



HAL
open science

A System For Retrieving and Classifying Images Extracted From Video Surveillance Cameras

Sirine Ammar

► **To cite this version:**

Sirine Ammar. A System For Retrieving and Classifying Images Extracted From Video Surveillance Cameras. Computer Vision and Pattern Recognition [cs.CV]. Université de La Rochelle; Université de Sfax (Tunisie), 2021. English. NNT : 2021LAROS010 . tel-03619976

HAL Id: tel-03619976

<https://theses.hal.science/tel-03619976v1>

Submitted on 25 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**LA ROCHELLE UNIVERSITÉ
UNIVERSITÉ DE SFAX**



Laboratoire Mathématiques, Image et Applications (MIA)
Multimedia, Information systems and Advanced Computing
Laboratory (MIRACL)

THÈSE présentée par :

Sirine AMMAR

soutenue le :

pour obtenir le grade de : **Docteur de l'université de La Rochelle & l'université de Sfax**

Discipline : **Informatique et Applications**

**A System For Retrieving and Classifying Images Extracted
From Video Surveillance Cameras**

JURY :

Laure TOUGNE	Professeure. Univ. de Lyon, Présidente du jury.
André BIGAND	Maître de Conférences (HDR), Univ. du Littoral Côte d'Opale, France. Rapporteur.
Antoine VACAVANT	Maître de Conférences (HDR), Univ. Clermont Auvergne, France. Rapporteur.
Michel BERTHIER	Professeur, Univ. de la Rochelle, France. Examineur.
Walid MAHDI	Professeur, Univ. de Sfax, Tunisie. Examineur.
Thierry BOUWMANS	Maître de Conférences (HDR), Univ. de La Rochelle, France, Directeur de thèse.
Mahmoud NEJI	Professeur. Univ. de Sfax, Tunisie, Co-directeur de thèse.
El-hadi ZAHZAH	Maître de Conférences (HDR), Univ. de La Rochelle, France, invité.
Nizar ZAGHDEN	Maître assistant. Univ. de Sfax, Tunisie, invité.



A System For Retrieving and Classifying Images Extracted From Video Surveillance Cameras

A thesis submitted by **Sirine Ammar** at Université de La Rochelle & Université de Sfax to fulfill the degree of **Doctor in Computer Science and Applications**.

La Rochelle

Directeur de thèse

Dr. Thierry Bouwmans

Laboratoire Mathématiques, Image et Applications
Université de La Rochelle (France)

Co-directeur de thèse

Pr. Mahmoud Neji

Multimedia, Information systems and Advanced Computing Laboratory
Université de Sfax (Tunisie)

Comité de thèse

Pr. Laure Tougne

Laboratoire LIRIS
Université de Lyon (France)

Pr. Michel Berthier

Laboratoire Mathématiques, Image et Applications
Université de La Rochelle (France)

Pr. Walid Mahdi

Multimedia, Information systems and Advanced Computing Laboratory
Université de Sfax (Tunisie)

Rapporteurs

Dr. André Bigand

Laboratoire LISIC
Université du Littoral Côte d'Opale (France)

Dr. Antoine Vacavant

Institut Pascal
Université Clermont Auvergne (France)

Invités

Dr. El-hadi Zahzah

Laboratoire L3i
Université de La Rochelle (France)

Dr. Nizar Zaghden

SETIT (ISBS)
Université de Sfax (Tunisie)

Acknowledgement

First I would like to express my gratitude to my supervisors, Dr. Thierry Bouwmans and Pr. Mahmoud Neji for their invaluable guidance and advices. Also, thank you to Catherine Choquet (director of the MIA) for accepting me in her lab and both institutions for providing me all the facilities to carry out this work in a great environment and atmosphere. Without their supports, this thesis is impossible.

I would like to thank Dr. André Bigand from Univ. of Littoral Côte d'Opale (France) and Dr. Antoine Vacavant from Univ. of Clermont Auvergne (France) for their acceptances to be the reviewers of this European thesis manuscript and for sharing interesting comments and criticism that helped improve this manuscript.

Sincere thanks to my dissertation examiners, Pr. Laure Tougne from Univ. of Lyon (France), Pr. Michel Berthier from Univ. of La Rochelle (France) and Pr. Walid Mahdi from Univ. of Sfax (Tunisia) for their interest in my work and for making their time available for me.

I would like to thank Dr. Nizar Zaghden from Univ. of Sfax (Tunisia) and Dr. El-hadi Zahzah from Univ. of La Rochelle (France) for their presence in my PhD defense.

I am very grateful to my family for their unconditional love and support without which this journey would not have been possible.

Last but not least, I would also like to thank my friends and colleagues from the University of La Rochelle and University of Sfax, especially those from the MIA and MIRACL labs. Their support has been invaluable throughout my Ph.D. study, making my time both memorable and enjoyable.

Contents

1	Introduction	1
1.1	Background subtraction	2
1.1.1	Challenges in scene modeling	2
1.1.2	Background subtraction process	5
1.2	Object classification	7
1.2.1	Object classification challenges	7
1.2.2	Object classification process	8
1.3	Face recognition	9
1.3.1	Face recognition challenges	9
1.3.2	Face recognition process	12
1.4	Contributions of this thesis	13
1.5	Thesis outline	15
2	Literature review	17
2.1	Background subtraction models	18
2.1.1	Mathematical models	18
2.1.2	Subspace models	20
2.1.3	Neural network modeling	21
2.1.4	Deep neural networks concepts	22
2.1.5	Signal processing models	23
2.1.6	Semantic concepts	24
2.2	Object classification	27
2.2.1	Conventional methods	27
2.2.2	Deep neural network methods	27
2.3	Face recognition methods	29
2.3.1	Holistic approaches	29
2.3.2	Local approaches	33
2.3.3	Hybrid approaches	35
2.3.4	Deep learning approaches	37
2.4	Solved and unsolved challenges	43
2.4.1	Background subtraction	43
2.4.2	Object classification	44
2.4.3	Face recognition	44
2.5	Conclusion	46

3	A novel deep detector classifier (DeepDC) for background subtraction in videos	47
3.1	Motivation	48
3.2	DeepSphere architecture	49
3.3	Proposed DeepDC descriptor	51
3.4	Experimental results and discussions	54
3.4.1	Description of the datasets	54
3.4.2	Quantitative and qualitative evaluation	55
3.5	Conclusion	78
4	A novel semi-supervised DCGAN model for object classification	79
4.1	Motivation	80
4.2	Generative Adversarial Networks	83
4.3	DCGANs architecture	84
4.4	Proposed approach	85
4.5	Experiments	90
4.5.1	Datasets	90
4.5.2	Experimental results and discussions	90
4.6	Conclusion	96
5	DCGAN-based data augmentation for face identification in images and video applications	97
5.1	Motivation	98
5.2	Face detection	99
5.3	Image data augmentation techniques	100
5.4	DCGANs	102
5.5	FaceNet	103
5.5.1	FaceNet model	103
5.5.2	Triplet Loss	105
5.6	Proposed approach	106
5.7	Experimental results	109
5.7.1	Description of the datasets	109
5.7.2	Qualitative and quantitative evaluation	110
5.8	Conclusion	120
6	Conclusions	121
6.1	Limitations	122
6.2	Future works	123
A	Notations and Symbols	125
B	List of Publications	127
	Bibliography	129

List of Tables

2.1	Background modeling methods: An overview (Part 1).	25
2.2	Background modeling methods: An overview (Part 2).	26
2.3	Comparative study of different object classification methods.	29
2.4	Face recognition using holistic approaches: An overview.	39
2.5	Face recognition using local approaches: An overview.	40
2.6	Face recognition using hybrid approaches: An overview.	41
2.7	Face recognition using deep-learning approaches: An overview.	42
3.1	FM of BS algorithms evaluated on five real videos of the VIRAT dataset [333]. The six best methods are underlined. The best methods in each category are in italic.	57
3.2	Number of extracted images for each class from five cameras of VIRAT_Video dataset [333].	58
3.3	FM of BS algorithms evaluated on 53 videos of the CDnet2014 dataset [460]. The six best methods are underlined. The best methods in each category are in italic.	61
3.4	Performance values of the proposed method compared to the other methods on eleven categories from CDnet2014 Dataset [460] (Part 1).	64
3.5	Performance values of the proposed method compared to the other methods on eleven categories from CDnet2014 Dataset [460] (Part 2).	65
3.6	Visual results on CDnet 2014 dataset: From left to right: Original images, Ground-Truth images, RPCA [84], DeepPBM [149], DeepSphere (ours).	66
3.7	Background subtraction results on seven frames from three video sequences of CDnet2014 dataset affected by illumination changes and dynamic backgrounds. Our algorithm successfully segments out the objects (person/vehicle) in all seven input frames.	67
3.8	Number of extracted images for each class from 53 cameras of CDnet2014 Video dataset [460].	71
3.9	Performance values of the proposed method compared to the other methods on 9 real videos from BMC 2012 Dataset [446]	73
3.10	Visual results on real-world videos of the BMC2012 dataset [446]: From left to right: Original images, Ground-Truth images, RPCA [84], DeepPBM [149], DeepSphere (ours).	74

3.11	Computational time for the BS task of our DeepSphere compared to RPCA [84] and DeepPBM [149] evaluated on the 6 short videos of BMC 2012 dataset [446]. For the fair comparison we ran the trained model using Intel Core i7 Hardware.	77
4.1	DCGAN-SSL model architecture	88
4.2	Baseline model architecture	88
4.3	Classifier accuracy in VIRAT video dataset [333].	92
4.4	Performance comparison (error rate, %) on VIRAT video dataset [333] of other models to DCGAN-SSL for different numbers of labeled subsets per class	94
4.5	Performance comparison (error rate, %) on CDnet2014 video dataset [460] of other models to DCGAN-SSL for different numbers of labeled subsets per class	94
5.1	The main face detection works	100
5.2	The main data augmentation works	102
5.3	Face recognition accuracy with DCGAN data augmentation using the proposed method.	111
5.4	Recognition performance with different methods using 10 classes from CDnet2014 dataset [460].	112
5.5	Recognition performance with different methods using 62 classes from LFW dataset [198].	113
5.6	Recognition performance with different methods using 20 classes from VG-Face2 dataset [85].	115
5.7	Face recognition accuracy with DCGAN data augmentation using the proposed method in video datasets.	115
5.8	Recognition performance with different methods using portal 1 from ChokePoint dataset [469].	115
5.9	Recognition performance with different methods using Youtube face dataset [468] (40 classes).	115
5.10	Training and classifying execution time using Intel Core i7 Hardware using ChokePoint dataset [469].	117
5.11	Training and classifying execution time using Intel Core i7 Hardware using Youtube face dataset [468].	119

List of Figures

1.1	Scenes captured from the same avenue under various conditions.	4
1.2	An overview of a background subtraction process.	5
1.3	Moving object tracking.	6
1.4	Object classification challenges. Note the high intra-class variations, significant amount of background clutter and difficult occlusions.	7
1.5	A general framework of object classification.	8
1.6	Face recognition challenges.	11
1.7	The three steps of face recognition process. (a) The output of face detection (the bounding box) (b) The extracted face patch (c) The extracted feature vector (d) Comparing the input feature vector with the vectors stored in the dataset by classification methods and find the most likely class (the red rectangle). Each face patch is described as a d-dimensional vector, the vector $x_{m,n}$ as the n_{th} in the m_{th} , and N_k represents the number of faces stored in the k_{th} class.	12
1.8	Schematic organization of the manuscript and the contributions.	16
2.1	An overview of background subtraction models.	19
3.1	DeepSphere architecture.	49
3.2	Illustration of the two-level anomaly discovery task.	51
3.3	The proposed architecture (Deep Detector Classifier). Step 1: background subtraction. Step 2: Object classification (chapter 4).	52
3.4	Examples of foreground activity detection in restaurant video dataset [91] :The top, middle and bottom rows represent normal situation, anomalous situation and detected results, respectively.	55
3.5	Network input and output in VIRAT video dataset [333] : (a) Background frame, (b) test frame, (c) output of DeepSphere, (d) segmentation mask of the proposed method.	56
3.6	Extracted images from our proposed background subtraction approach: (a) person, (b) car, (c) car but not the whole feature :etc, (d) Individual but not the whole body : etc	58
3.7	Network input and output in CDnet2014 dataset [460] : The first row is the background frame, the second row is the image test, the third row is the ground truth, the fourth row is the output of DeepSphere. The fifth row is the foreground mask of the proposed method.	59

3.8	Background subtraction obtained with the proposed scheme and the best five BS algorithms using the VIRAT scenes and CDnet2014 dataset. From the first to last row: input frame, region of interest, DeepSphere (ours), PBAS [196], DPWrenGABGS [470], MixtureOfGaussianV1BGS [228], LB-FuzzyAdaptiveSOM [295] and T2FGMM_UV [67].	62
3.9	F_measures obtained with the proposed scheme and other methods for the CDnet 2014 dataset [460]	68
3.10	Train and validation learning curves of DeepSphere using CDnet2014 dataset [460].	70
3.11	(a) Gain in performance between MOG [427], T2FGMM_UV [67] and DeepSphere [34] for the CDnet2014 dataset [460]. (b) Gain in performance between DeepSphere [34] and unsupervised models, RPCA [84] and DeepPBM [149] for the CDnet2014 dataset [460]. (c) Gain in performance between DeepSphere [34] and CNNs (supervised models) [42] [502] for the CDnet2014 dataset [460].	70
3.12	F_measures obtained with the proposed scheme and other methods on 9 real-world videos of the BMC 2012 dataset [446].	75
3.13	(a) Gain in performance between MOG [427], T2FGMM_UV [67] and DeepSphere [34] for the BMC 2012 dataset [446]. (b) Gain in performance between DeepSphere [34] and unsupervised models, RPCA [84] and DeepPBM [149] for the BMC 2012 dataset [446].	76
4.1	GAN high-level architecture.	83
4.2	The proposed Semi-supervised learning DCGAN (DCGAN-SSL) architecture for a 4 class classification problem.	85
4.3	VIRAT training dataset	91
4.4	Samples generated by DCGAN-SSL and GAN from the VIRAT video dataset [333]. DCGAN-SSL is on the left and GAN is on the right. The results are obtained after 200 epochs of training the models.	92
4.5	The test accuracy of the DCGAN-SSL over CNN for various amounts of labeled samples from the VIRAT video dataset [333].	92
4.6	Discriminator and generator losses using CDnet2014 dataset [460].	94
4.7	(a) Training and testing loss of Fully Connected (FC) classifier using CDnet2014 [460] (b) Training and testing loss of GAP using CDnet2014 [460].	95
4.8	(a) Standard CNN model accuracy using CDnet2014 dataset [460] (b) DCGAN-SSL accuracy using CDnet2014 dataset [460].	95
5.1	FaceNet model architecture, which consists of two modules : preprocessing and extraction of low-dimensional representation. The preprocessing module uses the Multi-task Cascaded CNN (MTCNN) [226] for the detection and alignment of samples. The low-dimensional representation extraction module consists of a batch input layer and a deep CNN which is followed by L2 normalization that provides the embedding of face. Next, the triplet loss is applied during training.	103
5.2	Inception module.	104
5.3	The triplet Loss.	105
5.4	An overview of the proposed face recognition pipeline. The CNN feature extractor generates 128-dimensional facial embeddings.	106

5.5	Face alignment using Multi-Task CNN, Facial Landmarks detection.	108
5.6	In each row some examples of representative images/frames of datasets used in this chapter: (a) Faces extracted from CDnet2014 dataset [460] (b) LFW dataset [198] (c) VGGFace2 dataset [85] (d) ChokePoint dataset [469] (d) and YouTube faces [468].	109
5.7	Generated Images using DCGANs on LFW dataset [198].	112
5.8	Generated Images using DCGANs on VGGFace2 dataset [85].	114
5.9	Recognition accuracy using LFW dataset [198], VGGFace2 dataset [85], ChokePoint dataset [469] and Youtube face dataset [468].	116
5.10	Face confidence using LFW dataset [198].	118
5.11	Face confidence using ChokePoint dataset [469].	118
5.12	(a) Training time using ChokePoint dataset [469] and Youtube face dataset [468]. (b) Classification time using ChokePoint dataset [469] and Youtube face dataset [468].	119

Chapter 1

Introduction

Artificial Intelligence (AI) is one of the most interesting and controversial technologies in the current world. Developers continue to work on improving machine learning solutions, and AI becomes increasingly advanced. Despite the evolution, AI still seems to struggle to render images. Therefore, object detection, classification and recognition are popular topics in the field of AI. Image recognition is a blend of image detection and classification. It defines the ability of AI to detect, classify and recognize the object. In this thesis, we propose a system to detect moving objects from video sequences (pedestrians, vehicles, etc), classify them and only then decide if the extracted human face belongs to a known faces. Background Subtraction (BS) plays a significant role in several computer vision applications. However, it is a challenging task, due to the real world constraints and dynamic weather conditions (e.g rainy day, high lighting, camera motion). Therefore, our proposed object detection system should be robust to these challenges. Multiple features have been extracted over the long history of BS, improved or even suggested to handle BS challenges. Highly discriminant features are extracted for each pixel, region or cluster in an image sequence. Once the moving objects have been extracted, the neural network must classify them by element type. Moving object classification aims to identify the category, called also label, of the detected object based on two main steps. First, several features are extracted from the detected objects. Second, they are fed to a learning-based classifier to specify the class of each object. Once the moving objects are classified, the last step is to recognize the extracted people. Face recognition is the best example of image recognition solutions and has acquired a significant position among all biometric systems. First, the system has to detect the face, classify it as human face and only then decide if it belongs to the target person.

This chapter presents an introduction about the BS, image classification and recognition tasks. First, we describe the challenges of BS in scene modeling, and then we detail the major steps in a BS algorithm. Second, we survey the challenges that may occur during object classification and face recognition processes. Third, we summarize the three main contributions of this thesis. Finally, we provide the thesis outline.

Contents

1.1	Background subtraction	2
1.1.1	Challenges in scene modeling	2
1.1.2	Background subtraction process	5
1.2	Object classification	7
1.2.1	Object classification challenges	7
1.2.2	Object classification process	8
1.3	Face recognition	9
1.3.1	Face recognition challenges	9
1.3.2	Face recognition process	12
1.4	Contributions of this thesis	13
1.5	Thesis outline	15

1.1 Background subtraction

1.1.1 Challenges in scene modeling

Background subtraction is an interesting area of research in computer vision. It covers a set of methods aimed at distinguishing moving objects, called "foreground" in the scene, from the static information, called "background". BS has been fed by many academic researchers and developers in the last 20 years. This is due to its potential applications and the large number of surveillance cameras installed in security-sensitive areas such as banks, railway stations, airports, and borders. Background subtraction can be used for surveillance systems in large spaces (such as football stadiums, and big shopping centers), in traffic surveillance (vehicle counting, vehicle detection and tracking), industrial applications (robot guidance, inspection and identification products) and natural environments (rivers) [27] [28]. In order to ensure a good operation of BS algorithms, three major conditions must be satisfied: the camera is fixed, the lighting is constant and the background is static, that is, pixels have a unimodal distribution and no background objects are moved/inserted in the scene. Under these ideal conditions, BS performs well. Usually, the appearance of an outdoor or indoor environment can be affected by a variety of changes that can happen over time. In general, it is difficult to build an excellent background model that can handle all these changes. There are numerous situations that can affect the appearance of the scene, which can reduce the accuracy of the BS methods. To our knowledge, the main challenges of background subtraction are [74], [501], [401]:

- **Camera jitter:** Generally, the camera jitter occurs in outdoor scenes. A sudden camera motion or camera jitter reduces the quality of images captured by cameras, resulting in blurry images. For example, heavy winds can cause a stationary camera to move

back and forth, causing nominal movement in scenes. This nominal camera motion is generally indiscernible from the movement caused by moving objects, leading to unwanted detection of foreground objects.

- **Camera automatic adjustments:** Automatic exposure (the amount of light falling on a camera's sensor) is a parameter available on most digital cameras. The light reflected by objects with homogeneous features (e.g. intensity, texture) is captured by cameras making segmentation a challenging task. The foreground aperture occurs when homogeneously parts of the moving object are part of the background rather than being classified as foreground pixels.
- **Pan-Tilt-Zoom (PTZ):** Most background subtraction research has focused on fixed cameras, while PTZ cameras have become more popular thanks to the wide area coverage. Traditional BS methods fail in the case of moving camera because foreground objects and background pixels are not stationary.
- **Video noise:** Generally, a video signal is covered with noise that appears during acquisition, coding, transmission and processing leading to disturbance of the original information which can result in artifacts, jagged edges, corners and invisible lines affecting the background scenes.
- **Intermittent object motion:** The intermittent motion occurs when a foreground object stops moving for long period of time or a background object begins to move, resulting in 'ghost' artifacts in the detected motion. Some videos include objects that suddenly stop to move, such as abandoned objects (parked vehicles and left-luggage). Handling this issue depends on the context of each situation. Several applications incorporate stationary foreground objects and others do not.
- **Dynamic backgrounds:** In a dynamic environment where the state of domain is changing continually, the transformation from one temporal stable to another is normally the result of an external event or a sequence of events (i.e. flowing water, moving leaves or plants). The background may include some elements which are not totally static as waving trees or water surface. These elements are considered as part of the background, even if they are not stationary. This dynamic environment does not provide a good background model because even some part of the scene that contains moving elements can be considered as background.
- **Presence of shadows:** Detecting cast shadows as a moving object is highly common and produces unwanted results. For example, the shadows are so different from the background that they can be wrongly considered as foreground.
- **Illumination changes:** Illumination changes often occur over time in indoor and outdoor environments, resulting in incorrect detections. For example, a wide range of lighting conditions, particularly those encountered during a typical 24-hour day-night cycle, cause gradual changes in appearance in outdoor environments. Additionally, turning on/off the light switch in an indoor scene can produce sudden illuminations. It is crucial that the background model be invariant or tolerates these kinds of changes.
- **Low frame rate:** Background changes and illumination changes are not consistently updated using a low frame rate and these variations seem to be more abrupt.



Figure 1.1: Scenes captured from the same avenue under various conditions.

- **Motion parallax:** 3D scenes with large variations in depth show parallax in images captured by a moving cameras, resulting in issues in background modeling and motion compensation.
- **Bootstrapping:** In a training video sequence, both background and foreground objects are present. Initial video data with only stationary objects is not always available, making it difficult to produce a representative background model. Thus, an initialization process is required to correctly model the background over time.
- **Camouflage:** Some moving objects appear a lot like the background called as camouflage impact. Foreground objects might have similar color with the background and become combined with it, resulting in a wrong discrimination between foreground and background.
- **Foreground aperture:** The presence of foreground objects can have the same motion characteristics. As a result, shadows usually distort the geometric shape of the foreground objects, and sometimes fuse moving objects. The homogenous part of foreground object may not be detected, causing false negatives.
- **Night scenes:** The videos taken at night are always a difficult task. Night scenes typically lead to high incorrect detections owing to the significant variation in lighting and the low-contrast between foreground and background.
- **Challenging weather:** In some situations, the BS algorithm must adapt to challenging weather conditions such as air turbulence or snow that affects the background scene.

To handle the above challenges, many developers have suggested diverse methods and its evaluation results have often been accessible by Change Detection website¹. Recent evaluation results have shown that the greatest challenge is the discrimination between the background and the foreground in the case of videos taken by PTZ cameras [501] and night videos. Another major problem would also be encountered if multiple challenges occur in the same scene. Figure 1.1 illustrates three cases at the same avenue. While Figures 1.1a and 1.1b display shadows and several variations in light, Figure 1.1c shows large reflections. Although all of these challenges are managed quietly today [60, 139, 343, 426, 458], they still disturb

¹<http://wordpress.jodoin.dmi.usherb.ca/results2014/>

the background subtraction process. Note that Figure 1.1 illustrates various situations such as large shadows, large reflections and illumination variations. At the present time, there is no background subtraction algorithm that can address all these kinds of issues at the same time, making the background subtraction domain even more difficult.

1.1.2 Background subtraction process

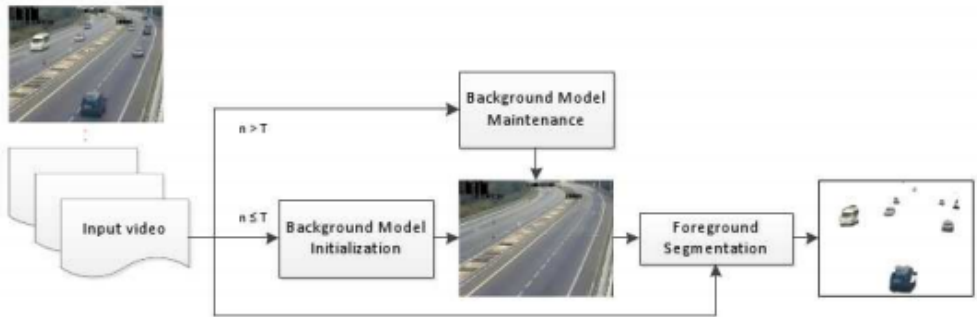


Figure 1.2: An overview of a background subtraction process.

In this section, we briefly remain the different steps related to background subtraction. Figure 1.2 shows an overview of these components. Essentially, background subtraction consists to initialize and update a model of the static scene, called the background (BG) model, and compare this model with the input frame to produce binary segmentation mask. Regions or pixels with a significant difference are assumed to be categorized as moving objects (they represent the foreground FG). A conventional background subtraction algorithm consists of four steps:

- **Background initialization:** (also called background extraction, background generation and background reconstruction) consists in calculating the initial background frame (also called reference frame).
- **Background modeling:** (also called background representation) constructs the model of a scene background.
- **Background maintenance:** relies to the mechanism of update used for the model to adapt itself to the changes occurred over time.
- **Foreground detection:** Foreground detection consists in categorizing pixels as 'background' or 'foreground'.

BS is generally an important first step in many computer vision applications as shown in Figure 1.3. The background maintenance and moving objects detection steps are performed repeatedly over time. A sub-entity of the reference image is compared with its corresponding sub-entity in the current image. This sub-entity may represent the size of a pixel, a cluster or a



Figure 1.3: Moving object tracking.

region. Additionally, this sub-entity is described by a "feature" which can be an edge feature, color feature, stereo feature, texture feature or motion feature [75]. In order to develop a background subtraction approach, engineers and researchers have to design each step and select the features based on the challenges they want to overcome in the involved applications.

Background initialization is a crucial step which computes an initial model of the background. It allows generating, extracting and constructing the background. Background model initialization has received little attention. This can be justified by the fact that initialization can be performed by exploiting certain clean frames at the beginning of the video sequence. Generally, in real scenarios, this assumption is not often satisfied due to the continued presence of clutter. Typically, the model is initialized using the first background image or an initial background model computed over a set of training frames, whether or not they contain moving objects. The background modeling (or representation) is the key step of any BS algorithm. The main idea behind such step is to create a static scene representation which is able to adapt to environmental changes in the background and to identify all foreground objects. In recent decades, a number of methodologies have been proposed to model and subtract the background, e.g. statistical methods [92, 427, 428], multilayer codebook based methods [179], methods for compressed streaming video [128], etc. The third step aims to update the background model which depends on the mechanism used to adapt the background model according to the changes on the scene over time. The background maintenance should be incremental (an online algorithm), as new data is streamed and so dynamically given. The flexible models employ robust updating mechanisms to deal with several challenges, such as noise, automatic camera settings and illumination changes in background. Additionally, this is where the updating mechanism is used, which determines if the inserted objects are integrated in the model, and if ghosts are updated or deleted. To handle these problems, several methods have been implemented [47, 205, 266, 308]. The final step is the foreground detection process, which compares the reference frame and the current image to assign foreground or background label to each pixel (or regions). This is a classification task, that can be performed by crisp [266] [362], statistical [20, 417] or fuzzy [90] classification techniques.

These different steps use methods that have various objectives and constraints. Therefore, they require algorithms with various features. Background initialization needs "offline" algorithms which are "batch" by using all the data at the same time. In contrast, background maintenance requires "online" algorithms which are "incremental" algorithms by using the incoming data one by one. Background initialization, representation and maintenance need reconstructive algorithms while foreground detection requires discriminative algorithms.

Deep convolutional neural networks (ConvNets) perform well in many computer vision applications including background subtraction [483] [78]. Additionally, it is generally easy to work with modern libraries (Caffe [219], Theano [52], Torch [13], etc.) with built-in architectures. In contrast, ConvNets are generally not appropriate for applications where few images are available. Training a deep ConvNets generally need a large number of images for a better generalization of the model. In addition, the computational cost for the training of ConvNets is high in terms of time and memory consumptions. Consequently, the study of new background subtraction techniques computationally simple is important in several real-life applications.

1.2 Object classification

1.2.1 Object classification challenges

Object classification, which is the process of assigning a semantic label to an object, is a core problem in computer vision and pattern recognition. It can be used as a building block for numerous other tasks such as localization, detection and full-scene labeling. Object classification is a challenging process that can be influenced by many factors. Since the classification results are the basis for many socio-economic and environmental applications, researchers and practitioners have made large efforts to develop advanced classification methods and techniques to improve classification performance. A good classification is very crucial, particularly in medicine. Thus, improved methods are required in this area. There are several challenges related to the object classification task as presented in Figure 1.4:



(<http://lear.inrialpes.fr>)

Figure 1.4: Object classification challenges. Note the high intra-class variations, significant amount of background clutter and difficult occlusions.

- ***Intra-Class Variation*** : Intra-class variation is a common challenge in object classification. The intra-class variation defines the image variations that occur between several images of one class. Two objects belong to the same class, but the system identifies them as a different class. Thus, the object classification system should be able to address the issue of intra-class variations.
- ***Scale Variation*** : Scale variation is a very big problem in object classification. It consists of having an image of the same object with several sizes. Scale variation affects the detection process, i.e. the objects of any size should be identified.
- ***Inter-Class Variation*** : Inter-class variation means two different objects appear to belong to the same class, but in reality they are not in the same class. It is easy for

machine to classify the object from labeled data but if a new object which is not already known is found then it will be difficult for machine to classify this object.

- **View-Point Variation** : Viewpoint variation occurs when an object is taken in several dimensions of rotation/orientation depending on how the object is captured in the image. Same object has different views from various viewpoints, therefore the classification system should consider all viewpoints.
- **Deformation** : Deformation of an object means the shape of the object is changed due to elasticity, stretching, etc. The classification system should consider the articulated object as belonging to the correct class.
- **Occlusion** : Occlusion of many objects in an image is a big challenge of object classification. There are many objects that we want to categorize in image can not be visualized entirely. Thus, large portion of the object is hidden behind another objects. The objects occluded may be of the same kind or it may be of a different kind.
- **Illumination** : The object classification system must be able to manage the illumination variation. Considering any image of several levels of brightness (illumination) to our image classification system, the system must be able to attribute them the same label.
- **Background Clutter** : It is defined when there are many objects in the image and for observers it is very difficult to segment whole objects and then to get the specific object. These images are very “noisy”.

1.2.2 Object classification process

The general framework of object classification is illustrated in Figure 1.5.



Figure 1.5: A general framework of object classification.

- **Feature extraction:** Typically comprises two main steps: image patch extraction which is performed by sampling local areas of images, generally in a sparse or dense manner and image patch representation which is performed via statistical analysis over pixels of image patches. The feature vectors of image patches are represented as local features including: 1) appearance-based features, e.g., scale-invariant feature transform (SIFT) [288], histogram of oriented gradients (HOG) [118]; 2) color-based features, e.g., color descriptors [390]; and 3) texture-based features, e.g., local binary pattern [334] and Gabor filter [269].
- **Building feature space:** Feature space is a collection of base vectors. There are three strategies for building the feature space.

The first one randomly selects patches from images as base vectors. This scheme is adopted in certain models of biological inspiration [395] [200]. It is fast but does not sufficiently represent the characteristics of the feature space.

The second one is based on supervised learning, i.e., the generation of dictionaries via supervised learning on local features. This method builds the relationship between features and labels, and represents the structure of the feature space well. However, it is time-consuming because it needs resolving dictionaries in an iterative way.

The third one is based on unsupervised learning, i.e., obtaining the base vectors through unsupervised learning over local features. This strategy strikes a good balance between speed and precision, and is widely used in current methods.

- **Describing features:** Describing features is a key component of object classification, and significantly influences image classification in terms of speed and accuracy. The coding strategies can be grouped into five categories:

Voting-based methods [107] [447] describe the distribution of local features with a histogram, indicating the occurrence information of visual codes.

Fisher coding-based methods [345] [346] calculate the distribution of local features with the Gaussian Mixture Models. Each Gaussian model reflects one pattern of local features.

Reconstruction-based methods [459] encode a feature by solving a least-square-based optimization problem with constraints on the number of codewords for reconstruction.

Local tangent-based methods [509] estimate the manifold of the feature space, based on which an exact description of local features is obtained.

Saliency-based methods [471] describe a local feature by the degree of saliency, for example, the ratio of the distances from a local feature to the codewords around it.

- **Classification:** is an important topic in machine learning. Various classifiers are used in object classification, e.g., Boosting, KNN and SVM. Additionally, kernel tricks, e.g., intersection kernel are usually used to improve overall performance.

1.3 Face recognition

1.3.1 Face recognition challenges

Face recognition is one of the most important tasks in computer vision and object recognition. It is pertinent in various fields such as in healthcare system, driving license system, monitoring operation, rail reservation system and passport authentication. In a big data set, face image identification task is often difficult. There are several biometric features that can be used to recognize humans like palm print, fingerprint, hand geometry, iris, speech, face, gaits and signature. However, these features need active intervention of human for authentication, while face recognition does not need active intervention of human. Thus, face recognition is much more appropriate than other biometrics. The human face is important for person's identity recognition and it is the characteristic which best distinguishes a person.

Face recognition is an issue that is initially very difficult to address with a computer. The almost unlimited ways that a face can appear in an image make this task very difficult. This task fails with a traditional computer system due to many challenges, such as, different lighting, angles and facial expressions. Face recognition has been the subject of a lot of research in the past and many solutions to this issue have been suggested. But, the simplest methods from the 2000s failed to address the issue in an unconstrained environment. Machine learning is a solution to several difficult tasks with traditional computer systems and in this situation, more precisely deep learning. Over the past few decades, an essential step in the development of face recognition methods has been the introduction of deep learning approaches using CNNs like FaceNet [393] and DeepFace [439] that outperform human accuracy in recognition dataset.

Face recognition is the process to recognize a face that has already been detected. Face recognition includes two general applications: verification and identification. The verification step can be presented as one to one match that correlates a face image with an available face image database whose personality is matched. Face identification is a one to N issue that matches a query face image against the images available in a database of faces. The third case is also taken into account when a query face may or may not be in the available database. In this case, the similarity score is computed and we can find out match based on the highest similarity score. Face detection and matching is important for face feature extraction and accuracy calculation. To the best of our knowledge, the challenges of face recognition process can be assigned to some certain factors such as:

- ***Illumination*** : Illumination represents the changes in light. The small variation in lighting conditions poses an important issue for automatic face recognition and can have an important effect on its results. If the lighting tends to vary, the same person is taken with the same sensor and with an almost identical face pose and expression, the results that emerge can seem quite different. Illumination drastically changes the appearance of the face. The difference between two same faces captured under various illumination conditions is greater than two different faces captured under the same illumination.
- ***Pose variation***: Face recognition systems are easily affected by variations in pose. The pose of the face changes when the movement of the head and the angle of view of the person vary. The different points of view of a camera or the head movements can always cause changes in the appearance of the face and generate intra-class variations causing a considerably drop in the automated face recognition rates. As the angle of rotation increases, identifying the actual face becomes difficult. This can result in incorrect recognition or no recognition if the database only has the frontal view of the face.
- ***Feature occlusion*** : Occlusion indicates blockage, and it happens when one or more parts of the face are blocked and the entire face is not available as an input image. Occlusion is examined to be one of the most significant issues in face recognition system. It happens due to moustache, beard and accessories including glasses, cap, mask, etc. It is widespread in real-world scenes. These components make the issue diverse and therefore make automatic face recognition process more difficult. Variability can be introduced by the presence of elements such as sunglasses, beards or hats. Faces can

also be masked in parts by objects or other faces. Facial features and facial expression also change due to various facial gestures.



(a) Illumination variations.



(b) Pose variations.



(c) Expression variations.

(<https://www.pathpartnertech.com/>)

Figure 1.6: Face recognition challenges.

- **Expressions** : Face is considered as one of the most important biometrics due to the significant role played by its unique features in human identity and emotions. Varying situations lead to several humours that can produce variable emotions and possibly changing facial expressions. Human expressions are especially macro-expressions which are sadness, happiness, disgust, anger, surprise, fear. Micro-expressions are fast and involuntary facial expressions, which display the fast facial patterns. Macro and micro expressions appear on person's face because of changes in his emotional state and as a result of such emotions, effective recognition becomes difficult.
- **Low Resolution** : Any standard image should have a resolution of at least $16*16$. The image with a resolution lower than $16*16$ is called a low-resolution image. These low-resolution images can be captured by small-scale standalone cameras such as street video surveillance cameras (called CCTV cameras), supermarket security cameras and ATM cameras. These cameras may capture a small portion of the face region and since the distance between the face and the camera is not very close, they can only capture the area of the face below $16*16$. Such a low-resolution image does not provide much information because most of them are lost. It can be a great problem in the face recognition process.

- **Ageing** : The texture/appearance of the face changes over time and causes ageing, which represents another issue in face recognition process. Human facial characteristics, lines/shapes, and other features change with the increasing age. It is made for image retrieval and visual observation after a long time. The recognition process depends on extracting features, basic characteristics such as hairstyles, wrinkles, marks, eyebrows, etc.
- **Imaging conditions** : Face appearance depends heavily to the quality of an image that can be affected by different cameras and environmental factors.

Figure 1.6 shows three situations of face recognition challenges. While Figures 1.6a and 1.6b show different light and pose variations, Figure 1.6c displays large expression variations.

Although all of these situations are handled quietly nowadays, they still disturb the face recognition process. It is crucial to note that, until now, there is no face recognition algorithm capable of solving all of these challenges at the same time, which makes the face recognition field more and more challenging.

1.3.2 Face recognition process

Usually, the face recognition process is divided into three steps: face detection, feature extraction, and face recognition. In Figure 1.7, we show an example of how these three steps work on an input image.

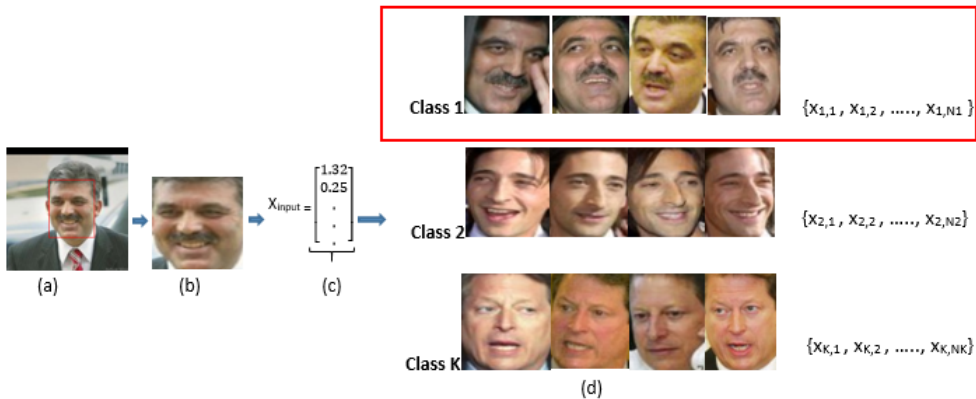


Figure 1.7: The three steps of face recognition process. (a) The output of face detection (the bounding box) (b) The extracted face patch (c) The extracted feature vector (d) Comparing the input feature vector with the vectors stored in the dataset by classification methods and find the most likely class (the red rectangle). Each face patch is described as a d-dimensional vector, the vector $x_{m,n}$ as the n_{th} in the m_{th} , and N_k represents the number of faces stored in the k_{th} class .

- **Face detection** : The principal goal of this step is to determine whether or not the given image/video contains human faces, and to locate these faces. As expected, this

step produces patches that contain each face in the input image. Face alignment is performed to justify the orientations and scales of these patches in order to make the face recognition system more robust and to facilitate its design. In addition to the pre-processing phase for face recognition, face detection could be used for ROI detection, image and video classification, retargeting, etc.

- **Feature extraction** : The face detection step is followed by the extraction of human face patches from images. There are some drawbacks to using these patches directly for facial recognition, firstly, each patch generally contains more than 1000 pixels, which are too large for building a robust face recognition system. Secondly, face patches can be captured from multiple camera alignments, with multiple face expressions, lighting, and may suffer from clutter and occlusion.

To overcome these disadvantages, feature extraction are carried out to perform tasks such as information packing, reduction in size, cleaning noise and main feature extraction. This step consists generally in transforming a face patch into a fixed dimensional vector or a set of landmarks with their corresponding locations. In some facial recognition literatures, face detection or face recognition includes a feature extraction step.

- **Face recognition** : After having formulated the representation of each face, the last step is to recognize the identities of these faces. A database of faces is required to perform automatic recognition. The features extracted from multiple images captured for each person are stored in the database. Then, when an input face image arrives, face detection and feature extraction are performed, and its features are compared to each face class stored in the database. There are two main applications the face recognition models perform, face identification and face verification. Face identification means, given an image of the face, we want the system to say who he/she is or the most likely identification, while in face verification given a face image and an estimate of the identification, we want the system to say true or false about the assumption.

1.4 Contributions of this thesis

Given the above importance of background subtraction, object classification and recognition, we present below the contributions of this thesis. The list of publications concerning this thesis can be found in Appendix B.

1. A novel deep based detector, namely Deep Detector Classifier (DeepDC). DeepDC is based on an unsupervised anomaly discovery framework called DeepSphere for moving objects detection and segmentation in videos (e.g. vehicles, pedestrians, etc). DeepSphere is more robust against the changing nature of anomalies in the training data (e.g., anomaly pollution, nested anomaly extent, spatio-temporal locality) or in the test data (data imbalance) to deal with the challenges enumerated in Section 1.1.1. DeepDC does not require any clean (outlier-free) or labeled data as input, while preserving consistent and robust performance.
2. A new semi-supervised classification method called DCGAN-SSL, which is an extension of the regular DCGAN to simultaneously learn a generative model and a semi-

supervised classifier for the object classification task. Our DeepDC based on DCGAN-SSL trains a multi-class classifier to categorize objects (pedestrians, vehicles, etc), extracted from video sequences, while making use of both labeled and unlabeled data, which is able to perform better than a standalone CNN model. It achieves good accuracy when trained with a few amount of labeled samples. Furthermore, DCGAN-SSL outperforms the baseline in proportion to the reduction in the training set, suggesting that forcing a weight-sharing between the discriminator and the classifier improves data efficiency. DCGAN-SSL works better than an isolated classifier on small training datasets. DCGAN-SSL discriminator can not only learn to distinguish real samples from fake one, but also to discriminate the class label. The proposed model improves classification performance on restricted datasets compared to a classifier without a generator component.

3. A new face recognition approach based on FaceNet model [393] to recognize the extracted people from video sequences through their faces and then to deal with illumination variations and dynamic backgrounds. Our method uses a deep convolutional network trained to directly optimize the embedding itself instead of an intermediate bottleneck layer as in traditional deep learning approaches. We extend our previous approach by proposing a novel data augmentation method based on DCGANs to improve face recognition accuracy. Our approach allows much greater representational efficiency achieving state-of-the-art face recognition performance using only 128-bytes per face.

1.5 Thesis outline

The rest of the thesis is organized as follows.

