on noisy, free-motion camera video. *IEEE Transactions on Computational Imaging*, 5(2):195–211, 2019.
- [MGRNO19] V Mondéjar-Guerra, J Rouco, J Novo, and M Ortega. An end-to-end deep learning approach for simultaneous background modeling and subtraction. In *British Machine Vision Conference (BMVC)*, Cardiff, 2019.
- [MH06] Yoshihiro Mitani and Yoshihiko Hamamoto. A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 27(10):1151–1159, 2006.
- [Mir15] Seyedali Mirjalili. How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1):150–161, 2015.
- [MKS<sup>+</sup>99] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, et al. Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop*, pages 249–252. Herndon, VA, 1999.
- [MM20] Vijaya Prabhagar Murugesan and Punniyamoorthy Murugesan. A new initialization and performance measure for the rough k-means clustering. *Soft Computing*, pages 1–15, 2020.
- [MML14] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

- [MMM18] Sebastián Maldonado, José Merigó, and Jaime Miranda. Redefining support vector machines with the ordered weighted average. *Knowledge-Based Systems*, 148:41–46, 2018.
- [MMN06] Davide A Migliore, Matteo Matteucci, and Matteo Naccari. A revaluation of frame difference in fast and robust motion detection. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 215–218, 2006.
- [MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [MP10] Lucia Maddalena and Alfredo Petrosino. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Computing and Applications*, 19(2):179–186, 2010.
- [MS04] Yan-Fen Mao and Peng-Fei Shi. Multimodal background model with noise and shadow suppression for moving object detection. *Journal of Southeast University (English Edition)*, 20(4):423–426, 2004.
- [MS15] Malay Mitra and RK Samanta. Hepatitis disease diagnosis using multiple imputation and neural network with rough set feature reduction. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, pages 285–293. Springer, 2015.
- [Nan09] Satyasai Jagannath Nanda. *Artificial immune systems: principle, algorithms and applications*. PhD thesis, 2009.
- [NHW<sup>+</sup>11] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems*, 50(3):559–569, 2011.
- [NKC11] Efendi Nasibov and Cagin Kandemir-Cavas. Owa-based linkage method in hierarchical clustering: Application on phylogenetic trees. *Expert Systems with Applications*, 38(10):12684–12690, 2011.
- [NSC<sup>+</sup>17] Aparajita Nanda, Pankaj Kumar Sa, Suman Kumar Choudhury, Sambit Bakshi, and Banshidhar Majhi. A neuromorphic person re-identification framework for video surveillance. *IEEE Access*, 5:6471–6482, 2017.

- [ORP00] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):831–843, 2000.
- [Özg07] Canan Özgen. *MULTI-CAMERA VIDEO SURVEILLANCE: DETECTION, OCCLUSION HANDLING, TRACKING AND EVENT RECOGNITION*. PhD thesis, MIDDLE EAST TECHNICAL UNIVERSITY, 2007.
- [Pal18] Luigi Leonardo Palese. A random version of principal component analysis in data clustering. *Computational biology and chemistry*, 73:57–64, 2018.
- [PDK16] Shubhangi D Patil, Ratnadeep R Deshmukh, and DK Kirange. Adaptive apriori algorithm for frequent itemset mining. In *System Modeling & Advancement in Research Trends (SMART), International Conference*, pages 7–13. IEEE, 2016.
- [PK17] Boris T Polyak and Mikhail V Khlebnikov. Robust principal component analysis: An irls approach. *IFAC-PapersOnLine*, 50(1):2762–2767, 2017.
- [PKY+08] Sinno Jialin Pan, James T Kwok, Qiang Yang, et al. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.
- [PMR10] Jorge J Palop, Lennart Mucke, and Erik D Roberson. Quantifying biomarkers of cognitive dysfunction and neuronal network hyperexcitability in mouse models of alzheimer’s disease: depletion of calcium-dependent proteins and inhibitory hippocampal remodeling. In *Alzheimer’s Disease and Frontotemporal Dementia*, pages 245–262. Springer, 2010.
- [PP18] Rupan Panja and Nikhil R Pal. Ms-svm: Minimally spanned support vector machine. *Applied Soft Computing*, 64:356–365, 2018.
- [PRM13] Mihir R Patel, Dipti P Rana, and Rupa G Mehta. Fapriori: A modified apriori algorithm based on checkpoint. In *Information Systems and Computer Networks (ISCON), 2013 International Conference on*, pages 50–53. IEEE, 2013.
- [QTYH13] Xueming Qian, Yuan Yan Tang, Zhe Yan, and Kaiyu Hang. Isaboost: A weak classifier inner structure adjusting based adaboost algorithm isaboost based application in scene categorization. *Neurocomputing*, 103:104–113, 2013.