- **Chapter 2** provides a discussion defining the solved and unsolved challenges in the context of background subtraction as well as presenting the different models suggested to address them. We also provide an overview of the different object classification methods. Furthermore, we review the main face recognition techniques used to identify people. The approaches are analyzed based on the facial representations they used.
- **Chapter 3** presents a Deep Detector Classifier (DeepDC) for moving object detection, that allows distinguishing foreground objects from background in video sequences. DeepDC is based on an unsupervised anomaly discovery framework called DeepSphere. The experiments conducted on real videos from the Background Modeling Challenge dataset (BMC 2012) [446], the Change Detection dataset (CDnet 2014) [460] and the VIRAT video dataset [333] show that the proposed DeepDC outperforms its competitors for the background subtraction task.
- **Chapter 4** describes a novel semi-supervised learning model based on DCGAN discriminator able to classify objects extracted from video sequences (pedestrians, vehicles, etc). The discriminator not only learns false from real images but also classify each real image to its corresponding category. In addition, our proposal allows to train the prediction task of the discriminator with almost small numbers of labeled samples with unlabeled samples to provide the network with further information. Results on VIRAT video dataset [333] and CDnet 2014 dataset [460] show the pertinence of the proposed approach.
- **Chapter 5** This chapter presents a novel face recognition descriptor based on FaceNet model [393]. Our proposed approach directly optimizes the embedding itself by training a deep convolutional network, instead of using an intermediate bottleneck layer as in conventional deep learning methods. Furthermore, our method allows a much greater representational efficiency. A high recognition performance is achieved using only 128-bytes per face. This not only increases the efficiency in terms of time and memory consumption, but also improves the recognition performance. Additionally, we extend our previous approach by a new data augmentation method based on DCGANs. The experiments conducted on Labeled Faces in the Wild dataset [], VGG face dataset [85], Youtube face dataset [468] and Chockepoint video dataset [469] show that the proposed descriptor outperforms other state-of-the-art descriptors. Furthermore, the application of data augmentation based on DCGANs improves the performance of face recognition.
- **Chapter 6** summarizes the thesis with remarks, advantages, and limitations of the proposed approaches. It also discusses the open issues and the promising future directions.

Figure 1.8 schematically illustrates the organization of this manuscript. In particular, the chapters in which the contributions of the thesis are presented, are highlighted.

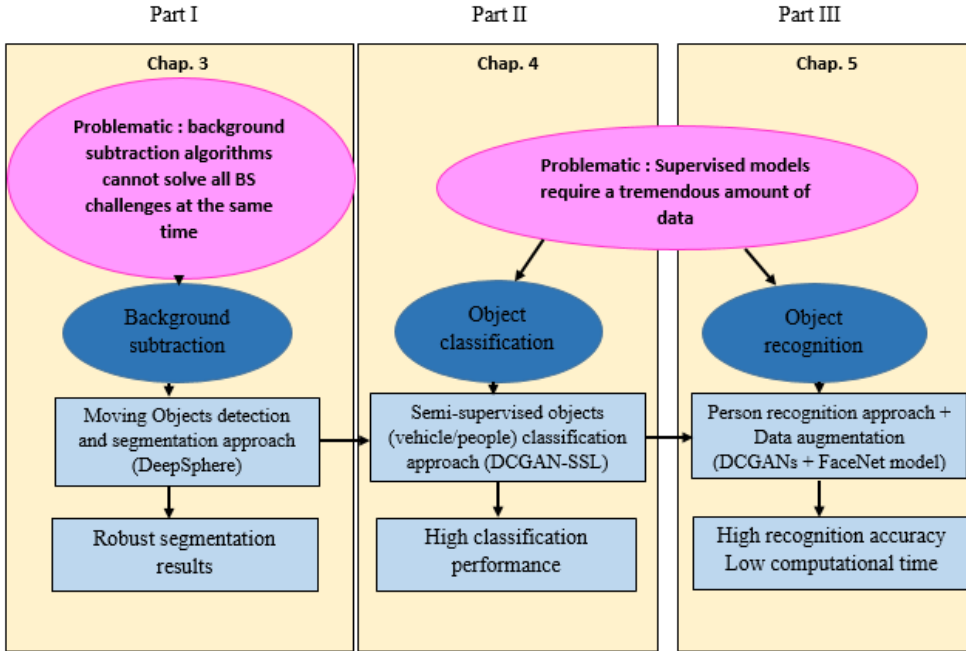


Figure 1.8: Schematic organization of the manuscript and the contributions.

Chapter 2

Literature review

This chapter begins with a brief introduction to the different solved and unsolved challenges encountered in the context of background subtraction, followed by a review of the traditional and recent approaches in this domain. Furthermore, we survey the representative studies in object classification applied once the moving objects are detected. Next, a review of the different algorithms used till now for the holistic-based, local-based, hybrid and deep learning face recognition methods, is provided.

This chapter corresponds to a concise version of our survey published in the International Conference on Cognition and Exploratory Learning in Digital Age (CELDA), Portugal, [32] and our recent survey published in Handbook on "Towards Smart World: Homes to Cities using Internet of Things" [30].

Contents

2.1	Background subtraction models	18
2.1.1	Mathematical models	18
2.1.2	Subspace models	20
2.1.3	Neural network modeling	21
2.1.4	Deep neural networks concepts	22
2.1.5	Signal processing models	23
2.1.6	Semantic concepts	24
2.2	Object classification	27
2.2.1	Conventional methods	27
2.2.2	Deep neural network methods	27
2.3	Face recognition methods	29
2.3.1	Holistic approaches	29
2.3.2	Local approaches	33
2.3.3	Hybrid approaches	35

2.3.4	Deep learning approaches	37
2.4	Solved and unsolved challenges	43
2.4.1	Background subtraction	43
2.4.2	Object classification	44
2.4.3	Face recognition	44
2.5	Conclusion	46

2.1 Background subtraction models

Foreground Segmentation in video streams is a major step in many visual surveillance applications for which background subtraction provides a suitable solution which offers a good compromise in terms of computation time and detection quality. The different steps of background subtraction use methods which have different objectives and limitations. Thus, they need algorithms with different features as presented in Figure 2.1 . Background initialization needs "offline" algorithms which are "batch" learning algorithms by taking all examples at one time. However, background maintenance requires "online" algorithms which are "incremental" algorithms by processing the data one by one. Background initialization, modeling and maintenance need reconstructive algorithms while foreground detection requires discriminative algorithms. Moving object detection methods can be divided into three main categories: the frame differencing method, the optical flow method and the background subtraction method. Frame difference algorithms [106] [188] [504] can be simply developed but they are highly sensitive to the challenges. Optical flow methods are more robust, while meeting real-time requirements remains a difficult task since it needs a lot of time. Background subtraction which is the common technique for detecting foreground objects offers a good compromise between robustness and real-time requirements. In the literature, several surveys [64] [68] [69] [75] [76] [307] and books [65] [73] can be found that handle the problem of moving objects detection by background subtraction.

The background model describes the model used to represent the background. A large variety of models resulting from signal processing techniques, mathematical concepts and machine learning methods have been proposed to model the background as presented in Figure 2.1, including crisp models [175] [265] [382], statistical models [86] [139] [427] [449], fuzzy models [45] [46] [48], Dempster-Schafer models [328], subspace learning models [146] [147] [310] [311] [335], robust learning models [84] [215] [216] [419] neural networks models [363] [364] [392] and filter based models [94] [103] [316] [445]. The main background modeling methods are shown in Table 2.1 and 2.2.

2.1.1 Mathematical models

Depending on mathematical concepts, the easiest way for modeling the background is to calculate the temporal mean [265], the temporal median [175] or the temporal histogram [382] which are the most popular techniques to generate a background and were extensively applied

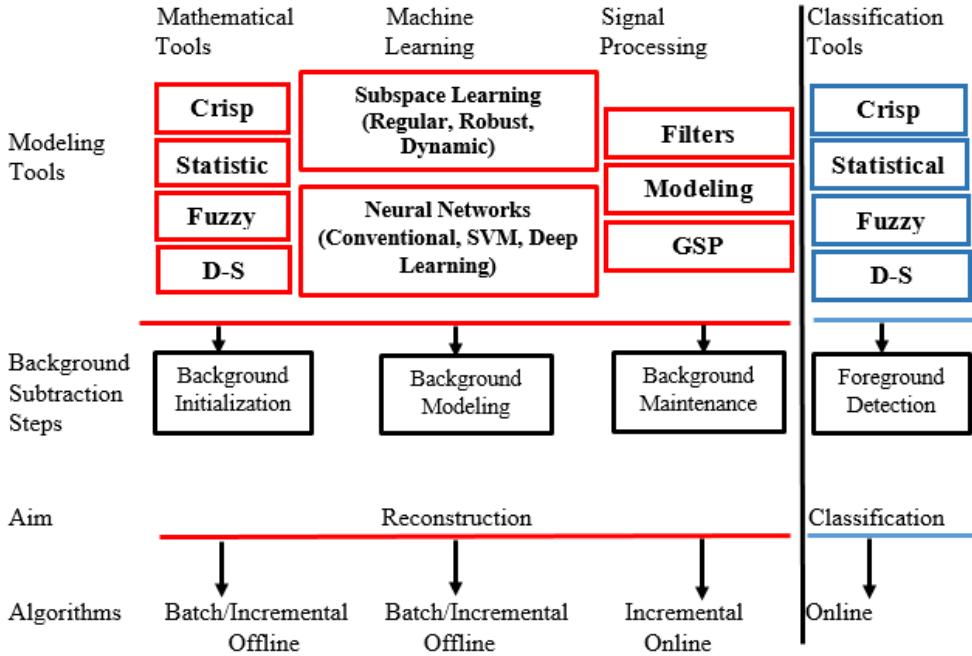


Figure 2.1: An overview of background subtraction models.

in the field of road traffic monitoring in 1990s, while they are very sensitive to the challenges encountered in video surveillance such as camera jitter, variations in lighting and dynamic backgrounds. These models are classified as crisp models. To tackle the imprecision, vagueness and incompleteness in the observed data (i.e. video), statistical models began to be implemented in 1999 like Kernel Density Estimation (KDE) [139] [516], single Gaussian (SG) [470], Gaussian Mixture Model (GMM) [86] [427]. These techniques based on Gaussian distribution models demonstrated their robustness to dynamic backgrounds [157] [353]. In the literature, more sophisticated statistical models have been implemented and can be categorized into those based on another distribution that mitigates the strict Gaussian constraint (i.e. general Gaussian distribution [140], Student's t-distribution [327] [180], Dirichlet distribution [186] [145], Poisson distribution [150] [514]), models based on co-occurrence [279] [280], [281] and confidence [378] [379], models with free distribution [50] [424] [425], and regression models [257] [444]. These methods have improved the robustness to a several challenges over time. The most successful techniques in the statistical category are sample-based methods named ViBe [50], SubSENSE [424] and PAWCS [425]. Another theory that manages inaccuracy, incompleteness and uncertainty is based on the fuzzy concepts. In 2006-2008, multiple authors used fuzzy models such as Type-2 fuzzy sets [45] [67], Sugeno integral [497], Crisp models [362] [265] [497] and Choquet integral [44] [48] [101] which are more robust against dynamic backgrounds [67]. Dempster-Schafer concept has also been used successfully in moving objects detection [328].

- **Statistical models** : It is interesting to take into consideration the improved versions of the current used models like MOG [427], codebook [239], ViBe [50] and PBAS [196]. Numerous improvements of MOG [151] [244] [293] [314] [381] [499] as well as codebook [252] [407] [500] [506], ViBe [187] [199] [498] [508] and IPBAS [217] algorithms are extensively used in real-time applications rather than the original versions of MOG. Six improvements of MOG evaluated by Goyal and Singhai [171] on the CD-net 2012 dataset showed that Shahet al.'s MOG [399] and Chen et Ellis' MOG [100], both published in 2014, reach significantly better detection in real applications compared to previously published MOG algorithms, which are MOG in 1999, Adaptive GMM P1C2-MOG-92 in 2003, Zivkovic–Heijden GMM [515] in 2004, and Effective GMM [267] in 2005. Additionally, there also exist real-time implementation of MOG [277] [348] [388] [389] [438], codebook [437], ViBe [248] [250] and PBAS [158] [249] [251]. Furthermore, robust background generation techniques [357] [440] as well as robust deep learned features [398] [397] with the MOG model could also be considered in the most challenging environmental conditions.
- **Fuzzy concepts** : Critical situations encountered in video surveillance generate uncertainties and inaccuracies throughout the background subtraction process. As a result, fuzzy concepts have been introduced in various steps of background subtraction [119] [120]. Bouwmans [72] presented a study on fuzzy concepts used in background subtraction. The advantage of background subtraction techniques based on fuzzy concepts is that they take no time. In 2008, Sigari et al. [410] [411] implemented a Fuzzy Running Average (FRA) method for background subtraction. Actually, a fuzzy classification based on saturating linear function allows obtaining a fuzzy membership value and then is used to calculate the learning rate for the background average model maintenance. Experimental results on road monitoring demonstrate the relevance of this method in the case of camouflage. In 2008, this method is adapted by Shakeri et al. [402] [403] in cellular automata for urban traffic applications. Cellular automata apply certain rules on pixels to model each frame sequence. The calculation is performed independently in all cells. Experimental results show that the fuzzy cellular running average achieves better performance compared to FRA. In 2011, Yeo et al. [489] [488] extend FRA for moving vehicle detection using infrared videos. Although these models are often studied as fuzzy background models, they should be more considered as fuzzy foreground detection models and fuzzy background maintenance because the fuzzy memberships only occur during these two steps. In 2012, Lu et al. [292] [291] successfully detect foreground vehicles based on Choquet integral.

2.1.2 Subspace models

In 1999, subspace learning methods have been used in an unsupervised way for background modeling in the idea of representing the content of online data while greatly reducing dimension. Subspace learning method (either local or global, linear or non linear) like Independent Component Analysis(ICA) [484], Principal Component Analysis (PCA) [133] [335] [134] [234] and Non-negative Matrix Factorization (NMF) [79] [80] provide a background subtraction framework, especially in the presence of illumination variations. These models are more robust to variations in lighting than statistical models [328]. In other methods, discrimina-

tive [146], [147], [310] and mixed [311] subspace learning models have been employed to improve the results of foreground detection. But, these standard techniques are not robust to outliers, making them very sensitive to the challenges encountered in video surveillance like camera jitter, noise, dynamic backgrounds and lack of data. To handle these limitations, since 2009, subspace learning have attracted renewed interest in this field due to the theoretical progress of robust PCA, created by Candès et al. [84], by decomposition into L+S matrices [215] [216] [419] that has been widely employed making them robust not only to variations in lighting but also to dynamic backgrounds [213] [218] [368] [369]. These techniques are more robust to standard subspace learning methods [183] [211] [212] [213] [214], but they are not applicable in real-time applications [70], since they need the implementation of batch algorithms. To deal with this problem, dynamic RPCA algorithms reviewed by Vaswani et al. [450] [451] are proposed to provide real time performance of methods based on RPCA. The incPCP algorithm [373] and its corresponding improvements [97] [341] [372] [374] [375] [376] [377] [412] as well as the ReProCS algorithm [355] and its multiple variants [178] [330] [356] are presented as online algorithms, provide both advantages in terms of detection, real-time and memory consumption. In road traffic monitoring, incPCP [376] was successfully tested for vehicle counting [358] [442], while an online RPCA algorithm was used for person and vehicle detection [480]. GRASTA [190], incPCP [376], ReProCS [178] and MEROP [330] are the most advanced algorithms in this subspace learning category. Nevertheless, methods based on tensor RPCA [135] [210] [289] [420] make it possible to take into account spatial and temporal constraints allowing more robustness against noise.

2.1.3 Neural network modeling

In 1996, neural networks have been used by Schofield et al. [392] for background representation and moving objects detection, with a Random Access Memory (RAM) neural network. This RAM-NN does not need a background maintenance step and requires a good background model. Information can no longer be changed once RAM-NN is trained with one pass of background images. Jimenez et al. [164] classify each portion of a video into one of the following categories: static, noisy and impulsive background categorie. This classification is made by a multilayer perceptron which requires a training set from specific parts of each training frame. Subsequently, Tavakkoli [441] design a neural network method on the basis of the novelty detection theory. In the training phase, the background is divided into blocks and each block is associated to a Radial Basis Function Neural Network (RBFNN). Each RBFNN is trained with background data that corresponds to its associated block. RBFNN is employed as a detector for close boundary generation for the defined category. In RBF-NN approaches, dynamic object detection can be observed as a single class problem and the dynamic background is learned. However, the representation of common background requires a large amount of data. In Wang et al. [462], a Probabilistic Adaptive Background Neural Network (ABPNN) model is presented which combines both a winner-take-all (WTA) and hybrid probabilistic networks. Each pixel is classified as background or foreground based on a Parzen estimation. The foreground regions are then categorized as a shadow or a motion region. But, initial threshold values should be defined for each of the considered videos. A feed-forward neural network based on an adaptive Bayesian model named Background Neural Network (BNN) is proposed by Culibrk et al. [109], to model the background. BNN is rep-

resented as a General Regression Neural Network (GRNN) and operates as a Bayesian classifier. Although the architecture is considered to be supervised, it can be extended as an unsupervised architecture in the background modeling domain. The network is composed of three subnets: classification, activation and replacement. The background/foreground features of a pixel are mapped using the classifier subnet based on a probability density function estimation. The network contains two summing neurons, one of them calculates the probability that the pixel values belong to the background and the other for estimating if it belongs to the foreground. The principal disadvantages are the high-complexity of the model and the needs of three nets to specify if a pixel belongs to the background. In a disruptive study, Maddalena and Petrosino [296] [297] [298] [299] a Self Organizing Background Subtraction (SOBS) approach allowing to preserve the spatial coherence of the pixel. This approach can be considered as a pixel-based and non-parametric method that simply address the multi-modality in background pixel distributions. The network can be automatically modeled through the network neurons weights and each pixel is represented by a neural map with weight vectors, that are initialized with pixel values in the HSV color channel. Next, each new pixel value from each new frame is classified either in the background or in the foreground by comparing it with its current model. Subsequently, enhanced variants of SOBS has been presented such as Multivalued SOBS [301], SOBS-CF [295], SC-SOBS [303], 3dSOBS+ [305], Simplified SOM [88], Neural-Fuzzy SOM [89] and MILSOBS [160]. These improvements allow SOBS being one of the principal methods on the CDnet 2012 dataset [173] for a long period. SOBS shows also great efficiency for the detection of stopped objects [300] [302] [304]. However, these SOBS-based methods require manually parameter adjustment.

2.1.4 Deep neural networks concepts

Since 2016, DNNs have also been successfully employed in background extraction [178] [357] [478] [479] [480] background subtraction [42] [53] [78] [104] [282], foreground detection improvement [496], ground-truth extraction [461] and deep spatial features learning [268] [332] [398] [397] [500]. Specifically, Restricted Boltzman Machines (RBMs) have been employed by Guo and Qi [181] and Xu et al. [478] to construct the background, to then detect moving objects using background subtraction. Furthermore, deep auto-encoder networks have been used by Xu et al. [479] [480] to perform similar task while Qu et al. [357] employed context-encoder to generate the background. Convolutional Neural Networks (CNNs) have also been employed by Braham and Droogenbroeck [78], Bautista et al. [53] and Cinelli [104] for background subtraction. Many improved variants of CNNs have been used such as cascaded CNNs [461], deep CNNs [42], structured CNNs [282] and two stage CNNs [505]. Robust spatial features are learned using Stacked Denoising Auto-Encoder (SDAE) by Zhang et al. [500] and the density analysis is applied to model the background, whereas Shafiee et al. [398] employ Neural Reponse Mixture (NeREM) to extract deep features employed in the Gaussian Mixture model [427]. Based on deep learning scene recognition model, Chan [93] suggested a scene-awareness algorithm for scene change detection allowing using the suitable background subtraction technique for the corresponding type of challenges. In 2019, Ammar et al. [29] employed a Deep Detector Classifier (DeepDC) to detect and classify moving objects in video sequences. An unsupervised anomaly discovery algorithm called DeepSphere is adapted to detect moving objects. In 2020, Ammar et al. [34]

suggested to employ and validate DeepSphere to detect and then segment moving objects in video sequences. DeepSphere uses both hypersphere learning and deep auto-encoders to reconstruct normal behaviors and remove anomaly pollution. Experimental results show that DeepSphere achieved higher accuracy compared to Deep Probabilistic Background Model (DeepPBM) [149] and Robust Principal Component Analysis (RPCA) [84].

All of these approaches were implemented by researchers who have not yet tested them for real applications. Only Bautista et al. [53] tested the convolutional neural network for detecting vehicles in low-resolution traffic video sequences. However, even their robustness in presence of the concerned unresolved challenges, recent deep learning methods still take too much time and memory to be actually used in real applications. Additionally, these methods need manually labeled data for the training and are generally scene specific. DNNs-based background subtraction can only treat a specific type of scene, and must be retrained for other video sequences [42]. Because the camera is stationary when recording similar scenes, this fact is often not a challenge. But, this may not be the case for some applications, as specified by Hu et al. [197]. Currently, methods based on deep learning seem to be only interesting in a theoretical point and not on a practical point. This current incompatibility [71] can be mitigated only by advances in online and unsupervised deep learning methods.

2.1.5 Signal processing models

The signal processing models take into account the temporal history of a pixel as one-dimensional signal. More precisely, many signal processing algorithms can be used: 1) signal estimation methods (i.e. filters), 2) transform-domain approaches, 3) sparse recovery functions (i.e. compressed sensing), and 4) Graph signal processing (GSP) approaches.

- **Estimation filter** : In 1990, Karmann et al. [233] proposed to estimate the background model of a scene using the Kalman filter. Each pixel with an important deviation from its predicted value is classified as foreground. Many enhancements have been suggested to make this method more robust to difficult situations such as illumination variations and varying backgrounds [62] [143] [316]. In 1999, a pixel level algorithm, called Wallflower, is presented by Toyama et al. [445] to perform probabilistic predictions of the background pixel values, estimated in the following frame by applying the Wiener filter. Chang et al. [94] [95] applied a Chebychev filtering to represent the background. All these filtering methods give important efficiency under slow variations in illumination but they are inefficient in the presence of complex backgrounds.
- **Transform domain models** : In 2005, Wren and Porikli [470] proposed a Fast Fourier Transform (FFT) based Waviz algorithm for background modeling using spectral signatures from multi-modal backgrounds. These signatures are then used to detect incoherent scene changes over time. They further introduce a Wave-Back method [351] which involves frequency decompositions of the historical pixel vector to model the background. For the reference and current frames, the Discrete Cosine Transform (DCT) coefficients are compared giving a distance maps, which are combined into a same DCT temporal window to be more robust to noise and a thresholding is applied to extract foreground objects. This method can address situations such as waving trees.

- ***Sparse signal recovery models*** : In 2008, Cevher et al. [87] were the first authors to propose a background subtraction method based on a compressed sensing technique. They learned and adapted a compressed background representation with a low-dimensionality rather than learning the entire background that is suitable to detect changes. Compressive samples allow foreground objects to be estimated directly without having to build an intermediate image. However, an auxiliary image is needed to simultaneously retrieve the appearance of objects using compressive measurements. To handle this limitation, many improvements have been presented in the literature [122] [325] [464] [465] [475] and important accuracy is reached using Bayesian compressive sensing methods [253] [254] [255].
- ***Graph Signal Processing (GSP) models***: Graph signal processing is an emerging field that tries to extend the concepts of classical digital signal processing to graphs. There is a lot of theoretical progress in recent years, and several applications in domains including machine learning and computer vision [336] [409]. Recently, Giraldo and Bouwmans [165] [166] proposed a semi-supervised background subtraction method called GraphBGS. This algorithm is based on the theory of reconstruction of graph signals [336] and it is very precise with respect to false positives. Unlike most methods of the state-of-the-art, GraphBGS shows competitive results on both static and moving camera sequences. GraphBGS thus lies in between the unsupervised and supervised techniques, leading to a new branch of background subtraction algorithms.

2.1.6 Semantic concepts

In 2017, Braham et al. [77] take advantage of object level semantics to deal with the diversity of difficult scenes for background subtraction. Combining the output of a semantic segmentation algorithm with the output of any background subtraction algorithm allows reducing false positive detections obtained by changes in illumination, dynamic backgrounds and shadows. Additionally, Braham et al. [77] suggest a fully semantic background representation to improve the detection of camouflaged moving objects. In 2019, a background subtraction algorithm is designed by Zeng et al. [495] with real-time semantic segmentation usable for real applications. While operating in real-time, this method achieves superior performance than unsupervised background subtraction algorithms and stills works better than some supervised methods. Semantic concepts have been also employed for background generation [259] [391] allowing their use in applications like privacy protection and video-impainting.

Table 2.1: Background modeling methods: An overview (Part 1).

Categories	Methods	Authors - Dates	Database
Crisp models	Temporal average	Lee et al. (2002) [265]	-
	Temporal median	Graszka et al. (2004) [175]	PETS2001 [1]
	Temporal histogram	Roy and Ghosh (2017) [382]	CDnet2014 [460] [445]
Statistical models	Single Gaussian (SG)	Wren et al. (1997) [470]	-
	Gaussian Mixture Model	Caseiro (2010) [86], Stauffer (1999) [427]	-
	Kernel Density Estimation (KDE)	Elgammal et al. (2000) [139], Zivkovic et al. (2006) [516]	-
	General Gaussian distribution	Elguebaly et al. (2013) [140]	OSU Thermal Pedestrian [123], OSU Color-Thermal [124]
	Student's t-distribution	Mukherjee [327], Guo et al. (2012) [180]	Caviar [2], Wallflower [445]
	Dirichlet distribution	Haines et al. (2012) [186], Fan et al. (2012) [145]	Wallflower [445]
	Poisson distribution	Faro et al. (2011) [150], Zin et al. [514]	PETS2006 [15]
	Models based on co-occurrence	Liang et al. (2015) [279], Liang et al. (2014) [280] [281]	PETS [1], AIST-INDOOR [278], Wallflower [445]
	Confidence	Rosell et al. (2008, 2010) [379] [378]	Wallflower [445]
	Free-distribution models	Barnich et al. (2009) [50], St-Charles et al. (2014, 2015) [424] [425]	CDnet2012 [173]
	Regression models	Tombari et al.(2009) [444], Lanza et al. (2010) [257]	-
	ViBe	Barnich et al. (2009) [50]	-
	SubSENSE	St-Charles et al. (2014) [426]	CDnet2012 [173]
	PAWCS	St-Charles et al. (2015) [425]	CDnet2012 [173]
	MOG	Stauffer et al. (1999) [427]	-
	Codebook	Kim et al. (2004) [239]	-
	PBAS	Hofmann et al. (2012) [196]	CDnet2012 [173]
	Improvements of MOG	Martins et al. (2017) [314], Lu et al. (2018) [293], Rout et al. (2017) [381], Kiran et al. (2017) [244], Farou et al. (2017) [151] [499], Shahet al.'s MOG [399], Chen Ellis'MOG [100]	fish4knowledge [3], underwaterchangedetection [4] Wallflower [445], UCSD [5], RGB-D Rigid Multi-Body [429]
	Improvements of codebook	Zhao et al. (2014) [506], Sharma et al. (2016) [407], Kusakunniran et al. (2017) [252], Zhang et al. (2015) [500]	CDnet2012 [173], CDnet2014 [460]
	Improvements of ViBe	Huang et al. (2014) [199] Han et al. (2014) [187], Zhang et al. (2014) [498], Zhou et al. (2014) [508]	CDnet2012 [173] CDnet2014 [460] RGB-D benchmark [421]
Improvements of PBAS	Javed et al. (2014) [217]	-	
Real-time implementation of MOG	Pham et al. (2010) [348], Li (2012) [277], Salvadori [388] [389] Tabkhi (2013) [438]	PETS2009 [154] VSSN 2006 [6]	
Real-time implementation of codebook	Szwoch et al. (2011) [437]	-	
Real-time implementation of ViBe	Kryjak et al. (2013) [248] [250]	CDnet2012 [173]	
Real-time implementation of PBAS	Kryjak et al. (2013, 2014) [251] [249], Lopez et al. [158]	CDnet2012 [173]	
Fuzzy concepts	Fuzzy Running Average	Sigari et al. (2008) [410] [411]	-
	Fuzzy cellular running Average	Shakeri et al. (2008) [402] [403]	-
	Fuzzy foreground detection	Yeo et al. (2011) [489] [488]	-
	Choquet integral	Lu et al. (2012, 2014) [292] [291] E. Baf et al. (2008) [48], [101]	Aquateque [138] PETS2001 [1]
	Type-2 FGMM	ElBaf et al. (2008) [45], Bouwmans et al. (2010) [67] Darwich et al. [120] [119]	PETS2006 [15] Terravic [318] CDnet2014 [460]
Sugeno integral	Zhang et al. (2006) [497]	PETS2001 [1]	
Dempster-Schafer	Dempster-Schafer theory	Munteanu et al. (2015) [328]	Aquateque [138]

Table 2.2: Background modeling methods: An overview (Part 2).

Categories	Methods	Authors - Dates	Database
Neural networks modeling	RAM neural network	Schofield et al. (1996) [392]	-
	Multilayer perceptron	Jimenez et al. (2003) [164]	-
	Radial Basis Function NN	Tavakkoli et al. (2005) [441]	-
	General Regression NN	Culibrk et al. (2007) [109]	-
	SOBS	Maddalena et al. [296], [299], [297], [298]	-
	Multivalued SOBS	Maddalena et al.(2009) [301]	-
	SOBS-CF	Maddalena et al. (2010) [295]	[7]
	SC-SOBS	Maddalena et al. (2012) [303]	CDnet2012 [173]
	3dSOBS+	Maddalena et al. (2014) [305]	BMC2012 [446]
	Simplified SOM	Chacon-Mugui et al. (2009) [88]	-
	Neural-Fuzzy SOM	Chacon-Mugui et al. (2013) [89]	-
MILSOBS	Gemignani et al. [160]	CDnet2012) [173]	
Deep Neural Network Modeling	Restricted Boltzman Machines	Guo (2013) [181], Xu et al. (2015) [478]	CDnet2012 [173]
	Deep auto-encoder networks	Xu et al. (2014) [480], [479]	Ocean [503] Watersurface [275]
	Context-encoder	Qu et al. (2016) [357]	-
	CNNs	Braham [78], Bautista [53] Cinelli [104]	CDnet2014 [460]
	Cascaded CNNs	Wang et al. (2016) [461]	-
	Deep CNNs	Babae et al. (2017) [42]	CDnet2014 [460], Wallflower [445], PETS2009 [154]
	Structured CNNs	Lim et al. (2017) [282]	CDnet2014 [460]
	Two stage CNNs	Zhao et al. (2017) [505]	CDnet2014 [460]
	SDAE, density analysis	Zhang et al. (2015) [500]	CDnet2012 [173]
	NeREM	Shafiee et al. (2015) [398]	CDnet2012 [173]
DeepSphere	Ammar et al. (2019, 2020) [29] [34]	[91] [333] [460] [446]	
Subspace models	ICA	Yamazaki et al. (2006) [484]	-
	PCA	Oliver et al. (2000) [335], Dong et al. (2011) [134] [133], Kawanishi et al. [234]	PETS2001 [1], VSSN2006 [6]
	NMF	Bucak et al. (2007, 2008) [80] [79]	PETS2001 [1]
	Robust PCA	Candès et al. (2011) [84] Xu et al. (2014) [480], Sobral et al.(2015) [419], [215] [216] [218] [213], [368] [369]	Watersurface [275], Ocean and Rain [503], UCSD [5], MarDT [8]
	Dynamic RPCA	Vaswani et al. (2018) [450] [451]	CDnet2012 [173]
	IncPCP algorithm	[373] [341] [372] [374] [376] [375] [377] [358] [442], [412], [97]	Lankershim [16], Neovison2 [17]
	ReProCS algorithm	Qiu et al. [355] [356], Guo [178], [330]	MR, SL [330]
	Discriminative subspace models	Farcas et al. [146] [147], Marghes [310]	Wallflower [445]
	Mixed subspace models	Marghes et al. (2012) [311]	Wallflower [445] PETS [153]
	GRASTA	He et al. (2012) [190]	[275]
	MEROP	Narayanamurthy et al. (2018) [330]	-
Tensor RPCA	Javed et al. (2015) [210], Sobral et al. [420], Lu et al. [289], Driggs (2019) [135]	CDnet2014 [460], BMC2012 [446]	
Signal Processing Models	Kalman filtering	Karmann et al. (1990) [233]	-
	Wiener filtering	Toyama et al. (1990) [445]	-
	Chebyshev filtering	Chang et al. (2004) [94] [95]	-
	Waviz algorithm, FFT	Wren and Porikli (2005) [470]	-
	Discrete Cosine Transform	Porikli (2005) [351]	-
	Compressive sensing method	Cevher et al. (2008) [87]	BSDS [312]
	Bayesian compressive sensing	Kuzin et al. [254], [254], [255], [253]	Convoy [463]
	Graph-based algorithm	Giraldo et al. (2020) [166] [165]	-
Semantic concepts	Semantic segmentation	Braham et al. (2017) [77]	CDnet2014 [460]
	Real-time semantic segmentation	Zeng et al. (2019) [495]	CDnet2014 [460]
	Semantic background initialization	Pierard et al. (2018) [259], Savakis et al. [391]	SBI [306], SBMnet [223]

2.2 Object classification

In this Section, we review the representative studies in object image classification and retrieval applied once the moving objects are detected as presented in Table 2.3. These objects can be categorized as humans, vehicles, etc.

2.2.1 Conventional methods

In the study of Zhu et al. [510], color and texture features are combined and fed into an Adaboost classifier for feature selection and classification. In Golle [168], an accuracy of 82.7 % was reached using an SVM classifier trained using the color and texture information. In 1999, Transductive SVMs (TSVMs) are proposed by Joachims [220] to classify a text. TSVMs consider a particular test set and attempt to decrease the misclassifications of these samples. In Shruti et al. [408], features are extracted using gabor filter coefficients and are fed into an SVM classifier. In 2011, Zaghden et al. [493] proposed a Fractal dimension method to differentiate Arabic and Latin ancient documents. In 2013, Zaghden et al. [494] combined a fractal dimension approach with local SIFT descriptors to categorize images. In their investigation, Ammar et al. [32] reviewed and categorized the representative classification methods into supervised and unsupervised techniques. In 2018, Ammar et al. [33] modeled each person by a pentagon built with the most representative skeleton joints. Feature vectors are extracted based on the distances between a subset of skeleton joints. Five Euclidean distances are computed using the vertices of two pentagons and SVM is used for classification. Jabri et al. [209] proposed two solutions for the detection and classification of moving vehicles. The first is a classical Adaboost method based on the extraction of Haar-like features, while the second manages a Local Binary Pattern descriptor which will be extracted with the Adaboost classifier. Results show that the Haar-like +Adaboost system is the most important. However, LBP+Adaboost has lower power consumption. Laopracha et al. (2019) proposed a method for selecting appropriate patterns of histograms of oriented gradients (HOGs) to detect vehicles. Indeed, the HOG method produces both ambiguous and redundant, which can bias the classification process. The selected features are tested using different classifiers including, SVM, random forest, K-nearest neighbor and deep neural network.

2.2.2 Deep neural network methods

- **Supervised deep learning methods:** Supervised learning is defined as a learning task that needs labeled training samples. There exist different methods based on deep learning for supervised classification. The potential ability of CNNs to classify images has been demonstrated in 1989 when LeCun et al. [263] classified handwritten zip code digits with only 5 % test error. In Long et al. [287], Fully connected networks (FCNs) are converted into convolutional ones to train an end-to-end CNN for image segmentation. In 2018, Babae et al. [43] modeled the background and extracted the relevant features from an image-background pair using DCNN, which are then fed into a classifier for segmentation. In 2014, handcrafted features are computed by Liu et al. [285] and a bag of words model is built. Both SVMs and Backpropagation Networks are

used for classification. In 2016, Braham and Droogenbroeck [78] proposed a background subtraction method based on CNN. The background is initialized by computing the temporal median on some frames. A patch is extracted around the pixel, transmitted into CNN and classified as background or foreground based on a threshold value. However, all these approaches perform in a supervised manner which requires a large amount of labeled data.

- ***Unsupervised deep learning methods:*** Unsupervised learning does not need labeled data and aims to exploit the large number of unlabeled data and define similarities between objects. In 2016, Li et al. [276] proposed an unsupervised classification method to process remote sensing images and map African land cover using the Stacked Autoencoder (SAE). Results show that SAE outperforms standard classifiers. In 2015, Zou et al. [517] propose a DBN to categorize remote sensing images. In 2014, a hybrid DCNN is used by Chen et al. [99], to detect vehicles in satellite images. In 2017, a Bidirectional GANs (BiGANs) is proposed by Donahue et al. [131], which adds an encoder module to the regular GAN that learns to map among latent and data space.
- ***Semi-supervised deep learning methods:*** The ever-growing size of current datasets combined with the problem of acquiring information on labels makes semi-supervised learning a major challenge of current data analysis. Semi-supervised learning addresses the problem of classification when only a small number of labeled data is available. To deal with this limitation, Ammar et al. [34] proposed a semi-supervised DCGAN (DCGAN-SSL) approach to simultaneously learn a generative model and a DCGAN discriminator classifier to categorize moving objects (humans/vehicles) extracted from VIRAT video dataset [333] and CDnet2014 dataset [460]. DCGAN-SSL enhances the classification performance on small data using a standard classifier without generative element. Rosenberg et al. [380] added unlabeled samples to the original labeled data to train the model in a semi-supervised way which achieves the same results as a standard model using a large amount of labeled data. In 2011, Diederik et al. [242] present a semi-supervised method with generative components that generalizes restricted labeled sets to large unlabeled data. In 2015, categorical GANs (catGANs) are proposed by Springenberg et al. [423] to combine a discriminative classifier from an unlabeled or partly labeled data with an adversarial generative model. In 2017, a semi-supervised virtual adversarial training (VAT) method [323] is proposed that searches for virtually examples to smooth the classifier outputs. Using a small number of labeled samples allows GANs to perform well [387], providing an efficient semi-supervised classification and high quality image generation. In 2018, the NLP language model [116] is used to improve sequence learning with recurrent networks using unlabeled data. The model results in weights used to train the model in a supervised way for data classification. In 2017, a recurrent language Neural Network, called multiplicative LSTM (mLSTM) [361] is trained in a semi-supervised way to estimate the subsequent character in the text. This model exceeded the advanced techniques using only small amount of labeled samples.

Table 2.3: Comparative study of different object classification methods.

Authors	Features/models	Classifier	Supervised	Un-supervised	Semi-supervised	Database
<i>Conventional Approaches</i>						
Zhu et al. (2013) [510]	Color and texture features	Adaboost classifier	✓			APIS 1.0 [510]
Golle et al. (2008) [168]	Color and texture features	SVM	✓			Asirra [141]
Shruti et al. (2014) [408]	Gabor features	SVM	✓			Yale-B [9]
Joachims et al. (1999) [220]	Word stem	Transductive SVMs	✓			-
Zaghden et al. (2013) [494]	Fractal dimensions, SIFT	K-means classifier		✓		-
Zaghden et al. (2011) [493]	Fractal dimensions method	K-means classifier		✓		-
Ammar et al. (2017) [32]	soft-biometric features	SVM	✓			MUCT [320] VIPeR [176]
Jabri et al. (2018) [209]	Haar-like, LBP	Adaboost	✓			GTI vehicle [10]
Laopracha et al. (2019) [258]	HOGs	SVM, KNN, random forest, DNN	✓	✓		GTI vehicle [10] CompCars [485] KITTI [159]
<i>Deep Neural Network Approaches</i>						
LeCun et al. (1989) [263]	Back-propagation network	CNN	✓			-
Long et al. (2015) [287]	CNN	FCNs	✓			PASCAL VOC [142]
Babaei et al. (2018) [43]	CNN	Multi Layer Perceptron (MLP)	✓			CDnet2014 [460]
Liu et al. (2014) [285]	Dense-SIFT features, CNN	Backpropagation Networks	✓			Asirra [141]
Braham (2016) [78]	ConvNets	Fully connected layer		✓		CDnet2014 [460]
Li et al. (2016) [276]	NDVI and MNDWI	Stacked Autoencoder (SAE)		✓		-
Zou et al. (2015) [517]	RBM	Deep Belief Network (DBN)		✓		RSSCN7 [11]
Chen et al. (2014) [99]	Hybrid DCNN	MLP		✓		-
Donahue et al. (2017) [131]	BIGAN	BIGAN discriminator		✓		ImageNet [384]
Ammar et al. (2020) [34]	DCGAN	DCGAN discriminator			✓	VIRAT [333], CDnet [460]
Rosenberg (2005) [380]	Wavelet transform	MSE, Mahalanobis distance			✓	-
Diederik et al. (2014) [129]	Deep Generative models	TSVM with RBF			✓	MNIST [481]
Springerberg et al. [423]	Catgan	Discriminative classifiers			✓	MNIST [481] CIFAR [246]
Miyato et al. (2017) [323]	VAT	NN classifier			✓	MNIST [481], CIFAR [246]
Salimans et al. (2016) [387]	GAN	standard classifier			✓	MNIST [481], SVHN [331]
Dai et al. (2018) [116]	NLP language model	LSTMs			✓	IMDB [294], DBpedia [270]
Radford et al. (2017) [361]	mLSTM RNN	logistic regression classifier			✓	MR [61]

2.3 Face recognition methods

2.3.1 Holistic approaches

Holistic approaches process the entire face area as a high-dimensional vector that is fed into a classifier. These approaches do not need to extract face areas or points of interest. However, they consider all pixels of the image with equal importance, which makes them costly in computation. In addition, these approaches generally ignore local information, so they are not very used for face identification. These approaches can be classified into linear and non-linear techniques according to the method used to represent the subspace in Table 2.4 .

Linear techniques

- **Eigenface and principal component analysis (PCA):** In Seo et al. [394], Locally Adaptive Regression Kernel (LARK) features are extracted to represent a face. A

self-similarity measure is calculated among a center and its neighboring pixels on the basis of a geodesic distance. The size of LARK is reduced using PCA, followed by a logistic function to make LARK features approximately binarized. The one-shot similarity measure is applied on the basis of a linear discriminative analysis (LDA) for the image restricted training. In Ghorbel et al. [163], the DoG filter is applied for image processing. The features are extracted using Eigenfaces and VLC techniques from the entire face image and matched using the chisquare distance. In 2012, Abdullah et al. [22] optimized the time complexity of Eigenfaces without affecting the recognition performance. In 2017, Johannes and Armin [224] have shown that Haar cascade classifiers exceed LBP classifiers in face detection. For face recognition, they demonstrated that Eigenfaces are better than Fisherfaces and LBP histograms. In 2016, Bhuiyan et al. [59] examined the eigenvectors of the covariance matrix of the key images to recognize a face. The features are extracted using Eigenfaces and identified using KNN. Lighting issues are surmounted by Root Mean Square (RMS) contrast stretching. The work of Abd Rahman et al. [21] was performed using PCA Eigenfaces approach to recognize a face in a single static, multiple static and dynamic images. The main idea in [386] was to use only the best Eigenfaces which represent the major variance in all facial images which leads to efficient calculations and speed.

- ***Fisherface and linear discriminative analysis (LDA)***: Fisher vectors are used by Simonyan et al. [414] to recognize a face. The authors proposed a discriminative reduction in dimensionality due to the high size of Fisher vectors. The Fisherface approach is more effective than the Eigenface method. On this basis, Li et al. [273] compared a dual-tree complex wavelet transform (DT-CWT) approach based on LDA with the DTCWT based PCA method. The face recognition efficiency of the Fisherface and the Eigenface are also compared in the DT-CWT area. In Abidin et al. [24], face expressions are recognized on the basis of a neural network using Fisherface. An integral projection method is adopted to segment and locate the face area. Neural network based on the back-propagation algorithm is applied to categorize facial expressions. In Gowda et al. [170], LPQ features are extracted from the face and iris regions and LDA is used for dimensionality reduction in order to achieve efficient computation. Both SVM and KNN are used for classification.
- ***Independent component analysis (ICA)*** : In Bartlett et al. [51], two architectures are proposed to represent facial images using ICA. The spatially local basis vectors are generated by ICA and are considered as a set of independent facial characteristics. In the second architecture, a factorial code is used to generate statistically independent compressed images. The performance of face recognition was evaluated by the KNN classifier and the cosine similarity measure. The authors reported that ICA-based representations outperformed PCA-based representations to recognize a face in sessions and changes in expression. Kong and Bing [245] used both ICA and SVM to recognize a face. Facial features are extracted using Informax algorithm and classified using Fast Least Squares SVM (FLS-SVM).
- ***Improvements of the PCA, LDA and ICA techniques*** : In order to deal with the large variations in appearance and the poor quality caused by approximate alignment of face images, Cui et al. [108] proposed a Spatial Face Region Descriptor (SFRD) to recognize a face by partitioning each image into various blocks in spatial domain, then

extracting the Token-Frequency characteristics from all regions by sum pooling the reconstructing coefficients over the patches of each block. Whitened Principal Component Analysis (WPCA) is applied to reduce the dimensionality of feature vectors to generate robust face descriptors which are combined using Pairwise-constrained Multiple Metric Learning (PMML). In 2018, Khan et al. [236] proposed to solve complex variations problem in face images by selecting the appropriate features from wavelet sub-bands based on particle swarm optimization (PSO). The LBP-DFT technique is proposed which used LBP features to deal with illumination and expression variations and Discrete Fourier Transform (DFT) to solve the issue of translational variance of the Discrete Wavelet Transform (DWT). In Dehai et al. [125], an ameliorated PCA method is introduced using Fast Fourier Transform (FFT) which fuses the amplitude spectrum of one image with the phase spectrum of another image to improve features, followed by the extraction of eigenvectors. Kernel SVM is used as a classifier. In Ridhi et al. [371], a modified PCA method is proposed for face recognition using certain components of the LDA algorithm. Experimental results show that LDA is better than PCA in face recognition. The work presented in Azeem et al. [38] aims to address the problem of partial occlusions in face recognition by using methods based on LDA, PCA, ICA, Local Non-Negative Matrix Factorization (LNMf) and Non-negative Matrix Factorization (NMF). Features extracted from eyes, nose or mouth region are used in the recognition phase. In [261], an approach is proposed which combines 2DPCA for face features extraction and SVM for classification.

- **Frequency domain analysis** : In Huang et al. [201], a patch strategy is acquired using 2D-DWT and an integral projection technology is used to extract facial features for face recognition. The overlapped patches are chosen to improve stability and maintain all local information. The classification is made by using the nearest neighbor classifier (NNC). In Sufyanu et al. [430], a method called ASDCT is proposed which combines anisotropic diffusion-based normalization technique (AS) and DCT. AS was used for preprocessing and DCT was adapted for feature extraction to address the issue of lighting variations and to improve the decorrelation ability of DCT to enhance face recognition. Performance measurements were evaluated using NNC. In Abdulrahman et al. [23], Eigenface and DWT are used for Face recognition. A 3-level DWT decomposition is applied to the images which are then transmitted to the PCA for dimensionality reduction. In Shanbhag et al. [405], the authors applied Spatial Differentiation (SD) technique and Wavelet Transform based Feature Extraction (WTFE) to preprocess the features by eliminating those which are irrelevant. 2D-SWT is applied with 2D-DWT, which, along with Twin Pose Testing Scheme (TPTS) extract pose invariant features which lead to high recognition rates. A Binary Particle Swarm Optimization (BPSO) is used to reduce the number of features.
- **Gabor filters** : In 2006, Perlibakas and Vytaitas [344] proposed to recognize a face based on both Log-Gabor features and PCA. Their algorithm aims to locate Log-Gabor characteristics with maximal magnitudes at only one scale and different orientations. The cosine similarity measure is used to obtain high recognition performance. In [185], an approach based on 2D face image features is proposed using a subset of uncorrelated and orthogonal gabor filters. The feature vector is reduced in size using LDA. The face image was enhanced and normalized to tackle variations in illumina-

tion. To overcome pose and facial expression changes, Ming et al. [322] proposed in 2012 a 3D Gabor Patched Spectral Regression (3D GPSR) method for face recognition which aims to solve least squares issues while using regularization, reduce noise and exploit the efficiency of the discriminant features. The identification of faces relies heavily on the difference among test and gallery images. To cope with this limitation, Cament et al. [83] updated the grid to extract Gabor features using a mesh to model the deformations of the faces. A statistical model is calculated on the basis of the scores using Gabor features to achieve high recognition rates across pose.

Non-linear techniques

- **Robust Kernel PCA (RKPCA)** : In 2019, Fan et al. [144] proposed an optimization of Kernel PCA algorithm called robust kernel PCA (RKPCA) based on a cost function that needs the reconstructed data point to be near to the original one and to the principal subspace to prevent the implicitness of the feature space. RKPCA remains the only unsupervised method that is robust to issues such as sparse noises and lack of data. In order to deal with the difficult optimization of RKPCA, ADMM+BTLS and PLM+AdSS methods are presented. To overcome the problem of ORB (Oriented-Fast and Rotated-Brief) [383] calculation, Vinay et al. [453] proposed in their approach called ORB-KPCA, an algorithm based on both ORB feature descriptor and KPCA [240]. ORB-KPCA is used for face recognition with Threshold Based Filtering (TBF) to filter out the wrong matches. Lu et al. [290] have taken into consideration the problem of the nonlinearity of face models distribution and the "small sample size" (SSS) and have proposed the kernel direct discriminant analysis (KDDA) which generalizes the direct-LDA (D-LDA). D-LDA is based on SVMs, KPCA and generalized discriminant analysis (GDA).
- **Gabor-KLDA** : In 2015, Vinay et al. [455] compare the Gabor-LDA (linear) and Gabor-KLDA (non-linear) to determine which technique is better adapted for face recognition tasks. Both LDA and Kernel Fisher Analysis KFA are used to reduce the dimensionality of facial features filtered by Gabor.
- **Multi-feature shape regression (MSR)** : In 2018, Yang et al. [486] proposed to improve the face recognition performance by adjusting the position of facial parameters using a face alignment algorithm based on multi-feature shape regression (MSR). The MSR uses gradient, color, and local features to improve the accuracy of the estimation of facial landmarks. A subspace projection optimizations (SPO) method is applied to recognize a face.
- **FDDL (Fisher Discrimination Dictionary Learning)** : To address the lack of training images in each class for a linear representation of the variability of the test, Ouanan et al. [338] proposed to extend the FDDL (Fisher Discrimination Dictionary Learning) model for face recognition based on the dictionary of occlusion variants. This dictionary is generated by calculating the difference of deep features among two face image pairs of the same individual.
- **Wavelet transform (WT), radon transform (RT), and cellular neural networks (CNN)**: In Vankayalapati et al. [448], the radon and wavelet transform approaches are com-

bined to extract non-linear features that are robust to facial expression and illumination changes. CNN is also used to extract non-linear facial features to ameliorate the recognition rate and the calculation speed.

- **2FNN (Two-Feature Neural Network)** : In 2010, 2FNN (Two-Feature Neural Network) method is proposed by Devi et al. [127] to recognize a face, which consists of extracting features using PCA and LDA that are merged based on wavelet fusion to enhance the LDA efficiency in case of a small number of images is accessible. Neural networks are used for classification.
- **Deep Dense Face Detector (DDFD)** : In 2015, Farfade et al. [148] have suggested an approach called Deep Dense Face Detector (DDFD) by refining the AlexNet model in the context of face detection.

2.3.2 Local approaches

Local approaches aim to extract specific features from the face image. These methods are sensitive to issues such as facial expressions, small occlusions, and pose changes. They can be categorized into methods based on local appearance which extract local features from sub-regions of the face image and methods based on key-points which extract features located on the points of interest detected in the face image as presented in Table 2.5.

Local Appearance-Based Techniques

- **Local binary pattern (LBP) and its variant** : In 2016, LBP and its extensions, Pyramid of Local Binary Pattern (PLBP) and Rotation Invariant Local Binary Pattern (RILBP) are evaluated by Khoi et al. [237] for face retrieval. The Grid LBP technique is used to split the face image into small regions and then the LBP feature vectors are concatenated into a histogram of spatially enhanced features. This system can support the increase in the size of the dataset without unexpected fall in Mean average precision (MAP). A local-appearance based method called LBP network (LBPNet) was proposed in [474]. The main contribution was to effectively extract hierarchical data representations. Results showed that LBPNet yields a higher accuracy compared to other unsupervised methods using FERET [349] and LFW [198] datasets. Laure et al. [260] used robust LBP for face features extraction to cope with large variations in expressions, lighting, and poses. KNN is applied for classification. One of the local approaches was the multi-scale LBP (MLBP) method proposed in Bonnen et al. [63], an extension of the standard LBP algorithm. Active Shape Models (ASM) are used to extract features and Procrustes Analysis is applied to preprocess MLBP components. Another variant of LBP is the LTP technique proposed in [367]. The similarities of the face components are fused to encode the differences among the central pixel and its corresponding neighbors into a trinary code using LTP to deal with noise. In Hussain et al. [202], Local pattern features are generalized in the local quantized pattern (LQP), using vector quantization and look-up table, which permits them to have deeper surroundings and additional levels of quantization to cope with difficult variations. LQP

acquires a part of the adaptability of visual word features and the calculation efficiency of LBP/LTP. Experimental results on FERET [349] and LFW [198] datasets showed that this representation enhances state of the art by about 3 %. Ghorbel et al. [163] used the DoG filter for preprocessing and the Uniform Local Binary Pattern (uLBP) to extract local features from face images.

- ***Histogram of oriented gradients (HOG)*** : There are lot of works using HOG features for face recognition. In 2015, Karaaba et al. [230] selected the similar regions of two face images by using a most similar region selection algorithm (MSRS) to deal with misalignment. A distances vector is constructed using multi-HOG algorithm. A mean of minimum distances (MMD) and a multi-layer perceptron based distance (MLPD) functions are used to recognize a face. Combined with MSRS, these techniques give high performance. In Arigbabu et al. [37], the face image is preprocessed using a bi-cubic interpolation re-sampling technique and noise removal. The shape of the face image is described locally using both Laplacian edge detector and Pyramid HOG (PHOG) descriptor to recognize human gender. SVM is used for gender classification. Experiments on LFW dataset [198] describe the effectiveness of this method. The work of Leonard et al. [272] showed the efficiency of the correlation filters for face recognition. The best filter is selected according to its robustness to the scale, noise and rotation changes.
- ***Correlation filters*** : Advanced face recognition systems provide sufficient efficiency in controlled environments and they are not very effective in the uncontrolled situations. Correlation filters have proven their effectiveness in pertinent methods under both controlled and uncontrolled settings. On the basis of this architecture, Napoléan and Alfalou [329], proposed to enhance the efficiency of a correlation approach to deal with illumination changes. The LBP-VLC correlator uses a particular Gaussian function for face image filtering to select the edges. A phase-only filters (POF) filter is used to approve the method. Experiments have shown the good efficiency of LBP-correlation methods under lighting changes. In a similar way, Heflin et al. [194] used an UMACE (Unconstrained Minimum Average Correlation Energy) filter based on an eye detection pipeline to decrease face misalignment, improving eye location precision. Experiments conducted on LFW [198] and FDHD [342] datasets demonstrated that this algorithm yields a high face recognition accuracy by giving more attention on the eye localization step. Proposed by Zhu et al. [512], a feature correlation filter (FCF) fuses the representations of faces with a correlation method to achieve the correlation on filter instead of pixel values. FCF can effectively decrease the need for storage with only a small number of features and reach significant performance. In 2013, Ouerhani et al. [339] proposed a correlation method to recognize a face based on a segmented composite POF filter, to increase the detection accuracy and reduce the correlation time. The target image is pre-processed and reconstructed on the basis of a spectral phase to achieve discriminant correlation and to tackle noise and face rotation. The comparison of the peak-to-correlation energy (PCE) to a specific threshold reduces the wrong alarm rate.
- ***Gabor features***: The complexity of the non-linear relation between the spaces of heterogenous face image is one of the drawbacks of heterogenous face recognition.

To address these limitations, Yi et al. [490] proposed an unsupervised Deep Learning method based on the extraction of local Gabor features at localized facial points. RBMs are used to learn locally shared representations which are processed by PCA and matched by cosine similarity.

Key Points Based Techniques

- **Scale invariant feature transform (SIFT)** : In 2015, a face recognition system is proposed in [271] using SIFT descriptor combined with Kepenekci method [235]. The locations of facial landmarks are acquired by Gabor wavelets responses in a dynamic way. A confidence metric based on the posterior probability is presented in a supervised manner to recognize poorly identified faces. The performance of the proposed approach is compared to the Kepenekci method using three public benchmarks, the LFW dataset [198], the AR dataset [313] and FERET dataset [349],
- **Speeded-up robust features (SURF)** : In 2009, Du et al. [136] applied SURF detectors and descriptors to extract image features for face recognition. A measure of similarity is used which contains the number of matched points, the mean value of the Euclidean distance, and the mean distance proportion of the total matched pairs. In 2015, Vinay et al. [454] adopted two variants of detector-descriptor, the SURF detector with SIFT descriptor and the SIFT detector with SURF descriptor, to increase the competence of face recognition systems. The Fast Library for Approximate Nearest Neighbour Search (FLANN) distance measure is used to determine the correspondance/miscorrespondance of the feature descriptors match. In 2016, a face recognition technique is proposed by Shah and Anand in [400] using SURF features and SVM classifier.
- **Binary robust independent elementary features (BRIEF)** : In 2011, Calonder et al. [82] adopted a binary descriptor named BRIEF to compare the descriptors extracted from feature points very quickly and with a low memory requirements. BRIEF leads to a similar recognition precision with SURF and SIFT, while performing fastly. KNN is used with the Hamming distance to match faces.
- **Fast retina keypoint (FREAK)** : To address the problems of insufficient memory and the complexity of the descriptors calculation, Alahi et al. [26] suggested a binary keypoint descriptor called FREAK, based on the distribution of ganglion cells in the retina. FREAK is represented by comparing a setting threshold with the difference in intensity between receptive fields pairs.

2.3.3 Hybrid approaches

Hybrid approaches combine simultaneously local and global features to recognize face images. The hybrid approaches that we presented in this section are summarized in Table 2.6.

- **color, texture, shape features and soft-biometric traits**:Methods fusing various features have received a lot of attention, such as the work of Ammar et al. [32] who

provided a brief knowledge of the different local and global approaches used for people re-identification. They also proposed an hybrid face identification system that combines color, texture and shape features as well as some soft-biometric traits (hair color, skin tone, eyes shape, eyes color, etc) to identify humans through their faces.

- ***Gabor wavelet and linear discriminant analysis (GW-LDA)***: Fathima et al. [152] proposed an approach called HGWLDA that combines both Gabor wavelet and LDA to recognize a face. The global face image is convolved with a gabor filter bank and different subspace variants of 2D-LDA are used to map the characteristics to a feature space. The KNN classifier is used to recognize a face.
- ***Over-complete LBP (OCLBP), LDA, and within class covariance normalization (WCCN)***: Barkan et al. [49] used over-complete LBP (OCLBP), which is an adjusted variant of the LBP with multiple scales. The faces are recognized based on a matrix-vector multiplication and the LDA technique is combined with Within Class Covariance Normalization (WCCN) to reduce large representations and recognize faces.
- ***Advanced correlation filters and Walsh LBP (WLBP)***: In 2015, Juefei et al. [225] presented a Walsh LBP (WLBP) face recognition technique, which uses one example per subject category to produce face images. In the training phase, a non-linear subspace is modeled by learning subject-dependent correlation filters, that is unresistant to pose variations.
- ***SIFT features, Fisher vectors, and PCA***: In 2013, Simonyan et al. [414] combined both SIFT features and Fisher vectors to recognize a face. The dimensionality of the Fisher vectors is reduced using PCA, which are projected linearly into a subspace of low dimension.
- ***CNNs and stacked auto-encoder (SAE) techniques***: One of the most popular hybrid face recognition methods, based on the combination of CNN and stacked auto-encoder (SAE), is presented in Ding and Tao [130], called multimodal deep face representation (MM-DFR). A face feature vector of high dimensionality is extracted using CNNs. The size of feature is reduced using three-layer SAE. Experiments on LFW [198] and CASIA-Web [114] datasets indicate that MM-DFR offers superior performance.
- ***PCA and ANFIS***: In 2015, Sharma et al. [406] presented a method called PCA-ANFIS using both PCA and ANFIS to extract face features under pose variations. The score value obtained by processing face images by PCA, is used by the ANFIS classifier in the recognition process. This neuro-fuzzy method gives a high recognition rate.
- ***DCT and PCA***: Face representation based on the Genetic Algorithm (GA) was known as one of the most successful methods. In 2018, Moussa et al. [326] developed a rapid face recognition system based on GA, DCT and PCA techniques. GA is used as a feature selection method and is combined with DCT-PCA to extract the most informative face features, remove irrelevant ones and then reduce the dimensionality.
- ***PCA, SIFT, and iterative closest point (ICP)***: In 2007, Mian et al. [317] presented a multimodal face recognition algorithm using a 3D spherical face representation in combination with SIFT features. The eyes, forehead and nose parts are used to tackle the impacts of face expressions to improve face recognition. An iterative closest point (ICP) algorithm is applied to match these regions and the matching scores are merged.

- **PCA and Gabor:** Bellakhdhar et al. [56] fuse the phase and magnitude of Gabor's representations to extract face features and they apply PCA for dimensionality reduction.
- **PCA, local Gabor binary pattern histogram sequence (LGBPHS), and GABOR wavelets:** In 2014, Cho et al. [102] suggested a face recognition algorithm, represented with the Local Gabor Binary Pattern Histogram Sequence (LGBPHS) and Gabor wavelets. PCA is used to reduce the dimensionality.
- **PCA and Fisher linear discriminant (FLD):** In 2012, Sing et al. [416] extracted local discriminant features from the face image sub-regions and global features from the entire image. PCA and Fisher linear discriminant (FLD) are applied to reduce the dimensionality of the combined feature vector.
- **SPCA-KNN:** In 2013, Kamencay et al. [229] introduced a face recognition approach called SIFT-PCA-KNN. Face images are preprocessed using a graph-based technique. Harris-Laplace and SPCA (SIFT-PCA) local features are extracted to construct the face descriptors. KNN is applied for classification.
- **Convolution operations, LSTM recurrent units, and ELM classifier:** In 2012, Sun et al. [431] presented a CNN-LSTM-ELM approach to achieve activity recognition with sequential algorithm. It is based on CNNs, Long-Short Term Memory (LSTM) layers and Extreme Learning Machine (ELM) classifier. This method is more convenient for classifying the extracted features and decreases the execution time.
- **SLBP and HOG:** In Annalakshmi et al. [36], the Spatially enhanced Local Binary Pattern (SLBP) is concatenated with the histogram of oriented gradients (HOG) to allow a robust representation of the face image and then to categorize the human gender with SVM. The choice of hybrid characteristics yields great precision by fusing features.

2.3.4 Deep learning approaches

Despite their decent results, machine learning techniques do not work well in unconstrained environments. This is principally due to the fact that machine learning techniques depend on handcrafted representations chosen by experts that can work well for one scenario and fail in other cases. Currently, a huge amount of research papers have been published based on DNNs in the area of facial biometrics with interesting results. A CNN, one of the most common DNNs, reveals a significant benefit on automatic extraction of visual features. Compared with conventional algorithms [482] for face recognition, CNNs are trained in a data-driven way. Additionally, CNN models combine both feature extraction and classification into one framework. Based on its weight-sharing capability, local connectivity and subsampling, CNNs are better able to extract features and make a significant progress in face recognition. Table 2.7 summarizes the main face recognition methods based on Deep Learning.

- **DeepFace** An approach is proposed, in Taigman et al. [439], for aligning faces to a 3D general shape model. They trained a multi-class network on about four thousand identities to recognize faces. A siamese network is also used to optimize the L1 distance between two face features. Their high accuracy on LFW [198] comes from an ensemble of three networks using various color channels and alignments. The predicted distances of these networks are combined using a non-linear SVM.

- **Convolutional Neural network (CNN)** In 2015, Li et al. [274] proposed a cascade of CNN face detectors with multiple resolutions. A calibration network is also proposed to enhance the quality of bounding boxes. CNNs trained on two-dimensional face samples can work successfully for three-dimensional face recognition by refining the CNN with three-dimensional facial scans [472]. Additionally, the three-dimensional context allows an invariance to lightening/make-up/camouflage situations.
- **FaceNet** : Schroff et al. [393] propose the FaceNet model to learn how to map from a face image towards an euclidean space embedding, in which the distances between the embeddings directly correspond to a measure of face similarity.
- **DeepID** : A DeepID model is developed by Sun et al. [432] that contains multiple CNNs rather than a single CNN, by which a strong feature extractor is built. The facial patches are fed into a DeepID which extracts features from various facial positions.
- **DeepID2** :Sun et al. [434] suggested an extension to DeepID named DeepID2, which employs both identification and verification signals to decrease intra-class variances while extending the inter-class discrepancy.
- **DeepID2+** : DeepID2+ [433] is proposed to enhance the DeepID2 performance by adding the supervision signals to all layers and augmenting the size of each layer.
- **VGG-16** : Simonyan et al. [415] present a DCNN model called VGG-16 and reach an accuracy of 98.95% using 2.6 million samples. This model needs less training data compared to DeepFace and FaceNet and employs a simpler network than DeepID2. But, the construction of such a large dataset exceeds the capabilities of academia groups.
- **DeepID3** : In 2015, two DNN architectures [113] are proposed, mentioned as DeepID3, for face recognition, which are reconstructed from the stacked convolutions of VGG and the inception layers of GoogLeNet. Supervisory signals are used to decrease the intra-personal face features variations. DeepID3 reached peak performance on both verification and identification tasks.
- **SphereFace** : Liu et al. [286] present an angular margin penalty to simultaneously impose extra intra-class compactness and inter-class separability.
- **ArcFace** : An additive Angular Margin Loss function is proposed by Deng et al. [203] which can successfully improve the discriminating power of feature embeddings learned through CNNs for face recognition.
- **CNNs and PCA and SVMs** : Zhu et al. [513] proposed to wrap faces into a canonical frontal view based on a deep network. First, CNN is trained and then, every face is categorized as corresponding to a known identity. A set of SVMs in conjunction with the dimensionality reduction technique PCA on the network output are used to perform face verification.
- **Center loss** : Wen et al. [466] were the pioneers of the center loss, which is a supervisory signal to learn a center for deep features of each class and penalizes the distances between each deep feature vector and its corresponding class center. However, it is very complicated to update the actual centers during training because the number of face classes available for training has grown considerably.

Table 2.4: Face recognition using holistic approaches: An overview.

Authors	Techniques	Database	Matching	Limitation	Advantage
<i>Linear Techniques</i>					
Seo et al. (2011) [304]	LARK + Eigenfaces and DoG filter	LFW [198]	L2 distance	Detection accuracy	Reducing Computational Complexity
Ghorbel et al. (2016) [163]	PCA	FERET [349]	Chi-square distance	Processing time	Reduce the representation complexity
Abdullah et al. (2012) [22]	PCA	FACE94 [121]	-	-	-
Johannes et al. (2017) [224]	PCA	AT & T [14]	Eigen classifier	-	Robust to changes in lighting, face rotation
Bhuiyan et al. (2016) [59]	PCA	Computer Vision Research	KNN	facial expressions, lighting conditions	High recognition accuracy, Robust under controlled environment
Rahman et al. (2014) [21]	PCA, Eigenfaces	-	-	-	-
Saha et al. (2014) [386]	Eigenfaces	FRAY Face DB	-	-	-
Simonyan et al. (2013) [414]	Fisherface	LFW [198]	Mahalanobis matrix	Single feature type	Robust
Li et al. (2009) [273]	DF-CWT-based Fisherface	PCA, AT & T [14]	probabilistic reasoning model (PRM) classifier	-	superior classification performance, high recognition rate
Abidin et al. (2012) [24]	Fisherface	Japanese Female Facial Expression (JAFPE)	feed forward neural network	hard discrimination of ambiguous expressions	high recognition rate
Gowda et al. (2018) [170]	LPO and LDA	MEPCO	SVM	Computation time	Good accuracy
Bartlett et al. (2002) [51]	Eigenface and ICA	local dataset	KNN classifier, cosine similarity	-	high accuracy and success recognition rate
Kong, Rui et al. (2011) [245]	ICA	ORL [14]	SVM	-	-
Cui et al. (2013) [108]	Bow	AR [313], ORL [14], FERET [349]	ASM	Occlusions	Robust
Khan et al. (2018) [236]	PSO and DWT	CK, MMI, JAFFE	Euclidean distance	noise	Robust to illumination
Dehat et al. (2013) [125]	PCA and FFT	YALE [55]	SVM	Complexity	Discrimination
Ridhi et al. (2017) [371]	PCA	YALE [55]	Nearest Neighbor (NN)	Registration errors, bad expression variations	High recognition rate
Azeem et al. (2014) [38]	curvelet transform	ORL [14]	Least Square SVM	High recognition, Fast computational speed	Robustness
Le et al. (2011) [261]	2-D-PCA	FERET [349], AT & T [14]	SVM	-	-
Huang et al. (2015) [201]	2D-DWT	FERET [349], LFW [198]	KNN	Pose	Frontal facial images
Sufyanu et al. (2016) [430]	DCT	ORL [14], YALE [55]	NCC	High memory	Controlled and uncontrolled data
Abdulrahman et al. (2014) [23]	PCA	YALE [55]	Euclidean distance	-	reduce computation complexity, fast convergence
Shanbhag et al. (2014) [405]	DWT BPPO	-	-	Rotation	Significant reduction in the number of features
Peritbakas et al. (2006) [344]	PCA and Gabor filter	FERET [349]	Cosine similarity	Precision	Pose
Haréz et al. (2015) [185]	Gabor filter and LDA	ORL [14], YaleB [9]	DNGC	Pose	Good recognition performance
Ming et al. (2012) [322]	3D GPRS	FRGC [350], CASIA [114]	Nearest Neighbor (NN)	Registration errors, bad expression variations	High recognition rate
Ciment et al. (2015) [83]	Gabor jets computation	FERET, CMU-PIE Sim2003	Borda count threshold	Some alignment steps	Robust to pose variations
<i>Non-linear techniques</i>					
Fan et al. (2019) [144]	RKPCA	MNIST, ORL [14], COIL20, YaleB [9]	RDF Kernel	Works only with data have nonlinear structures	Robustness to sparse noises
Vinay et al. (2018) [453]	ORB and KPFA	ORL [14]	FL-ANN matching	Processing time, Low recognition rate	Robust, Reduce the dimensionality complexity
Kim et al. (2005) [240]	Kernel Hebbian features	Yale Face DatabaseB	projection method	slow convergence	reduce time complexity, super-resolution and denoising
Lu et al. (2003) [290]	KPCA and GDA	UMIST face [174]	SVM	High error rate	Excellent performance
Vinay et al. (2015) [455]	Gabor-LDA, Gabor-KFA	ORL [14]	MAHCOS, Euclidean Block distances	City, Low studies compared	Reduce the dimensionality reduction
Yang et al. (2018) [486]	PCA and MSR	HELEN face	ESR	Complexity	Utilizes color gradient and regional information
Onanan et al. (2018) [338]	FDDL	AR [313]	CNN	Occlusions	Orientations, expressions
Vankayalapati et al. (2009) [448]	wavelet transform, cellular neural networks	ORL [14]	-	Pose	High recognition rate
Devi et al. (2010) [127]	2FNN	ORL [14]	NN classifier	Complexity	Low error rate
Farfale et al. (2015) [148]	Deep Dense Face Detector (DDFD)	AFLW, FRGC2.0 [350]	CNN	occlusion	handle occlusion, pose variations, minimal complexity

Table 2.5: Face recognition using local approaches: An overview.

Authors	Techniques	Database	Matching	Limitation	Advantage
Local Appearance-Based Techniques					
Khoi et al. (2016) [237]	LBP	TDF, LFW [198]	CF1999, MAP	Skwness in face image	Robust feature in frontal face
Xi et al. (2016) [474]	LBPNet	FERET LFW [198]	[349], Cosine similarity	Complexities of CNN	High Recognition Accuracy
Laure et al. (2017) [260]	robust LBP	LFW [198], CMU-PIE [413]	KNN	Illumination conditions	Robust
Bonnen et al. (2012) [63]	MRF and MLBP	AR [313]	Cosine similarity	Failure of Landmark extraction	Robust to changes in facial expression
Ren et al. (2013) [367]	Relaxed LTP	CMU-PIE Yale B	[413], Chisquare distance	Noise level	Superior performance compared with LBP, LTP
Hussain et al. (2012) [202]	LPQ	FERET LFW [198]	[349], Cosine Similarity	Lot of discriminative information	Robust to illumination variations
Karaaba et al. (2015) [230]	HOG and MMD	FERET [349]	MMD/MLPD	Low recognition accuracy	Aligning difficulties
Arigbabu et al. (2017) [37]	PHOG	LFW [198]	SVM	Complexity and time of computation	Head pose variations
Leonard et al. (2012) [272]	VLC correlator	PHPID	ASPOF	The low number of the reference image used	Robustness to noise
Napoléon et al. (2014) [329]	LBP and VLC	YaleB Extended [161]	[9], YaleB POF	Illumination	Rotation + Translation
Heflin et al. (2012) [194]	UMACE correlation filter	LFW [198]/PHPID	PSR	Some pre-processing steps	More effort on the eye localization stage
Zhu et al. (2007) [512]	PCA-FCF	CMU-PIE FRGC2.0 [350]	[413], Correlation filter	Use only linear method	Occlusion-insensitive
Ouerhami et al. (2013) [339]	VLC	PHPID	PCE	POWER	Processing time
Yi et al. (2015) [490]	PCA	RBM	Cosine similarity	Low level expressions	Robustness to variations
Key-points based techniques					
Lenc et al. (2015) [271]	SIFT	FERET [349], AR [313], LFW [198]	A posterior probability	Still far to be perfect	Sufficiently robust on lower quality real data
Du et al. (2009) [136]	SURF	LFW [198]	FLANN distance	Processing time	Robustness and distinctiveness
Vinay et al. (2015) [454]	SURF+SIFT	LFW [198], Face94 [121]	FLANN distance	Processing time	Robust in unconstrained scenarios
Shah et al. (2016) [400]	SURF	UMIST faces [55]	[174], Yale SVM	pre-processing	High recognition rate, Fast computational speed
Calonder et al. (2011) [82]	BRIEF		KNN	Low recognition rate	Low processing time
Alahi et al. (2012) [26]	FREAK	[319]	Hamming distance	Overlapping between receptive fields	Reduces computational cost and memory
Xie et al. (2009) [476]	curvelet transform	ORL [14]	Least Square SVM	High recognition, Fast computational speed	Robustness

Table 2.6: Face recognition using hybrid approaches: An overview.

Authors	Features/Techniques	Database	Matching	Limitation	Advantage
Ammar et al. (2017) [32]	color, texture and shape, soft-biometric features	MUCT [320], VIPeR [176]	SVM		Robustness, High accuracy
Fathima et al. (2015) [152]	GW-LDA	AT & T [14]	KNN	High processing time	Illumination invariant, reduce dimensionality
Barkan et al. (2013) [49]	OC-LBP, LDA, WCCN	LFW 2018	WCCN	-	Reduce the dimensionality
Juefei et al. (2015) [225]	ACF and WLBP	LFW [198]	-	Complexity	Pose conditions
Simonyan et al. (2013) [414]	Fisherface + SIFT	LFW [198]	Mahalanobis matrix	Single feature type	Robust
Sharma et al. (2015) [406]	PCA-ANFIS ICA-ANFIS LDA-ANFIS	ORL [14]	ANFIS	Sensitivity, Specificity	Pose conditions
Moussa et al. (2018) [326]	DCT-PCA, genetic algorithm	Yale [55], ORL [14] UMIST [174]	Euclidean distance	-	Fast, High classification rates
Mian et al. (2007) [317]	Hotelling transform, SIFT, and ICP	FRGC [350]	ICP	Processing time	Facial expressions
Cho et al. (2014) [102]	PCA-LGBPHS GABOR Wavelets	Extended Face [161]	Yale Bhattacharyya distance	Illumination condition	Complexity
Sing et al. (2012) [416]	PCA-FLD	CMU FERET AR [313], [413], [349],	SVM	Robustness	Pose illumination and expression
Kamenay et al. (2013) [229]	SPCA-KNN	ESSEX	KNN	Processing time	Expression variation
Sun et al. (2018) [431]	CNN-LSTM-ELM	OPPORTUNITY	TM/ELLSM	High processing time	Automatically learn feature representations
Ding et al. (2015) [130]	CNNs and SAE	LFW [198]	-	Complexity	High recognition rate
Bellakhdhar et al. (2013) [56]	PCA, LGBPHS, DPL	YALE face [55]	Bhattacharyya distance	-	High recognition, computation time reduction
Bellakhdhar et al. (2013) [56]	Magnitude and phase of Gabor, PCA	ORL [14]	SVM	-	High recognition rate
Annalakshmi et al. (2019) [36]	ICA and LDA	LFW [198]	Bayesian classifier	Sensitivity	Good accuracy
Annalakshmi et al. (2019) [36]	PCA and LDA	LFW [198]	Bayesian classifier	Sensitivity	Specificity

Table 2.7: Face recognition using deep-learning approaches: An overview.

Authors	Features/Techniques	Database	Matching	Limitation	Advantage
Sun et al. (2015) [113]	DeepID3	LFW [198]	-	Small training data	High accuracy rate
Li et al. (2015) [274]	CNN cascade	AFLW	CNNs	Complexity	Low error rate, Much faster detection
Schroff et al. (2015) [393]	FaceNet model	LFW [198] YFD [468]	SVM	Big-sized network	Reduce error rate, High accuracy
Liu et al. (2017) [286]	Sphereface	LFW [198] YFD [468] Megaface [321]	Nearest neighbor classifier	-	high open-set face recognition accuracy
Deng et al. (2019) [203]	ArcFace	CASIA [114] VG- GFace2 [85] LFW [198]	-	-	High verification accuracy
Sun et al. (2014) [432]	DeepID	LFW [198]	ConvNets	Small training sets	High recognition accuracy with only weakly aligned faces
Sun et al. (2014) [434]	DeepID2	LFW [198]	-	Much shallower recent compared DNN	Low error rate
Sun et al. (2015) [433]	DeepID2+	LFW [198] YFD [468]	-	Much shallower recent compared DNN	Robust to occlusions
Li et al. (2017) [274]	CNNs	3D-TEC [452] BU-3DFE [491]	KNN	Depends on the number of objects in the scene	Robustness, Low processing time
Taigman et al. (2014) [439]	DeepFace	LFW [198] YFD [468]	fully-connected layer	-	Rapid computation time
Zhu et al. [513]	CNN and PCA	LFW [198]	SVM	Complexity	High performance
Simonyan et al. (2014) [415]	VGG-16	UCF101 [422]	SVM	sensitivity to camera motion	Does not require significant hand-crafting, reduce computational time
Wen et al. (2016) [466]	CenterFace	LFW [198] YFD [468]	Fully connected layer	-	Robust face recognition

2.4 Solved and unsolved challenges

2.4.1 Background subtraction

In order to adequately evaluate and compare videos, the challenges presented in CDnet 2014 [460], which is a part of the Change Detection Workshop (CDW 2014), are taken into consideration. This database includes all the CDnet 2012 [173] videos plus 22 additional ones taken by cameras covering five various categories that include supplementary challenges that were not solved in the CDnet 2012 database [173]. The categories are called as follows: “dynamic backgrounds”, “baseline”, “shadows”, “camera jitter”, “thermal”, “intermittent object motion”, “low frame-rate”, “challenging Weather”, “PTZ”, “turbulence” and “night videos”. Additionally, while ground truths for all frames of the CDnet 2012 dataset [173] were made available publicly for test and evaluation purposes, ground truths of only the first half of each video sequence in the five additional categories from the CDnet 2014 dataset [460], are made available publicly for test. However, the assessment will cover all frames for all the video sequences in CDnet 2012 [173]. All challenges presented in these different categories have several temporal and spatial characteristics. Therefore, it is crucial to identify both the solved and unsolved challenges. The CDnet 2012 [173] and CDnet 2014 [460] datasets help to highlight when it is difficult to provide robust moving objects detection for current BS methods. A very important observations are provided by Jodoin [222], in 2015, regarding both the solved and unsolved challenges based on the experimental results conducted on the CDnet 2014 dataset [460]. Challenges encountered in “baseline” and “bad weather” videos can be effectively addressed by current background subtraction algorithms. The “camera jitter”, “thermal” and “dynamic backgrounds” categories are an available challenge for the best background subtraction methods. The “low frame-rate”, “Night videos” and “PTZ” sequences are highly difficult. In a valuable study, Bouwmans et al. [71] provide a survey about the progression made over the recent years from the MOG model [427] designed in 1999 to the current DNNs models developed in 2019. This study reveals that the big difference was reached by DNNs algorithms compared to SuBSENSE with 32.92% and 24.31% using respectively FgSegNet-V2 and Cascaded CNN. The gap of 1.55% that persists between FgSegNet-V2 and the best algorithm is lower than the difference of 6.93% between Cascaded CNN and FgSegNet-V2. However, the large gap obtained by FgSegNet-V2 and Cascaded CNN is usually due to their supervised appearance, and a necessary limitation of training using labelled data.

However, when no labeled samples are available, considerable attention should be focused on unsupervised approaches as well as unsupervised GAN, robust subspace tracking [330] [352] [374] [376] [450] and semantic background subtraction [77] [495] that are often fascinating in the background subtraction domain. In addition, deep learning methods effectively identify the changed areas in images with fixed backgrounds but still suffer from multiple challenges such as varying backgrounds and camera jitter, even if they offer a higher efficiency than conventional approaches [231]. Generally, experiments conducted on the “IOM” and the “PTZ” categories are prevented. Additionally, these categories usually give low F-measure. As a result, it appears that the recently evaluated DNNs have problems in these categories, possibly due to the difficulties of dealing with changes occurred at moving cameras and learning the sleep period of moving objects. Finally, despite the

progress of background subtraction models developed for stationary cameras, camera jitter and PTZ cameras, with many RPCA [97] [191] [192] [374] [412] [467] and deep learning models [283] [284] they can only handle small jitter issues or rotation and translation motions. Thus, more particular algorithms and models are required for moving objects detection. Once the objects are detected, their classification can be performed for subsequent processing modules such as tracking and recognition.

2.4.2 Object classification

Object classification is an active area of research in computer vision [35]. However, we still do not have a computer vision system that can achieve human-level classification ability for images. Object classification is still a challenge due to the tremendous variations in images such as translation, rotation and changes in scale and illumination. CNN is the current state-of-the-art object classification method [193] [247]. It has been used in many object classification competitions [105] [247] [436]. It has been proven that CNN can even outperform humans in recognizing 1000 objects [436]. However, CNN presents a serious problem: it requires a large amount of labeled samples. The lack of labeled training examples is the most challenging problem of the image classification tasks. Additionally, the acquisition of labeled data is very expensive and time-consuming. In order to reduce the dependence of CNN on labeled data, the field of unlabeled data should be considered. Unlike labeled data, unlabeled samples are numerous and can be obtained inexpensively. Learning from unlabeled samples is an unsupervised learning task. The pursuit of unsupervised learning for image classification began in 2006 [195]. Although intensive research has been carried out on this topic, recent state-of-the-art image classification method, CNN, is a purely supervised learning method. The present success of supervised learning techniques is mainly due to the current large datasets and the existing labels [177]. However, unsupervised learning methods will become the main considered solution with a quick rise in data complexity and size [366]. Unsupervised learning methods such as sparse coding and pre-training are unnecessary for obtaining high-performance image classification [105] [247] [436]. Labeling large amount of images is unrealistic and time consuming for many image classification applications. The need to develop semi-supervised techniques, which allow training a system with only a few labeled samples together with large amounts of unlabeled samples increase faster. The latter being widely available and inexpensive, this could considerably help the classification of objects. Once objects detection and classification (humans, vehicles, etc) is done, a face recognition step is needed in order to identify extracted people.

2.4.3 Face recognition

LFW dataset [198] was published in 2007 and contains 13,233 face images of 5749 people. As the most famous benchmark used for evaluating the performance of the deep learning techniques under unconstrained conditions, its accuracy has reached almost 100% [365]. However, the faces in LFW dataset [198] are mostly frontal without extreme pose or severe illumination, while there are no difficult situations. VGG-Face2 [85] includes 3.32M from 9131 identities. Compared with LFW [198], this dataset is not publicly available and it

contains faces with pose variations. Most of the global face recognition techniques such as PCA [133] [134] [234] [335] and LDA [484] are used to reduce the dimensions and to select the useful information. However, these approaches are not effective in the unconstrained environments where pose, illumination and occlusion are uncontrolled. Local approaches are considered as a robust approaches in the unconstrained cases compared with global approaches. Recently, CNNs have shown excellent performance in various face recognition tasks [113] [189], e.g., Rajeev et al. [365] and Schroff et al. [393] presented that their proposed method achieved the accuracy of 99.78% and 99.63% on the QLFW dataset [232], respectively. However, it remains difficult for them to obtain sufficient precision on faces under uncontrolled environment with variations in illumination, pose and occlusion, among which occlusion has been considered the most difficult. On the one hand, data imbalance in face datasets should be one possible reason for this phenomenon. Although most facial recognition datasets contain a huge amount of identities, they still suffer from the lack of occluded facial images. It seems that, without training with a large number of occluded face images, DCNNs cannot perform well due to the higher inter-class similarity and the larger intra-class variation caused by occlusions [162]. To solve this problem, more occluded face images should be involved into the CNN training process. On the other hand, the loss function could also have great impact on the training of CNN for face verification and results in poor performance as it could be biased to the data distribution. For example, softmax loss, which was not specifically designed for complex samples, would neglect occluded faces by increasing the conditional probability of all samples. To deal with this problem, numerous loss functions and constraints on the traditional loss functions have been presented [365] [203] [204]. A straightforward way to get better CNN model performance under partial occlusion is to train the network with occluded faces. Challenges caused by unconstrained illumination and environmental degradation such as blurring and problems resulting from large stand-offs and poor image quality can be effectively resolved by incorporating a sensitivity term into a DCNN cost function [19]. This method has been shown to be effective in day and night time images and at different stand-off distances on the Long Distance Heterogeneous Face dataset [112], however it has only been tested on a small, augmented dataset. Another methodology which achieved competitive results without the benefit of large-scale annotated datasets was presented by [40] which used deep convolutional belief networks based on local convolutional restricted Boltzmann machines. Unsupervised representations were learned from unlabeled samples and then transferred to a classification model like SVM and metric learning algorithms for recognition task. The performance of the facial recognition system depends mostly on image acquisition conditions, mainly when the posture changes and because the acquisition techniques themselves may include artifacts. In this case, the challenge of face recognition systems is to distinguish individuals from images captured using cameras, presenting low-resolution, block artifacts, or faces with variable poses. This challenge remains unsolved and requires further research. What's worse, most methods aimed at treating just one aspect of unconstrained facial changes only, such as pose, lighting or expression. There was no any technique to deal with these unconstrained challenges in an integral way. Therefore, "shallow" methods only improved the accuracy of the LFW dataset to approximately 95% [110] and are insufficient to extract stable identity feature invariant to real-world changes. Due to the insufficiency of this technique, facial recognition systems were often resulted in unstable performance in real-world applications. The single sample face recognition (SSFR) [155] represents one of the most difficult face recognition problems, where there is

only one face representation per individual for training. Approaches based on deep learning require large training data to function properly [155]. SSFR remains an unresolved issue and is among the most common topics in industry or academia. While there are several risks with facial recognition, it also offers numerous solutions for future and upcoming technologies.

2.5 Conclusion

As discussed in this chapter, numerous approaches of background subtraction, object classification and recognition have been proposed until the present date. However, there still exist open research questions to be investigated, as for example no traditional algorithm today still seem to be able to simultaneously address all the key challenges of illumination variation, dynamic camera motion, cluttered background and occlusion. We believe that one way to solve this issue is by the systematic investigation regarding the role and importance of features within foreground detection, object classification and recognition. In the following chapters of this thesis, we tackle the problem by beginning proposing a new deep detector classifier based on an unsupervised anomaly discovery framework, that unlike the general foreground-background separation task, detects moving objects (vehicules/pedestrians...) without any additional image processing or background learning. Furthermore, we present an object classification approach to categorize the extracted objects in a semi-supervised way using the discriminator network of DCGANs as a classifier. In addition, we propose a new face recognition approach to identify the extracted faces based on FaceNet model [393] with DCGANs data augmentation to achieve high recognition accuracy.