- [Qui86] J Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- [Qui93] J Ross Quinlan. C4. 5: Programming for machine learning. *Morgan Kaufmann*, 38:48, 1993.
- [RACM16] Graciela Ramírez-Alonso and Mario I Chacón-Murguía. Auto-adaptive parallel som architecture with a modular analysis for dynamic object segmentation in videos. *Neurocomputing*, 175:990–1000, 2016.
- [RKW09] Chinta Rambabu, Kiyoung Kim, and Woontack Woo. Fast and accurate extraction of moving object silhouette for personalized virtual reality studio@ home. *Journal of Real-Time Image Processing*, 4(4):317, 2009.
- [RMK95] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer, 1995.
- [Sat11] G Sathyadevi. Application of cart algorithm in hepatitis disease diagnosis. In *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, pages 1283–1287. IEEE, 2011.
- [SC17] Hasan Sajid and Sen-Ching Samson Cheung. Universal multimode background subtraction. *IEEE Transactions on Image Processing*, 26(7):3249–3260, 2017.
- [SCBB15a] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. A self-adjusting approach to change detection based on background word consensus. In *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 990–997. IEEE, 2015.
- [SCBB15b] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2015.
- [SCJ17] Tao Sun, Lizhi Cheng, and Hao Jiang. Greedy method for robust linear regression. *Neurocomputing*, 243:125–132, 2017.
- [SCM17] Kamal Sehairi, Fatima Chouireb, and Jean Meunier. Comparative study of motion detection methods for video surveillance systems. *Journal of Electronic Imaging*, 26(2):023025, 2017.

- [SDLH14] José A Sáez, Joaquín Derrac, Julián Luengo, and Francisco Herrera. Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers. *Pattern Recognition*, 47(12):3941–3948, 2014.
- [SG99] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.
- [SH00] Thomas Stützle and Holger H Hoos. Max–min ant system. *Future generation computer systems*, 16(8):889–914, 2000.
- [Sha16] Ashish Shah. Association rule mining with modified apriori algorithm using top down approach. In *Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on*, pages 747–752. IEEE, 2016.
- [SJ10] Veit Schwämmle and Ole Nørregaard Jensen. A simple and fast method to determine the parameters for fuzzy c–means cluster analysis. *Bioinformatics*, 26(22):2841–2848, 2010.
- [SJ19] Rimbun Siringoringo and Jamaluddin Jamaluddin. Initializing the fuzzy c-means cluster center with particle swarm optimization for sentiment clustering. In *Journal of Physics: Conference Series*, volume 1361, page 012002. IOP Publishing, 2019.
- [SKCH17] Iqbal H Sarker, Muhammad Ashad Kabir, Alan Colman, and Jun Han. An improved naive bayes classifier-based noise detection technique for classifying user phone call behavior. In *Australasian Conference on Data Mining*, pages 72–85. Springer, 2017.
- [SL08] Danijel Skočaj and Aleš Leonardis. Incremental and robust learning of subspace representations. *Image and Vision Computing*, 26(1):27–38, 2008.
- [SLD<sup>+</sup>06] Paolo Spagnolo, M Leo, Arcangelo Distanto, et al. Moving object segmentation by background subtraction and temporal analysis. *Image and Vision Computing*, 24(5):411–423, 2006.
- [SLL20] Jiaqi Song, Fei Li, and Ruxiang Li. Improved k-means algorithm based on threshold value radius. In *IOP Conference Series: Earth and Environmental Science*, volume 428, page 012001. IOP Publishing, 2020.