Chapter 3

A novel deep detector classifier (DeepDC) for background subtraction in videos

In this chapter, we propose a Deep Detector Classifier (DeepDC) for moving objects detection and segmentation in videos. Our proposal consists of adapting an anomaly discovery framework called "DeepSphere" to the foreground-background separation task. By combining the strengths of hypersphere learning and deep auto-encoders, DeepSphere appears to be robust in dealing with the changing nature of anomalies in the training data (e.g., pollution of anomalies, spatiotemporal locality, extent of nested anomalies) or in the test data (data imbalance). Experiments conducted on VIRAT dataset ¹ [333], real videos from BMC2012 dataset ² [446] of outdoor scenes and the Change Detection 2014 dataset ³ [460] under several conditions show that the proposed DeepDC outperforms its competitors for the background subtraction task. Results show that DeepDC is less sensitive to noise and to the dynamic nature of the background and produces a good segmentation masks, while preserving robustness to illumination changes. Results also indicate that DeepDC is able to detect foreground objects without additional image processing.

The work presented here was published at the International Symposium on Visual Computing (ISVC), Nevada, USA (oral presentation) [29] and the IET image processing journal [34]. The reader can found the related source code on Python ⁴.

Contents

3.1	Motivation	48
3.2	DeepSphere architecture	49
3.3	Proposed DeepDC descriptor	51
3.4	Experimental results and discussions	54

¹<https://viratdata.org/>

²<https://pgram.com/dataset/background-models-challenge-bmc/>

³<http://changedetection.net/>

⁴https://github.com/ammarsirine/BS_DeepSphere

3.4.1	Description of the datasets	54
3.4.2	Quantitative and qualitative evaluation	55
3.5	Conclusion	78

3.1 Motivation

Moving objects detection plays a significant role in many computer vision applications that allow to monitor the traffic, recognize actions and count people. Background subtraction is a common approach to this problem. There are three main steps in a background subtraction method: Background initialization which aims to model the background using a certain number of video frames and can be represented in different ways. The background model is used as a reference to be compared with the current video frames. The next step is feature extraction which involves selecting the appropriate features representing the relevant information to compare the reference frame with the following frames. Once the features are extracted over pixels or block of pixels of the background and the current frames, a similarity measure is calculated. Each pixel is classified as belonging to the 'background' or the 'foreground' based on this similarity threshold value. All these steps make it possible to build an entire segmentation system. Recently, due to the availability of big labeled data, it is important to maintain the performance of videos to only retrieve relevant information. Unwanted information embedded in video sequences comes at a high cost in terms of the big amount of data stored. They also contain several inter-dependent and time-varying components. Thus, it is important to select only the pertinent information, such as cars or people, to exploit those resources with a better performance. Additionally, it is important to understand the normal schemes of systems and to automatically identify abnormal behaviours in videos in order to intervene as soon as possible to ensure system stability. Recently, the video surveillance domain has received a lot of attention, but it still covers several issues, like the occlusion of objects in videos, the noise resulting from light variations, the background and the current frames are usually with different illumination, resulting in misclassification. In outdoor environments, these issues increase significantly. Videos are generally of poor quality due to the large distance between the objects and the camera, resulting in high sensitivity to variations in illumination. Therefore, the background is usually dynamic. As a result, some background parts of the current frame do not overlap with the corresponding sections of the reference frame, resulting in no pixel-by-pixel correspondence between the input and the background images. Additionally, if the background image and the foreground object are with the same color, the detected object is misclassified. Therefore, it is important to deal with these issues by developing high performance algorithms to implement a powerful video surveillance system. To overcome the previous limitations, in this thesis, we exploit the power of an unsupervised anomaly discovery framework called DeepSphere proposed by Teng et al. [443] and adapt it to perform moving objects detection task.

The rest of this chapter is organized as follows. The DeepSphere architecture is presented in Section 3.2. The new descriptor that we propose is described in Section 3.3. Comparative results obtained on both synthetic and real videos are given in Section 3.4. Finally, the conclusion drawn at the last section closed the Chapter 3.

3.2 DeepSphere architecture

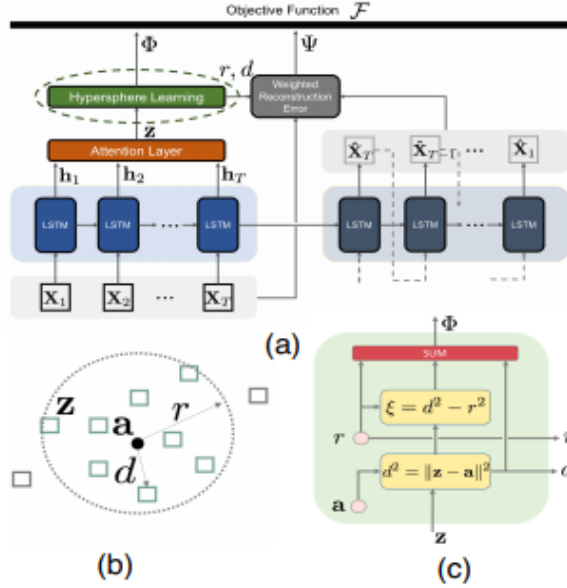


Figure 3.1: DeepSphere architecture.
(<https://www.ijcai.org>)

In recent years, deep learning based anomaly detection techniques are more widely adopted and have been applied to different types of tasks. DeepSphere proposed by Teng et al. [443] is an unsupervised, unified and end-to-end algorithm that can detect anomalies in dynamic networked systems. It can perform two goals: (i) the detection of anomalies at the case level, i.e. to determine if the network is abnormal, (ii) the discovery of anomalies at the nested level, i.e. the exploration of the abnormal structure of localized cases in spatial and temporal context, when anomalies take place and how they deviate from the normal situation. DeepSphere does not need any labeled data or clean data (outlier-free) as input, it is still able to reconstruct normal behaviors. In this thesis, we propose to adapt and validate DeepSphere to perform foreground objects segmentation in video surveillance applications.

DeepSphere [443] aims to both identify anomalous cases and explore the abnormal structure in dynamic networks located in spatial and temporal context. DeepSphere exploits deep autoencoders and hypersphere learning to exclude pollution from anomalies and reconstruct normal behaviors. It allows to capture the spatio temporal dependencies among components and across time steps, to flexibly learn non-linear entity representation, and reconstruct normal behaviors from anomalous incoming data. The high-quality representations learned by auto-encoder allow hypersphere to better differentiate abnormal cases.

A deep autoencoder is a neural network that is composed of two components: the encoder α_ϕ and the decoder β_ϕ , which are highly nonlinear mapping functions developed via neural

networks with parameters θ and ϕ , respectively. The encoder which maps an input image into a compact representation stored in the low dimensional internal layer $z = \alpha_\theta(X)$, while the decoder maps from the internal layer into the output layer to reconstruct the original data $\hat{X}_k = \beta_\phi(z)$. The hypersphere can be characterized by two elements, a centroid a and a radius r and the group of data points is represented as $\{z_k, k = 1, \dots, m\}$.

Figure 3.1(a) shows the whole DeepSphere architecture. In DeepSphere, a sample case χ is divided into a sequence of matrices $\{X_t, t = 1, \dots, T\}$ corresponding to a series of graphs [443]. They are transmitted into an LSTM encoder [54] and a sequence of internal states $\{h_t, t = 1, \dots, T\}$ can be produced. LSTM autoencoder is used for better capturing the structural relationships and the potential temporal dependencies in dynamic graphs. The h_t allows capturing the source sequence information X_t , comprising long and short term dependencies. The attention mechanism is employed to assign several attention to different h_t , i.e., $z = P_t w_t h_t$, where z represents the embedded representation, and w_t is the attention weight at timestep t . In the hidden space, an hypersphere learning layer is considered which learns a spherically shaped boundary around the encodings z_k to separate anomaly pollution (Figure3.1(b)). The hypersphere learning layer internal structure is shown in Figure 3.1(c). The input of the hypersphere learning layer is z_k , the two parameters r and a are considered as nodes, the distance d and the outlier penalty ξ are calculated by functions which are considered as two non-linear neurons. To reduce the risk of accepting abnormal cases, the objective function is defined as:

$$\Phi = r^2 + \gamma \sum_{k=1}^m \xi_k + \frac{1}{m} \sum_{k=1}^m \|z_k - a\|^2 \quad (3.1)$$

All normal tensors must be mapped to the centroid a of the hypersphere. The 3rd element is added to minimize the average distance between z_k and a . Finally, the hypersphere learning layer generates ϕ , d and r . The latent representations z outside the hypersphere over long distances are processed as anomalous, while those located inside the hypersphere at short distances tend to be normal. The reconstruction error for the LSTM autoencoder is adapted as follows :

$$\Psi = \sum_{k=1}^m \eta_k \|\chi_k - \hat{\chi}_k\|^2 \quad (3.2)$$

where χ_k is reconstructed via the LSTM decoder, and η_k represents the case-wise weights calculated using a heuristic function $\eta \{d_k, r\}$. It is recommended to create latent representations z located close to a , while penalizing anomalous cases outside the hypersphere.

The overall objective function is the combination of the hypersphere component ϕ and the penalized reconstruction difference Ψ :

$$\min_{\Theta} F = \min_{\Theta} \{\Phi + \lambda \Psi\} \quad (3.3)$$

where λ is the compromise parameter between these two elements, and $\Theta = \{a, r, w, \theta, \phi\}$ is the set of parameters containing the centroid of the hypersphere a , the radius r , the attention parameter w , and the neural network parameters θ , Φ for the LSTM encoder and decoder. Adam Optimizer [129] is selected to train the DeepSphere model. Since DeepDC has been trained and has a new unseen sample χ_k ($k > m$), the detection of anomalies at the case level

can be done according to its distance from a and making a decision accordingly. Additionally, DeepDC is able to reconstruct its normal behavior $\hat{\chi}_k$ even if χ_k is an anomalous entry. By calculating the reconstruction difference $\Delta(\chi_k) = \chi_k - \hat{\chi}_k$, we can find nested anomalies located in temporal and spatial dimensions.

Our proposed Deep Detector Classifier (DeepDC) uses DeepSphere framework to detect and then extract moving objects from videos. Once DeepSphere algorithm is applied, foreground activities are detected and moving objects are segmented according to a global image threshold.

3.3 Proposed DeepDC descriptor

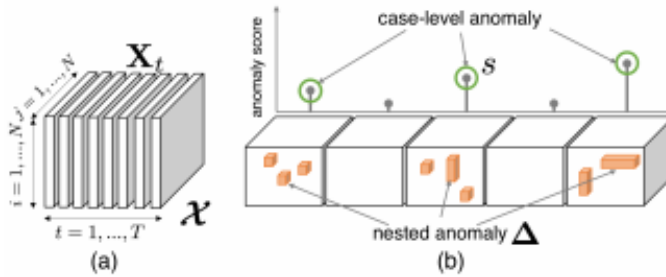


Figure 3.2: Illustration of the two-level anomaly discovery task. (<https://www.ijcai.org>)

The standard DeepSphere algorithm proposed by Teng et al. [443] has proven to be a powerful and robust anomaly discovery framework that simultaneously would satisfy these two conditions, identifying abnormal cases and further exploring the anomalous structure of cases localized in spatial and temporal context. DeepSphere exploits the strengths of hypersphere learning and deep auto-encoders, to exclude anomaly pollution and reconstruct normal behaviors. DeepSphere is not based on manually labeled data and can generalize to unseen data. First, the goal is of two-level, the model can satisfy both transparency and warning requirements. Second, the model must be inductive, it can be generalized to test data.

Figure 3.2 shows the main concept: For a dynamic graph, considering a group of observation samples, each characterized as a tensor representing the inner spatio-temporal structure, Figure 3.2 (a), the model allows inductively identifying the anomalous sample cases and discovering the nested anomalies located in the anomalous tensors, Figure 3.2 (b).

A dynamic graph is described as $G(t) = \{V, E, x(t)\}$. where V indicates the vertex set, E represents the edge set and $x(t)$ denotes the function mapping every edge e_{ij} with a time series $\{x_{ij}(t), t = 1, \dots, T\}$. Figure 3.2 shows that one observation case of $G(t)$ can be represented by a third-order tensor $\chi \in^{N \times N \times T}$, and the slices along the time dimension are the adjacency matrices of the graph at several time steps, designed as $\{X_t, t = 1, \dots, T\}$. A set of cases can be characterized as $\{\chi_k, k = 1, 2, \dots\}$. The dynamic graph contains a group of observation sam-

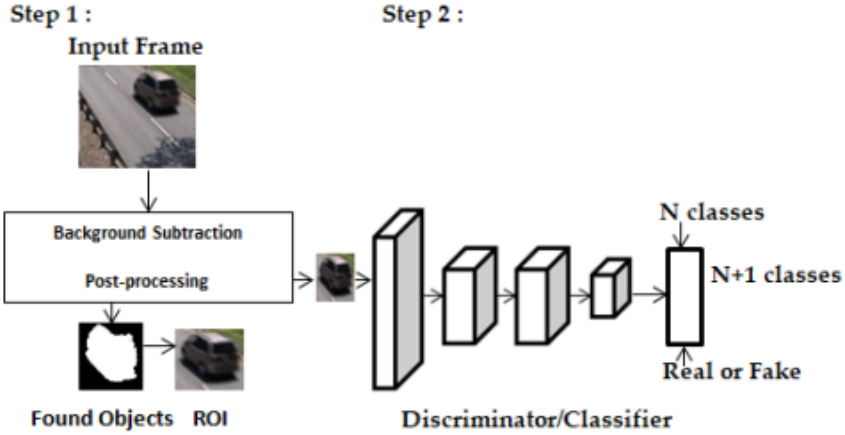


Figure 3.3: The proposed architecture (Deep Detector Classifier). Step 1: background subtraction. Step 2: Object classification (chapter 4).

ples (i.e., the training data) $\{\chi_k, k = 1, \dots, m\}$. A model is trained based on the training data, and then inductively the trained model is applied to unseen data (i.e., test data) $\{\chi_k, k > m\}$. The issue is of two levels (Figure 3.2 (b)): Case level anomaly detection which aims to identify the anomalous observation cases (i.e., tensors) in the test data, defined as $\{\chi_u, u > k\} \subset \{\chi_k, k > m\}$. This task deals with the warning requirement by computing an anomaly score $s(\chi_k)$ for each case χ_k transmitting a signal predicting if the system is normal or not and nested anomaly discovery to discover the abnormal cells nested within the abnormal tensors in the test data and evaluating the deviation from the expected normal behaviours. This task aims to provide certain transparency, a difference $\Delta(\chi_k)$ would be calculated, demonstrating how the abnormal tensor deviate from the expected normal tensor.

In this chapter, we propose to adapt DeepSphere to the foreground-background separation task. In our framework, the data (video) are transformed into a tensor. First, a model is trained based on the training data (normal situation) and then the trained model is inductively applied to the test data (anomalous situation). DeepDC aims to identify which tensors (observation cases) in the test data are anomalous, in our case, referred to the moving people or vehicles that appear along with the video and then to discover the anomalous cells nested within the anomalous tensors in test data. As DeepSphere has been trained, given a new test frame, we can perform case-level anomaly detection by examining its distance towards the center of the hypersphere and make decision accordingly. By computing the reconstruction difference between the original input frame and the reconstructed image, we can detect and then segment foreground objects. A global image threshold is applied using Otsu's method [337] to perform automatic thresholding. By combining the strengths of deep autoencoders and hypersphere learning, our approach based on DeepSphere appears to be robust to illumination changes, dynamic background, and produces a good segmentation results. Without additional image processing steps, the foreground activities are well captured by DeepSphere. Deep autoencoders have proven a strong ability to learn nonlinear

representations, which allows capturing the patterns in input data [264]; But, unlabelled input samples are not strictly free from anomalies, that means, they could be polluted by certain anomalous samples, called ‘‘anomaly pollution’’. The learning process can be affected by anomaly pollution, which can significantly decrease the quality of neural network. To solve this issue, hypersphere learning is proposed which learns a compact limit to separate normal and abnormal samples to exclude anomaly pollution.

Our proposed approach is achieved by incorporating autoencoders with hypersphere learning in a mutually supportive way. DeepSphere does not only inherit the ability of hypersphere learning to separate anomalies, which improves the quality of autoencoders; but also it presents the benefits of autoencoders to be able to capture spatio-temporal dependencies between components and through timesteps, for flexible learning of nonlinear feature representation, and to rebuild normal behaviors from possibly anomalous input data. Outliers can be detected and excluded by learning a compact hypersphere. The hypersphere can be characterized by its centroid a , its radius r , and the group of data points represented as $\{z_k, k = 1, \dots, m\}$ as shown in Figure 3.1 (b). The error function must be minimized:

$$\Phi(a, r) = r^2 + \gamma \sum_k \xi_k \quad (3.4)$$

with the constraints,

$$\|z_k - a\|^2 \leq r^2 + \xi_k, \xi_k \geq 0, \forall i, \quad (3.5)$$

where ξ_k are slack variables allowing the probability of anomalies in the samples. The distance from z_k to a is not necessarily less than r^2 but greater distance must be penalized (the samples outside the limit are considered as anomaly pollution). Furthermore, the parameter γ controls the compromise between penalization and sphere volume. The radius r and the centroid a can be obtained by minimizing Eq. 3.4. Our proposed DeepDC model contains an hypersphere learning element which allows separating normal and anomalous representations, excluding and penalizing anomaly pollution included in the input data to detect moving objects in video sequences.

An autoencoder learns high quality non-linear representations, which allows a good distinction of anomalous cases by hypersphere learning. Our approach consists of detecting and then segmenting foreground objects from video sequences using DeepSphere. DeepSphere is an unified and unsupervised learning process that does not need outliers or labeled training data. It aims to detect anomalies in dynamic graphs and to identify anomalous sample cases and nested anomalies in the abnormal tensor. We leverage DeepSphere and adapt it to detect and then segment moving objects in video sequences. Our proposed approach consists on two steps as presented in Figure 3.3: First, moving objects are detected based on DeepSphere without additional image processing steps. Then, foreground objects are segmented by simply thresholding the difference between the original input frame and the reconstructed image. Second, deep features are extracted from the segmented objects to classify them using a semi-supervised classifier, which consists of a DCGAN discriminator network as mentioned in chapter 4.

3.4 Experimental results and discussions

Several experiments were conducted to illustrate both the qualitative and quantitative results of the proposed DeepDC descriptor. We evaluated the performance of DeepDC in four widely used datasets including restaurant video dataset [91], CDnet2014 dataset [460], VIRAT video dataset [333] and BMC2012 dataset [446] which includes both real and synthetic videos of outdoor environments acquired with a fixed camera, under different weather conditions like wind, real or sun [446].

3.4.1 Description of the datasets

We give a brief introduction of these datasets as follows:

- **Restaurant video dataset:** The restaurant video dataset [91] is a set of frames taken in a restaurant. This dataset includes video background representation and activity detection consisting of snapshots of restaurant activities.
- **VIRAT video dataset:** The VIRAT video dataset is proposed by Oh et al. [333], which presents a greater variety of events and contains events involving interactions between several individuals, vehicles, and facilities. The VIRAT video dataset [333] includes two large categories of activities (one-object and two-objects) that implicate both vehicles and humans. There are three types of interactions that are presented:
 1. person events: standing, walking, throwing, running, carrying, gesturing, loitering and picking up.
 2. Events concerning people and vehicles: getting in or out of the vehicle, opening or closing the trunk, bicycling, dropping off, loading, unloading.
 3. Person and facility events: entering or leaving the facility.

The VIRAT video dataset [333] contains a rich set of actions between multiple objects and includes several types of person-vehicle interactions, labeled in detail with numerous examples per category.

- **Change detection (CDnet2014) dataset:** To test the proposal, we chose also the CDnet2014 [460] dataset considered the largest dataset for foreground segmentation and background modeling. This dataset is made up of 53 videos divided into eleven categories. Each category represents a different challenge for the segmentation algorithms, such as dynamic backgrounds, illumination changes, shadows, camera instability, night scenes, camouflage, etc. This dataset contains 10 videos which mainly contain pedestrians.
- **Background Models Challenge (BMC2012) dataset:** BMC 2012 dataset [446] is created for the Background Models Challenge of the ACCV 2012 conference. It is composed of 29 outdoor videos, some of which are synthetic. Ten synthetic videos are available representing two scenes: a roundabout and a street and their associated ground truth. These videos show different challenging situations, mainly related to the different lighting conditions. Despite only a small subset of images have been labeled, ground truth of real images is available.

3.4.2 Quantitative and qualitative evaluation

- **Restaurant video dataset:**

Our proposed approach is evaluated on several datasets to assess the performance of DeepDC. The restaurant video dataset [91] is a collection of images recorded in a restaurant. In chalapathy et al. [91], background modeling, as well as activity detection are assessed using the restaurant video dataset [91]. Background represents the relatively stationary scenes, while foreground activity incorporates snapshots of restaurant activities, can be guests who come, talk at the reception and exit. DeepSphere algorithm detects foreground activities without the need of additional image processing, unlike the standard foreground-background separation task.

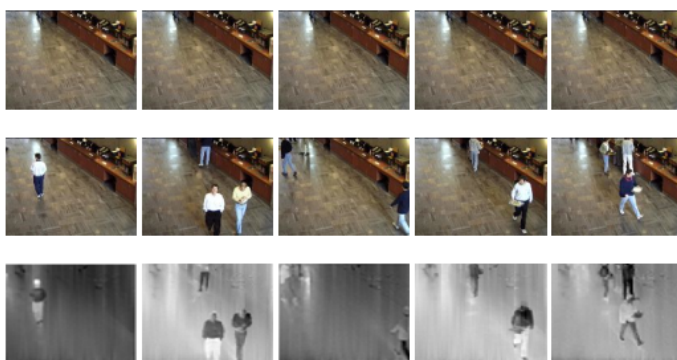


Figure 3.4: Examples of foreground activity detection in restaurant video dataset [91] :The top, middle and bottom rows represent normal situation, anomalous situation and detected results, respectively.

Figure 3.4 shows the results of activity detection in the restaurant dataset [91]. Two cases of activities are planned for people coming and leaving. The top, middle and bottom rows represent the normal situation, the anomalous situation and the results detected separately. The foreground activities are well captured by DeepSphere without additional image processing. The results suggest that DeepSphere has an extended application with tasks similar to discover anomalies in video surveillance applications.

- **VIRAT video dataset:**

We also present the results of background subtraction using the proposed DeepDC to well capture moving objects in VIRAT video sequences [333]. Once the foreground objects are detected, a good segmentation results are obtained. Figure 3.5 shows that foreground activities are well captured in VIRAT video dataset [333] by using DeepSphere technique without additional image processing steps. This results in a good segmentation of foreground objects used as input in GAN classification in Chapter 4.

Additionally, we compare our proposed DeepDC with the 29 algorithms implemented in the Background Subtraction Library, BGSLibrary [418]. BGSLibrary [418] provides a simple C++ framework for performing background subtraction. The library

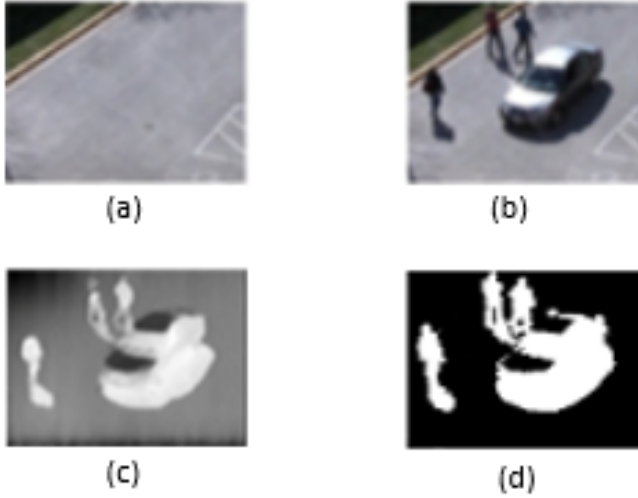


Figure 3.5: Network input and output in VIRAT video dataset [333] : (a) Background frame, (b) test frame, (c) output of DeepSphere, (d) segmentation mask of the proposed method.

includes 29 background subtraction algorithms. The OpenCV2 library must be installed for using the BGSLibrary [418]. We evaluate the proposed detector on five real outdoor video sequences from VIRAT dataset [333].

This process is carried out using the metrics Recall, Precision and F-measure. These metrics are based on the numbers of true positive TP pixels (correctly detected foreground pixels), false positive FP pixels (background pixels detected as foreground ones), false negative pixels FN (foreground pixels detected as background ones), and true negative pixels (correctly detected background pixels). Recall represents the percentage of foreground pixels detected correctly in relation to the total pixels in the foreground of the groundtruth. Precision represents the percentage of foreground pixels detected correctly in relation with to the total number of pixels detected as foreground and F-measure represents a balance between the metrics Recall and Precision.

$$\begin{aligned}
 - \text{Recall} &= \frac{TP}{TP + FN} \\
 - \text{Precision} &= \frac{TP}{TP + FP} \quad \text{and} \\
 - \text{F-measure} &= 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
 \end{aligned}$$

The F-measures obtained using the proposed DeepDC based on DeepSphere, compared to the large range of BGSLibrary [418] are given in Table 3.1. The average F-measure results across the algorithms show that our DeepDC based on DeepSphere outperforms the 29 algorithms of BGSLibrary [418].

Table 3.1: FM of BS algorithms evaluated on five real videos of the VIRAT dataset [333]. The six best methods are underlined. The best methods in each category are in italic.

<i>ID Method</i>	<i>Recall</i>	<i>Precision</i>	<i>F-Measure</i>
<i>Basic methods, mean and variance over time</i>			
StaticFrameDifferenceBGS	0.6903	0.8661	0.7682
FrameDifferenceBGS	0.6028	0.7956	0.6860
WeightedMovingMeanBGS	0.6829	0.8060	0.7393
WeightedMovingVarianceBGS	0.7289	0.8237	0.7734
<i>AdaptiveBackgroundLearning</i>	<i>0.7632</i>	<i>0.8963</i>	<i>0.8244</i>
DPMeanBGS	0.7056	0.8718	0.7800
DPAdaptiveMedianBGS [315]	0.6734	0.8870	0.7656
DPPratiMediodBGS [81]	0.6901	0.8370	0.7904
<i>Fuzzy-based methods</i>			
<i>FuzzySugenIntegral</i> [497]	<i>0.8067</i>	<i>0.9229</i>	<i>0.8509</i>
FuzzyChoquetIntegral [45]	0.7899	0.8901	0.8370
LBFuzzyGaussian	0.7312	0.9105	0.8110
<i>Statistical methods using one Gaussian</i>			
<i>DPWrenGABGS</i> [470]	<i>0.8347</i>	<i>0.9249</i>	<i>0.8775</i>
LBSimpleGaussian [57]	0.6321	0.9107	0.7462
<i>statistical methods using multiple Gaussians</i>			
DPGrimsonGMMBGS [427]	0.7226	0.8873	0.7965
<i>MixtureOfGaussianV1BGS</i> [228]	<i>0.8188</i>	<i>0.9285</i>	<i>0.8702</i>
MixtureOfGaussianV2BGS [516]	0.8110	0.8500	0.8301
DPZivkovicOfGaussians [516]	0.7937	0.9120	0.8475
LBMixtureOfGaussians [66]	0.8750	0.8340	0.8480
<i>Type-2-fuzzy-based methods</i>			
T2FGMM_UM [67]	0.6611	0.9350	0.7745
<i>T2FGMM_UV</i> [67]	<i>0.8362</i>	<i>0.9751</i>	<i>0.8548</i>
T2FMRF_UM [507]	0.6100	0.8821	0.7212
T2FMRF_UV [507]	0.8692	0.6788	0.7623
<i>Statistical methods using colour and texture features</i>			
MultiLayerBGS [487]	0.7159	0.8657	0.7837
<i>Non-parametric methods</i>			
<i>PixelBasedAdaptiveSegmenter</i> [196]	<i>0.852</i>	<i>0.923</i>	<i>0.885</i>
GMG [167]	0.9470	0.7030	0.8031
VuMeter [172]	0.7195	0.9055	0.8019
<i>Methods based on eigenvalues and eigenvectors</i>			
<i>DPEigenbackgroundBGS</i> [335]	<i>0.8790</i>	<i>0.6584</i>	<i>0.7475</i>
<i>Neural and neuro-fuzzy methods</i>			
LBAdaptiveSOM [347]	0.8056	0.9017	0.8509
<i>LBFuzzyAdaptiveSOM</i> [295]	<i>0.8064</i>	<i>0.9273</i>	<i>0.8626</i>
Proposed approach: DeepSphere	0.8880	0.9742	0.9291



Figure 3.6: Extracted images from our proposed background subtraction approach: (a) person, (b) car, (c) car but not the whole feature :etc, (d) Individual but not the whole body : etc

Table 3.2: Number of extracted images for each class from five cameras of VIRAT_Video dataset [333].

Class	no. of images
person	856
vehicle	870
etc	549

Finally, the desired object areas are extracted (vehicles, pedestrians, etc), which are represented by rectangles. If the size of the detected object is less than 5×5 pixels, it is considered as noise and is removed. Figure 3.6 illustrates some examples of the extracted objects. Figure 3.6a corresponds to the first class objects which represents 'person'. Figure 3.6b represents the second class, 'vehicle'. Figure 3.6c and Figure 3.6d show a partially detected vehicle and people. In many cases, although the whole body is visible, it is partially detected due to the same color as the background frame or static regions of the body. In some cases, our method detects only certain regions of the object (vehicle/person) when it is partially masked or in case of movement in the blind spots of the camera. These objects are unwanted and are removed using a semi-supervised DCGAN classification as presented in Chapter 4.

Table 3.2 illustrates the number of images taken from five different cameras from the VIRAT video dataset [333], which are used for training and testing. Three categories of objects are defined: 'person', 'car' and 'etc'. The class 'person' represents the full body. The class 'car' represents the entire cars and the 'etc' class indicates images that are difficult to classify. This class contains only partially detected objects such as cars or people when the background includes the image. The most extracted regions of interest (ROIs) are people or cars and the other images captured by background subtraction represent objects that are incorrectly selected. We have used five cameras from VIRAT dataset [333] to build the training and testing datasets. Once the background subtraction based on DeepSphere is achieved, we have obtained 2275 images of objects extracted from VIRAT video sequences (Table 3.2). In Chapter 4, we aim to categorize the extracted objects into three classes, 'person', 'car' and 'etc' based on a semi-supervised classifier, which consists of the DCGAN discriminator.

- **Change detection (CDnet2014) dataset:**

Figure 3.7 shows the foreground detection results using our proposed DeepDC on individual frames from two video sequences of CDnet2014 dataset [460]: CameraJitter/Traffic (frame #1247), CameraJitter/Traffic (frame #1546) and Baseline/Pedestrians (frame #566).

Traffic (frame #1247) Traffic (frame #1546) Pedestrians (frame #566)

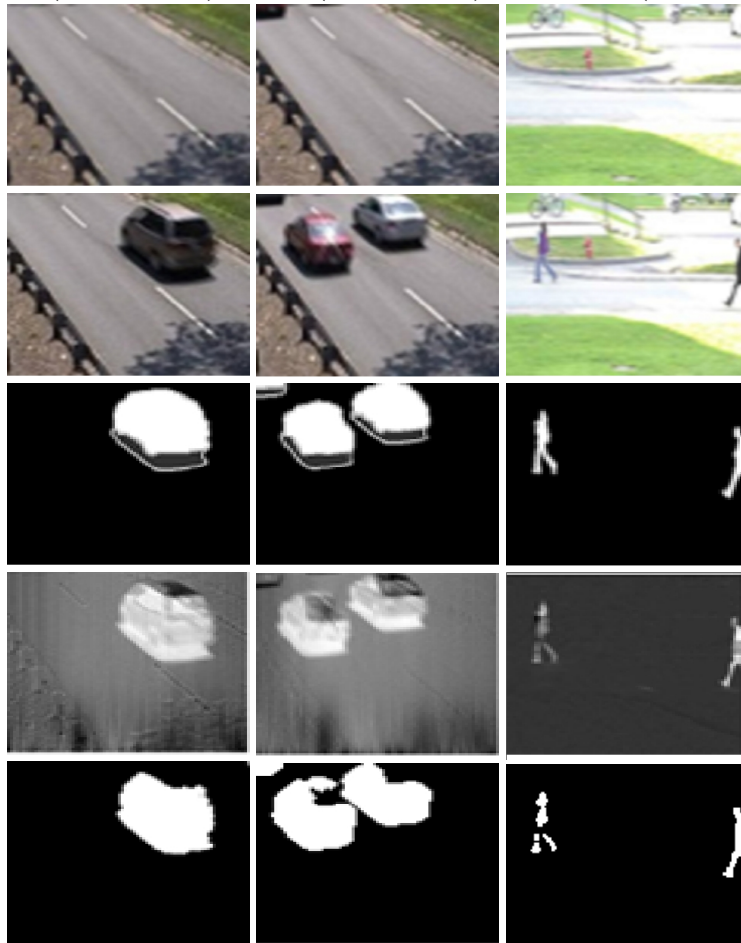


Figure 3.7: Network input and output in CDnet2014 dataset [460] : The first row is the background frame, the second row is the image test, the third row is the ground truth, the fourth row is the output of DeepSphere. The fifth row is the foreground mask of the proposed method.

Our proposed DeepDC based on DeepSphere clearly appears to be robust and less sensitive to the background subtraction method and shows greater performance in CDnet2014 [460] scenes.

Table 3.3 shows the average F-measure values of the different BGSLibrary algorithms [418] and the proposed DeepDC detector based on DeepSphere on 53 videos of CDnet2014 dataset [460]. Best F-measures are underlined. The proposed DeepDC algorithm gives the highest value compared to the large range of BGSLibrary algorithms [418].

We have extracted the six best methods, according to the results obtained using video sequences provided by VIRAT [333] and CDnet2014 dataset [460], that clearly overcome the other ones. These methods cover a long period of time in the literature, the GMM improvement proposed by Kaewtrakulpong and Bowden [228], LBFuzzy AdaptiveSOM [295] and T2FGMM.UV [67] are very good BS methods. PFinder (DPWrenGABGS) [470] and PBAS [196] have showed an interesting robustness since it has been possible to find a good compromise between the increase of true positive (TP) pixels and the increase of false positive (FP). DeepDC outperforms these algorithms implemented in BGSLibrary [418] with 92.91% and 96.39%, in VIRAT [333] and CDnet2014 datasets [460], respectively.

The top six ranking algorithms can be confirmed through the visual analysis as presented in Figure 3.8. Without using additional image processing, we can observe that our approach based on DeepSphere shows consistently better performance in different scenarios.

Figure 3.8 illustrates sample results of applying DeepDC and the best five BGSLibrary algorithms on videos from CDnet2014 [460] and VIRAT [333] datasets. In this Figure, the test frames are displayed in the first row, the ground truth are shown in the second row, and the results obtained with the proposed method are displayed in the third row. The results obtained with the other methods are shown in the fourth to eighth rows of Figure 3.8. As observed, DeepDC is enough successful in detecting foreground objects in these scenes and outputs acceptable foreground masks. DeepDC clearly appears less sensitive to the background subtraction challenges, whereas the five others fall in detecting moving objects, unless applying a strong post-processing step.

One of the most famous statistical-based approaches to model the reference frame is the Gaussian Mixture Model (GMM), which is originally proposed by Stauffer and Grimson [427]. On the basis of GMM, Kim et al. [238] proposed a background subtraction algorithm to extract moving object areas from each video frame based on Gaussian Mixture Model (GMM). We demonstrate that DeepDC achieves better results compared to the work of Kim et al. [238].

Table 3.3: FM of BS algorithms evaluated on 53 videos of the CDnet2014 dataset [460]. The six best methods are underlined. The best methods in each category are in italic.

<i>ID Method</i>	<i>Recall</i>	<i>Precision</i>	<i>F-Measure</i>
<i>Basic methods, mean and variance over time</i>			
StaticFrameDifferenceBGS	0.8452	0.6007	0.7023
FrameDifferenceBGS	0.6736	0.9252	0.7796
WeightedMovingMeanBGS	0.2744	0.7346	0.3996
<i>WeightedMovingVarianceBGS</i>	<i>0.8103</i>	<i>0.8641</i>	<i>0.8363</i>
AdaptiveBackgroundLearning	0.8742	0.7548	0.8102
DPMeanBGS	0.7605	0.5930	0.6664
DPAdaptiveMedianBGS [315]	0.6611	0.9350	0.7745
DPPratiMediodBGS [81]	0.6128	0.9269	0.7378
<i>Fuzzy-based methods</i>			
FuzzySugenoIntegral [497]	0.5488	0.8817	0.6766
<i>FuzzyChoquetIntegral [45]</i>	<i>0.7621</i>	<i>0.9160</i>	<i>0.8370</i>
LBFuzzyGaussian	0.7372	0.5778	0.6479
<i>Statistical methods using one Gaussian</i>			
<i>DPWrenGABGS [470]</i>	<i>0.7946</i>	<i>0.9445</i>	<i>0.8631</i>
LBSimpleGaussian [57]	0.8354	0.6038	0.7010
<i>statistical methods using multiple Gaussians</i>			
DPGrimsonGMMBGS [427]	0.8793	0.7647	0.8180
<i>MixtureOfGaussianV1BGS [228]</i>	<i>0.8779</i>	<i>0.9572</i>	<i>0.8546</i>
MixtureOfGaussianV2BGS [516]	0.6295	0.9368	0.7530
DPZivkovicOfGaussians [516]	0.6547	0.9385	0.7714
LBMixtureOfGaussians [66]	0.8750	0.8076	0.8402
<i>Type-2-fuzzy-based methods</i>			
T2FGMM_UM [67]	0.5306	0.6021	0.5641
<i>T2FGMM_UV [67]</i>	<i>0.7729</i>	<i>0.9463</i>	<i>0.8509</i>
T2FMRF_UM [507]	0.6100	0.8821	0.7212
T2FMRF_UV [507]	0.6787	0.7723	0.7225
<i>Statistical methods using colour and texture features</i>			
MultiLayerBGS [487]	0.7552	0.7878	0.7712
<i>Non-parametric methods</i>			
<i>PixelBasedAdaptiveSegmenter [196]</i>	<i>0.8429</i>	<i>0.9193</i>	<i>0.8794</i>
GMG [167]	0.8793	0.7647	0.8180
VuMeter [172]	0.8258	0.8621	0.8436
<i>Methods based on eigenvalues and eigenvectors</i>			
<i>DPEigenbackgroundBGS [335]</i>	<i>0.8880</i>	<i>0.7691</i>	<i>0.8244</i>
<i>Neural and neuro-fuzzy methods</i>			
LBAdaptiveSOM [347]	0.6460	0.8272	0.7254
<i>LBFuzzyAdaptiveSOM [295]</i>	<i>0.8304</i>	<i>0.8804</i>	<i>0.8568</i>
Proposed approach: DeepSphere	0.9440	0.9847	0.9639

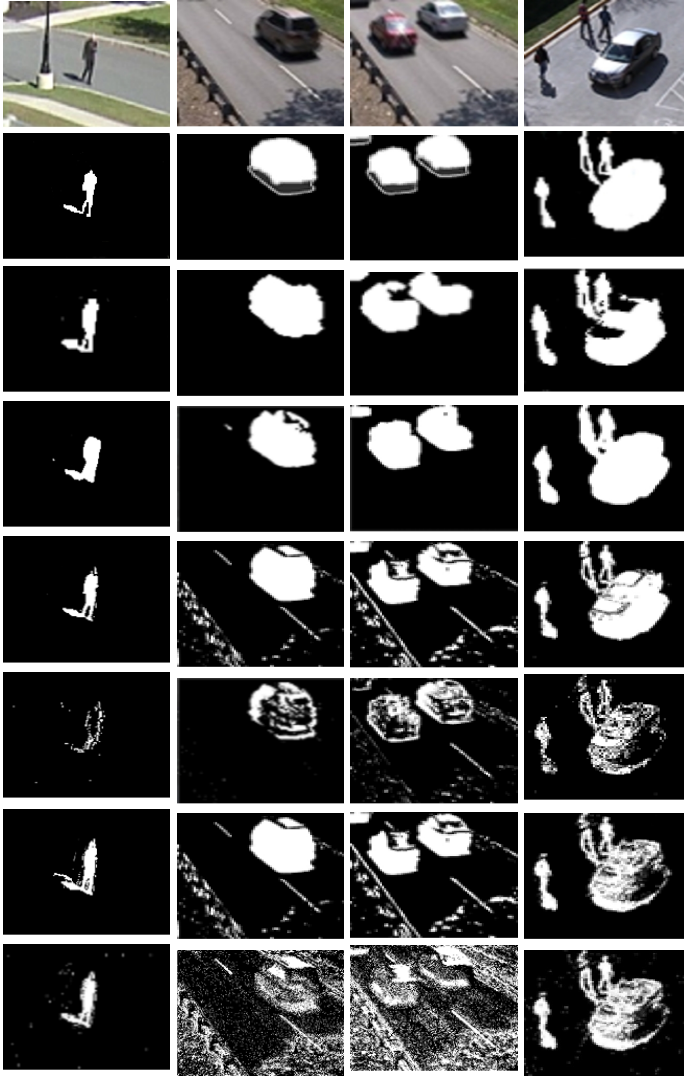


Figure 3.8: Background subtraction obtained with the proposed scheme and the best five BS algorithms using the VIRAT scenes and CDnet2014 dataset. From the first to last row: input frame, region of interest, DeepSphere (ours), PBAS [196], DPWrenGABGS [470], MixtureOfGaussianV1BGS [228], LBFuzzyAdaptiveSOM [295] and T2FGMM_UV [67].

We compare DeepDC with the best five algorithms of the BGSLibrary [418]. Moreover, two unsupervised foreground detection methods, which both estimate a deterministic low dimensional representation of the background in videos, the Robust PCA (RPCA) model [84] and the Deep Probabilistic Background Model (DeepPBM) [149], and two supervised foreground segmentation methods based on Deep Learning, Deep-

CNN [42] and DPDL [502], were chosen to compare the segmentation results of our proposal. The average quantitative results for each category are reported in Tables 3.4 and 3.5. It can be seen that in most of the challenges, the highest F-measure values across the eleven categories of CDnet2014 dataset (highlighted in bold) are obtained using our proposal. As representative example, that demonstrate the robustness of this research work, we have the "Dynamic-B" and "B-Weather" categories where despite the highly dynamic background regions due to partial obstruction of objects of interest and snow fall, we obtained the best segmentation results. Another representative example corresponds to the "thermal/ Corridor" scenario, where despite the camouflage caused by the nature of thermal images combined with the morphology of the human body where the limbs represent small objects with fine details, a large proportion of the human silhouette is successfully segmented. In both challenges, there is a noticeable difference between the average F-measure values obtained compared to the other algorithms, even against Deep Learning based methods such as Deep CNN [42], DPDL [502] and DeepPBM [149]. Moreover, in categories with less dynamic backgrounds or less camouflage problem (for example turbulence, IO-Motion), the performance of our proposal continues to be superior in the segmentation of small-sized foreground objects. However, the combination of several challenges (severe camouflage, highly dynamic background, the reduced dimension of the object of interest, shadow, jitter, etc), can compromise the performance of the proposed method. For example, in the "shadow / Cubicle" scenario, in which the shadows of the foreground objects is considered as foreground, the proposed algorithm DeepDC was superseded by the Deep CNN [42] method. A similar situation occurred in the "Night" videos, our proposal was slightly surpassed the DeepPBM algorithm [149] because the camouflage caused by the night condition and the illumination changes due to the very strong headlights, complicated the segmentation of the cars. Furthermore, in the "PTZ" category, DPDL [502] was exceeded DeepSphere in F-measure because of the small-sized moving objects presented in the scenes.

Our proposal outperforms the previous methods, with the highest average F-measure values on almost all categories, except for "PTZ" scenes, for which DPDL has achieved the best value and "Shadow" category, for which Deep CNN [42] has obtained the best segmentation results. Note that both DPWrenGABGS and RPCA give lower F-measure than MoG, T2FGMM.UV, PBAS, LBFuzzyAdaptiveSOM, DeepPBM [149], Deep CNN and DPDL for some videos. To quantify the results, the output binary masks generated by the background subtraction methods are compared with the ground truth images taken from CDnet2014 [460] dataset.

Table 3.6 shows the qualitative segmentation results across the eleven CDnet2014 [460] categories. Graphical results demonstrate that our proposal is more robust than other methods to the background subtraction challenges. DeepDC is tolerant to lighting changes as RPCA [84] is whereas DeepPBM [149] is not, and robust to noise and the dynamic nature of the background as DeepPBM [149] is whereas RPCA [84] is not. Figure 3.9 demonstrates that our approach outperforms all other methods in almost all the CDnet 2014 [460] categories.

Table 3.4: Performance values of the proposed method compared to the other methods on eleven categories from CDnet2014 Dataset [460] (Part 1).

Videos	Method	Recall	Precision	F-measure
B-Weather	DPWrenGABGS [470]	0.4554	0.5631	0.5022
	MixtureOfGaussianV1BGs [228]	0.7929	0.8402	0.8159
	T2FGMM_UV [67]	0.5877	0.8809	0.7051
	PixelBasedAdaptiveSegmenter [196]	0.5698	0.9351	0.7081
	LBfuzzyAdaptiveSOM [295]	0.3163	0.9123	0.4697
	RPCA [84]	0.2927	0.7334	0.4185
	DeepPBM [149]	0.7953	0.8638	0.8281
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.7517	0.9494	0.8301
	DPDL (CNN) (Temporal-wise) [502]	0.7614	0.9244	0.8350
	DeepSphere (ours)	0.8066	0.9518	0.8726
Baseline	DPWrenGABGS [470]	0.4230	0.8864	0.5727
	MixtureOfGaussianV1BGs [228]	0.4896	0.9382	0.6434
	T2FGMM_UV [67]	0.637	0.838	0.723
	PixelBasedAdaptiveSegmenter [196]	0.6859	0.9250	0.7877
	LBfuzzyAdaptiveSOM [295]	0.2282	0.7318	0.3479
	RPCA [84]	0.4382	0.9073	0.5910
	DeepPBM [149]	0.4842	0.7890	0.6001
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.8063	0.6515	0.7207
	DPDL (CNN) (Temporal-wise) [502]	0.8577	0.8863	0.8718
	DeepSphere (ours)	0.8625	0.9271	0.8918
C-Jitter	DPWrenGABGS [470]	0.3399	0.9096	0.4949
	MixtureOfGaussianV1BGs [228]	0.4407	0.8256	0.5747
	T2FGMM_UV [67]	0.4512	0.8736	0.5951
	PixelBasedAdaptiveSegmenter [196]	0.4624	0.9489	0.6218
	LBfuzzyAdaptiveSOM [295]	0.3806	0.7516	0.5053
	RPCA [84]	0.3839	0.5638	0.4568
	DeepPBM [149]	0.5759	0.6866	0.6264
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.9131	0.8348	0.8722
	DPDL (CNN) (Temporal-wise) [502]	0.8788	0.9319	0.8990
	DeepSphere (ours)	0.8905	0.9867	0.9361
Dynamic-B	DPWrenGABGS [470]	0.2140	0.6914	0.3268
	MixtureOfGaussianV1BGs [228]	0.4189	0.7691	0.5424
	T2FGMM_UV [67]	0.5279	0.8663	0.6560
	PixelBasedAdaptiveSegmenter [196]	0.8718	0.7144	0.7853
	LBfuzzyAdaptiveSOM [295]	0.4090	0.8921	0.5609
	RPCA [84]	0.2532	0.3783	0.3033
	DeepPBM [149]	0.5570	0.8476	0.6722
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.852	0.782	0.8142
	DPDL (CNN) (Temporal-wise) [502]	0.732	0.845	0.7860
	DeepSphere (ours)	0.8543	0.9083	0.8761
IO-Motion	DPWrenGABGS [470]	0.3290	0.8027	0.4667
	MixtureOfGaussianV1BGs [228]	0.3064	0.9851	0.4674
	T2FGMM_UV [67]	0.3473	0.8111	0.4864
	PixelBasedAdaptiveSegmenter [196]	0.2982	0.7043	0.4190
	LBfuzzyAdaptiveSOM [295]	0.5295	0.9839	0.6885
	RPCA [84]	0.3518	0.8415	0.4961
	DeepPBM [149]	0.5135	0.8032	0.6265
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.5735	0.8251	0.6098
	DPDL (CNN) (Temporal-wise) [502]	0.7870	0.9935	0.8783
	DeepSphere (ours)	0.7896	0.9975	0.8815
Low-F	DPWrenGABGS [470]	0.2137	0.8250	0.3394
	MixtureOfGaussianV1BGs [228]	0.1927	0.8099	0.3113
	T2FGMM_UV [67]	0.3060	0.8583	0.4511
	PixelBasedAdaptiveSegmenter [196]	0.3701	0.8797	0.5210
	LBfuzzyAdaptiveSOM [295]	0.2019	0.7539	0.3185
	RPCA [84]	0.2880	0.4788	0.3597
	DeepPBM [149]	0.6108	0.3406	0.4374
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.5924	0.9675	0.6002
	DPDL (CNN) (Temporal-wise) [502]	0.711	0.9200	0.8050
	DeepSphere (ours)	0.7292	0.9677	0.8295
Night videos	DPWrenGABGS [470]	0.6271	0.9336	0.7503
	MixtureOfGaussianV1BGs [228]	0.3235	0.7204	0.4465
	T2FGMM_UV [67]	0.4627	0.9319	0.6184
	PixelBasedAdaptiveSegmenter [196]	0.4015	0.8377	0.5428
	LBfuzzyAdaptiveSOM [295]	0.5254	0.9691	0.6814
	RPCA [84]	0.8457	0.8522	0.8489
	DeepPBM [149]	0.7742	0.9497	0.8530
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.5315	0.8366	0.5835
DPDL (CNN) (Temporal-wise) [502]	0.659	0.792	0.7191	
DeepSphere (ours)	0.7796	0.9711	0.8649	

Table 3.5: Performance values of the proposed method compared to the other methods on eleven categories from CDnet2014 Dataset [460] (Part 2).

Videos	Method	Recall	Precision	F-measure
PTZ	DPWrenGABGS [470]	0.4067	0.9408	0.5679
	MixtureOfGaussianV1BGS [228]	0.3828	0.5305	0.4447
	T2FGMM.LUV [67]	0.4369	0.8564	0.5786
	PixelBasedAdaptiveSegmenter [196]	0.3610	0.9056	0.5162
	LBFuzzyAdaptiveSOM [295]	0.4439	0.7800	0.5658
	RPCA [84]	0.3430	0.7326	0.4673
	DeepPBM [149]	0.751	0.780	0.765
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.4630	0.9610	0.6249
	DPDL (CNN) (Temporal-wise) [502]	0.8905	0.7205	0.7965
	DeepSphere (ours)	0.6076	0.9977	0.7553
Shadow	DPWrenGABGS [470]	0.3317	0.9652	0.4938
	MixtureOfGaussianV1BGS [228]	0.3254	0.8552	0.4714
	T2FGMM.LUV [67]	0.5218	0.9572	0.6754
	PixelBasedAdaptiveSegmenter [196]	0.3846	0.9347	0.5450
	LBFuzzyAdaptiveSOM [295]	0.5120	0.9044	0.6538
	RPCA [84]	0.3178	0.7022	0.4376
	DeepPBM [149]	0.3083	0.9769	0.4687
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.901	0.962	0.9304
	DPDL (CNN) (Temporal-wise) [502]	0.8023	0.9575	0.8730
	DeepSphere (ours)	0.9590	0.8938	0.9252
Thermal	DPWrenGABGS [470]	0.3056	0.9652	0.4642
	MixtureOfGaussianV1BGS [228]	0.3946	0.9143	0.5513
	T2FGMM.LUV [67]	0.608	0.556	0.5808
	PixelBasedAdaptiveSegmenter [196]	0.691	0.566	0.6223
	LBFuzzyAdaptiveSOM [295]	0.5382	0.9489	0.6868
	RPCA [84]	0.767	0.781	0.7747
	DeepPBM [149]	0.3716	0.8810	0.5227
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.6637	0.9257	0.7583
	DPDL (CNN) (Temporal-wise) [502]	0.8172	0.7946	0.8075
	DeepSphere (ours)	0.720	0.966	0.8201
Turbulence	DPWrenGABGS [470]	0.8180	0.8275	0.8226
	MixtureOfGaussianV1BGS [228]	0.4118	0.8864	0.5623
	T2FGMM.LUV [67]	0.4706	0.8099	0.5953
	PixelBasedAdaptiveSegmenter [196]	0.8028	0.9375	0.8650
	LBFuzzyAdaptiveSOM [295]	0.8033	0.9305	0.8618
	RPCA [84]	0.5366	0.7085	0.6107
	DeepPBM [149]	0.7378	0.9293	0.6643
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.8226	0.8430	0.8330
	DPDL (CNN) (Temporal-wise) [502]	0.7979	0.9082	0.8455
	DeepSphere (ours)	0.829	0.951	0.8857
Average	DPWrenGABGS [470]	0.4063	0.8582	0.5274
	MixtureOfGaussianV1BGS [228]	0.4977	0.8371	0.5301
	T2FGMM.LUV [67]	0.4870	0.8399	0.6059
	PixelBasedAdaptiveSegmenter [196]	0.5340	0.8477	0.6303
	LBFuzzyAdaptiveSOM [295]	0.4443	0.8689	0.5764
	RPCA [84]	0.4379	0.6981	0.5240
	DeepPBM [149]	0.6236	0.7988	0.6422
	Deep CNN (DeepBS) (Pixel-wise) [42]	0.6933	0.8958	0.7433
	DPDL (CNN) (Temporal-wise) [502]	0.7975	0.8705	0.8287
	DeepSphere (ours)	0.8016	0.9568	0.8671

Table 3.6: Visual results on CDnet 2014 dataset: From left to right: Original images, Ground-Truth images, RPCA [84], DeepPBM [149], DeepSphere (ours).





















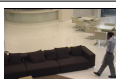


































Categories	Original	Ground Truth	RPCA	DeepPBM	DeepSphere (ours)
B-Weather Skating (Frame #1866)					
Baseline Highway (Frame #1370)					
C-Jitter Traffic (Frame #1538)					
Dynamic-B Fall (Frame #2533)					
I-O-Motion Sofa (Frame #625)					
lowFramerate turnpike (Frame #1075)					
NightVideos tramstation (Frame #1006)					
PTZ TwoPosition (Frame #1024)					
Shadow Cubicle (Frame #5529)					
Thermal Corridor (Frame #5013)					
Turbulence T-3 (Frame #939)					

Table 3.7: Background subtraction results on seven frames from three video sequences of CDnet2014 dataset affected by illumination changes and dynamic backgrounds. Our algorithm successfully segments out the objects (person/vehicle) in all seven input frames.




Categories	Original	Ground Truth	RPCA	DeepPBM	DeepSphere (ours)
Shadow Cubicle (Frame #1186)					
Shadow Cubicle (Frame #5529)					
Shadow Cubicle (Frame #5544)					
Shadow Cubicle (Frame #5566)					
Shadow Cubicle (Frame #7065)					
Dynamic-B Fall (Frame #2533)					
Dynamic-B Overpass (Frame #2467)					

Table 3.7 shows the results of background subtraction for five frames from the CDnet2014 "shadow"/Cubicle video (frame #1186, frame #5529, frame #5544, frame #5566, frame #7065) and two frames from the "dynamic-Background" category (Fall frame #2533, Overpass frame #2467). Our proposal clearly appears to be more tolerant to the background subtraction method, whereas RPCA [84] and DeepPBM [149] are very useless in detecting moving objects. Table 3.7 illustrates the effectiveness of the proposed descriptor in dealing with changing lighting and dynamic backgrounds conditions. Our model achieves a more accurate foreground segmentation. Compared to RPCA [84], DeepSphere can better deal with noise and dynamic backgrounds. Compared to DeepPBM [149], DeepSphere achieves better foreground segmentation because it can cope with illumination changes.

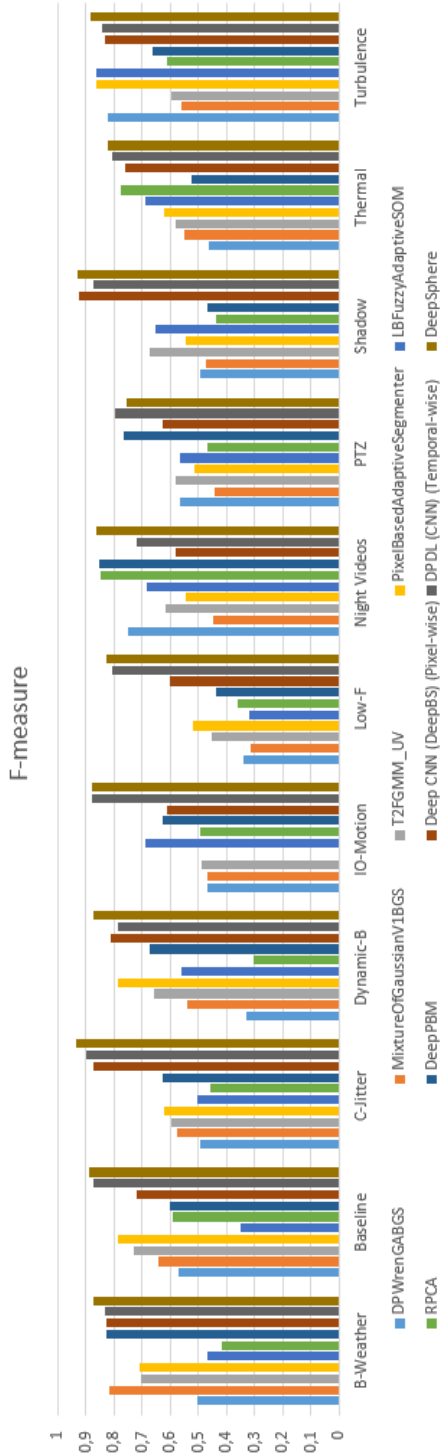


Figure 3.9: F-measures obtained with the proposed scheme and other methods for the CDnet 2014 dataset [460]

Figure 3.10 shows the training and validation loss of DeepSphere using CDnet2014 dataset [460]. It reveals that the train and validation loss decreases to a point of stability with a minimal difference between the two final loss values, proof of a good fit. Model loss is almost lower on the training dataset than the validation dataset.

Figure 3.11(a), Figure 3.11(b) and Figure 3.11(c) display graphics of the F-measures for various methods, from MOG to our proposed model, DeepSphere. In these figures, the closer the method curve is to a circle with radius 1, the more the method is robust over the 11 categories of CDnet 2014 dataset [460]. By looking at Figure 3.11(a), we can first see that the fuzzy gaussian model, namely, Type-2 Fuzzy Gaussian Mixture Model (T2FGMM.UV) [67] slightly outperforms the MOG [427] basic statistical model, implemented in 1999 except on the "Bad Weather" category. However, the average F-measure did not exceed 0.7 %, which is relatively low. Only for the "baseline" and "B-Weather" categories, the F-measure exceeded 0.7 %, which makes these methods usable in applications with not too complex environments. Second, we can see that the proposed DeepDC neural network based on DeepSphere significantly increases the F-measure under "baseline", "camera jitter", "intermittent object motion", and "turbulence" categories.

In Figure 3.11(b), we can see that the unsupervised DeepPBM [149] model which allows a deterministic low dimensional representation of the background in videos achieves higher performance than RPCA [84], known as one of the standard and well-performed subspace learning methods, except in the "Thermal" category. Second, we can see that the proposed DeepSphere algorithm achieves better performance than RPCA [84] and DeepPBM [149] in almost all the categories of CDnet 2014 dataset [460], except in the "PTZ" category, where DeepPBM [149] outperforms DeepSphere.

Figure 3.11(c) compares DeepSphere with two supervised CNNs based methods. The DPDL [502] (Temporal-wise) model provides a better performance than the Deep CNN (DeepBS) (Pixel-wise) model [42] in almost all categories, except in the "Dynamic-B" and "Shadow" categories. This can be explained by the fact that the DPDL [502] is a temporal-wise algorithm which imposes temporal coherence by modeling the dependencies between adjacent temporal pixels, while the Deep CNN [42] is a pixel-wise method that does not take into account temporal or spatial restrictions. In addition, in Figure 3.11(c), we can also see an increase in performance between DeepSphere and DPDL model [502] which was designed in 2018, thereby showing the important improvement made by our proposed method. DeepSphere significantly increase the F-measure under "Baseline", "Camera jitter", "Intermittent object motion" and "Turbulence" categories. The gain in F-measure was approximately 10 %. This good performance of DeepSphere based methods is due to their ability to take into account both spatial and temporal constraints, which are extremely important in this field. DeepSphere enforces both spatial and temporal coherence by modeling the dependencies between adjacent temporal and spatial pixels, resulting in better performance.

Table 3.8 illustrates the number of images extracted from 53 cameras from CDnet 2014 dataset [460], which are used for training and testing in Chapter 4. In total, 7630 images are extracted belonging to three categories: 'person', 'vehicle' and 'etc'.

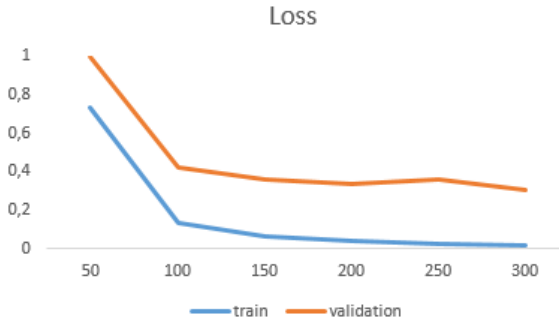
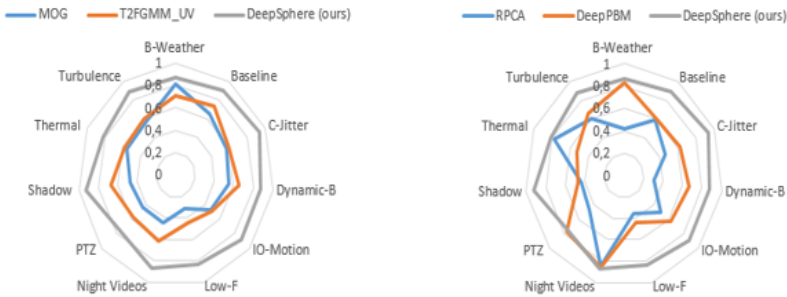


Figure 3.10: Train and validation learning curves of DeepSphere using CDnet2014 dataset [460].

a) GAP MOG-T2FGMM_UV-DeepSphere b) GAP DeepSphere-RPCA-DeepPBM



c) GAP DeepSphere-CNNs

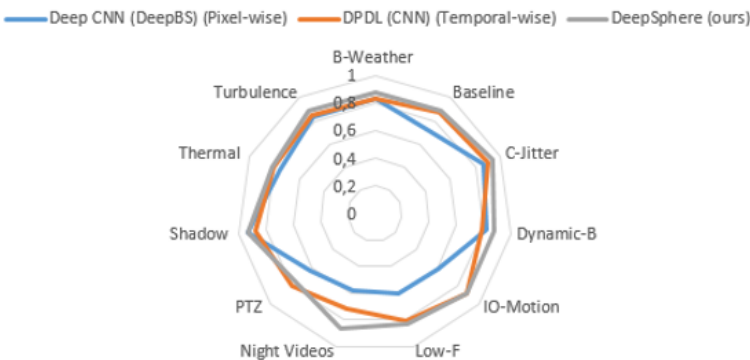


Figure 3.11: (a) Gain in performance between MOG [427], T2FGMM_UV [67] and DeepSphere [34] for the CDnet2014 dataset [460]. (b) Gain in performance between DeepSphere [34] and unsupervised models, RPCA [84] and DeepPBM [149] for the CDnet2014 dataset [460]. (c) Gain in performance between DeepSphere [34] and CNNs (supervised models) [42] [502] for the CDnet2014 dataset [460].

Table 3.8: Number of extracted images for each class from 53 cameras of CD-net2014 Video dataset [460].

Class	no. of images
person	3200
vehicle	2780
etc	1650

- Background Models Challenge (BMC2012) dataset:** In BMC2012 dataset [446], we compare our proposal on 9 real-world videos including challenges, such as cast shadows, the presence of dynamic backgrounds, the presence of a continuous flow of cars, intermittent object motion, general climatic conditions (sunny, rainy and snowy conditions), color saturation, lighting conditions and the presence of big objects. The average F-measure obtained using our proposal compared to the five best algorithms of BGSLibrary [418] and the two unsupervised foreground detection algorithms, RPCA [84] and DeepPBM [149] are reported in Table 3.9. We highlighted in bold the best F-measure values in each category. Our algorithm achieved the highest F-measure on almost all the categories, except in the "Train in the tunnel" category, where PBAS algorithm [196] obtained the best results, These results are attributed to the small-size of the foreground objects, which prevented DeepDC from effectively distinguishing these pixels. Results show that the average F-measure values across the 9 categories of the BMC2012 dataset are more stable in our proposal (DeepDC) respect to the rest of the methodologies. RPCA [84] is less performant than DeepPBM [149]. Our method outperforms RPCA [84] and DeepPBM [149] on BMC 2012 dataset [446] with 28% and 15% in average in F-measure score, respectively.

Numerous experiments were carried out to illustrate both the qualitative and quantitative performances of the proposed DeepDC. First, we present results of background subtraction on individual frames from nine video sequences: Parking (frame #1563), Big trucks (frame #64), Wandering students (frame #250), Rabbit in the night (frame #215), Snowy christmas (frame #17097), Beware of the trains (frame #699), Train in the tunnel (frame #1454), Traffic during windy day (frame #140) and one rainy hour (frame #15555). Table 3.10 shows the visual results obtained using DeepPBM [149], RPCA [84] and our proposed method on 9 real videos from BMC 2012 dataset [446]. Results confirm that our proposed DeepDC based on DeepSphere algorithm clearly improves the foreground mask by reducing false positives and negative detections.

We can remark that DeepSphere outperforms both DeepPBM [149] and RPCA [84] on almost all categories of BMC 2012 dataset [446], except in the "Train in the tunnel" category. Our method is more successful in detecting foreground objects in these videos, and provides acceptable results, while RPCA [84] and DeepPBM [149] both fail to detect efficient foreground masks. Our descriptor clearly appears to be more robust to noise and to the dynamic nature of the background as DeepPBM [149] is whereas RPCA [84] is not, and robust to illumination changes as RPCA [84] is whereas DeepPBM [149] is not.

It can be observed in Tables 3.9 and 3.10, that, in the case of real video sequences, DeepDC generally achieves the highest accuracy, in terms of recall, precision and F-measure metrics. Specifically, DeepDC well handles typical background maintenance challenges, such as waving trees in "Boring parking" and "Traffic during windy day" categories and gradual light changes in "Boring parking" category, despite the tiny size of moving objects such as those in "Rabbit in the night", "Beware of the trains" and "One rainy hour" and the large moving objects, such as those in "Big trucks" category. However, the combination of several challenges can compromise the performance of the proposed method. For example, in the "train in the tunnel" category, the presence of strong reflections combined with the small-sized foreground objects, the proposed algorithm DeepDC was superseded by the PBAS method [196].

Figure 3.12 compares the performance of the five best algorithms of BGSLibrary [418], DPWrenGABGS, MixtureOfGaussianV1BGS, T2FGMM_UV, LBFuzzyAdaptiveSOM and PBAS as well as RPCA [84], DeepPBM [149] and the proposed descriptor. Graphical values of our proposed method are enough successful than other methods as illustrated in Figure 3.12. Our proposal attains the highest quantitative values such as the F-measure exceeds 68 % in the majority of the categories.

By analyzing Figure 3.13(a), we can see that the fuzzy Gaussian model, T2FGMM_UV slightly outperforms the statistical model, MOG in almost all the categories of BMC 2012 dataset [446], except in the "snowy christmas" category. However, the F-measure did not exceed 0.75 on average, which is relatively low. Second, we can also see in Figure 3.13(a) that the proposed DeepSphere neural network model achieves better performance than T2FGMM_UV and MOG in all the categories. The gain in F-measure score was approximately 15%.

By looking at Figure 3.13(b), we can first see that DeepPBM [149] achieves better performance than RPCA [84]. We also remark that the proposed model based on DeepSphere outperforms both RPCA [84] and DeepPBM [149] algorithms in almost all the categories of BMC 2012 dataset [446], except in the "Train in the tunnel" category, where the proposed algorithm DeepSphere was superseded by the DeepPBM [149] method. The gain in F-measure score was approximately 10%. The average F-measure of DeepSphere was roughly 0.78, which becomes more acceptable in terms of reliable use in real conditions.

Table 3.9: Performance values of the proposed method compared to the other methods on 9 real videos from BMC 2012 Dataset [446]

Videos	Method	Recall	Precision	F-measure
Boring parking, active bkgk	DPWrenGABGS [470]	0.536	0.508	0.5196
	MixtureOfGaussianV1BGS [228]	0.563	0.504	0.5306
	T2FGMM_UV [67]	0.4356	0.8640	0.5792
	PixelBasedAdaptiveSegmenter [196]	0.659	0.756	0.7051
	LBfuzzyAdaptiveSOM [295]	0.4056	0.7682	0.5351
	RPCA [84]	0.7169	0.370	0.4854
	DeepPBM [149]	0.5587	0.8907	0.6867
	DeepSphere (ours)	0.6290	0.9318	0.7510
Big trucks	DPWrenGABGS [470]	0.5527	0.7368	0.6316
	MixtureOfGaussianV1BGS [228]	0.538	0.405	0.4631
	T2FGMM_UV [67]	0.4553	0.7793	0.5741
	PixelBasedAdaptiveSegmenter [196]	0.5079	0.7090	0.5918
	LBfuzzyAdaptiveSOM [295]	0.488	0.535	0.511
	RPCA [84]	0.630	0.8010	0.6834
	DeepPBM [149]	0.852	0.881	0.8602
	DeepSphere (ours)	0.8578	0.8918	0.8735
Wandering students	DPWrenGABGS [470]	0.6866	0.3764	0.4075
	MixtureOfGaussianV1BGS [228]	0.3855	0.6119	0.4105
	T2FGMM_UV [67]	0.6226	0.4986	0.4760
	PixelBasedAdaptiveSegmenter [196]	0.5603	0.6342	0.5389
	LBfuzzyAdaptiveSOM [295]	0.2351	0.7014	0.3522
	RPCA [84]	0.8403	0.90	0.8723
	DeepPBM [149]	0.9287	0.851	0.9432
	DeepSphere (ours)	0.929	0.9890	0.9590
Rabbit in the night	DPWrenGABGS [470]	0.6011	0.6666	0.5640
	MixtureOfGaussianV1BGS [228]	0.4651	0.6530	0.4718
	T2FGMM_UV [67]	0.800	0.747	0.662
	PixelBasedAdaptiveSegmenter [196]	0.51	0.82	0.59
	LBfuzzyAdaptiveSOM [295]	0.3475	0.8963	0.5008
	RPCA [84]	0.3179	0.8339	0.4603
	DeepPBM [149]	0.3579	0.9820	0.5246
	DeepSphere (ours)	0.5267	0.9827	0.6858
Snowy christmas	DPWrenGABGS [470]	0.6700	0.7045	0.5745
	MixtureOfGaussianV1BGS [228]	0.6955	0.8326	0.6829
	T2FGMM_UV [67]	0.6964	0.7079	0.6596
	PixelBasedAdaptiveSegmenter [196]	0.8824	0.5969	0.7122
	LBfuzzyAdaptiveSOM [295]	0.7108	0.7012	0.6623
	RPCA [84]	0.1538	0.6850	0.2513
	DeepPBM [149]	0.5385	0.9255	0.6808
	DeepSphere (ours)	0.7840	0.8160	0.7532
Beware of the trains	DPWrenGABGS [470]	0.716	0.503	0.5750
	MixtureOfGaussianV1BGS [228]	0.7373	0.7586	0.7220
	T2FGMM_UV [67]	0.7283	0.8922	0.7556
	PixelBasedAdaptiveSegmenter [196]	0.7882	0.7179	0.7159
	LBfuzzyAdaptiveSOM [295]	0.4485	0.9677	0.6129
	RPCA [84]	0.6151	0.7820	0.6814
	DeepPBM [149]	0.877	0.811	0.836
	DeepSphere (ours)	0.9133	0.8143	0.8597
Train in the tunnel	DPWrenGABGS [470]	0.55	0.38	0.40
	MixtureOfGaussianV1BGS [228]	0.3064	0.7583	0.4364
	T2FGMM_UV [67]	0.4465	0.4482	0.4480
	PixelBasedAdaptiveSegmenter [196]	0.72	0.62	0.64
	LBfuzzyAdaptiveSOM [295]	0.5553	0.4544	0.5037
	RPCA [84]	0.2321	0.2321	0.3038
	DeepPBM [149]	0.5714	0.6184	0.5945
	DeepSphere (ours)	0.573	0.495	0.5376
Traffic during windy day	DPWrenGABGS [470]	0.4673	0.4724	0.4444
	MixtureOfGaussianV1BGS [228]	0.3323	0.3232	0.3277
	T2FGMM_UV [67]	0.629	0.544	0.5837
	PixelBasedAdaptiveSegmenter [196]	0.6485	0.8413	0.7330
	LBfuzzyAdaptiveSOM [295]	0.2408	0.4083	0.3029
	RPCA [84]	0.3914	0.5202	0.4482
	DeepPBM [149]	0.79	0.74	0.7638
	DeepSphere (ours)	0.6535	0.9490	0.7740
One rainy hour	DPWrenGABGS [470]	0.71	0.63	0.65
	MixtureOfGaussianV1BGS [228]	0.85	0.65	0.59
	T2FGMM_UV [67]	0.751	0.619	0.679
	PixelBasedAdaptiveSegmenter [196]	0.663	0.6927	0.5902
	LBfuzzyAdaptiveSOM [295]	0.40	0.37	0.39
	RPCA [84]	0.677	0.503	0.5769
	DeepPBM [149]	0.671	0.837	0.74
	DeepSphere (ours)	0.838	0.907	0.867
Average	DPWrenGABGS [470]	0.6099	0.5536	0.5297
	MixtureOfGaussianV1BGS [228]	0.5414	0.6107	0.515
	T2FGMM_UV [67]	0.6183	0.677	0.6069
	PixelBasedAdaptiveSegmenter [196]	0.6599	0.7264	0.6417
	LBfuzzyAdaptiveSOM [295]	0.4257	0.6447	0.4823
	RPCA [84]	0.5082	0.6252	0.529
	DeepPBM [149]	0.6602	0.8096	0.7299
	DeepSphere (ours)	0.7449	0.8640	0.7845

Table 3.10: Visual results on real-world videos of the BMC2012 dataset [446]: From left to right: Original images, Ground-Truth images, RPCA [84], DeepPBM [149], DeepSphere (ours).

Categories	Background image	Input image	ground truth	RPCA	DeepPBM	DeepSphere (ours)
Boring parking, active bkgg (Frame #1563)						
Big trucks (Frame #64)						
Wandering students (Frame #250)						
Rabbit in the night (Frame #215)						
Snowy christmas (Frame #17097)						
Beware of the trains (Frame #699)						
Train in the tunnel (Frame #1454)						
Traffic during windy day (Frame #140)						
One rainy hour (Frame #15555)						

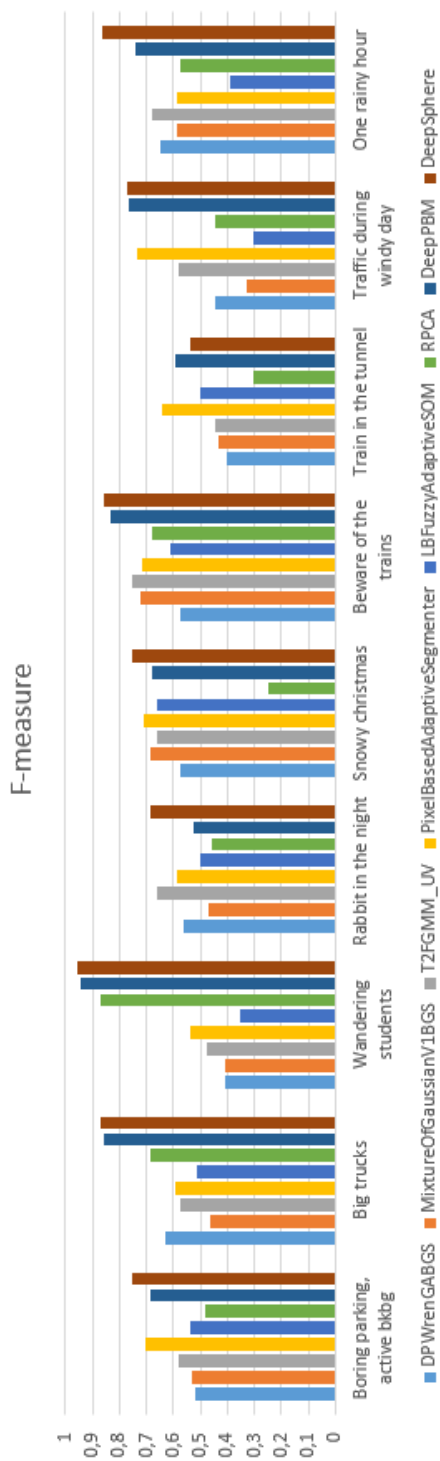


Figure 3.12: F-measures obtained with the proposed scheme and other methods on 9 real-world videos of the BMC 2012 dataset [446].

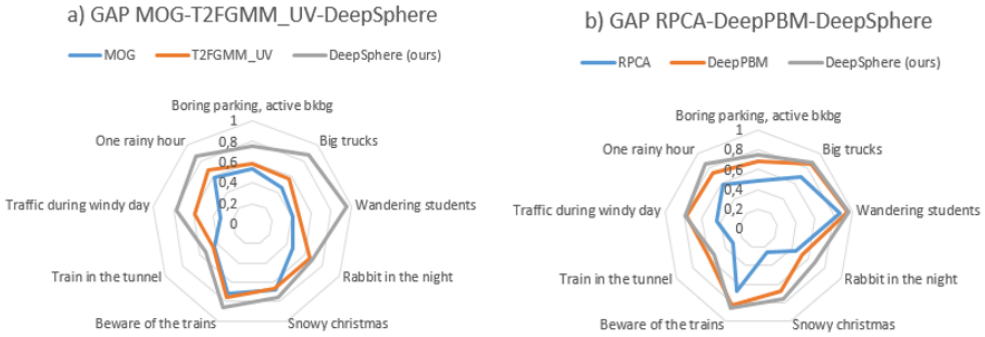


Figure 3.13: (a) Gain in performance between MOG [427], T2FGMM_UV [67] and DeepSphere [34] for the BMC 2012 dataset [446]. (b) Gain in performance between DeepSphere [34] and unsupervised models, RPCA [84] and DeepPBM [149] for the BMC 2012 dataset [446].

The final results we give is about the computational time which is an important factor for some applications. We collected the computational times needed to detect and segment the foregrounds using RPCA [84], DeepPBM [149] and the proposed model based on DeepSphere on the six short real videos of BMC 2012 dataset [446] as well as the average computational time (in minutes). We used 3.5 GHZ Intel Core i7. Results are summarized in Table 3.11. Our DeepDC model based on DeepSphere shows better time performance than both RPCA [84] and DeepPBM [149]. According to the reported results, background subtraction using the trained model based on DeepSphere can be done in more than 8 times and 3 times faster than RPCA [84] known as one of the standard and well-performed subspace learning methods and DeepPBM [149], respectively.

Table 3.11: Computational time for the BS task of our DeepSphere compared to RPCA [84] and DeepPBM [149] evaluated on the 6 short videos of BMC 2012 dataset [446]. For the fair comparison we ran the trained model using Intel Core i7 Hardware.

Algorithm	Run Time
<i>Big trucks - 1498 frames</i>	
RPCA [84]	38 min
DeepPBM [149]	15.4 min
DeepSphere (ours)	6.26 min
<i>Wandering students - 795 frames</i>	
RPCA [84]	30 min
DeepPBM [149]	9.2 min
DeepSphere (ours)	3.36 min
<i>Rabbit in the night - 1896 frames</i>	
RPCA [84]	45 min
DeepPBM [149]	23.6 min
DeepSphere (ours)	8 min
<i>Beware of the trains - 1065 frames</i>	
RPCA [84]	35.5 min
DeepPBM [149]	13.5 min
DeepSphere (ours)	7.43 min
<i>Train in the tunnel - 1726 frames</i>	
RPCA [84]	43 min
DeepPBM [149]	20.2 min
DeepSphere (ours)	7.3 min
<i>Traffic during windy day - 793 frames</i>	
RPCA [84]	25 min
DeepPBM [149]	8.4 min
DeepSphere (ours)	3.26 min
<i>Average over all the videos</i>	
RPCA [84]	42.09 min
DeepPBM [149]	15.04 min
DeepSphere (ours)	5.8 min

3.5 Conclusion

In this chapter, a new deep detector classifier called DeepDC for moving objects detection and segmentation is proposed. DeepDC is based on an anomaly discovery framework called DeepSphere, which combines the strengths of deep autoencoders and hypersphere learning to detect anomalies in dynamic networked systems. We propose to adapt and validate DeepSphere to perform foreground objects segmentation. DeepDC generates good segmentation results without additional image processing. It is also tolerant to illumination changes as RPCA is whereas DeepPBM is not and robust to noise and the dynamic nature of the background as DeepPBM is whereas RPCA is not. We compared the proposed DeepDC model to the 29 algorithms implemented in the BGSLibrary as well as to RPCA and DeepPBM on real videos from VIRAT video dataset [333], CDnet 2014 dataset [460] and BMC 2012 dataset [446]. Experimental results show that the proposed model qualitatively and quantitatively outperforms the mentioned methods in both time efficiency and accuracy, making it a serious candidate for the background subtraction task. In the next chapter, we will describe our semi-supervised approach for moving objects classification to deal with the lack of labeled data. The proposed approach classifies the extracted objects using the discriminator network of DCGANs in a semi-supervised manner.

Chapter 4

A novel semi-supervised DCGAN model for object classification

This chapter presents a novel semi-supervised learning approach based on deep convolutional generative adversarial networks, DCGANs, able to extract suitable features to classify objects. Our proposal called DCGAN-based semi-supervised learning (DCGAN-SSL) is an extension of the DCGAN architecture for training a classifier while making use of labeled and unlabeled data. In addition, it allows to learn a generative model and a classifier simultaneously. A DCGAN is originally intended for unsupervised learning, we adapt it for semi-supervised learning classification task. Results on VIRAT video dataset [333] et CDnet 2014 dataset [460] show the relevance of the proposed approach. DCGAN-SSL classifier outperforms not only three standard models (TSVM, CatGAN, VAT, etc) as expected but also its recent competitor, the CNN model, which was especially designed for the object classification task.

The work presented here was published in the IET Image Processing Journal [34].

Contents

4.1	Motivation	80
4.2	Generative Adversarial Networks	83
4.3	DCGANs architecture	84
4.4	Proposed approach	85
4.5	Experiments	90
4.5.1	Datasets	90
4.5.2	Experimental results and discussions	90
4.6	Conclusion	96

4.1 Motivation

Nowadays, digital image processing techniques have evolved rapidly with the development of image classification and recognition technologies. Due to the high non-linear approximation capacity of these image classification technologies, CNN have proved to be the efficient way, attracting more and more attention and obtaining various applications in image classification. However, the large application of CNNs to high-resolution images is still been prohibitively expensive, despite the relative efficiency of their local architecture and their appealing qualities. In addition, CNN requires a large amount of labeled data to process and train the neural network which, while acquiring additional data or labeling all the data, remains expensive or even impossible. To overcome these problems, most image classification approaches have applied GANs because of the benefits that GANs can work well with the lack of data and its super-resolution. In this chapter, due to the good performance of the convolutional operation, we propose to apply an extension to GAN, called DCGANs on unlabeled samples to enhance the accuracy of object classification using a small number of labeled samples. Compared to GAN, DCGAN focuses on using Deep Convolutional networks in place of fully-connected networks. DCGAN is more suitable for handling multimedia and image data, as two-dimensional convolution and deconvolution operations can be performed. Convolutional networks generally search for spatial correlations in an image. This means that a DCGAN would likely be more suitable for image/video data than a conventional GAN. In addition, when it comes to categorizing images, fully connected layers need a lot of weights in the first hidden layer. Networks with a big amount of parameters encounter numerous problems, for example, chances of overfitting, slower training time, etc.

Various imaging-based applications, including object detection, object segmentation, image classification and image recognition take advantage from deep learning networks. One of the reasons for the success of deep learning applications is that the model can learn from a huge amount of labeled training samples. Supervised learning has been at the center of deep research. It involves learning from labeled training samples, where each individual sample consists of the instance problem with its label. For example, in a classification task, the data element to be categorized is represented as a feature vector and the class is assigned as a categorical label. The set of samples, also known as a training set or a labeled set, is used to create the classifier which can be used to categorize any new given sample. However, all supervised object classification methods based their approaches on the assumption of availability of large labeled dataset. For many areas of interest, data collection is relatively easy while labelling it by human experts is expensive and time-consuming. However, it can be argued that the most widespread framework is where a big number of unlabeled data exists, but we want to train some supervised predictor. Generally, labeling all the data is extremely expensive, therefore the labeled dataset is usually several orders of magnitude smaller. Researchers are interested in minimizing the cost of obtaining labeled training examples, and there are several studies are underway with unsupervised and semi-supervised deep learning. In many real world applications such as text processing and image processing, where there is an abundant amount of unlabeled samples, requiring people to label unlabeled data, is an

expensive task. In these applications, labeled data is sparse. Supervised learning is a cost effective and time consuming process, since it needs a big number of labeled training samples. In contrast, unsupervised learning does not need any labeled data and groups the data depending on the similarity of data points by using either maximum likelihood or clustering approach. However, this approach can not accurately cluster an unknown data. Unsupervised learning is more complicated than supervised learning, because we lack the ground truth to assess the results. To solve these problems, Semi-supervised learning (SSL) has been suggested by researchers, which can learn with a few number of training data, can label the training data and treats the remaining samples as test data. Semi-supervised image classification leverages both labeled and unlabeled data to increase classification performance. It is an intermediate between supervised and unsupervised learning that incorporates the ability to use partially labeled dataset. According to different learning tasks, many semi-supervised classification methods and semi-supervised clustering methods are available in the literature [34] [96] [98] [220] [242] [309] [323] [380] [423] [387] [511]. Existing GAN-based image classification methods are still unsupervised, as these methods do not employ label information and the images produced by the generator which are used to train GAN are also unlabeled. The classification accuracy could be increased. The need to create models that can learn from less data is increasing faster. Consequently, in this chapter, we present a DCGAN based semi-supervised learning model, called DCGAN-SSL to classify objects extracted from video sequences from VIRAT video dataset [333] and CDnet2014 dataset [460] using a very small labeled training set. Semi-supervised learning is a technique in which both labeled and unlabeled samples are employed to train a classifier.

Until recently, to our best knowledge, there have been no previous methods which processed altogether semi-supervised learning and DCGAN networks for multi-class object classification problem. State-of-the-art object classification approaches operate on an unsupervised way, ignoring the supervised learning. Goodfellow et al. [169] proposed a novel framework to estimate generative models via a contradictory learning. The GAN generator can be used to learn the actual distribution of unsupervised data. Radford et al. [360] demonstrate that DCGANs are an excellent candidate for unsupervised learning. A hierarchy of representations from object regions to scenes is learned by a deep convolutional adversarial pair in both the generator and discriminator. In addition, the learned features are employed for further tasks, proving that they are applicable as common representations of images. The supervised and unsupervised learning are two significant techniques used for object classification. Using generative models for semi-supervised learning is not a new idea. Kingma et al. [242] expand work on variational generative models [241] [370] to do just that. Here, we are trying to do something similar with GANs. We are not the first to apply GANs in a semi-supervised context. GANs have been widely used and have obtained competitive results for semi-supervised learning [387] [423] [117] [262] [132] [256] [243] [58] [477]. The CatGAN [423] modifies the objective function to take into account mutual information between observed examples and their predicted class distribution. Salimans et al. [387] report a way to utilize GANs for a classification task with K classes. More precisely, they propose to extend the vanilla GAN where the labeled data is increased with examples generated from the generator. The discriminator learns to predict the original classes and one fake class of the generated data. This assists the discriminative model by increasing a small amount of labeled data with a large amount of unlabeled data of real and generated samples. Their work presents

various new architectural features and training processes, such as feature matching and mini-batch discrimination functions, to help the convergence of GANs. In this way, GAN not only produces a large amount of samples and expands the training dataset, but also improves the ability of the networks to extract features and the generalization accuracy of the classifier via the adversarial training method. In Salimans et al. [387], a fully connected generator network was employed. In our work, we replace it with a DCGAN and achieve a superior performance. As one of the previous works, Diederik [129] uses the deep generative model in a semi-supervised learning way through the maximization of the variational lower bound of the unlabeled data likelihood and assumes an additional latent variable in the directed generative model, which is corresponding to the classification label. In Donahue et al. [131], an adversarial formulation with a third element is described, called the “encoder”. The encoder tries to encode real data to some latent space, while the generator allows mapping a simple distribution in latent space to data space. They demonstrated that this encoder learns for inverting the generator, and can be employed as a featurizer for a supervised training. On the autoregressive side, Dai and Le [116] explored the idea of first “pretraining” a sequence model to perform a task on unlabeled text data. Then, they employed these pretrained weights for training supervised models for text classification. Their results show improved learning stability and model generalization. In 2017, Radford et al. [361] learned a language model by training an mLSTM RNN on Amazon reviews and then employed its internal cell state from the last time step as features for the following supervised task of sentiment analysis of Amazon reviews. This allowed the authors to match the state-of-the-art in their sentiment analysis dataset with far fewer labeled samples and to outperform it with the full training set.