- [SMM15] Shahrzad Saremi, Seyedeh Zahra Mirjalili, and Seyed Mohammad Mirjalili. Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications*, 26(5):1257–1263, 2015.
- [SND11] Satyam Srivastava, Ka Ki Ng, and Edward J Delp. Crowd flow estimation using multiple visual features for scenes with changing crowd densities. In *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 60–65. IEEE, 2011.
- [SO11] Yiyuan She and Art B Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011.
- [SQH<sup>+</sup>18] Jie Su, Linbo Qing, Xiaohai He, Hang Zhang, Jing Zhou, and Yonghong Peng. A new regularized matrix discriminant analysis (r-mds) enabled human-centered eeg monitoring systems. *IEEE Access*, 2018.
- [SS05] Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1778–1792, 2005.
- [SS13] Ruxandra Stoean and Catalin Stoean. Modeling medical decision making by support vector machines, explaining by rules of evolutionary algorithms with feature selection. *Expert Systems with Applications*, 40(7):2677–2686, 2013.
- [SSSS14] Aarti Sharma, Rahul Sharma, Vivek Kr Sharma, and Vishal Shrivatava. Application of data mining—a survey paper. *International Journal of Computer Science and Information Technologies*, 5(2):2023–2025, 2014.
- [Sun20] Li-na Sun. An improved apriori algorithm based on support weight matrix for data mining in transaction database. *Journal of Ambient Intelligence and Humanized Computing*, 11(2):495–501, 2020.
- [SW06] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *Asian Conference on Computer Vision*, pages 581–590. Springer, 2006.
- [SYH<sup>+</sup>19] Lei Shang, Fucheng You, Chao Han, Xuwei Wang, and Shuai Zhao. Optimization of three-frame difference method and improvement of pedestrian detection code book. In *Journal of Physics: Conference Series*, volume 1302, page 022014. IOP Publishing, 2019.

- [SYN<sup>+</sup>20] Moch Arief Soeleman, S Yogi, Aris Nurhindarto, W Karis, RA Pramunendar, et al. An improvement for background modelling using a mixture of gaussian and region growing in moving objects detection. In *Journal of Physics: Conference Series*, volume 1430, page 012032. IOP Publishing, 2020.
- [Tav05] A Tavakkoli. Automatic video object plane extraction using non-parametric kernel density estimation. *Mathematical Methods in Computer Vision, University of Nevada, Reno, NV*, 2005.
- [TD05] Dusan Teodorovic and Mauro Dell’Orco. Bee colony optimization—a cooperative learning approach to complex transportation problems. *Advanced OR and AI methods in transportation*, pages 51–60, 2005.
- [TG AJ15] Shweta Taneja, Charu Gupta, Sakshi Aggarwal, and Veni Jindal. Mfz knn a modified fuzzy based k nearest neighbor algorithm. In *Cognitive Computing and Information Processing (CCIP), 2015 International Conference on*, pages 1–5. IEEE, 2015.
- [TGL07] Peng Tang, Lin Gao, and Zhifang Liu. Salient moving object detection using stochastic approach filtering. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 530–535. IEEE, 2007.
- [THH<sup>+</sup>08] J Timmis, E Hart, A Hone, M Neal, A Robins, S Stepney, and A Tyrrell. *Immuno-engineering* (vol. 268), 2008.
- [TKBM99] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 1, pages 255–261. IEEE, 1999.
- [TL08] Du-Ming Tsai and Shia-Chih Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE Transactions on image processing*, 18(1):158–167, 2008.
- [TLL07] Xiao-jun Tan, Jun Li, and Chunlu Liu. A video-based real-time vehicle detection method by classified background learning. *World transactions on engineering and technology education*, 6(1):189, 2007.
- [TM09] Hui-Xin Tian and Zhi-Zhong Mao. An ensemble elm based on modified adaboost. rt algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Transactions on Automation Science and Engineering*, 7(1):73–80, 2009.