In this chapter, we introduce DCGANs to learn useful representations during the adversarial training process and the learned features are used to classify images with relatively small number of training samples, so as to use both the labeled training data and the unlabeled generated samples to train DCGAN for object classification task. We exploit the power of an unsupervised representation learning using DCGANs to build an image classifier which can be trained with relatively small amount of labeled training samples as compared to a fully supervised process. Our DCGAN-SSL classification model aims to improve the feature extraction ability of the discriminator and the classification performance. We believe the combination of semi-supervised learning with DCGANs networks may provide useful information for object classification. By fusing semi-supervised learning and DCGANs, the derived DCGAN-SSL extracts more detailed information from the objects to be classified. Our contributions can be summarized as follows

- A robust combination of DCGANs and semi-supervised learning that allows us to be more robust on feature extraction and object classification task. We extend DCGANs to the semi-supervised learning context by forcing the discriminator network to output class labels. DCGAN-SSL allows to learn a generative model and a classifier simultaneously and shows that it can be used to create a more data-efficient classifier.
- A detailed comparative evaluation of our proposed DCGAN-SSL model against other four state-of-the-art models on two large scale datasets that are CDnet2014 dataset [460] and VIRAT video dataset [333]. We show that DCGAN-SSL improves classification performance on restricted data sets over a baseline classifier with no generative component. DCGANs can significantly improve the quality of the generated samples.

The rest of this chapter is organized as follows. The GAN architecture is illustrated in Section 4.2. In Section 4.3, we describe the DCGANs architecture. Our proposed DCGAN-SSL model which trains DCGANs in a semi-supervised way is presented in Section 4.4. Comparative results on real world videos are given in Section 4.5. Finally, the conclusion is drawn at the last section closed the Chapter 4.

4.2 Generative Adversarial Networks

GAN is a class of artificial neural networks newly developed by GoodFellow et al. [169], which trains two adversarial neural networks as presented in Figure 4.1. The first neural network consists of the generator, which takes as input a random noise and produces new data samples. The second network, named the discriminator, obtains input from both the generator and the original training data. The discriminator examines samples, and determines whether the data belongs to the actual training dataset or comes from the generator. A point will be achieved when the generator captures the whole distribution of training samples. Thus, the discriminator is not able to distinguish whether the inputs come from the generator or not. It is said, at this time, that the GAN is fully trained.

Specifically, the generator (G) takes as input a random noise vector z and produces a sample $X_{fake} = G(z)$. The discriminator D input consists of samples produced by the generator and original samples and it outputs a probability distribution of the data possible sources. Equation 4.1 illustrates the complete GAN training process. The discriminator D focuses on maximizing the log-likelihood of assigning the correct label, while the generator (G) is trained to maximize the probability that D makes an error (second term in the equation).

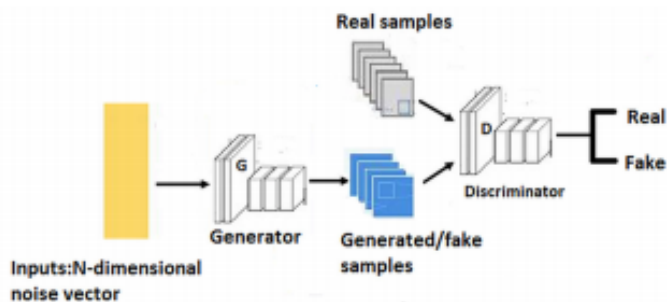


Figure 4.1: GAN high-level architecture.

(<https://jglobal.jst.go.jp>)

$$L = E[\log P(Y = \text{real} | X_{\text{real}})] + E[1 - \log P(Y = \text{fake} | X_{\text{fake}})] \quad (4.1)$$

GANs are known to be hard in train and unstable. This causes the generator in many cases to output poor samples. As a result, many researchers have focused on improving the

stability of training. DCGAN [360] is one of the common GAN extensions, which generates peak performance. We used the DCGAN architecture and its improvement by Salimans et al. [387], which is based on the feature matching concept. Feature matching is a technique to deal with GAN instability by identifying a new generator goal. In the feature matching process, the generator is trained to match the expected value of the features on an intermediate layer of the discriminator, unlike the conventional GAN, where the generator is trained to directly maximize the discriminator output. This results in an improved stability in situations where the conventional GAN is unstable. Let $f(x)$ indicate activations on an intermediate layer of the discriminator, the new objective for the generator is described in Equation 4.2.

$$L_G = \|E_{x \sim p_{data}} f(x) - E_{z \sim p_z(z)} f(G(z))\|_2^2 \quad (4.2)$$

4.3 DCGANs architecture

DCGAN is an improved version of the original GAN architecture with deep convolutional networks (CNNs). Compared to the original GAN, DCGAN almost entirely uses the convolutional layer rather than the fully connected layer. In this thesis, we exploit the ability of DCGAN's discriminator [360] to classify the extracted objects from video sequences. The idea is to simultaneously train two adversarial networks. The first network is a discriminator that learns to determine if the sample comes from the data distribution. The second is a generative model that aims to generate "fake" images that attempts to fool the discriminator. After several stages of training, the optimization will achieve a stable point where the discriminator will be difficult to discern whether the data was "fake" or not. Mathematically, the training process of DCGANs can be seen as a minimax game. The generator $G(z)$ takes an input z from a uniform distribution. The discriminator $D(\cdot)$ takes x as input, being either images, from selected database or output of generator $G(z)$. During training, the discriminator tries to distinguish between selected database and $G(z)$, i.e. attempts to maximize, $\log(D(x)) + \log(1 - D(G(z)))$. Simultaneously, the generator attempts to fool the discriminator by minimizing $\log(1 - D(G(z)))$. The optimization will achieve a point of equilibrium where the discriminator is unable to distinguish between x and $G(z)$, after multiple steps of training.

DCGANs were the first major advancement on the original GAN architecture. The architecture of DCGAN can be summarized as follows:

- Replace all max pooling layers with strided convolutions for both the discriminator and the generator networks
- Use transposed convolution for upsampling
- Remove fully connected (FC) hidden layers
- Use Batch normalization (BN) for both the discriminator and generator networks.
- Use Rectified linear unit (ReLU) activation in the generator except for the output which uses tanh
- Use LeakyReLU on all layers for the discriminator

4.4 Proposed approach

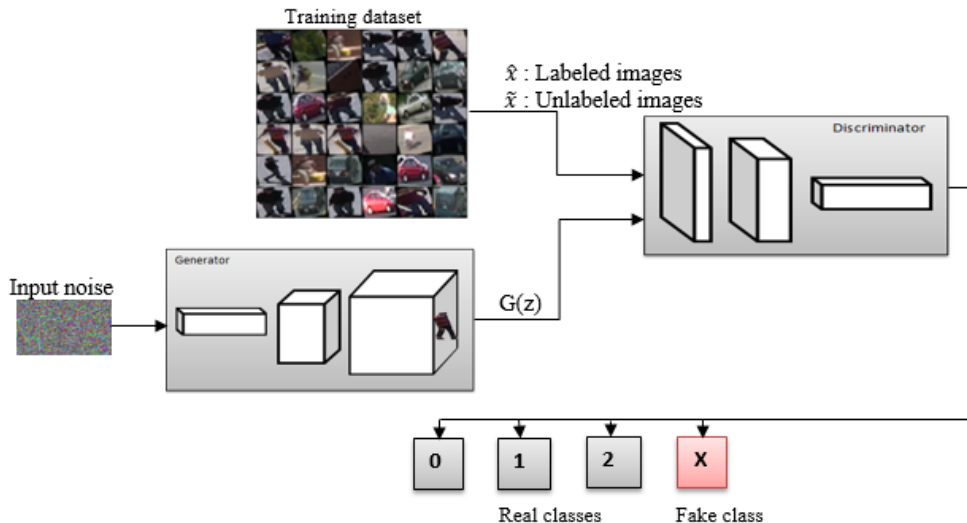


Figure 4.2: The proposed Semi-supervised learning DCGAN (DCGAN-SSL) architecture for a 4 class classification problem.

The ordinary DCGAN descriptor introduced by Radford et al. [360] has proved to be a powerful candidate for unsupervised learning. The practically unrestricted number of unlabeled images and videos can be employed in order to learn a strong intermediate data representations, which can be then used on various supervised learning tasks like image classification. We propose that one way to create powerful image representations is to train DCGANs [360], and later to reuse parts of the generator and discriminator nets as supervised feature extractors. In a conventional supervised learning environment, a standard classifier model is usually needed to categorize an input x into one of the possible N classes. The classifier outputs class probabilities $P_{model}(y|x)$, and is then trained to reduce the cross-entropy between the labels and the predictive probability distribution of the classifier. Deep neural networks are typically trained on vast quantities of labelled data and it has been difficult to apply deep models to datasets with limited labels.

In this thesis, owing to the need to create models that can take advantage of fewer data, we propose a DCGAN-SSL model to classify the objects extracted from video sequences (Chapter 3), which include vehicles, people, small objects and images containing some small parts of the body, etc. Our DCGAN-SSL classification model is applied on unlabeled samples to achieve better accuracy in supervised object classification using only a few quantities of labeled training samples. A shared discriminator/classifier is applied that distinguishes real samples from false ones and predicts the class label. We adapt and extend DCGANs to a semi-supervised classification of the extracted objects, by replacing the traditional discriminator with a multi-class classifier, which, instead, of predicting whether a sample belongs to the

data distribution (it is real or not), it affects at each pixel of the input image a label from the K real classes or mark it as a fake sample (additional $K + 1$ class). Therefore, $P_{model}(y = K + 1 | x)$ is used to denote the probability that the given input x is fake. This allows the model to learn from unlabeled samples, as it can be deduced that the model input falls into one of the K original dataset classes by maximizing $\log P_{model}(y \in \{1, \dots, K\} | x)$. The generator functions as a source of diverse information from which the discriminator obtains unlabelled training samples. In our approach, the intuition is utilizing the samples generated by DCGAN generators to improve the classification tasks. These samples are added to the dataset, thereby, increasing the class labels of the original dataset. The images obtained by background subtraction are of low resolution, as shown in Figure 4.3. Therefore, it is crucial to define the convenient size of entered images to train a performant DCGAN network. Our goal is to categorize the selected ROIs. We assume that the use of DCGAN discriminator network can easily handle the object classification task, as we have eliminated a large number of undesired images in Chapter 3. We use feature matching for the generator loss [423].

A brief overview of the proposed framework is presented in Figure 4.2. The RGB color input images are 32×32 pixels belong to three categories: 'person', 'cars' and 'etc'. The last class 'etc' includes all badly detected objects (non-vehicles/ non-person). An artificial "fake" class is added, corresponding to the class $K + 1$. The discriminator has two functions. It acts as a supervised classifier and it distinguishes real and fake images simultaneously. We used the DCGAN discriminator as a $K + 1$ (in our case = 4) class classifier. It recognized the K different classes of labelled data, as well as the $(K + 1)^{th}$ fake class that represents the output of the generator.

There are three different sources of training data for our DCGAN-SSL discriminator:

- Real images with labels \hat{x} : These are pairs of image label as in any conventional supervised classification issue.
- Real images without labels \tilde{x} : The classifier only learns them as real.
- Images from the generator $G(z)$: The discriminator learns to classify them as fake.

Combining these different data sources will allow the classifier to learn from a wider perspective, allowing the model to make inferences much more accurately than it would be when using only the labeled samples for training. Let p_{data} denotes the data distribution and p_z be the model distribution implicitly defined by $G(z)$ when $z \sim p_z$.

Therefore, the discriminator losses consist of the following components:

- Supervised loss: calculates the individual real class probabilities using a softmax cross entropy function from the estimated distribution on $K = 3$ object categories, which represents 'person', 'vehicles' and 'etc'.

$$L_{supervised} = -E_{\hat{x}, y \sim p_{data}(\hat{x}, y)} [\log P_{model}(y | \hat{x}, y < K + 1)] \quad (4.2)$$

- Unsupervised loss: the discriminator must distinguish between real training images and fake images produced by the generator. It represents the loss resulting from the

classification of unlabelled samples as real, and the loss from categorizing generated images as fake.

- The loss from classifying inputs as real:

$$-E_{\tilde{x} \sim p_{data}(\tilde{x})} [\log(1 - P_{model}(y = K + 1 | \tilde{x}))] \quad (4.3)$$

- The loss from classifying produced samples as fake:

$$-E_{z \sim p_z(z)} P_{model}(y = K + 1 | G(z)) \quad (4.4)$$

The discriminator loss is the sum of both the supervised loss and the unsupervised loss. The individual real class probabilities are calculated by the supervised loss. As this is a multi-class classification problem, it is optimized using a softmax cross entropy function. Unsupervised loss is computed using sigmoid cross entropy. On the other hand, generator loss is obtained by using feature matching techniques. To do that, features are taken after the GAP layer, when real data is processed by the discriminator. A moment is calculated when the discriminator analyzes generated fake samples from the generator. The average absolute difference between the two moments represents the loss of the generator. The loss function for training the generator network G is defined as:

$$L(G) = \|E_{\tilde{x} \sim p_{data}(\tilde{x})} f(\tilde{x}) - E_{z \sim p_z(z)} f(G(z))\|_2^2 \quad (4.5)$$

where $f(\cdot)$ represents the output of the feature layer. The losses are adjusted in a way that the discriminator can help the generator learning how to generate realistic samples, therefore, the discriminator must distinguish between real and fake samples and to use the generator's images, as well as the labeled and unlabeled training samples, to help categorize the dataset.

Algorithm 1 illustrates our proposed DCGAN-SSL approach.

Algorithm 1 Minibatch Stochastic Gradient Descent (SGD) Training of DCGAN-SSL

- 1: **for** number of training iterations **do**
 - 2: Sample mini-batch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_z(z)$
 - 3: Sample mini-batch of m labeled examples $\{(\hat{x}^{(1)}, y^{(1)}), \dots, (\hat{x}^{(m)}, y^{(m)})\}$ from data generating distribution $p_{data}(\hat{x}, y)$.
 - 4: Sample mini-batch of m unlabeled examples $\{\tilde{x}^{(1)}, \dots, \tilde{x}^{(m)}\}$ from data generating distribution $p_{data}(\tilde{x})$.
 - 5: Update the discriminator by descending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m (-\log P_{model}(y^{(i)} | \hat{x}^{(i)}, y^{(i)} < K + 1) - [\log(1 - P_{model}(y = K + 1 | \tilde{x}^{(i)})) + \log P_{model}(y = K + 1 | z^{(i)})])$$
 - 6: Sample mini-batch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_z(z)$
 - 7: Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \|f(\tilde{x}^{(i)}) - f(G(z^{(i)}))\|_2^2$$
 - 8: **end for**
-

Table 4.1 represents the architecture and parameters used to build our DCGAN-SSL model. Our implementation closely mirrored the conventional implementation presented in the DCGAN paper by Salimans et al. [387].

Table 4.1: DCGAN-SSL model architecture

Model	Architecture details
Generator	
Layer1	Dense, output: 8192, batch normalization, activation: LeakyReLU
Layer2	Reshape layer, output shape: (32,32,3)
Layer3	2D transpose convolution layer, 256 (5×5) filters, unit strides, same padding, batch normalization, LeakyReLU activation
Layer4	2D transpose convolution layer, 128 (5×5) filters, unit strides, same padding, batch normalization, LeakyReLU activation
Layer5	2D transpose convolution layer, 64 (5×5) filters, unit strides, same padding, tanh activation
Discriminator	
Layer1	2-dimensional (2D) convolutional layer, 64 (5×5) filters, unit strides, same padding, LeakyReLU activation.
Layer2	2D convolutional layer, 128 (5×5) filters, unit strides, same padding, batch normalization, dropout (rate = 0.5), LeakyReLU activation
Layer3	2D convolutional layer, 256 (5×5) filters, unit strides, same padding, LeakyReLU activation
Layer4	Global average pooling 2 D
Layer5	Dense layer, output: 4, softmax activation.

Table 4.2: Baseline model architecture

Model	Architecture details
CNN	
Layer1	2D convolutional layer 64 (5×5) filters, unit strides, same padding, relu activation, maxPooling (2×2)
Layer2	2D convolutional layer 128 (5×5) filters, unit strides, same padding, relu activation, maxPooling (2×2)
Layer3	Dense (384), Dense (192), dropout (rate = 0.5), Dense layer, output: 4, softmax activation

The first layer of the generator is a dense layer, which takes in a seed of random noise, and reshapes it into a 4-D tensor. This layer is then preceded by a sequence of transpose convolutions, batch normalization and LeakyReLU functions. The sequence of operations upsamples the size of the input until the desired size is achieved. In our case, the desired image size is 32×32 , which is squeeze between values -1 and 1 through the hyperbolic tangent function. The discriminator works like a normal CNN classifier, it contains a sequence of convolution layers with batch normalization. However, rather than applying fully connected layer on top of convolution stack for the last layer, a global average pooling (GAP) operation, which is a regularisation technique, is applied. We apply GAP thanks to some advantages over traditional fully connected layers, which include greater robustness for spatial translation and fewer over-fitting problems as presented in Figure 4.7. In GAP, the average over the spatial dimensions of a feature map is computed which gives one value. Afterwards, a fully connected layer is applied to output the final logits, which represents the number of classes we want to predict. The logits are transmitted to a softmax function, which outputs the probabilities of classification. However, for modeling the binary classification value (the probability of an input being real or fake), a LogSumExp function is employed. The final logits are transformed into a sigmoid logits. This implementation ends with a softmax output layer with one unit for each of the classes. The discriminator could also output four units corresponding to (class-1, class-2, class-3, fake). Each input image is categorised based on the semi-supervised model.

Our baseline is a CNN. In order to reach an accurate comparison of classification efficiency, we adopt the same implementation for the discriminator of DCGAN experiments. For the baseline model, we use the architecture proposed in Kim et al. [238]. The number of labelled data points needed to achieve a level of performance similar to that of the baseline model is a powerful indicator that we benefit from unlabelled data.

Our baseline CNN classifier contains two convolutional layers, two pooling layers, two fully connected layers, and finally an output layer. Each convolutional layer is followed by a ReLU function and a pooling layer and contains 64 and 128 feature maps, respectively and the stride is fixed to 1. A max pool with 2×2 filters with stride 2 is used. The activation function ReLU and dropout are both employed at each fully connected layer. The two fully connected layers have 384 and 192 nodes, respectively. We used batches of size 50 and the learning rate is fixed to 0.001 for Adam optimiser. We use the architecture and parameters presented in Table 4.2 as the baseline model.

Let $D = \{X, Y, X_0\}$ be a dataset where (X, Y) are the labeled points, and X_0 is the rest of the unlabeled data, which is often orders of magnitude larger than X . Our DCGAN-SSL model requires a generative model and a discriminator being trained simultaneously using all of $\{X, Y, X_0\}$. The discriminator attempts to compute both the adversarial loss and the classification loss. Our proposed approach combines the DCGAN model and the work of Salimans et al. [387] to create a semi-supervised DCGAN network. The baseline model is trained only on the available labeled data $S_0(X, Y)$. The models are evaluated in their accuracy and the amount of labeled data required to converge to good results. We used a DCGAN discriminator to classify moving objects extracted from VIRAT video dataset [333] and CDnet2014 dataset [460] using a very small labeled training set. Both the generator and the discriminator are trained at the same time when building a DCGAN for generating images. After training,

the discriminator can be discarded because it is used only to train the generator. The generator is only employed to assist the discriminator during training. It behaves like a varied source of information including the unlabeled training samples used by the discriminator. These unlabeled data are essential to increase the performance of the discriminator. In addition, by turning the discriminator into a semi-supervised classifier, it has not only to compute the probability of whether its inputs are real or not as in regular image generation GAN, but also it has to learn the probabilities of each of the original dataset classes. For each input image, the discriminator has to learn the probabilities to determine its class. The discriminator returns a signal to the generator as a function of this probability, to adjust its parameters during training, which improves its ability to create realistic images. We have converted the discriminator of a regular GAN into a 4 class classifier. To do that, we turn its sigmoid output into a softmax with 4 class outputs. The first 3 for the original class probabilities of the VIRAT dataset [333] and the CDnet2014 dataset [460] (person/vehicle/etc), and the 4th class for all the fake images that come from the generator.

The discriminator acts in part as any other conventional classifier. For this reason, it can suffer from the same issues as any classifier if it is not properly constructed. One of the most likely disadvantages that can be encountered when a big classifier is trained on a very restricted dataset, is the huge of over-fitting. Overtrained classifiers generally show a significant difference between the smaller training error and the higher test error. This situation demonstrates that the model succeeded in capturing the structure of the training data. But, because it believed too much in the training data, it could not generalize for test samples. To avoid that, a large use of regularization is made through GAP and dropout, which allows reducing over-fitting in DCGAN-SSL as presented in Figure 4.8.

4.5 Experiments

4.5.1 Datasets

The performance of our proposed descriptor was evaluated on two public large datasets, the CDnet2014 dataset [460] and the VIRAT video dataset [333] dedicated to the evaluation of change and motion detection. We assess the performance of our method on moving objects extracted from these two datasets as presented in Chapter 3.

4.5.2 Experimental results and discussions

In this chapter, we propose a semi-supervised learning approach using DCGAN to categorize our image dataset. The main idea is to use the samples produced by DCGAN generators together with the unlabeled data to increase the performance of a classifier trained on a small number of labeled samples. Therefore, mitigating the challenges associated with collecting and labeling large dataset. The proposed descriptor can build a more efficient classifier and it generates higher quality samples compared to a regular GAN. Due to the need to create models that can take advantage from fewer data, we tried to use semi-supervised DCGANs to classify extracted objects in Chapter 3 (people, car, certain parts of the body, etc). In our

DCGAN-SSL classification method, we use the DCGAN discriminator to categorize objects extracted from VIRAT video dataset [333] using a very small labeled training set. A GAN with a classification discriminator exploits both the unlabeled and labeled data. The unlabeled data allow to simply distinguish fake from real. The labeled data allow for the optimization of the classification accuracy. We adapt and extend DCGANs, by replacing the conventional discriminator with a multi-class classifier, which, instead of, predicting whether a sample belongs to the data distribution (it is real or not), it assigns to each pixel of the input image a label from the K original classes or considered it as a fake sample ($K + 1$ class). We try to simultaneously solve a semi-supervised classification problem and learn a generative model. We performed semi-supervised experiments on ROIs extracted from Chapter 3 to see whether



Figure 4.3: VIRAT training dataset

the classifier component of the DCGAN-SSL would perform better than an isolated classifier on restricted training sets. Figure 4.3 shows some examples of the extracted ROIs from VIRAT video dataset [333]. We train semi supervised DCGAN without ever updating the Generator. We used different quantities of training data as labeled examples to test our semi-supervised DCGAN method, considering setups with 20, 50, 100, and 120 labeled samples per class. Results are averaged over 3 random subsets of labeled data, each chosen to have a balanced number of examples from each class. The remaining training images are provided without labels. Table 4.3 summarizes our results. The use of unlabeled data allows us to obtain good accuracy especially with much fewer labeled samples. The images produced by GAN generator do not look visually appealing and are completely indistinguishable from dataset images. DCGAN-SSL can enhance their visual quality and outputs images clearly distinguished from images coming from VIRAT dataset [333] as presented in Figure 4.4.

We focus on evaluating the effectiveness of our DCGAN-SSL method using varying amounts of labeled samples vs baseline CNN [238]. Figure 4.5 illustrates the effect of varying the number of labeled samples on the test accuracy when applying DCGAN-SSL vs CNN.

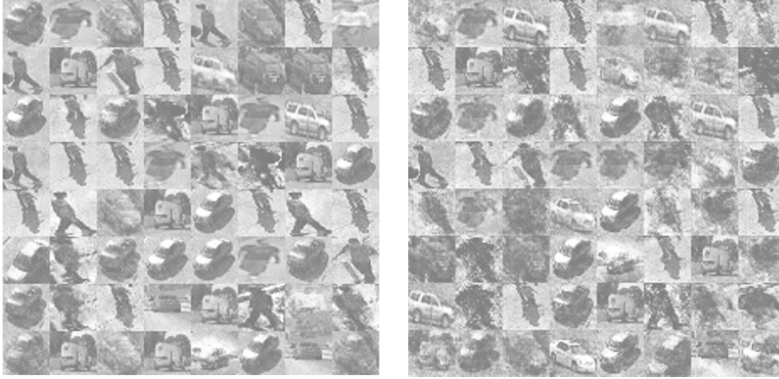


Figure 4.4: Samples generated by DCGAN-SSL and GAN from the VIRAT video dataset [333]. DCGAN-SSL is on the left and GAN is on the right. The results are obtained after 200 epochs of training the models.

Table 4.3: Classifier accuracy in VIRAT video dataset [333].

Examples	CNN [238]	DCGAN-SSL (Proposed method)
120	0.8912	0.9382
100	0.8616	0.8917
50	0.7833	0.8129
25	0.7013	0.7645

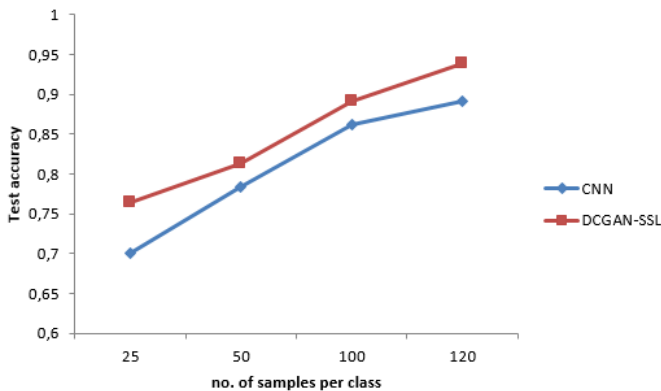


Figure 4.5: The test accuracy of the DCGAN-SSL over CNN for various amounts of labeled samples from the VIRAT video dataset [333].

To achieve the highest accuracy, we performed a random search on the value of learning rate. It can be seen that the use of a semi-supervised classification allows a highly better use of few quantities of labeled samples. At the lowest amount, using only 25 training examples, the semi-supervised model gives a high test accuracy of 0.76 vs 0.7 using only the supervised objective. This represents 6 % rise in efficiency when training with only 25 samples. DCGAN-SSL exceeds the baseline in proportion to shrinkage of the training set, suggesting that forcing the discriminator and the classifier to share weights improves the efficiency of the data. We extend the CNN classification presented in [238] to categorize extracted ROIs from the video and propose a method exploiting DCGAN discriminator as a multi-class classifier for semi-supervised classification purposes. Our proposed method outperforms the work of Kim et al. [238] which is based on supervised classification using CNN and demonstrates the effectiveness of semi-supervised classification learning applied on objects extracted from VIRAT video dataset [333]. This implementation reaches train accuracy of 0.95 and a test accuracy of roughly 0.93 using only 120 labeled examples.

In Figure 4.5, we can see that using a semi-supervised objective which mainly multi-tasks between the supervised CNN objective and the standard DCGAN objective allows much better use of smaller amounts of labels. Results show that the use of a semi-supervised classifier is useful for object classification, because of the large quantity of richer information that it can extract from the video. It is because our method uses a small amount of labeled samples. DCGAN-SSL performs much greater than CNN. CNN considers only supervised classification. For a fair comparison on the accuracy of the classification, the same architecture for the discriminator for CNN experiments is used. Next, we show that the proposed approach allows to improve the performance of a supervised object classification method, as compared to CNN and other existing models in the filed of object classification.

We have compared our proposed method with other models, namely TSVM [220], CatGAN [423], VAT [323], CNN [238] and calculate the average error rate on VIRAT video dataset [333] over 3 random splits of labeled samples. Results summarized in Table 4.4 show that our method based on DCGAN-SSL achieved its superior performance when there are only a few labeled examples. We train the DCGAN-SSL model on CDnet2014 dataset [460], on sets of labeled samples of size 100, 500 and 1000. Table 4.5 summarizes the results using CDnet2014 dataset [460]. We observe that with the different amount of labeled data, VAT [323] performs better than CatGAN [423] and TSVM [220]. Using 1000 labeled samples, DCGAN-SSL achieved the best performance compared to the state-of-the-art methods on CDnet2014 dataset [460], with the error rate of 1.04%.

Table 4.4: Performance comparison (error rate, %) on VIRAT video dataset [333] of other models to DCGAN-SSL for different numbers of labeled subsets per class

Examples	120	100	50	25
TSVM [220]	3.83 ± 0.77	3.864 ± 6.66	4.940 ± 0.36	5.62 ± 4.02
CatGAN [423]	2.70 ± 0.84	2.80 ± 4.2	3.87 ± 0.36	5.6 ± 3.92
VAT [323]	2.684 ± 1.2	2.77 ± 4.87	3.819 ± 0.6	4.50 ± 0.9
CNN [238]	2.27 ± 0.62	2.34 ± 1.23	3.79 ± 0.56	4.25 ± 0.85
DCGAN-SSL	1.80 ± 0.41	1.92 ± 0.22	3.38 ± 0.50	4.10 ± 0.65

Table 4.5: Performance comparison (error rate, %) on CDnet2014 video dataset [460] of other models to DCGAN-SSL for different numbers of labeled subsets per class

Examples	1000	500	100
TSVM [220]	10.78 ± 6.6	12.08 ± 0.19	12.96 ± 0.67
CatGAN [423]	1.73 ± 0.82	2.32 ± 0.16	2.4 ± 0.86
VAT [323]	1.45 ± 0.2	1.60 ± 0.80	2.23 ± 0.05
CNN [238]	1.26 ± 0.7	1.5 ± 0.11	1.8 ± 0.82
DCGAN-SSL	1.04 ± 0.41	1.15 ± 0.07	1.52 ± 0.33

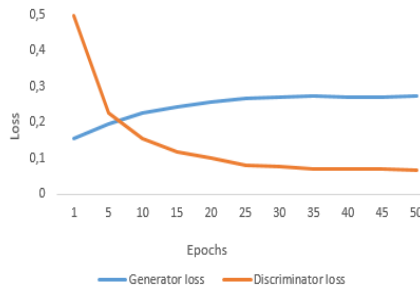
**Figure 4.6:** Discriminator and generator losses using CDnet2014 dataset [460].

Figure 4.6 presents the DCGAN-SSL discriminator and generator losses using CDnet2014 dataset [460]. It can be seen that the generator loss increases. This means that the DCGAN-SSL model successfully generates images that the discriminator fails to discriminate.

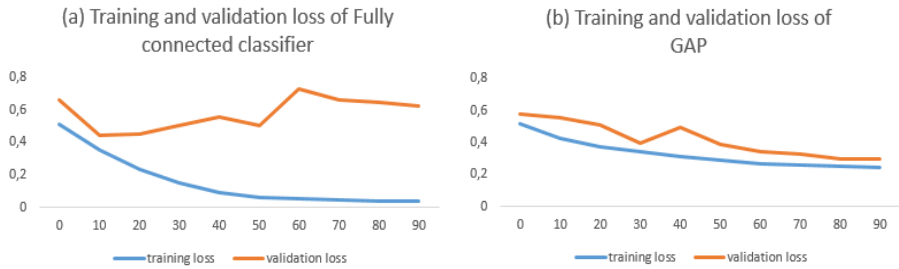


Figure 4.7: (a) Training and testing loss of Fully Connected (FC) classifier using CDnet2014 [460] (b) Training and testing loss of GAP using CDnet2014 [460].

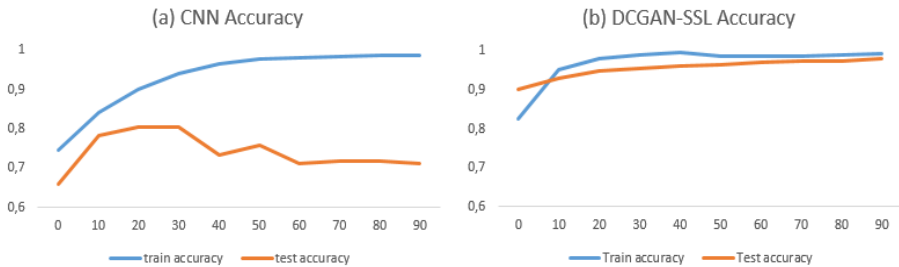


Figure 4.8: (a) Standard CNN model accuracy using CDnet2014 dataset [460] (b) DCGAN-SSL accuracy using CDnet2014 dataset [460].

As can be observed in Figure 4.7 (a), with a standard FC classifier, the training loss continues to decrease and the validation loss decreases to a point and begins to increase in Epoch 50, a sure sign of overfitting. There's a big gap between the training and the validation curves. The model strongly overfits. In Figure 4.7 (b) the training and validation loss decreases to a point of stability with a small gap between the two final loss values. The validation loss of GAP stagnates around 0.3. The model doesn't overfit as much as in the previous case. The GAP has a lower validation test by the 40th epoch than the FC model. More specifically, GAP demonstrated a decrease in validation loss over the FC classifier by around 3%.

It can be observed from Figure 4.8 that the replacement of FC layers in classical CNNs with GAP in DCGAN-SSL can boost performance with a smaller labeled training size. The training accuracy of the standard CNN achieves high values but it overfits regarding the validation dataset. These results illustrate that by adding GAP instead of a stack of FC layers on top of the feature maps of DCGAN-SSL, the performance of CNN is increased without requiring additional training data. One advantage of GAP compared to the FC layers is that it enforces correspondences between feature maps and categories, which makes it more native to the convolution structure. Additionally, there is no parameter to optimize in the GAP, so overfitting is prevented at this layer. As can be seen from the graphs above, the DCGAN-SSL model has a higher validation accuracy than the CNN and a lower training accuracy, but this is obviously due to over-fitting being reduced in the DCGAN-SSL model.

4.6 Conclusion

In this chapter, a new DCGAN-SSL approach for classifying moving objects extracted from video sequences is proposed. It combines the strengths of the generative models and the semi-supervised learning framework and trains jointly supervised and unsupervised models. Motivated by the ability of the DCGAN discriminator to classify data well, we have suggested to use DCGAN discriminator to extract deep features and then categorize extracted images. Then, we compared the DCGAN-SSL classifier with CNN as well as some existing models on real videos of CDnet2014 dataset [460] and VIRAT video dataset [333]. Combining the supervised loss with the unsupervised loss of DCGAN permitted us to achieve a test accuracy of 0.93 with only 120 samples. The experimental results show that our DeepDC based on DCGAN-SSL outperforms the CNN model and other three traditional methods. In addition, our model is easy to use and it enables us to apply it into various applications including facial expression, activity recognition, etc. As a future work, we suggest improving generative models and applying large amounts of unlabeled samples to reach high efficiency on supervised methods and to contribute to the efficient use of very small amounts of labeled samples. These semi-supervised methods are successful and thus allow creating an accurate supervised learning model that requires the collection of a sufficient amount of labeled samples, which is relatively expensive.

Chapter 5

DCGAN-based data augmentation for face identification in images and video applications

This chapter presents a Deep Convolutional Generative Adversarial Net (DCGAN) able to increase training data for better face recognition performance. Our proposal is based on FaceNet model to extract high-quality features from faces, called embeddings, that can be used to train our face identification system. Additionally, a DCGAN-based data augmentation method is used to reduce overfitting while maintaining the robustness of a classifier and then improving the accuracy of face recognition. Results on two datasets show the pertinence of the proposed approach. This chapter is based on the paper presented in the International Symposium on Visual Computing (ISVC 2020) [31].

Contents

5.1	Motivation	98
5.2	Face detection	99
5.3	Image data augmentation techniques	100
5.4	DCGANs	102
5.5	FaceNet	103
5.5.1	FaceNet model	103
5.5.2	Triplet Loss	105
5.6	Proposed approach	106
5.7	Experimental results	109
5.7.1	Description of the datasets	109
5.7.2	Qualitative and quantitative evaluation	110
5.8	Conclusion	120

5.1 Motivation

In recent years, face recognition has been an important area of research in the field of computer vision and pattern recognition. Compared to conventional machine learning methods, deep learning algorithms have shown promising performances in terms of image recognition accuracy and processing speed. In particular, Convolutional Neural Network (CNN) shows the highest performance in image recognition field [247] [435]. Compared to conventional algorithms for face recognition, CNNs are trained in a data-driven way. Additionally, CNNs models combine both feature extraction and classification into one framework. A CNN model incorporates mainly convolutional layers, pooling layers, fully-connected layers, as well as an input and an output layer. Based on its weight-sharing capability, local connectivity and sub-sampling, CNNs are better able to extract features and make a significant progress in face recognition. CNN's performance is affected by the network structure, its parameters, and the number of training images. However, in these approaches, a classification layer [433] [439] is trained on a set of known face identities. Then, an intermediate bottleneck layer is used to represent the input as a signature vector with reduced dimensionality in order to generalize recognition over all identities used in training. But, these approaches have many downsides, such as their indirect nature and their inefficiency. Typically, a face representation is very large (thousands of dimensions) with the use of the bottleneck layer and cannot generalize well to new identities. To reduce the dimensionality, Sun et al. [433] applied PCA to achieve only a linear transformation that can simply be learned in a network layer. Unlike these methods FaceNet directly trains a 128-D compact embeddings using a loss function based on an online triplet mining method based on LMNN [227].

Despite the exceptional efficiency of CNNs in image recognition, it still faces difficult challenges, such as the difficulty of getting enough training images, because CNN requires a large amount of data for learning. Generally, a large volume of training samples is useful to achieve high recognition accuracy. Because a CNN has a powerful learning ability, it needs different facial views for each subject. However, it is sometimes difficult to provide sufficient number of images for CNN training. Obtaining such a dataset for one class is not only time consuming, but also impractical. Moreover, it is often necessary to train samples of faces in different lighting, poses, and occlusion situations. To deal with the issue of lack of samples, an efficient method is suggested is the data augmentation technique. The principal purpose of data augmentation is to increase the size of the training dataset in order to achieve high accuracy [105], robustness of a classifier and decrease over-fitting. The increase in data size is achieved by applying label-preserving transformations to transform the accessible images. Generally, the advanced approaches used to increase the number of images in the database are affine transformation (cropping, inversion, rotation, translation....), the brightness changes of the image, adding Gaussian noise and the application of various filtering operations. A small number of samples in the dataset may not be appropriate in a complex scenes because the most discriminant features of each elements are probably different.

In this chapter, we propose a face recognition approach based on FaceNet model with DCGANs data augmentation, that can mitigate the issues discussed above. Like other recent works that use deep networks [432] [439], our method is a completely data-driven approach which allows to learn the representation directly from face pixels. Instead of using engi-

neering features, we get an extended dataset of labelled faces using DCGANs to achieve the convenient in variances to illumination, pose and other variational situations. FaceNet contains two different deep network architectures that have been recently used with great success in the field of computer vision. Both are deep convolutional networks [481] [137]. The first architecture is based on the Zeiler & Fergus [115] model which contains several interleaved convolutional layers, max-pooling layers, non-linear activations and local response normalisations. The second deep network architecture used the Inception model of Szegedy et al. [435] which was recently used as the winning approach for ImageNet 2014 [435]. These networks employ mixed layers that run various convolutional and pooling layers in parallel and combine their responses. The proposed face recognition model uses a complex system of multiple steps based on FaceNet model, that combines the output of a deep convolutional network and an SVM for classification. We evaluate our approach based on DCGANs using images from CDnet2014 dataset [460], LFW dataset [198], VGGFace2 dataset [85], Choke-Point dataset [469] and Youtube face dataset [468]. In addition, we investigate the effect of generated images quality on face identification. In the field of automatic image generation, DCGAN is known to generate high quality images [360]. Convolution between generator and discriminator leads to obtain the high-performance image generator. Relatively little approach based on *feature extraction* has been proposed for face recognition task.

Our contributions can be summarized as follows:

1. A robust combination of FaceNet model for feature extraction and DCGANs for data augmentation that allows us to be more robust on face recognition task. The proposed approach aims to identify people extracted from video sequences through their faces and to increase face image dataset by generating synthetic human faces which efficiently expand the training data, handling the effects of misalignment, lighting variations, partial occlusions, variations in pose and to avoid over-fitting during training.
2. We evaluate the impact of the combination of these two methods in face recognition performance on two image face datasets that are the LFW dataset [198] and the VGGFace2 dataset [85] as well as two video face datasets, the ChokePoint dataset [469] and the Youtube face dataset [468].

The rest of this chapter is as follows. In Section 5.2, we give a review of the different face detection methods. Section 5.3 provides a brief review of data augmentation techniques. Section 5.5 discusses the architecture of FaceNet model based on Triplet loss. The construction of the new face recognition method which combines FaceNet model and DCGAN data augmentation is described in Section 5.6. Comparative results obtained on both static and video face datasets are given in Section 5.7. Finally, the conclusion is shown in Section 5.8.

5.2 Face detection

In recent years, human detection beings in a video-surveillance sequence has attracted more and more attention because of its large area of applications. In the literature, many methods are presented for detecting humans. To recognize a human being, it is crucial to detect his face as the most representative part of the human body. Face detection and alignment are

crucial for numerous applications, such as the recognition of faces and the analysis of facial expressions. Nevertheless, these tasks are challenging due to the wide visual variations of faces, such as various occlusions, poses and extreme illumination variations. In 2004, Viola and Jones [457] suggested a cascade face detector using Haar-Like features and AdaBoost to train cascaded classifiers which are more performant and effective in real time applications. However, some subsequent studies, [41] [348] [354], show that it cannot maintain a continuous competitiveness in real word applications with greater visual variations of human faces that influence the visual coherence of faces. In addition to the cascading structure, the deformable part models (DPM) presented in [208] [324] [473] for face detection can reach outstanding performance. However, they require high computation cost and may generally need costly annotation in the training step. Face detection has been enhanced with the development of robust feature extraction techniques such as HOG (histogram of oriented gradients) [18] and LBP (local binary patterns) [221] and their variants. Subsequently, deep CNNs are used for face detection. Yang et al. [385] presented deep neural networks (DNNs) for face detection. But, this algorithm is consumely in time under real conditions. In 2015, Li et al. [274] proposed a CNN cascade face detectors with several resolutions. The authors also tried to improve the quality of bounding boxes through a calibration network. An OpenCV-based deep learning face detector was used in [182] to locate faces in images using a pre-trained OpenCV and Dlib models based on the Single-Shot-Detector (SSD) with a ResNet-10 network. However, the Dlib face detector lacks some of the difficult examples (partial occlusions, silhouettes, etc.). This makes the model less efficient on other benchmarks. To deal with these limitations, in this chapter, we propose to apply a cascaded face landmark detector called Multi-task CNN in the preprocessing module [226], which consists of three layers of deep convolutional networks, to detect and align the sample set.

The main face detection works, including our proposal are shown in Table 5.1.