- [TMA14] Sina Tabakhi, Parham Moradi, and Fardin Akhlaghian. An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, 32:112–123, 2014.
- [TNB06] Alireza Tavakkoli, Mircea Nicolescu, and George Bebis. Robust recursive learning for foreground region detection in videos with quasi-stationary backgrounds. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 315–318. IEEE, 2006.
- [TNBN08] Alireza Tavakkoli, Mircea Nicolescu, George Bebis, and Monica Nicolescu. A support vector data description approach for background modeling in videos with quasi-stationary backgrounds. *International journal on artificial intelligence tools*, 17(04):635–658, 2008.
- [TNH99] Jon Timmis, Mark Neal, and John Hunt. Data analysis using artificial immune systems, cluster analysis and kohonen networks: some comparisons. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, volume 3, pages 922–927. IEEE, 1999.
- [TNH00] Jon Timmis, Mark Neal, and John Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1-3):143–150, 2000.
- [TSA14] Divya Tomar, Shubham Singhal, and Sonali Agarwal. Weighted least square twin support vector machine for imbalanced dataset. *International Journal of Database Theory and Application*, 7(2):25–36, 2014.
- [TSAT07] Tatsuya Tanaka, Atsushi Shimada, Daisaku Arita, and Rin-ichiro Taniguchi. A fast algorithm for adaptive background model construction using parzen density estimation. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 528–533. IEEE, 2007.
- [TTD<sup>+</sup>20] Dan Tang, Liu Tang, Rui Dai, Jingwen Chen, Xiong Li, and Joel JPC Rodrigues. Mf-adaboost: Ldos attack detection based on multi-features and improved adaboost. *Future Generation Computer Systems*, 2020.
- [UM17] S Umadevi and KS Jeen Marseline. A survey on data mining classification algorithms. In *2017 International Conference on Signal Processing and Communication (ICSPC)*, pages 264–268. IEEE, 2017.
- [Vap63] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.



- [Vap98] Vladimir Naumovich Vapnik. Vladimirvapnik."statistical learning theory", vol. 1, 1998.
- [Vic17] Neil J Vickers. Animal communication: when i'm calling you, will you answer too? *Current biology*, 27(14):R713–R715, 2017.
- [VM20] Midhula Vijayan and R Mohan. A universal foreground segmentation technique using deep-neural network. *MULTIMEDIA TOOLS AND APPLICATIONS*, 2020.
- [WADP97a] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785, 1997.
- [WADP97b] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785, 1997.
- [Wal96] Peter Walley. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–57, 1996.
- [WB02] Andrew Watkins and Lois Boggess. A new classifier based on resource limited artificial immune systems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1546–1551. IEEE, 2002.
- [WBN<sup>+</sup>09] Junxian Wang, George Bebis, Mircea Nicolescu, Monica Nicolescu, and Ronald Miller. Improving target detection by coupling it with tracking. *Machine Vision and Applications*, 20(4):205–223, 2009.
- [WD14] Bin Wang and Piotr Dudek. A fast self-tuning background subtraction algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 395–398, 2014.
- [WF20] Rong Wang and Yue Feng. Evaluation research on green degree of equipment manufacturing industry based on improved particle swarm optimization algorithm. *Chaos, Solitons & Fractals*, 131:109502, 2020.
- [WFC<sup>+</sup>18] Jie Wen, Xiaozhao Fang, Jinrong Cui, Lunke Fei, Ke Yan, Yan Chen, and Yong Xu. Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

- [WGW18] Kunfeng Wang, Chao Gou, and Fei-Yue Wang.  $m^4cd$ : A robust change detection method for intelligent visual surveillance. *IEEE Access*, 6:15505–15520, 2018.
- [Win18] Uci machine learning repository. <https://archive.ics.uci.edu/ml/datasets.php>, Last accessed on 20-12-2018.
- [Win21a] Cours-data-mining. <http://www.lifl.fr/~talbi/Cours-Data-Mining.pdf>, Last accessed on 14-01-2021.
- [Win21b] Data mining. <https://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/Alex/>, Last accessed on 14-01-2021.
- [Win21c] Data mining and its importance. <https://www.loginworks.com/blogs/217-data-mining-and-its-importance/>, Last accessed on 14-01-2021.
- [Win21d] 12 most useful data mining applications of 2021. <https://www.upgrad.com/blog/12-most-useful-data-mining-applications-of-2020/>, Last accessed on 18-01-2021.
- [WJM<sup>+</sup>20] Dingding Wang, Jing Jiang, Jiaqing Mo, Jun Tang, and Xiao-yi Lü. Express: Rapid screening of thyroid dysfunction using raman spectroscopy combined with an improved support vector machine. *Applied Spectroscopy*, page 0003702820904444, 2020.
- [WJP<sup>+</sup>14] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: an expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 387–394, 2014.
- [WK19] Yuanni Wang and Tao Kong. Air quality predictive modeling based on an improved decision tree in a weather-smart grid. *IEEE Access*, 7:172892–172901, 2019.
- [WLJ17] Yi Wang, Zhiming Luo, and Pierre-Marc Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 96:66–75, 2017.
- [WLS20] Chun-feng Wang, Kui Liu, and Pei-ping Shen. A novel genetic algorithm for global optimization. *Acta Mathematicae Applicatae Sinica, English Series*, 36(2):482–491, 2020.

- [WLYT20] Yue Wang, Hao Liu, ZhongXin Yu, and LiangPing Tu. An improved artificial neural network based on human-behaviour particle swarm optimization and cellular automata. *Expert Systems with Applications*, 140:112862, 2020.
- [WM03] H-Y Wang and K-K Ma. Automatic video object segmentation via 3d structure tensor. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–153. IEEE, 2003.
- [WS07] Brandyn White and Mubarak Shah. Automatically tuning background subtraction parameters using particle swarm optimization. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1826–1829. IEEE, 2007.
- [WTB04] Andrew Watkins, Jon Timmis, and Lois Boggess. Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317, 2004.
- [WW20] Quanmin Wang and Xuan Wei. The detection of network intrusion based on improved adaboost algorithm. In *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, pages 84–88, 2020.
- [WWW20] Xiaochun Wang, Xiali Wang, and Don Mitchell Wilkes. Enhancing hierarchical linkage clustering via boundary point detection. In *Machine Learning-based Natural Scene Recognition for Mobile Robot Localization in An Unknown Environment*, pages 109–128. Springer, 2020.
- [WZX<sup>+</sup>20] Qin Wan, Xiaolin Zhu, Yueping Xiao, Jine Yan, Guoquan Chen, and Mingui Sun. An improved non-parametric method for multiple moving objects detection in the markov random field. *Computer Modeling in Engineering & Sciences*, 124(1):129–149, 2020.
- [XGS08] Zhifei Xu, Irene Yu-Hua Gu, and Pengfei Shi. Recursive error-compensated dynamic eigenbackground learning and adaptive background subtraction in video. *Optical Engineering*, 47(5):057001, 2008.
- [XJZ20] Yiping Xu, Hongbing Ji, and Wenbo Zhang. Coarse-to-fine sample-based background subtraction for moving object detection. *Optik*, page 164195, 2020.