Table 5.1: The main face detection works

<i>Authors/Date</i>	<i>Features</i>
<i>Conventional methods</i>	
Viola and Jones (2004) [457]	cascade face detector using Haar-Like features and AdaBoost
Mathias et al. (2014) [324], Yan et al. (2014) [208], Zhu et al. (2012) [473]	deformable part models (DPM)
Albiol et al. (2008) [18]	HOG (Histogram of Oriented Gradients)
JoChang-yeon (2008) [221]	LBP (Local Binary Patterns)
<i>Deep Learning methods</i>	
Yang et al. (2015) [385]	Deep CNNs
Li et al. (2015) [274]	CNN cascade face detectors
Vikas Gupta (2018) [182]	OpenCV-based deep learning face detector, Dlib models based on SSD, ResNet-10
Proposed approach	Multi-task CNN (MTCNN) [226]

5.3 Image data augmentation techniques

A number of data augmentation techniques have been suggested to enlarge the image data artificially. The main data augmentation works are summarized in Table 5.2. This section

reviews the existing works that analyzed the conventional and in-depth data augmentation techniques. Vincent et al. [456] added Gaussian noise, Masking noise and Salt-and-pepper noise to obtain more noisy images to train Stacked Denoising Autoencoders. Howard et al. [156] applied cropping and flipping to extend the training dataset, which is broadly used in the subsequent studies [114] [115] [483] even combined the original face image and its mirror to improve the performance of face recognition based representation. To synthesize a great number of corrupted images, Xie et al. [207] added Gaussian noise to images. Wu et al. [359] introduced a number of techniques, such as color casting that modifies the intensities of the RGB channels, the vignetting which decreases the image's brightness towards the periphery compared to the image center and the distortion of the lens which is a deviation from rectilinear projection caused by the camera lens.

Data increasing methods specific to face images were also presented. Jiang et al. [111] suggested an effective 3-D reconstruction method for generating face images with various poses, expressions and illuminations. Mohammadzade and Hatzinakos [184] suggested an expression subspace projection method that by projecting an image with an arbitrary expression into the expression subspace, new expression images are generated for each person. A more accurate estimation of the within-subject variability was achieved. Seyyedsalehi et al. [396] use a nonlinear manifold separator neural network (NMSNN) to extract identity and expression manifolds for face images. However, most of them are complex and attached to constrained environments. As shown in last studies [25] [247] [359], data increasing methods assist the trained Deep CNN model implemented with a robust generalization capability to detect invisible but similar noise patterns in the training data. A landmark perturbation technique is suggested by Shan et al. [404] to extend the training dataset to solve the problem of misalignment. However, they only disturbed the eye coordinates of each face image with eight neighbors. O'Donnell and Bruce [340] have demonstrated that hairstyle is an extremely significant feature to recognize faces. But, criminals generally use various hairstyle masks to hide their hairs or other disguises when they commit crimes. Many people, particularly woman change their hair styles regularly. Additionally, because of different hair styles with different fringes occlude the forehead or even part of eyes, which would influence the performance of face recognition. Lv et al. [206] proposed five data augmentation techniques devoted for face recognition, covering landmark perturbation and four synthesis methods (hairstyles, poses, glasses, illuminations).

The traditional data augmentation methods, such as rotation, flip and translation, are severely limited, which cannot achieve good generalization results. To improve the recognition accuracy of facial images, in this chapter, a new method of data augmentation based on DCGANs is proposed for face recognition. By using images generated by DCGANs and images in the original dataset as input, this model can achieve the top average identification accuracy.

Experiments on face identification show that DCGAN can generate data that approximate to real images, which can be used to (1) provide a larger data set for the training of large neural networks, and improve the performance of the recognition model through highly discriminating image generation technology; (2) reduce the cost of data collection; (3) enhance the diversity of data and the generalization ability of the recognition models. Generated faces by DCGANs extend the training dataset, which mitigates the effects of misalignment, illu-

mination variations, changes in pose and partial occlusions, as well as the overfitting during training.

Table 5.2: The main data augmentation works

<i>Authors/Date</i>	<i>Methods</i>
Vincent et al. (2010) [456]	Gaussian noise, Masking noise and Salt-and-pepper noise
Howard et al. (2013) [156]	cropping and flipping
Xie et al. (2012) [207]	Gaussian noise
Wu et al. (2015) [359]	color casting, vignetting, the distortion of the lens
Jiang et al. (2005) [111]	3-D reconstruction method
Mohammadzade (2013) [184]	expression subspace projection method
Seyyedsalehi et al. (2014) [396]	nonlinear manifold separator neural network (NMSNN)
Lv et al. (2017) [206]	A landmark perturbation and four synthesis methods (hairstyles, poses, glasses, illuminations)
Proposed approach	DCGANs

5.4 DCGANs

In this thesis, we exploit the ability of DCGAN’s generator [360] to artificially generate more facial images similar to the original faces in the training dataset. The idea is to simultaneously train two adversarial neural networks. The first network is a discriminator that learns to determine whether the sample comes from the data distribution. The other is a generative model that aims to generate “fake” images that attempts to fool the discriminator. After several stages of training, the optimization will achieve a stable point where the discriminator will be difficult to discern whether the data was “fake” or not.

Mathematically, the training process of DCGANs can be seen as a minimax game. The generator $G(z)$ takes a sampled input z from a uniform distribution. The discriminator $D(\cdot)$ takes x as input, being either images, from the database or output of generator $G(z)$. During training, the discriminator tries to distinguish between selected database and $G(z)$, i.e. attempts to maximize, $\log(D(x)) + \log(1 - D(G(z)))$. Simultaneously, the generator attempts to deceive the discriminator by minimizing $\log(1 - D(G(z)))$. The optimization will achieve a point of equilibrium where the discriminator is unable to distinguish between x and $G(z)$, after multiple steps of training. The generator takes a noise vector as input, followed by a fully connected layer containing 8192 neurons and resized to the dimension of $4 \times 4 \times 1024$. Next, 4 transposed convolutional layers are used with stride of 2 and padding resulting in a reduction of the channels and an up-sampling of the features by factor of 2. The output image’s size is of $64 \times 64 \times 3$. The input image with dimension $64 \times 64 \times 3$ is transmitted through 4 consecutive convolutional layers with the final output of dimension $4 \times 4 \times 512$. The last fully connected layer produces final output classes by a softmax activation function. It generates the probability that x is sampled from the true distribution. The final classification is done by attributing the class with the highest probability to a given image.

5.5 FaceNet

5.5.1 FaceNet model

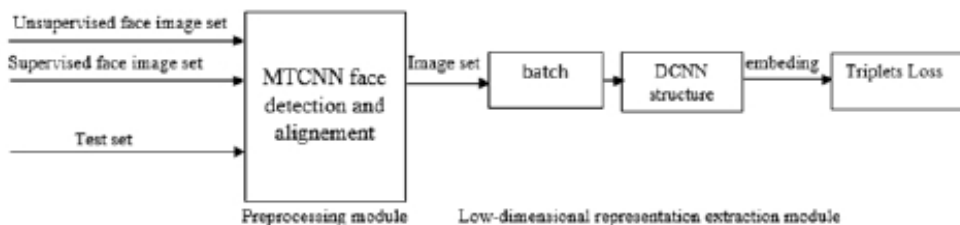
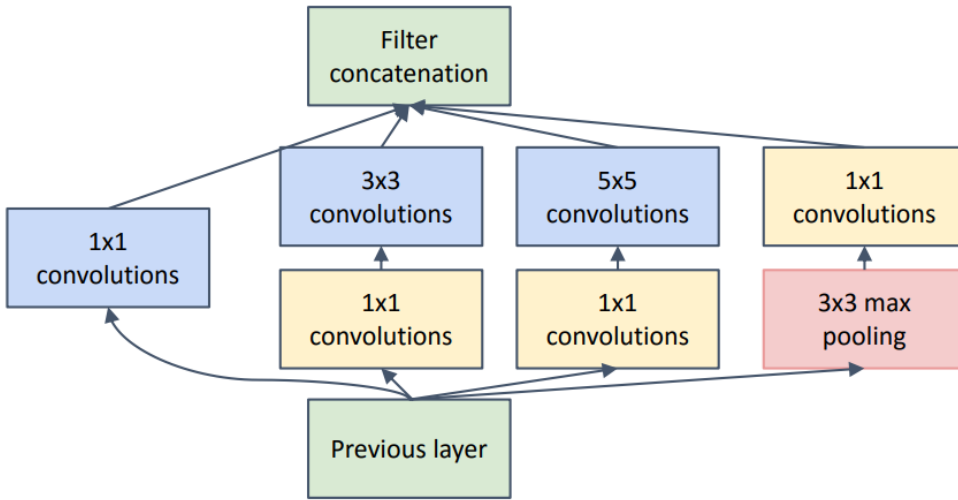


Figure 5.1: FaceNet model architecture, which consists of two modules : preprocessing and extraction of low-dimensional representation. The preprocessing module uses the Multi-task Cascaded CNN (MTCNN) [226] for the detection and alignment of samples. The low-dimensional representation extraction module consists of a batch input layer and a deep CNN which is followed by L2 normalization that provides the embedding of face. Next, the triplet loss is applied during training.

FaceNet is a deep CNN implemented since 2015 by Google researchers to successfully deal with the difficulties in face detection and verification. Figure 5.1 presents the structure of the FaceNet architecture. The FaceNet network transforms the face image into 128-D Euclidean space. Therefore, FaceNet model aims to identify the similarities and differences on the image data set when is trained for triplet loss. The encodings with 128-D are used to cluster faces in an efficient way. FaceNet encodings are used as feature vectors for face recognition and verification, after creating the vector space. The distances for the "same" images would be much closer than the non similar random images. FaceNet [393] generally consists of two different basic architectures based on CNNs. The first category adds $1 \times 1 \times d$ convolutional layers between the standard convolutional layers of the Zeiler & Fergus [115] architecture, then gets a 22 layers NN1 model. The second category consists of Inception models based on GoogLeNet [435]. Figure 5.2 represents the network structure of an Inception module. It contains 4 branches from the left to right. It employs convolution with 1×1 filters as well as 3×3 and 5×5 filters and a 3×3 max pooling layer. Each branch uses a 1×1 convolution to achieve time complexity reduction. FaceNet model is a deep CNN trained via a triplet loss technique that allows vectors for the same identity to become more similar (smaller distance), while vectors for different identities should become less similar (larger distance). The key advantage of this model is that it uses DCNN trained to directly optimize the embedding itself rather than extracting them from an intermediate bottleneck layer in other deep learning approaches. The most important part of FaceNet model is the end-to-end learning of the entire system.



<https://www.cs.toronto.edu>

Figure 5.2: Inception module.

Therefore, the triplet loss is applied to directly reflect what we want to achieve in face verification, recognition and classification. The triplet loss [434] assists to project all faces with a similar identity onto a single point in the embedding space. But, the triplet loss attempts to impose a margin between every pair of faces from one identity to all others, which enables the faces for one subject to live on a manifold, whereas still imposing a distance and therefore discrimination to other subjects. The subsequent section describes this triplet loss and how it can be learned effectively on a large scale.

In this chapter, the same pre-processing is performed for all training and testing samples. First, face detection is carried out using MTCNN algorithm [226], then five key points are located for each sample. A similar transformation is made depending to the position of the keypoints that are located. Finally, all faces are cropped into images of a certain dimensions. After applying face alignment and cropping, the input face is passed through the deep neural network. The FaceNet deep learning model is applied to extract the 128-d feature vector called embeddings that quantify each face in an image. The computation of the face embedding lies in the training process, including the input data into the network and the triplet loss technique. The neural network calculates the encoding vector of size 128 for each face. Then, adjusts the network weights through the triplet loss function. The objective of the triplet loss is to push the 128-d encodings of two images of the "same" identity (Anchor and Positive) closer to each other. At the same time, it tries to pull the encodings of the negative image farther apart. In this way, the network learns to quantify faces and output highly discriminating and robust embeddings adapted to face recognition. The model allows computing encodings for each face and finally an SVM classifier is trained on top of the face embeddings.

5.5.2 Triplet Loss



<https://www.computer.org/csdl>

Figure 5.3: The triplet Loss.

To train a face recognition model, each input batch of data contains three images : the anchor, the positive image and the negative image.

Triplet loss process is applied to minimize the distance between the anchor and the sample if the sample is positive and signifies the same person; also, to maximize the distance between the encodings of images (the anchor and the negative sample), which signifies a different identity. Thus, triplet loss is one of the best ways to learn a good 128-D encoding for each face. The anchor image represents the reference image that we took from that dataset to calculate the triplet loss.

Let's $f(x) \in \mathbb{R}^d$, where $f(x)$ represents the embedding which maps an image x into a d -dimensional Euclidean space. This encoding is restrained to live on the d -dimensional hypersphere, *i.e.* $\|f(x)\|_2 = 1$. This loss is motivated in [227] in the context of the nearest neighbor clustering. An anchor image x_i^a of one person must be closer to all other images x_i^p of the "same" person (positive) than to any image x_i^n of a different person (negative). This is visualized in Figure 5.3.

Thus, this equation must be satisfied

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \quad (5.1)$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \tau \quad (5.2)$$

where α is a margin that is imposed between positive and negative examples. Let N be the cardinality of τ , the set of all possible triplets in the training set. The generation of all possible triplets results in numerous triplets that are satisfied in a easy way (*i.e.* reach the restriction in Eq. 5.1). The process minimizes a loss on triplets that measures triplet satisfaction. These triplets, as they would still be transmitted across the network, they would lead to a slower convergence and they would not contribute to the training. It is important to choose hard triplets, that are active and can thus improve the model.

5.6 Proposed approach

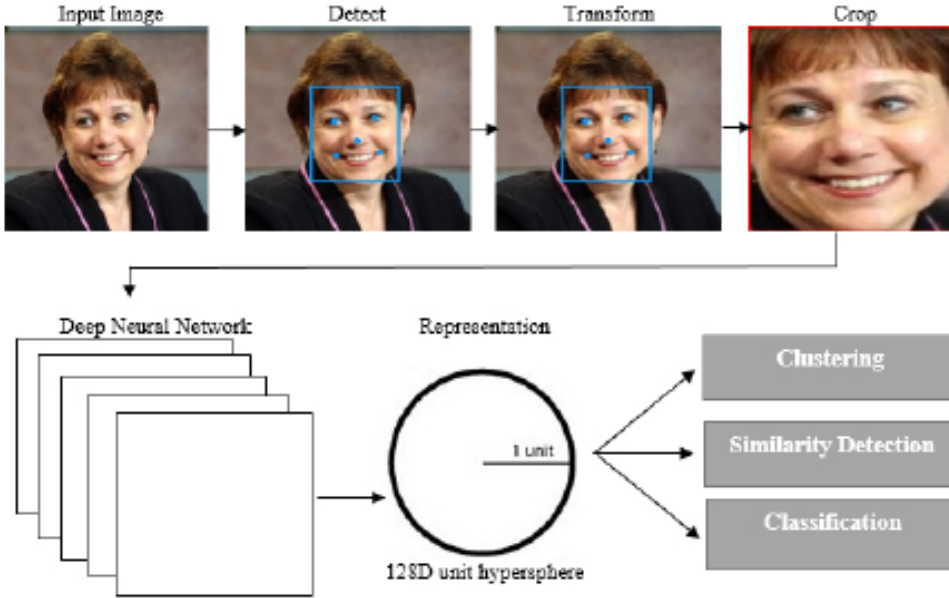


Figure 5.4: An overview of the proposed face recognition pipeline. The CNN feature extractor generates 128-dimensional facial embeddings.

Our proposed approach is based on DCGANs for data augmentation and FaceNet model for face classification. Sometimes, we have to use a small number of dataset for CNN. With more images, CNN would perform better. In this chapter, we propose a data augmentation technique based on the application of DCGANs to tackle the problem of samples collection difficulties. Therefore, we add similar images which are produced by DCGANs to the original training face dataset to increase the data. The images generated by the generator of DCGANs cannot be distinguished by the discriminator of CNN if they are real or not. Thus, we assume that generated images by DCGANs have similar CNN features, function like similar images and help a small number of dataset. The proposed approach is carried out in the following steps: 1) DCGANs are trained for each class. 2) Images are generated by trained DCGAN models. And 3) Generated images are added to the original dataset. We compare our proposed data augmentation method based on DCGANs to add generated images for human identification through their faces with the work of Pei et al. [492] who use standard data augmentation methods (rotation, translation, Gaussian noise addition and brightness change). Additionally, we present an approach based on both OpenCv and deep learning for face verification (is this the claimed person?), identification (who is this person?) and clustering (finding common people between these faces). Our system includes several important steps, fast and accurate face detection, face processing and cropping by computing facial landmarks using MTCNN face detector and 128-d face encodings extraction by

applying FaceNet deep learning model, training a face recognition model on the embeddings and finally applying SVM to classify and recognize faces in images and video streams. Our face recognition pipeline is presented in Figure 5.4. Firstly, the proposed face identification system takes an input image or video frame, detects the location of a face in the image using a cascaded face landmark detector called Multi-task CNN in the pre-processing module [226], which contains three layers of deep convolutional networks to detect and align the sample set. Multi-task CNN is used for joint face detection and alignment, which combines these two tasks based on an unified cascaded CNNs by multi-task learning. It consists of the following three steps. In the first step, a candidate windows is quickly obtained thanks to a shallow CNN. The windows are refined allowing the rejection of a great amount of non-faces windows through a more complex CNN. Then, a more effective CNN is trained to refine the result and estimate the positions of facial landmarks. This multi-task learning can significantly enhance the performance of the algorithm. The face is pre-processed and aligned by computing facial landmarks based on MTCNN as presented in Figure 5.5. Face alignment is the process of identifying the geometric structure of the faces and tries to perform a canonical face alignment based on rotation, translation and scale. It has been shown that the face alignment increases the precision of face recognition. The principal goal of MTCNN is to re-scale the corresponding face image to a range of different sizes called an image pyramid. The Proposal Network (P-Net) is a fully convolutional network used to get the candidate windows and their corresponding bounding box regression vectors. The non-maximum suppression (NMS) is applied to merge highly overlapped regions and refine the output. This can be considered as a two-class classification problem that can be solved by the cross-entropy loss.

$$L_i^{det} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i))) \quad (5.3)$$

$$y_i^{det} \in \{0, 1\} \quad (5.4)$$

In Eq 5.3, y_i is the input image, p_i is the probability produced by the network y_i that represents a sample being a face. Eq 5.4 indicates the label of ground-truth.

The Refine Network (R-Net) aims to filter the bounding boxes to eliminate a large number of rough facial windows. This objective can be considered as a problem of bounding box regression, also overcome by the Euclidean loss.

$$L_i^{box} = \|\hat{y}_i^{box} - y_i^{box}\|_2^2 \quad (5.5)$$

In Eq 5.5, \hat{y}_i^{box} and y_i^{box} represent the regression target calculated by the network and the corresponding real coordinate, respectively.

The Output Network (O-Net) identifies face areas with more supervision. It outputs five facial features' coordinates. The positions of the faces are obtained to realize face detection and alignment. The detection of facial features belongs to the regression problem that minimizes the defined Euclidean loss:

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \quad (5.6)$$

In Eq.5.6, $\hat{y}_i^{landmark}$ and $y_i^{landmark}$ are the coordinates of the predicted facial landmarks with the trained network and the actual condition for the i -th input image, respectively. The facial landmarks correspond to five feature points on the face, which cover the left mouth,

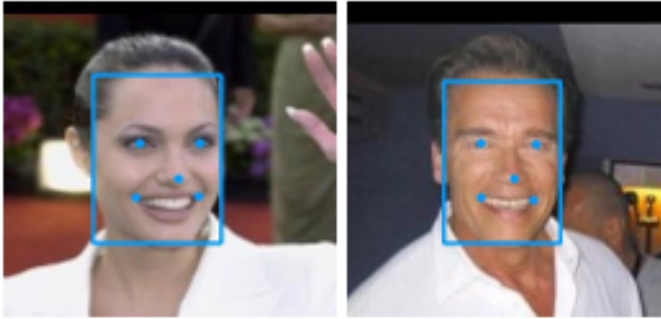


Figure 5.5: Face alignment using Multi-Task CNN, Facial Landmarks detection.

right mouth, nose, left eye as well as the right eye.

The next step introduces the encoding process using FaceNet [393], as presented in Section 5.5. An OpenCV Deep learning Torch embedding model is used to extract the encodings. Each face is represented by a DNN on a 128-d unit hypersphere. Our method uses deep CNN to learn the mapping from face images to an Euclidean space where the distances correspond to the face similarity measures. This model encodes a face image into a vector of 128-D. Compared to other face representations, this embedding has the benefit that a larger distance between two face embeddings signifies that the faces are probably of different people. Training of the network requires a face triplets, the face image of the target person, the test face image of the target person and the face image of a different person. This advantage facilitates similarity detection and classification compared to other face recognition methods where Euclidean distance between features is not important. OpenFace library [39] was used with pre-trained FaceNet model to train this DCNN. In this chapter, we propose to use FaceNet model, which can increase the accuracy of the CNN (VGG-16) model, almost halve the execution time, decrease the deep neural network training time and also improve the alignment process by removing a redundant face detection. Furthermore, we propose to train our model with a small version of the original FaceNet network nn4, called nn4.small2 as it reduces the number of parameters, has an input size of only 96x96 which considerably reduces the CPU requirements (285M FLOPS vs 1.6B for NN2). The nn4.small2 version not only reduces the input size, but also it does not use 5x5 convolutions in the higher layers because the receptive field is already too small. This version contains a structure similar to the FaceNet architecture, but with the removal of layers 4b, 4c and 4d and with smaller 5a and 5b layers. It consists of a mixture of regular pooling layers, convolutional layers and inception layers.

The final step of our face recognition model is to train a classifier on top of the embeddings previously generated from face dataset by using deep CNN. An Euclidean embedding is learned per image using a deep convolutional network. The network is trained in such a manner that the squared L2 distances between the embeddings corresponds to face similarity. Faces of the "same" person have close distances and faces of different people have great distances. Once this encoding has been generated, the distance between the two encodings is thresholded for face verification. Finally, we use Support Vector Machine (SVM) for face classification task.

5.7 Experimental results

The experiments were carried out on faces taken from the 'person' category of the CDnet2014 dataset [460] obtained in Chapter 4, LFW dataset [198] and VGGFace2 dataset [85] as well as two video face datasets, ChokePoint dataset [469] and Youtube face dataset [468].

5.7.1 Description of the datasets

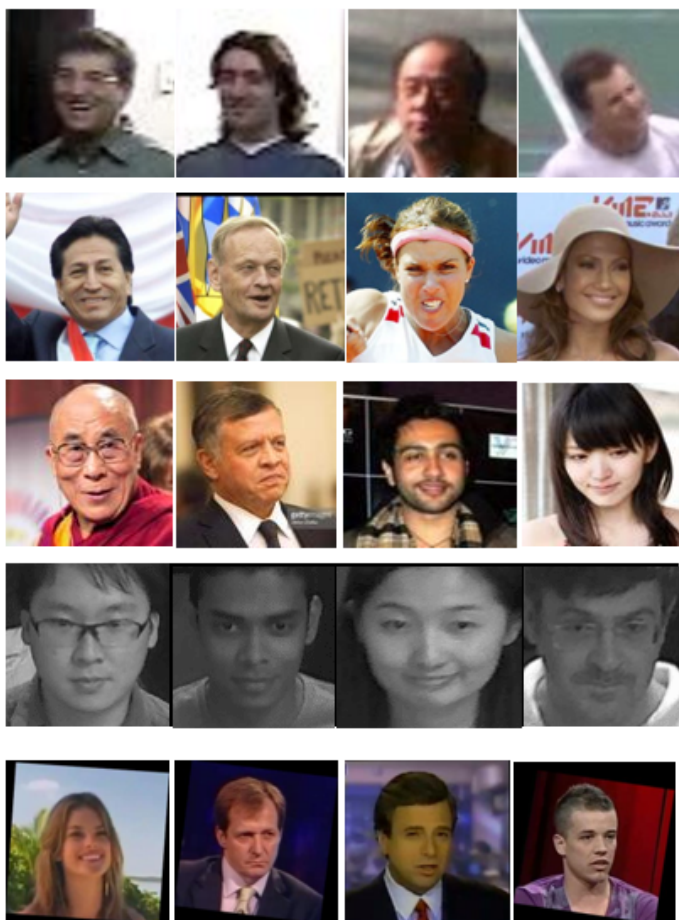


Figure 5.6: In each row some examples of representative images/frames of datasets used in this chapter: (a) Faces extracted from CDnet2014 dataset [460] (b) LFW dataset [198] (b) VGGFace2 dataset [85] (c) ChokePoint dataset [469] (d) and YouTube faces [468].

- **Faces extracted from CDnet2014 dataset**

We collected 1000 facial ROIs from the 'person' class obtained from CDnet2014 [460] images of Chapter 4, corresponding to 10 identities with different illumination, pose and facial expressions, to train the proposed face recognition approach. There are approximately 100 faces captured for each identity belonging to subjects whose faces appear clearly.

- **Labeled Faces in the Wild (LFW) dataset**

The LFW dataset [198] is the standard benchmark for face verification and recognition. This dataset includes 13,233 facial images of 5,749 subjects. These images present several challenges related to face pose, expression, illumination, and partial occlusion. This dataset has a limitation is that only 1,680 identities out of a total of 5,749 subjects have more than one face image. A subset of the dataset consisting of 3137 images belonging to 62 subjects was used during the experiments, by selecting the subjects with 20 or more images.

- **VGGFace2 dataset**

The VGGFace2 dataset [85] includes 9000 identities. The distribution of faces for different subjects is varied, from 87 to 843, with a mean of 362 images for each subject. Because of time reason, we did not manage to run experiments on the whole dataset. In our experiments, we choose a subset from VGGFace2 dataset [85] by randomly selecting 20 subjects to evaluate the performance of our method. The selected subset includes 12 men and 8 women. The constructed VGG-based image set contains 7746 images.

- **ChokePoint dataset**

The ChokePoint video dataset [469] is designed for verification/identification experiments of people in real world surveillance situations using current techniques. Faces have variations in terms of pose, lighting, sharpness, as well as mis-alignment owing to the automatic localization/detection of faces. The ChokePoint video dataset consists of 25 identities (19 men and 6 women) in portal 1 and 29 identities (23 men and 6 women) in portal 2. We used portal 1 for our experiments.

- **Youtube face dataset**

YouTube Faces Database (YFD) [468] consists of 3425 videos of 1595 different people with an average of 2.15 videos per subject, with video clips ranging from 48 to 6070 frames. This dataset provides a set of videos and labels for subject recognition from videos. We evaluate our approach with 40 identities from Youtube face dataset [468].

5.7.2 Qualitative and quantitative evaluation

In our experiments, we start with recognizing faces in images and then move on to recognizing faces in video streams. We also use a label encoder which contains the name for the people our model can recognize. We filter weak detections and extract the face ROI to recognize faces in the image. The results of our experiments are carried out on different datasets.

- **Faces extracted from Change Detection dataset (CDnet2014)** To augment the diversity of the original images and reduce overfitting, we expand the original dataset through our proposed data augmentation technique based on DCGANs. We add 100, 250 and 500 generated images per one class of the CDnet2014 dataset [460]. Results summarized in Table 5.3 show that when adding 100 images per class, the accuracy can achieve 94.5%. In addition, after a period of collecting more data, the accuracy improves to 96.11%. We also compare our proposed face recognition approach based

Table 5.3: Face recognition accuracy with DCGAN data augmentation using the proposed method.

	Number of augmented samples per class			
	+0	+100	+250	+500
CDnet2014 dataset [460]	0.91	0.945	0.951	0.9611
LFW dataset [198]	0.64	0.781	0.895	0.9212
VGGFace2 dataset [85]	0.65	0.678	0.88	0.9583

on DCGAN data augmentation with standard data augmentation methods. Results reported in Table 5.4 demonstrate the superiority of DCGAN based data augmentation. Our proposed face recognition approach showed higher performance when applying DCGAN data augmentation over traditional data augmentation techniques. While standard data augmentation techniques result in accuracy above 92%, DCGAN data augmentation achieves an accuracy of 96.11%. Filter operations have relatively better performance compared to geometric transformations and brightness augmentation methods, but still are inferior to DCGAN-based data augmentation. Table 5.4 provides also evidence that FaceNet model results in improved accuracy than VGG-16 model. The combination of FaceNet for face recognition with DCGAN data augmentation outperforms the work of Pei et al. [492] based on VGG-16 network for face recognition and standard data augmentation methods (see Table 5.4). To prove the effectiveness of our method, which is based on the augmented training images, our approach is compared with traditional face recognition techniques such as PCA and LBPH. Compared with PCA and LBPH, our face recognition approach based on FaceNet model with DCGAN data augmentation can achieve 96.11%.

- **Labeled Faces in the Wild (LFW) dataset** In our experiments, we use DCGANs as data augmentation technique. Figure 5.7 and 5.8 show that by using DCGANs as data augmentation method, several quality images were produced. Various unrealistic images that could not be seen faces were generated. This emergence of many unrealistic images is caused by the lack of training images for DCGANs. Realistic and unrealistic images are picked up through our subjective assessment. The criterion was whether we could see them as human faces or not. As it can be seen, DCGANs are able to produce images that are similar to the original faces with a small modifications. Then, we add realistic images per one class on both datasets. We add 100, 250 and 500 generated images per one class for both LFW dataset [198] and VGGFace2 dataset [85].

Table 5.4: Recognition performance with different methods using 10 classes from CDnet2014 dataset [460].

PCA method	33.3%
LBPH method	42%
Geometric transformation and brightness augmentation method (CNN) [492]	76.67%
Filter operation augmentation method (CNN) [492]	91.37%
DCGANs augmentation method (CNN)	95.7%
Geometric transformation and brightness augmentation method (ours)	78.17%
Filter operation augmentation method (ours)	92.69%
DCGANs augmentation method (ours)	96.11%

**Figure 5.7:** Generated Images using DCGANs on LFW dataset [198].

Results reported in Table 5.3 show that the more training samples are used for fine-tuning, the higher the accuracy and performance of the model are. Results show that our method based on DCGANs data augmentation achieves an accuracy of 78.1% and 67.8% with LFW dataset [198] and VGGFace2 dataset [85], respectively, when adding 100 samples per class. Furthermore, with adding 500 samples per class, the accuracy can achieve 92.12% in LFW dataset [198] and 95.83% in VGGFace2 dataset [85]. Our proposal is compared with two typical face recognition algorithms, namely, Principal Component Analysis (PCA) and Local Binary Patterns Histograms (LBPH). PCA is generally employed to reduce the dimensionality of datasets while maintaining the values which contribute most to variance. The covariance matrix is decomposed to obtain the main components of the data (i.e., eigenvector) and their corresponding eigenvalues. The LBPH face recognition method is based on the Local Binary Patterns (LBP), which is an efficient texture description method. The occurrences of the LBP codes are represented in a histogram for texture classification. The classification is then carried out by calculating the similarity between histograms. Additionally, we compare our proposed face recognition method based on FaceNet model and DCGANs data augmentation with the work of Pei et al. [492] based on standard data augmentation techniques (Translation, rotation, inversion, brightness change, and Gaussian noise addition) and VGG-16 model for face identification. Our method is also compared with the combination of CNN for face recognition and DCGANs model for data augmentation. The recognition results of the tested methods using 62 classes from LFW

dataset [198] are reported in Table 5.5. As it can be seen, the proposed approach achieves better performance than traditional face recognition methods (PCA/LBPH) using only a small amount of samples. Furthermore, with DCGANs data augmentation, our face recognition approach based on FaceNet model outperforms all others standard techniques used in the work of Pei et al. [492] for data augmentation. DCGANs data augmentation is used to enlarge the number of original training samples for fine-tuning the proposed model. DCGANs allow generating human faces that are similar to the faces in the original dataset with small modifications. In addition, the generated faces appear fairly like realistic images with small noise. DCGANs has the ability to complete the details of the face and generate human faces that appear authentic and similar to the original face, with very low resolution human face images as input. The proposed approach efficiently expands the training data, mitigating the effects of misalignment, pose variations, lighting changes and over-fitting. Table 5.5 shows also that FaceNet model is more effective than CNNs.

Table 5.5: Recognition performance with different methods using 62 classes from LFW dataset [198].

PCA method	50%
LBPH method	37%
Geometric transformation and brightness augmentation method (CNN) [492]	69.21%
Filter operation augmentation method (CNN) [492]	81.94%
DCGANs augmentation method (CNN)	84.26%
Geometric transformation and brightness augmentation method (ours)	70.94%
Filter operation augmentation method (ours)	86.57%
DCGANs augmentation method (ours)	92.12%

- **VGGFace2 dataset**

Table 5.6 summarizes the results of the comparison of the proposed approach with PCA, LBPH and the use of CNNs models for face recognition. To prove the efficiency of FaceNet model in the face recognition task, we compare it with the work of Pei et al. [492] based on CNN for face recognition with data augmentation through geometric transformation, image brightness change, and the application of different filter operations. These methods are evaluated on 20 classes from VGGFace2 dataset [85]. The best accuracy is on bold. Once again, our face recognition model achieved the best accuracy compared to all the methods, even when using the CNN model which extracts highly robust and discriminant features.

Experimental results show that our approach based on DCGANs data augmentation and face classification using FaceNet gives better results than conventional data augmentation methods proposed in the work of Pei et al. [492] and face classification using VGG-16 model. Results also demonstrate that our face recognition approach based on FaceNet model outperforms both PCA and LBPH as well as the use of CNNs for face classification. FaceNet model quantitatively outperforms the mentioned techniques, making it a serious candidate for the face recognition task in computer vision applications. The proposed method based on FaceNet model gives more accuracy than using



Figure 5.8: Generated Images using DCGANs on VGGFace2 dataset [85].

VGG-16 model with a difference of 7.86 % using LFW dataset [198] and 8.93% using VGGFace2 dataset [85]. Table 5.6 also show that the application of filter operations as data augmentation methods gives higher performance than using geometric transformation and brightness augmentation methods (cropping, rotation, translation,.....).

- ChokePoint dataset & Youtube face dataset** To evaluate the impact of data augmentation based on DCGANs in face recognition, we add 100 images for each class of the ChokePoint video dataset [469] and Youtube face dataset [468]. Results are summarized in Table 5.7. As It can be seen, the face recognition accuracy is higher with a difference of 0.47% and 0.12 %, respectively, when adding only 100 images per class. Table 5.8 and 5.9 show the recognition accuracies of the proposed approach, PCA, LBPH, the use of CNN for face classification and the methods described in Pei et al. [492] using ChokePoint video dataset [469] and Youtube face dataset [468], respectively. Results are summarized in Figure 5.9. Experimental results have shown that the proposed method, which combines FaceNet model for face recognition and DCGANs for data augmentation outperforms the other techniques with 95.18 % and 99.65 %, respectively. Data augmentation using DCGANs gives higher accuracy than the use of standard data augmentation methods (geometric transformation and brightness augmentation method, filter operation method..) in the work of Pei et al. [492]. We also compare the proposed approach with CNN for face recognition and we conclude that our method based on FaceNet model achieves higher accuracy with a difference of 12.11% and 0.52%, respectively. In addition, our proposed approach outperforms both PCA and LBPH standard techniques.

Table 5.6: Recognition performance with different methods using 20 classes from VGGFace2 dataset [85].

PCA method	40%
LBPH method	32%
Geometric transformation and brightness augmentation method [492] (CNN)	81.85%
Filter operation augmentation method (CNN) [492]	85.23%
DCGANs augmentation method (CNN)	86.90%
Geometric transformation and brightness augmentation method (ours)	82.39%
Filter operation augmentation method (ours)	87.20%
DCGANs augmentation method (ours)	95.83%

Table 5.7: Face recognition accuracy with DCGAN data augmentation using the proposed method in video datasets.

	Number of augmented samples per class	
	+0	+100
ChokePoint dataset [469]	94.71%	95.18%
Youtube face dataset [468]	99.53%	99.65%

Table 5.8: Recognition performance with different methods using portal 1 from ChokePoint dataset [469].

PCA method	50.4%
LBPH method	34.09%
Geometric transformation and brightness augmentation method (CNN) [492]	72.66%
Filter operation augmentation method (CNN) [492]	70.83%
DCGANs augmentation method (CNN)	83.07%
Geometric transformation and brightness augmentation method (ours)	75.26%
Filter operation augmentation method (ours)	82.18%
DCGANs augmentation method (ours)	95.18%

Table 5.9: Recognition performance with different methods using Youtube face dataset [468] (40 classes).

PCA method	60.2%
LBPH method	50.6%
Geometric transformation and brightness augmentation method (CNN) [492]	81.16%
Filter operation augmentation method (CNN) [492]	97.7%
DCGANs augmentation method (CNN)	99.08%
Geometric transformation and brightness augmentation method (ours)	86.62%
Filter operation augmentation method (ours)	98.64%
DCGANs augmentation method (ours)	99.65%

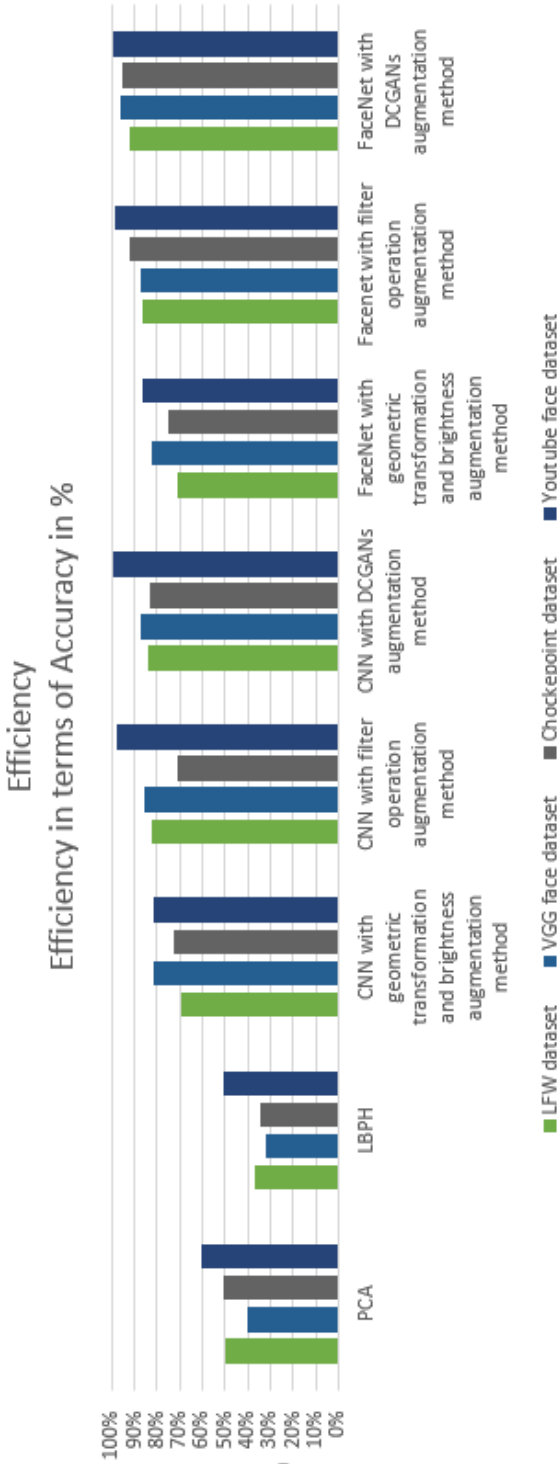


Figure 5.9: Recognition accuracy using LFW dataset [198], VGGFace2 dataset [85], ChokePoint dataset [469] and Youtube face dataset [468].

Figure 5.10 shows that the face of *Andre Agassi* from LFW dataset [198] is recognized with 72.32%. However, the confidence is higher with data augmentation based on DCGANs achieving 77.08%. In Figure 5.11 (a), we can see that the face prediction has only 50.71% confidence when using the ChokePoint dataset [469]; however, this confidence is higher when applying data augmentation with DCGANs achieving 91.63% as shown in Figure 5.11 (b). It is the same in Figure 5.11 (c) & Figure 5.11 (d) with an increase of 1.8% adding only 100 images per class. The experimental results on LFW database [198], VGGFace2 database [85], ChokePoint face database [469] and Youtube face database [468] show that the proposed approach has improved the face recognition performance with better recognition results.

The running time of an algorithm depends on the size of the input images. One of the ways for obtaining a fixed-dimensional input image is to resize the face in the bounding box to 96x96 pixels. A potential problem is that faces can look in different directions. To handle this, we propose to reduce the size of the input space by pre-processing the faces with alignment. We align faces by finding the locations of the eyes and nose with a cascaded face landmark detector called Multi-task CNN, then we perform an affine transformation to make the eyes and nose appear in roughly the same place. Additionally, to further improve the computational time, we propose to train our model with a small version of Facenet called nn4.small2 as it reduces the number of parameters. This improvement allows for negligible alignment time and reduced neural network execution time. The almost halved execution time is the result of using a neural network that is smaller than the original FaceNet’s nn4 network with the idea that a small model will perform better.

To evaluate the classification execution time of CNN and our proposed face recognition method based on FaceNet model, we use the CPU Intel Core i7 7500u. We collected the elapsed CPU time for training and classifying. Table 5.10 and Table 5.11 show the results using ChokePoint dataset [469] and Youtube face dataset [468], respectively. As shown in Figure 5.12, for the same sample image, we tried both CNN and the proposed method based on FaceNet using CPU. Using ChokePoint video dataset [469], CNN takes about 8 mins for training, when FaceNet takes only 6 mins. For testing, CNN takes about 0.280 ms while FaceNet takes only 0.16 ms. Using the Youtube face dataset [468], CNN takes about 21 mins while FaceNet takes only 8 mins. For testing, CNN takes about 0.280 ms while FaceNet takes only 0.22318 ms. The proposed method based on FaceNet using nn4.small2 was able to run 2 times faster.

Table 5.10: Training and classifying execution time using Intel Core i7 Hardware using ChokePoint dataset [469].

Method	Training	Classifying an image (64*64)
CNN (VGG-16)	10 minutes	0.50 ms
Proposed approach based on FaceNet	6 minutes	0.16 ms

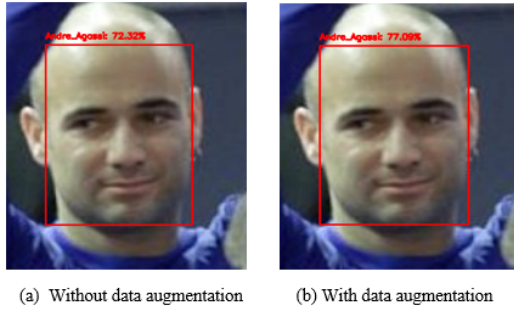
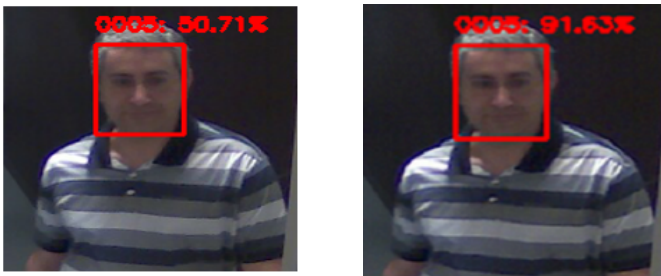
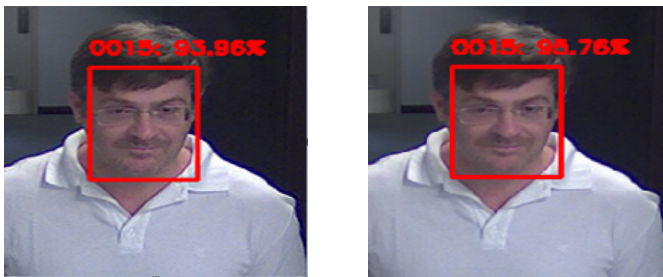


Figure 5.10: Face confidence using LFW dataset [198].



(a) Without data augmentation (b) With data augmentation



(c) Without data augmentation (d) With data augmentation

Figure 5.11: Face confidence using ChokePoint dataset [469].

Table 5.11: Training and classifying execution time using Intel Core i7 Hardware using Youtube face dataset [468].

Method	Training	Classifying an image (64*64)
CNN (VGG-16)	21 minutes	0.28 ms
Proposed approach based on FaceNet	8 minutes	0.22318 ms

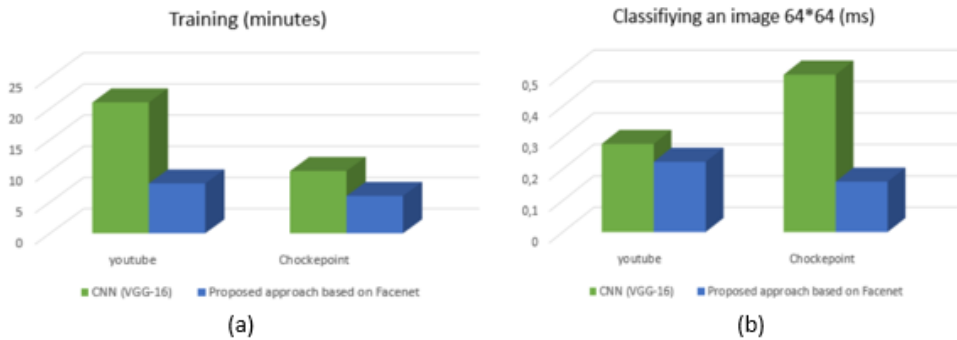


Figure 5.12: (a) Training time using ChokePoint dataset [469] and Youtube face dataset [468]. (b) Classification time using ChokePoint dataset [469] and Youtube face dataset [468].