- [XRC<sup>+</sup>20] Xibo Xu, Mengyan Ren, Jianfei Cao, Quanyuan Wu, Peiyuan Liu, and Jianshu Lv. Spectroscopic diagnosis of zinc contaminated soils based on competitive adaptive reweighted sampling algorithm and an improved support vector machine. *Spectroscopy Letters*, 53(2):86–99, 2020.
- [XSG06] Zhifei Xu, Pengfei Shi, and Irene Yu-Hua Gu. An eigenbackground subtraction method using recursive error compensation. In *Pacific-Rim Conference on Multimedia*, pages 779–787. Springer, 2006.
- [XSH16] Haiying Xia, Shuxiang Song, and Liping He. A modified gaussian mixture background model via spatiotemporal distribution with shadow detection. *Signal, Image and Video Processing*, 10(2):343–350, 2016.
- [XWC<sup>+</sup>19] Jie Xue, Chaozhong Wu, Zhijun Chen, PHAJM Van Gelder, and Xinpeng Yan. Modeling human-like decision-making for inbound smart ships based on fuzzy decision trees. *Expert Systems with Applications*, 115:172–188, 2019.
- [YCL19] Cheng-Hong Yang, Li-Yeh Chuang, and Yu-Da Lin. Epistasis analysis using an improved fuzzy c-means-based entropy approach. *IEEE Transactions on Fuzzy Systems*, 2019.
- [YGW<sup>+</sup>20] Xinhui Yu, Weidong Guo, Nan Wu, Liang Zou, and Meng Lei. Rapid discrimination of coal geographical origin via near-infrared spectroscopy combined with machine learning algorithms. *Infrared Physics & Technology*, page 103180, 2020.
- [YHY<sup>+</sup>10] Sheng Chen Yu, Ying Hu, Gui Xian Yu, Xu Ling Jin, Li Nang Zhang, and Tie Jun Shao. Ecg t wave detector based on neural network improved by genetic algorithms. In *2010 Second WRI Global Congress on Intelligent Systems*, volume 1, pages 355–358. IEEE, 2010.
- [YJD11] Huda Yasin, Tahseen A Jilani, and Madiha Danish. Hepatitis-c classification using data mining techniques. *International Journal of Computer Applications*, 24(3):1–6, 2011.
- [YKG<sup>+</sup>14] Chen Yin, Ren Kan, Gu Guohua, Qian Weixian, and Xu Fuyuan. Moving object detection based on improved single gaussian background model. *Chinese J Lasers*, 41(11):1109002, 2014.
- [YLY<sup>+</sup>13] Yukai Yao, Yang Liu, Yongqing Yu, Hong Xu, Weiming Lv, Zhao Li, and Xiaoyun Chen. K-svm: An effective svm algorithm based on k-means clustering. *JCP*, 8(10):2632–2639, 2013.

- [YIYh07] Gao Yue-lin and Duan Yu-hong. A new particle swarm optimization algorithm with random inertia weight and evolution strategy. In *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, pages 199–203. IEEE, 2007.
- [YN17] Miin-Shen Yang and Yessica Nataliani. Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. *Pattern Recognition*, 71:45–59, 2017.
- [YX09] Chao Yang and Ruzhi Xu. Adaptation learning rate algorithm of feed-forward neural networks. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–3. IEEE, 2009.
- [YYL19] Tianming Yu, Jianhua Yang, and Wei Lu. Dynamic background subtraction using histograms based on fuzzy c-means clustering and fuzzy nearness degree. *IEEE Access*, 7:14671–14679, 2019.
- [Z<sup>+</sup>03] Jing Zhong et al. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 44–50. IEEE, 2003.
- [ZBC02] Ming Zhao, Jiajun Bu, and Chun Chen. Robust background subtraction in hsv color space. In *Multimedia systems and Applications V*, volume 4861, pages 325–333. International Society for Optics and Photonics, 2002.
- [ZCX<sup>+</sup>20] Mengyang Zhao, Chorng Hwa Chang, Wenbin Xie, Zhou Xie, and Jinyong Hu. Cloud shape classification system based on multi-channel cnn and improved fdm. *IEEE Access*, 8:44111–44124, 2020.
- [ZFJ<sup>+</sup>11] Mingyuan Zhao, Chong Fu, Luping Ji, Ke Tang, and Mingtian Zhou. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, 38(5):5197–5204, 2011.
- [ZFX Y09] Li Zhihua, Zhou Fan, Tian Xiang, and Chen Yaowu. High efficient moving object extraction and classification in traffic video surveillance. *Journal of Systems Engineering and Electronics*, 20(4):858–868, 2009.
- [Zha19] Fei Zhang. Modular configuration of service elements based on the improved k-means algorithm. *Expert Systems*, 36(5):e12344, 2019.

- [ZHWC20] Dehua Zheng, Zhen Hong, Ning Wang, and Ping Chen. An improved lda-based elm classification for intrusion detection algorithm in iot application. *Sensors*, 20(6):1706, 2020.
- [ZHWG08] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time kd-tree construction on graphics hardware. In *ACM Transactions on Graphics (TOG)*, volume 27, page 126. ACM, 2008.
- [ZJYK19] Junhui Zuo, Zhenhong Jia, Jie Yang, and Nikola Kasabov. Moving target detection based on improved gaussian mixture background subtraction in video images. *IEEE Access*, 7:152612–152623, 2019.
- [ZL10] Huaxiang Zhang and Jing Lu. Creating ensembles of classifiers via fuzzy clustering and deflection. *Fuzzy sets and Systems*, 161(13):1790–1802, 2010.
- [ZLL20] Yun-tao Zhao, Wei-gang Li, and Ao Liu. Improved grey wolf optimization based on the two-stage search of hybrid cma-es. *Soft Computing*, 24(2):1097–1115, 2020.
- [ZPH<sup>+</sup>19] Honghao Zhang, Yong Peng, Lin Hou, Guangdong Tian, and Zhiwu Li. A hybrid multi-objective optimization approach for energy-absorbing structures in train collisions. *Information Sciences*, 481:491–506, 2019.
- [ZXL16] Shibing Zhou, Zhenyuan Xu, and Fei Liu. Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. *IEEE Transactions on Neural Networks and learning systems*, 28(12):3007–3017, 2016.
- [ZYL14] Bichen Zheng, Sang Won Yoon, and Sarah S Lam. Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. *Expert Systems with Applications*, 41(4):1476–1482, 2014.
- [ZYT<sup>+</sup>18] Guian Zhang, Zhiyong Yuan, Qianqian Tong, Mianlun Zheng, and Jianhui Zhao. A novel framework for background subtraction and foreground detection. *Pattern Recognition*, 84:28–38, 2018.
- [ZYZ09] Yong Zeng, Yupu Yang, and Liang Zhao. Pseudo nearest neighbor rule for pattern classification. *Expert Systems with Applications*, 36(2):3587–3595, 2009.

- [ZZZ18] Geng Zhang, Chengchang Zhang, and Huayu Zhang. Improved k-means algorithm based on density canopy. *Knowledge-Based Systems*, 145:289–297, 2018.

# Author's publications

## International publications

---

**Wafa Nebili**, Brahim Farou and Hamid Seridi, "Background subtraction using artificial immune recognition system and single gaussian (airs-sg)", *Multimedia Tools and Applications*, 2020, vol.79, no 35, p.26099-26121.

**Wafa Nebili**, Brahim Farou and Hamid Seridi, "Using Resources Competition and Memory Cell Development to Select the Best GMM for Background Subtraction", *International Journal of Strategic Information Technology and Applications (IJSITA)*, 2019, vol. 10, no 2, p. 21-43.

**Wafa Nebili**, Samir Hallaci and Brahim Farou, "Background subtraction based on a Self-Adjusting MoG", *International Journal of Informatics and Applied Mathematics*, 2019, vol. 2, no 1, p. 73-84.

**Wafa Nebili**, Hamid Seridi and Mohamed Nadjib Kouahla, "A New Process for Selecting the Best Background Representatives based on GMM", *International Journal of Informatics and Applied Mathematics*, 2019, vol. 1, no 1, p. 35-46.

## International Communications

---

**Wafa Nebili**, Ala-Eddine Benrazek, Brahim Farou, Muhammet Kurulay and Mohamed Amine Ferrag, "Using A Bio Inspired Approach To Improve Uavs Vision Coverage", *CITCS'2019, Guelma, Algeria*, 2019.



## National Communications

---

**Wafa Nebili** Brahim Farou, Hamid Seridi, "A New Process for Selecting the Best Background Representatives based on GMM", 1st Conference on Informatics and Applied Mathematics IAM'2018, Guelma, Algeria, 2018.

**Wafa Nebili** Brahim Farou and Hamid Seridi, "GMM with Dynamic Management of the Number of Gaussians based on AIRS", In: JERI, Saida, Algeria, 2019.

Nebili Charefeddine, **Wafa Nebili**, Brahim Farou and Hamid Seridi, "Selection of the best GMM for background representation based on the AIRS", 2nd Conference on Informatics and Applied Mathematics IAM'2019, Guelma, Algeria, 2019.