5.8 Conclusion

In summary, a new face recognition approach based on FaceNet model with DCGAN data augmentation is proposed. It combines the strengths of deep neural networks for image generation and recognition. Then, we compared the proposed method with CNN on LFW and VGGFace2 datasets as well as on videos from ChokePoint and Youtube face datasets. The experimental results have shown that DCGAN data augmentation for FaceNet face recognition outperforms CNN and that DCGAN data augmentation produces faces similar to the original ones in the training dataset which improves the recognition performance. The proposed approach described in this chapter only reaches the highest accuracy when the number of training images is huge. Moreover, the use of alignment for pre-processing and smaller neural network models reduces the FaceNet execution time. Compared with CNN, our proposal allows reducing the execution time while maintaining a high recognition accuracy.

Chapter 6

Conclusions

In this thesis, we set out to improve background subtraction by focusing on detecting foreground objects without using additional image processing or background learning. Background subtraction is a crucial task in many computer vision applications including surveillance devices in public spaces, traffic monitoring and industrial machine vision. We focused on developing robust descriptor to deal with illumination changes, noise, and produces a good segmentation masks. In addition, we present an efficient approach able to classify the extracted objects in a semi-supervised way. Both labeled and unlabeled data are used to train a classifier. This type of classifier takes a tiny portion of labeled data and a much larger amount of unlabeled data. The goal is to combine these sources of data to train a Deep Convolutional Neural Network (DCNN) to learn an inferred function capable of mapping a new image to its desirable outcome. Finally, we propose a face recognition approach to identify the extracted people. We combine both data augmentation using DCGANs and face recognition using FaceNet model to improve the recognition accuracy, robustness of a classifier and decrease overfitting. The key contributions of the thesis are as follows.

- **A Deep Detector Classifier (DeepDC).** A DeepDC is introduced in this thesis. It exploits the strength of an anomaly discovery framework called DeepSphere, which leverages both deep autoencoders and hypersphere learning methods, having the capability of isolating anomaly pollution and reconstructing normal behaviors, to detect foreground objects. We adapt DeepSphere to the context of foreground-background separation. The new DeepDC produces a good segmentation results without additional image processing and background learning. It is also tolerant to illumination changes as RPCA [84] is whereas DeepPBM [149] is not and robust to noise and the dynamic nature of the background as DeepPBM [149] is whereas RPCA [84] is not. Experimental results show that DeepSphere is robust under scenes ranging from dynamic background to changing illuminations.
- **A Semi-supervised classification approach.** DeepDC also allows to classify the extracted images. We propose a semi-supervised learning method called DCGAN-SSL to classify the extracted objects based on the discriminator of DCGAN. The discrimi-

nator is transformed into a multi-class classifier, which takes a tiny portion of labeled data and a much larger amount of unlabeled data. The experiments carried out on videos show that our classification approach which uses a generative model is more efficient in terms of accuracy than CNNs.

- **A face recognition method based on FaceNet model.** The last contribution of the thesis is the proposed face recognition approach for people identification. Our proposal is able to extract 128-dimensional face embedding to represent the face, to handle the large face representation which cannot generalize well to new identities. It trains a model to create embeddings directly, rather than extracting them from an intermediate layer. Our method uses FaceNet model that not only increases the efficiency in terms of time and memory consumption, but also can improve the recognition accuracy. Experiments conducted on challenging image and video datasets show that this approach is more efficient in terms of time and recognition accuracy than previous methods. We extend our previous approach by proposing a DCGANs data augmentation technique for increasing the size of the dataset by generating more realistic images similar to the original faces. By fusing DCGANs data augmentation and FaceNet model for face recognition, the derived method allows improving the face recognition accuracy compared to standard data augmentation methods.

6.1 Limitations

- Our proposed DeepDC model is designed to detect moving objects in videos. The drawback of our DeepSphere-based algorithm is that the potential interactions among spatial and temporal dimensions are neglected. Furthermore, DeepSphere does not take into account the structural information of the graph, since it mainly allows extracting information in the time dimension. Therefore, it is only applicable to dynamic graphs. In addition, considering the adjacency matrices as input signifies that its input dimension is equal to the square of the number of vertices. This limits the scalability of the proposed approach.
- Our proposed DCGAN used for objects classification and data augmentation demonstrates that adversarial networks learn good image representations. The major drawback is that there are still some forms of model instabilities. We have observed that these models need a long time to train. Further work is required to address this instability.
- In this thesis, we use FaceNet model to handle deep metric learning issues and generate feature embeddings. FaceNet is based on a triplet model to minimize the distance between samples of the same class and maximize the distance between samples of various classes. However, the triplet network contains a large number of parameters, which requires sampling a large number of triplets from the training data in order to learn a robust model. However, sampling all possible triplets from the training data can quickly become difficult, where most of those samples may generate small costs that result in slow convergence. The vast majority of the training samples will produce gradients with magnitudes that are close to zero which can compromise the training

convergence of the triplet model. For example, suppose that a training set contains N samples, therefore the set of triplets has complexity size $O(N^3)$, as a result its training is impractical even for data sets of limited sizes.

6.2 Future works

- We plan to extend our DeepDC framework based on DeepSphere to exploit multiple data sources. We propose to investigate the possibility to process large scale and dynamic streaming data to detect foreground objects, to allow a more robust background subtraction task. We also intend to extend DeepDC to include the potential interactions among spatial and temporal dimensions as well as the structural information of the graph to further improve the detection results.
- In this thesis, we propose to use DCGANs for data augmentation and object classification. However, DCGANs are trained with long time. We intend to reduce the model size and the computational requirements to reduce the time of DCGANs. The long-training time can be also improved by varying curriculum learning and mining offline.
- As future work, we will try to better understand the error cases, further improve the model, and also more reduce the model size. We propose to extend our face recognition approach based on FaceNet model by developing a sampling technique that stochastically sub-samples the set of triplets and allows using sufficient samples to ensure that a some fraction of the hard negatives and positives are available for training. Moreover, taking into account the great complexity involved in the search of hard positive and negative examples, we propose to implement a training procedure to train samples with high gradient magnitudes: the loss functions that consider the overall structure of the embedding space will be incorporated. We propose to extend our approach by exploring the overall embedding structure and the hard negative/positive mining. We will extend the triplet loss used in our approach with a global loss that supposes that the distribution of distances between anchor and negative samples and anchor and positive samples is based on a Gaussian distribution. Additionally, we propose to compare our proposal with recent models like RetinaFace [126] and on a more extensive database such as WiderFace [12].

Appendix A

Notations and Symbols

α_θ	encoder
β_ϕ	decoder
X	input data
z	internal layer, random noise vector
\hat{X}_k	original data
a	centroid of a hypersphere
r	radius of a hypersphere
$\{\chi_k, k = 1, \dots, m\}$	historical observation samples; training data
$\{\chi_k, k > m\}$	unseen test data
χ	sample case, tensor
X_t	series of matrices; series of graphs
h_t	internal states
w_t	weight at timestep t
z_k	encodings, embedded representations, data points
ξ	outlier penalty, slack variables
Φ	objective function, error function
Ψ	reconstruction error
χ_k	output of LSTM decoder
η_k	case-wise weights
$\eta \{d_k, r\}$	heuristic function
F	overall objective function
λ	compromise parameter
Θ	set of parameters
$\Delta(\chi_k)$	reconstruction difference
$G(t)$	dynamic graph
V	vertex set
E	edge set
$x(t)$	mapping function
e_{ij}	edge

$x_{ij}(t)$	time series
TP	true positive
FP	false positive
FN	false negative
FN	false negative
G	generator network
D	discriminator network
$X_{fake}, G(z)$	output of the generator
$f(x)$	activations of an intermediate layer, embedding
x	input of the discriminator
FC	Fully Connected
BN	Batch Normalization
$ReLU$	Rectified Linear Unit
P_{model}	class probability
N	number of classes, cardinality of τ
K	number of real classes
$K + 1$	additional class
$L_{supervised}$	supervised loss
(X, Y)	labeled points
X_0	rest of the unlabeled data
I	number of total iterations
$p_g(z)$	noise prior
$p_d(x)$	data generating distribution
x_i^a	anchor image
x_i^p	positive exemplar
x_i^n	negative exemplar
α	margin
τ	set of all possible triplets in the training set
x_i^{det}	cross-entropy loss
p_i	probability produced by the network
y_i	a sample being a face
y_i^{det}	label of ground-truth
L_i^{box}	bounding box regression objective
\hat{y}_i^{box}	regression target
y_i^{box}	real coordinate
$L_i^{landmark}$	regression problem
$\hat{y}_i^{landmark}$	coordinates of the predicted facial landmarks
$y_i^{landmark}$	actual condition of the i -th input image

Appendix B

List of Publications

This dissertation has led to the following communications:

Journal Papers

- Ammar S., Bouwmans T., Zaghden N. and Neji M. “Deep Detector Classifier (DeepDC) for moving objects segmentation and classification in video surveillance”. IET Image Processing, 2020 (published).

Book chapters

- Ammar S., Bouwmans T., Zaghden N. and Neji M. “From Moving Objects Detection to Classification and Recognition : A Review for Smart Environments”. Chapter on the handbook “Towards Smart World: Homes to Cities using Internet of Things”, 2020 (published)

Conferences

- Ammar S., Zaghden N. and Neji M. “A Framework for People Re-Identification in Multi-Camera Surveillance Systems”. In the Proceedings of the 14th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA), Vilamoura, Portugal, October, 2017. (published)
- Ammar S., Zaghden N. and Neji M. “An Effective Approach Based on a Subset of Skeleton Joints for Two-Person Interaction Recognition”. In the Proceedings of the

23rd Iberoamerican Congress on Pattern Recognition (CIARP), Madrid, Spain, December, 2018. (published)

- Ammar S. and Bouwmans T., Zaghden N. and Neji M. “Moving Objects Segmentation Based on DeepSphere in Video Surveillance”. In the Proceedings of the International Symposium on Visual Computing (ISVC), California, USA (oral presentation), October, 2019. (published)
- Ammar S., Bouwmans T., Zaghden N. and Neji M. “Towards an Effective Approach for Face Recognition with DCGANs Data Augmentation”. In the Proceedings of the International Symposium on Visual Computing (ISVC), California, USA, (oral presentation) October, 2020.(published)

Bibliography

- [1] <http://www.cvg.reading.ac.uk/PETS2001>. 25, 26
- [2] <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>. 25
- [3] <https://groups.inf.ed.ac.uk/f4k/>. 25
- [4] <http://underwaterchangedetection.eu/Videos.html>. 25
- [5] http://www.svcl.ucsd.edu/projects/background_subtraction/. 25, 26
- [6] <https://aimagelab.ing.unimore.it/vssn06/>. 25, 26
- [7] <http://cvrr.ucsd.edu/aton/shadow/index.html>. 26
- [8] <http://www.dis.uniroma1.it/~labrococo/MAR/index.htm>. 26
- [9] <https://computervisiononline.com/dataset/1105138686>. 29, 39, 40
- [10] https://www.gti.ssr.upm.es/data/Vehicle_database.html. 29
- [11] <https://sites.google.com/site/qinzoucn/documents>. 29
- [12] <http://shuoyang1213.me/WIDERFACE/>. 123
- [13] Torch. www.torch.ch. Accessed: 2015-03-11. 7
- [14] The database of faces. *ATT Laboratories Cambridge*, 2002. 39, 40, 41
- [15] <http://www.cvg.reading.ac.uk/PETS2006/data.html>, 2006. 25
- [16] Lankershim boulevard dataset. <http://ngsim-community.org/>, 2007. 26
- [17] Use neovision2 project. <http://ilab.usc.edu/neo2/>, 2007. 26
- [18] Albiol A., Monzo D., Martin A., and Sastre J. Face recognition using HOG-EBGM. *Publisher, City*, 2008. 100
- [19] Jalali A., Mallipeddi R., and Lee M. Sensitive deep convolutional neural network for face recognition at large standoffs with small dataset. *Expert Systems With Applications*, pages 304–315, 2017. 45
- [20] T. Aach, L. Dumbgen, R. Mester, and D. Toth. Bayesian illumination-invariant motion detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 640–643, 2001. 6

- [21] A. Abd Rahman, M. A. Noah, Safar R. S., and Kamarudin N. Human face recognition : An eigenfaces approach. *IntelSys*, pages 42–47, 2014. 30, 39
- [22] M. Abdullah, M. Wazzan, and Bo-Saeed S. Optimizing face recognition using PCA. *arXiv*, 2012. 30, 39
- [23] M. Abdulrahman, Y.G. Dambatta, Muhammad A.S., and Mamat M. Face recognition using eigenface and discrete wavelet transform. *International Conference on Advances in Engineering and Technology*, 2014. 31, 39
- [24] Z. Abidin and A. Harjoko. A neural network based facial expression recognition using Fisherface. *International Journal of Computer Applications*, 2012. 30, 39
- [25] T. Ahonen, E. Rahtu, V. Ojansivu, and Heikkila J. Recognition of blurred faces using local phase quantization. *International Conference on Pattern Recognition*, pages 1–4, 2008. 101
- [26] A. Alahi, Ortiz R., and F. Vandergheynst. Freak : Fast retina keypoint. *IEEE CVPR*, pages 510–517, 2012. 35, 40
- [27] I. Ali, J. Mille, and L. Tougne. Space-time spectral model for object detection in dynamic textured background. In *Pattern Recognition Letters*, pages 1710–1716, 2012. 2
- [28] I. Ali, J. Mille, and L. Tougne. Adding a rigid motion model to foreground detection: Application to moving object detection in rivers. In *Pattern Analysis and Applications*, pages 1–20, 2013. 2
- [29] S. Ammar, T. Bouwmans, N. Zaghden, and Neji M. Moving objects segmentation based on DeepSphere in video surveillance. In *International Symposium on Visual Computing (ISVC)*, pages 307–319, 2019. 22, 26, 47
- [30] S. Ammar, T. Bouwmans, N. Zaghden, and Neji. M. From moving objects detection to classification and recognition : A review for smart environments. In *Towards Smart World: Homes to Cities using Internet of Things*, 2020. 17
- [31] S. Ammar, T. Bouwmans, N. Zaghden, and M. Neji. Towards an effective approach for face recognition with DCGANs data augmentation. *International Symposium on Visual Computing, ISVC 2020*, 2020. 97
- [32] S. Ammar, Zaghden N., and Neji M. A framework for people re-identification in multi-camera surveillance systems. *Advances in neural information processing systems*, pages 319–322, 2017. 17, 27, 29, 35, 41
- [33] S. Ammar, Zaghden N., and Neji M. An effective approach based on a subset of skeleton joints for two-person interaction recognition. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2018*, 2018. 27
- [34] S. Ammar, Bouwmans T., Zaghden N., and Neji M. A deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance. *IET Image processing*, 2020. vi, 22, 26, 28, 29, 47, 70, 76, 79, 81
- [35] A. Andreopoulos and Tsotsos. J. K. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, pages 827–891, 2013. 44

- [36] M. Annalakshmi, S. Roomi, and A. Naveedh. A hybrid technique for gender classification with slbp and hog features. *Clust. Comput.*, pages 11–20, 2019. 37, 41
- [37] O. Arigbabu, S. Ahmad, and W. Adnan. Soft biometrics: Gender recognition from unconstrained face images using local feature descriptor. *arXiv*, 2017. 34, 40
- [38] A. Azeem. A survey : face recognition techniques under partial occlusion. *Int. Arab J. Inf. Technol.*, pages 1–10, 2014. 31, 39
- [39] Amos B., Ludwiczuk B., and Satyanarayanan M. Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science, Tech. Rep.*, pages 16–118, 2016. 108
- [40] Huang. G. B., H. Lee, and Learned-Miller.E. Learning hierarchical representations for face verification with convolutional deep belief networks. pages 2518–2525, 2012. 45
- [41] Yang B., Yan J., Lei Z., and Li S. Z. Aggregate channel features for multi-view face detection. *International Joint Conference on Biometrics.*, pages 1–8, 2014. 100
- [42] M. Babae, D. Dinh, and G. Rigoll. A deep convolutional neural network for background subtraction. *Preprint*, 2017. vi, 22, 23, 26, 63, 64, 65, 69, 70
- [43] M. Babae, D. Tung Dinh, and G. Rigoll. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, pages 63–649, 2018. 27, 29
- [44] F. E. Baf, T. Bouwmans, and Vachon B. Foreground detection using the choquet integral. *IEEE WIAMIS*, pages 187–190, 2008. 19
- [45] F. E. Baf, T. Bouwmans, and B. Vachon. Type-2 Fuzzy Mixture of Gaussians Model : Application to background modeling. *ISVC*, pages 772–781, 2008. 18, 19, 25, 57, 61
- [46] F. E. Baf, T. Bouwmans, and B. Vachon. Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. *IEEE-Workshop OTCBVS*, pages 60–65, 2009. 18
- [47] F. El Baf, T. Bouwmans, and B. Vachon. A fuzzy approach for background subtraction. In *IEEE International Conference on Image Processing (ICIP)*, pages 2648–2651, 2008. 6
- [48] F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. *IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1729–1736, 2008. 18, 19, 25
- [49] O. Barkan, J. Weill, L. Wolf, and Aronowitz H. Fast high dimensional vector multiplication face recognition. *IEEE ICCV*, pages 1960–1967, 2013. 36, 41
- [50] O. Barnich and M.V. Droogenbroeck. Vibe: a powerful random technique to estimate the background in video sequences. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 945–948, 2009. 19, 20, 25
- [51] M.S. Bartlett, J.R. Movellan, and Sejnowski T.J. Face recognition by independent component analysis. *IEEE Trans. Neural Netw.*, pages 1450–1464, 2002. 30, 39

- [52] F. Bastien, R. Lamblin, P. and Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements, 2012. 7
- [53] C. Dy Bautista, R. O. Manalac, and M. Cordel. Convolutional neural network for vehicle detection in low resolution traffic videos. *TENCON*, 2016. 22, 23, 26
- [54] M. Baytas and Xiao C. Patient subtyping via time-aware LSTM networks. *SIGKDD*, pages 65–74, 2017. 50
- [55] P. N. Belhumeur, Hespanha J. P., and Kriegman D. J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, pages 711–720, 1997. 39, 40, 41
- [56] F. Bellakhddhar, K. Loukil, and Abid M. Face recognition approach using Gabor Wavelets, PCA and SVM. *IJCSI International Journal of Computer Science Issues*, pages 201–206, 2013. 37, 41
- [57] Y. Benezeth, P.-M. Jodoin, and B. Emile. Review and evaluation of commonly-implemented background subtraction algorithms. *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 1–4, 2008. 57, 61
- [58] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *In Neural Information Processing Systems*, 2019. 81
- [59] M. Bhuiyan. Towards Face Recognition Using Eigenface. *International Journal of Advanced Computer Science and Applications*, 2016. 30, 39
- [60] S. Bianco, G. Ciocca, and R. Schettini. How far can you get by combining change detection algorithms? *Computing Research Repository (CoRR)*, 2015. 4
- [61] B. Bo, L. Lillian, J. Howell, and J. Saul. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *In Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124, 2005. 29
- [62] M. Boninsegna and Bozzoli A. A tunable algorithm to update a reference image. *Signal Processing : Image Communication*, 2000. 23
- [63] K. Bonnen, B. Klare, and A.K. Jain. Component-based representation in automated face recognition. *IEEE Trans. Inf. Forensics Secur.*, pages 239–253, 2012. 33, 40
- [64] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *In Computer Science Review*, pages 31–66, 2014. 18
- [65] T. Bouwmans, N. Aybat, and Zahzah E. Handbook on robust low-rank and sparse matrix decomposition : Applications in image and video processing. *Taylor and Francis Group*, 2016. 18
- [66] T. Bouwmans and F. El Baf. Background modeling using mixture of gaussians for foreground detection, a survey. *Recent Patents Comput. Sci.*, pages 219–237, 2008. 57, 61

- [67] T. Bouwmans and F. El Baf. Modeling of dynamic backgrounds by type-2 Fuzzy Gaussians Mixture Models. *Journal of Basic and Applied Sciences*, pages 265–276, 2010. vi, 19, 25, 57, 60, 61, 62, 64, 65, 69, 70, 73, 76
- [68] T. Bouwmans, B. Hoferlin, F. Porikli, and A. Vacavant. Recent approaches in background modeling for video surveillance. *Handbook Background Modeling and Foreground Detection for Video Surveillance, Taylor and Francis Group*, 2014. 18
- [69] T. Bouwmans, B. Hoferlin, F. Porikli, and A. Vacavant. Traditional approaches in background modeling for videosurveillance. *Handbook Background Modeling and Foreground Detection for Video Surveillance, Taylor and Francis Group*, 2014. 18
- [70] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo. On the applications of robust PCA in image and video processing. *Proc. IEEE*, 2018. 21
- [71] T. Bouwmans, Z. Javed, M. Sultana, and S. Jung. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Netw.*, 2019. 23, 43
- [72] T. Bouwmans, S. Pal, A. Petrosino, and L. Maddalena. Background subtraction for visual surveillance: A fuzzy approach. *Handbook on Soft Computing for Video Surveillance, Taylor and Francis Group*, pages 103–139, 2012. 20
- [73] T. Bouwmans, F. Porikli, B. Hoferlin, and Vacavant A. Handbook on background modeling and foreground detection for video surveillance. *CRCPress, Taylor and Francis Group*, 2014. 18
- [74] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant. Background modeling and foreground detection for video surveillance. In *Chapman & Hall/CRC*, 2014. 2
- [75] T. Bouwmans, C. Silva, C. Marghes, S. Zitouni, H. Bhaskar, and C. Frélicot. On the role and the importance of features for background modeling and foreground detection. In *Computer Science Review*, 2016. 6, 18
- [76] T. Bouwmans and E.H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 2014. 18
- [77] M. Braham, S. Pierard, and M.V. Droogenbroeck. Semantic background subtraction. *IEEE International Conference on Image Processing, ICIP*, 2017. 24, 26, 43
- [78] M. Braham and M. Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, 2016. 7, 22, 26, 28, 29
- [79] S. Bucak and B. Günsel. Incremental subspace learning and generating sparse representations via non-negative matrix factorization. *Pattern Recognit.*, 2008. 20, 26
- [80] S. Bucak, B. Günsel, and O. Gursoy. Incremental non-negative matrix factorization for dynamic background modeling. *International Workshop on Pattern Recognition in Information Systems*, 2007. 20, 26

- [81] S. Calderara, Melli R., and Prati A. Reliable background suppression for complex scenes. *ACM Int. Workshop on Video Surveillance and Sensor Networks (VSSN)*, pages 211–214, 2006. 57, 61
- [82] M. Calonder, Lepetit V., M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF : Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1281–1298, 2011. 35, 40
- [83] L. Cament, F. Galdames, K. Bowyer, and Perez C. Face recognition under pose variation with local gabor features enhanced by active shape and statistical models. *Pattern Recognition*, 2015. 32, 39
- [84] E. Candes, X. Li, Y. Ma, and Wright J. Robust principal component analysis ? *Journal of ACM*, 2011. iii, iv, vi, 18, 21, 23, 26, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 76, 77, 121
- [85] Q. Cao, L. Shen, and Parkhi O. M. VGGFace2: A dataset for recognising face across pose and age. *International Conference on Automatic Face and Gesture Recognition*, 2018. iv, vii, 15, 42, 44, 99, 109, 110, 111, 112, 113, 114, 115, 116, 117
- [86] R. Caseiro, P. Martins, and J. Batista. Background modelling on tensor field for foreground segmentation. *BMVC*, 2010. 18, 19, 25
- [87] V. Cevher, Reddy D., M. Duarte, A. Sankaranarayanan, R. Chellappa, and R. Baraniuk. Compressive sensing for background subtraction. *ECCV*, 2008. 24, 26
- [88] M. Chacon-Muguia, S. Gonzalez-Duarte, and P. Vega. Simplified SOM-neural model for video segmentation of moving objects. *IJCNN*, pages 474–480, 2009. 22, 26
- [89] M. Chacon-Muguia, G. Ramirez-Alonso, and S. Gonzalez-Duarte. Improvement of a neural-fuzzy motion detection vision model for complex scenario conditions. *IJCNN*, 2013. 22, 26
- [90] M. Chacon-Murguia and S. Gonzalez-Duarte. An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems. *IEEE Transactions on Industrial Electronics (TIE)*, pages 3286–3298, 2012. 6
- [91] R. Chalapathy, Menon K., and Chawla S. Robust, deep and inductive anomaly detection. *European Conference On Machine Learning Principles and Practice of Knowledge Discovery, ECML PKDD*, 2017. v, 26, 54, 55
- [92] A. Chan, V. Mahadevan, and N. Vasconcelos. Generalized stauffer–grimson background subtraction for dynamic scenes. *Machine Vision and Applications (MVA)*, pages 751–766, 2011. 6
- [93] Y. Chan. Deep learning-based scene-awareness approach for intelligent change detection in videos. *Journal of Electronic Imaging*, 2019. 22
- [94] T. Chang, T. Ghandi, and M. Trivedi. Vision modules for a multi sensory bridge monitoring approach. *ITSC*, pages 971–976, 2004. 18, 23, 26
- [95] T. Chang, Ghandi T., and M. Trivedi. Computer vision for multi-sensory structural health monitoring system. *ITSC*, 2004. 23, 26

- [96] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 2009. 81
- [97] G. Chau and P. Rodriguez. Panning and jitter invariant incremental principal component pursuit for video background modeling. *International Workshop RSL-CV 2017 in conjunction with ICCV 2017*, 2017. 21, 26, 44
- [98] N. Chawla and G. Karakoulas. Learning from labeled and unlabeled data: an empirical study across techniques and domains. *J Artif Intell Res*, 2005. 81
- [99] X. Chen, S. Xiang, C.L. Liu, and C.H. Pan. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.*, pages 1797–1801, 2014. 28, 29
- [100] Z. Chen and T. Ellis. A self-adaptive gaussian mixture model. *Comput. Vis. Image Comput. CVIU*, pages 35–46, 2014. 20, 25
- [101] P. Chiranjeevi and S. Sengupta. Interval-valued model level fuzzy aggregation-based background subtraction. *IEEE Transactions on Cybernetics*, 2016. 19, 25
- [102] H. Cho, R. Roberts, B. Jung, Choi O., and Moon S. An efficient hybrid face recognition algorithm using pca and gabor wavelets. *Int. J. Adv. Robot. Syst.*, 2014. 37, 41
- [103] G. Cinar and J. Principe. Adaptive background estimation using an information theoretic cost for hidden state estimation. *IJCNN*, 2011. 18
- [104] P. Cinelli. Anomaly detection in surveillance videos using deep residual networks. *Master Thesis, Universidade de Rio de Janeiro*, 2017. 22, 26
- [105] D. Cire san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 44, 98
- [106] R. Collins, Lipton A., and Kanade T. A system for video surveillance and monitoring. *IEEE T-PAMI*, 2000. 18
- [107] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In: *Proc. Eur. Conf. Comput. Vis.*, 2004. 9
- [108] Z. Cui, Li W., Xu D., Shan S., and Chen X. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *CVPR*, pages 3554–3561, 2013. 30, 39
- [109] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht. A neural network approach to bayesian background modeling for video object segmentation. *IEEE Transactions on Neural Networks*, pages 1614–1627, 2007. 21, 26
- [110] Chen. D., X. Cao, Wen. F., and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification., *IEEE conference on computer vision and pattern recognition.*,, pages 3025–3032, 2013. 45
- [111] Jiang D., Hu Y., Yan S., Zhang L., Zhang H., and Gao W. Efficient 3D reconstruction for face recognition. *Pattern Recongition*, pages 787–798, 2005. 101, 102

- [112] Kang D. Nighttime face recognition at large standoff: Cross-distance and crossspectral matching. *Pattern Recognition*, pages 3750–3766, 2014. 45
- [113] Sun. Y.and Liang. D., Wang. X., and Tang. X. Deepid3 : Face recognition with very deep neural networks. *arXiv*, pages 44–51, 2015. 38, 45
- [114] Yi D., Lei Z., Liao S., and Li S. Z. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 36, 39, 42, 101
- [115] Zeiler M. D. and Fergus R. Visualizing and understanding convolutional networks. *European Conference on Computer Vision*,, pages 818–833, 2014. 99, 101, 103
- [116] A.M. Dai and Le Q. V. Semi-supervised sequence learning. *International Conference on Learning Representations, ICLR 2018*, 2018. 28, 29, 82
- [117] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov. Good semi supervised learning that requires a bad gan. *Neural Information Processing Systems*, 2017. 81
- [118] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 8
- [119] A. Darwich, P. Hebert, A. Bigand, and Y. Mohanna. Background Subtraction under Uncertainty using a Type-2 Fuzzy Set Gaussian Mixture Model. *International Conference on Computer Science, Computer Engineering, and Education Technologies, CSCEET 2017*, pages 1–6, April 2017. 20, 25
- [120] A. Darwich, P. Hebert, A. Bigand, and Y. Mohanna. Background Subtraction Based on a New Fuzzy Mixture of Gaussians for Moving Object Detection. *MDPI Journal of Imaging*, 2018. 20, 25
- [121] Libor Spacek’s Facial Image Database. Face94 database. <http://cswwww.essex.ac.uk/mv/allfaces/faces94.html>, 2015. 39, 40
- [122] R. Davies, L. Mihaylova, N. Pavlidis, and N. Eckley. The effect of recovery algorithms on compressive sensing background subtraction. *Sensor Data Fusion : Trends, Solutions, and Applications*, 2013. 24
- [123] J. Davis and M. Keck. A two-stage approach to person detection in thermal imagery. *In Workshop on Applications of Computer Vision, IEEE OTCBVS WS Series Bench*, page 364–369, 2005. 25
- [124] J. Davis and V. Sharma. Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding*, page 162–182, 2007. 25
- [125] Z. Dehai, Da D., Jin L., and Qing L. A PCA-based face recognition method by applying fast fourier transform in pre-processing. *ICMT*, 2013. 31
- [126] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild, 2019. 123
- [127] B. Devi, N. Veeranjanyulu, and Kishore K. A novel face recognition system based on combining eigenfaces with fisher faces using wavelets. *Procedia Comput. Sci.*, pages 44–51, 2010. 33

- [128] B. Dey and M. K. Kundu. Robust background subtraction for network surveillance in H.264 streaming video. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, pages 1695–1703, 2013. 6
- [129] K. Diederik and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 29, 50, 82
- [130] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Trans. Multimed.*, pages 2049–2058, 2015. 36, 41
- [131] J. Donahue, Krähenbühl J., and Darrel T. Adversarial feature learning. *International Conference on Learning Representations, ICLR2017*, 2017. 28, 29, 82
- [132] J. Dong and T. Lin. MarginGAN: Adversarial training in semi-supervised learning. *Neural Information Processing Systems*, 2019. 81
- [133] Y. Dong and G. DeSouza. Adaptive learning of multi-subspace for foreground detection under illumination changes. *Comput. Vis. Image Underst.*, 2011. 20, 26, 45
- [134] Y. Dong, T. Han, and G. DeSouza. Illumination invariant foreground detection using multi-subspace learning. *J. Int. Knowl. Based Intell. Eng. Syst.*, pages 31–41, 2010. 20, 26, 45
- [135] D. Driggs, S. Becker, and J. Boyd-Graberz. Tensor robust principal component analysis : Better recovery with atomic norm regularization. *Preprint*, 2019. 21, 26
- [136] G. Du, Su F., and Cai A. Face recognition using SURF features. *Pattern Recognition and Computer Vision*, 2009. 35, 40
- [137] Rumelhart D. E., Hinton G. E., and Williams R. J. Learning representations by back-propagating errors. *Nature*, pages 533–536, 1986. 99
- [138] F. El Baf and T. Bouwmans. Comparison of background subtraction methods for a multimedia learning space. *International Conference on Signal Processing and Multimedia*, 2007. 25
- [139] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767, 2000. 4, 18, 19, 25
- [140] T. Elguebaly and N. Bouguila. Finite asymmetric generalized gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 2013. 19, 25
- [141] J. Elson, J. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. *J. Mach. Learn. Res.*, pages 366–374, 2007. 29
- [142] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge 2011 (VOC2011) results. <http://www.pascalnetwork.org/challenges/VOC/voc2011/workshop/index.htm>. 29
- [143] D. Fan, Cao M., and C. Lv. An updating method of self-adaptive background for moving objects detection in video. *International Conference on Audio, Language and Image Processing*, pages 1497–1501, 2008. 23

- [144] J. Fan and T.W. Chow. Exactly robust kernel principal component analysis. *IEEE Trans. Neural Netw. Learn. Syst.*, 2019. 32, 39
- [145] W. Fan and N. Bouguila. Online variational learning of finite dirichlet mixture models. *Evolving Systems*, 2012. 19, 25
- [146] D. Farcas and T. Bouwmans. Background modeling via a supervised subspace learning. *IVPCV*, 2010. 18, 21, 26
- [147] D. Farcas, C. Marghes, and T. Bouwmans. Background subtraction via incremental maximum margin criterion : A discriminative approach. *Machine Vision and Applications*, pages 1083–1101, 2012. 18, 21, 26
- [148] S. Farfade, M. Saberian, and Li L. Multi-view face detection using deep convolutional neural networks. *ICMR*, 2015. 33
- [149] A. Farnoosh, B. Rezaei, and S. Ostadabbas. DeepPBM : deep probabilistic background model estimation from video sequences. *Preprint*, 2019. iii, iv, vi, 23, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 76, 77, 121
- [150] A. Faro, D. Giordano, and Spampinato C. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, pages 1398–1412, 2011. 19, 25
- [151] B. Farou, M. Kouahla, and H. Akdag. Efficient local monitoring approach for the task of background subtraction. *ng. Appl. Artif. Intell.*, pages 1–12, 2017. 20, 25
- [152] A. Fathima, S. Ajitha, V. Vaidehi, M. Hemalatha, R. Karthigaiveni, and R. Kumar. Hybrid approach for face recognition combining gabor wavelet and linear discriminant analysis. *CGVIS*, pages 220–225, 2015. 36, 41
- [153] Ferryman. <http://www.cvg.rdg.ac.uk/pets2006/data.html>, 2006. 26
- [154] J. Ferryman and A. Shahroki. An overview of the pets 2009 challenge. *Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009. 25, 26
- [155] Guo. G. and N. Zhang. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.*, 2019. 45, 46
- [156] Howard A. G. Some improvements on deep convolutional neural network based image classification. *arXiv:1312.5402*, 2013. 101, 102
- [157] B. Garcia-Garcia, F. Gallegos-Funes, and A. Rosales-Silva. A gaussian-median filter for moving objects segmentation applied for static scenarios. *IntelliSys*, pages 478–493, 2018. 19
- [158] Lopez P. Garcia-Lesta D., Brea V. and Cabello D. Impact of analog memories non-idealities on the performance of foreground detection algorithms. *IEEE International Symposium on Circuits and Systems, ISCAS*, pages 1–5, 2018. 20, 25
- [159] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3354–3361, 2012. 29

- [160] G. Gemignani and A. Rozza. A novel background subtraction approach based on multi-layered self organizing maps. *EEE International Conference on Image Processing*, 2015. 22, 26
- [161] A. Georghiadis, Belhumeur P., and Kriegman D. from few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence (PAMI)*, pages 643–660, 2001. 40, 41
- [162] M. Ghazi and Hazim. E. A comprehensive analysis of deep learning based representation for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–41, 2016. 45
- [163] A. Ghorbel, I. Tajouri, Aydi W., and Masmoudi N. A comparative study of GOM, uLBP, VLC and fractional Eigenfaces for face recognition. *International Image Processing, Applications and Systems (IPAS)*, pages 5–7, 2016. 30, 34, 39
- [164] P. Gil-Jimenez, S. Maldonado-Bascon, R. Gil-Pita, and H. GomezMoreno. Background pixel classification for motion detection in video image sequences. *International Work Conference on Artificial and Natural Neural Network, IWANN*, pages 718–725, 2003. 21, 26
- [165] J. H. Giraldo and T. Bouwmans. GraphBGS: Background subtraction via recovery of graph signals. *arXiv:2001.06404*, 2020. 24, 26
- [166] J. H. Giraldo and T. Bouwmans. Semi-supervised background subtraction of unseen videos: Minimization of the total variation of graph signals. *IEEE International Conference on Image Processing*, 2020. 24, 26
- [167] A.B. Godbehere, A. Matsukawa, and K. Goldberg. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. *American Control Conf. (ACC)*, pages 4305–4312, 2012. 57, 61
- [168] P. Golle. Machine learning attacks against the asirra captcha. *15th ACM conference on Computer and com-munications security*, pages 535–542, 2008. 27, 29
- [169] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and A. Courville. Generative adversarial networks. *ArXiv e-prints*, 2014. 81, 83
- [170] H. Gowda, G. Kumar, and Imran M. Multimodal biometric recognition system based on nonparametric classifiers. *Data Anal. Learn.*, pages 269–278, 2018. 30, 39
- [171] K. Goyal and J. Singhai. Review of background subtraction methods using gaussian mixture model for video surveillance systems. *Artif. Intell. Rev.*, pages 241–259, 2018. 20
- [172] Y. Goyat, T. Chateau, and L. Malaterre. Vehicle trajectories evaluation by static video sensors. *IEEE Int. Conf. on Intelligent Transportation Systems*, pages 864–869, 2006. 57, 61
- [173] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. pages 4663–4679, 2012. 22, 25, 26, 43
- [174] D. Graham and Allinson N. Web site of umist multi-view face database. *Image Engineering and Neural Computing Lab, UMIST, UK*, 1998. 39, 40, 41

- [175] P. Graszka. Median mixture model for background-foreground segmentation in video sequences. *WSCG*, 2014. 18, 25
- [176] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition and tracking. *IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance*, 2007. 29, 41
- [177] K. Gregor, Y. LeCun, and WI. Omnipress. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pages 399–406, 2010. 44
- [178] H. Guo, C. Qiu, and N. Vaswani. Practical reprocs for separating sparse and low-dimensional signal sequences from their sum. *Preprint*, 2013. 21, 22, 26
- [179] J.-M. Guo, C.-H. Hsia, Y.-F. Liu, and M.-H. Shih. Fast background subtraction based on a multilayer codebook model for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, pages 1809–1821, 2013. 6
- [180] L. Guo and M. Du. Student’s t-distribution mixture background model for efficient object detection. *ICSPCC*, pages 410–4144, 2012. 19, 25
- [181] R. Guo and H. Qi. Partially-sparse restricted boltzmann machine for background modeling and subtraction. *ICMLA*, pages 209–214, 2013. 22, 26
- [182] V. Gupta. Face detection opencv, dlib and deep learning. <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/>, 2018. 100
- [183] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection by robust PCA solved via a linearized alternating direction method. *International Conference on Image Analysis and Recognition, ICIAR*, 2012. 21
- [184] Mohammadzade H. and Hatzinakos D. Projection into expression subspaces for face recognition from single sample per person. *IEEE Transactions on Affective Computing*, pages 69–82, 2013. 101, 102
- [185] S. Hafez, Selim M., and Zayed H. 2D face recognition system based on selected gabor filters and linear discriminant analysis lda. *ArXiv*, 2015. 31, 39
- [186] T. Haines and T. Xiang. Background subtraction with dirichlet processes. *European Conference on Computer Vision, ECCV*, 2012. 19, 25
- [187] G. Han, J. Wang, and X. Ca. Improved visual background extractor using an adaptive distance threshold. *J. Electron. Imaging*, 2014. 20, 25
- [188] I. Haritaoglu, Harwood D., and Davis L. W4 : Real time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 80–85, 2000. 18
- [189] M. A. Hasnat, J. Bohn, and J. Milgram. von Mises-Fisher mixture model-based deep learning: application to face verification. *arXiv: 1706.04264*, 2017. 45
- [190] J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. *CVPR*, 2012. 21, 26

- [191] J. He, Zhang D., and L. Balzano. Iterative online subspace learning for robust image alignment. *IEEE Conference on Automatic Face and Gesture Recognition*, 2013. 44
- [192] J. He, D. Zhang, L. Balzano, and T. Tao. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing*, 2013. 44
- [193] K. He, Zhang. X., S. Ren, and J. Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 44
- [194] B. Heflin, W. Scheirer, and T.E. Boult. For your eyes only. *In Proceedings of the IEEE WACV*, pages 193–200, 2012. 34, 40
- [195] G. E. Hinton, Osindero. S., and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation.*, pages 1527–1554, 2006. 44
- [196] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The Pixel-Based Adaptive Segmenter. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 38–43, 2012. vi, 20, 25, 57, 60, 61, 62, 64, 65, 71, 72, 73
- [197] Z. Hu, T. Turki, N. Phan, and J. Wang. 3d atrous convolutional long short-term memory network for background subtraction. *IEEE Access*, 2018. 23
- [198] G. B. Huang, M. Mattarand Berg Tamara, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition.*, October 2008. iv, vii, 33, 34, 35, 36, 37, 39, 40, 41, 42, 44, 99, 109, 110, 111, 112, 113, 114, 116, 117, 118
- [199] L. Huang, Q. Chen, J. Lin, and Lin H. Block background subtraction method based on ViBe. *Appl. Mech. Mater.*, 2014. 20, 25
- [200] Y. Huang, K. Huang, D. Tao, L. Wang, T. Tan, and X. Li. Enhanced biological inspired model. *In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008. 9
- [201] Z. Huang, W.J. Li, J. Shang, J. Wang, and T. Zhang. Non-uniform patch based face recognition via 2D-DWT. *Image Vision Comput.*, pages 12–19, 2015. 31, 39
- [202] S.U. Hussain, T. Napoléon, and F. Jurie. Face recognition using local quantized patterns. *British Machine Vision Conference BMVC*, 2012. 33, 40
- [203] Deng J., Guo J., Xue N., and Zafeiriou S. Arcface: Additive angular margin loss for deep face recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, pages 4690–4699, 2019. 38, 42, 45
- [204] Deng J., Zhou Y., and Zafeiriou S. Marginal loss for deep face recognition. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. 45
- [205] Lindstrom. J, F. Lindgren, K. Åström, J. Holst, and U. Holst. Background and foreground modeling using an online em algorithm. *In IEEE International Workshop on Visual Surveillance at ECCV*, pages 9–16, 2006. 6

- [206] Lv J., Shao X., Huang J., Zhou X., and Zhou X. Data augmentation for face recognition. *Neurocomputing*, 2017. 101, 102
- [207] Xie J., Xu L., and Chen E. Image denoising and inpainting with deep neural networks. In: *Advances in Neural Information Processing Systems, NIPS'12*, pages 341–349, 2012. 101, 102
- [208] Yan J., Lei Z., Wen L., and Li S. The fastest deformable part model for object detection. in *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 2497–2504, 2014. 100
- [209] S. Jabri, M. Saidallah, A. el Alaoui, and A. El Fergougui. Moving vehicle detection using haar-like, lbp and a machine learning adaboost algorithm. *IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pages 121–124, 2018. 27, 29
- [210] S. Javed, T. Bouwmans, and S. Jung. Stochastic decomposition into low rank and sparse tensor for robust background subtraction. *ICDP*, 2015. 21, 26
- [211] S. Javed, T. Bouwmans, and S. Jung. Improving or-pca via smoothed spatially-consistent low-rank modeling for background subtraction. *ACM Symposium on Applied Computing, SAC*, 2017. 21
- [212] S. Javed, A. Mahmood, S. Al-Maadeed T.B., and S. Jung. Moving object detection in complex scene using spatiotemporal structured-sparse RPCA. *IEEE Trans. Image Process.*, 2019. 21
- [213] S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Background-foreground modeling based on spatiotemporal sparse subspace clustering. *IEEE Trans. Image Process.*, 2017. 21, 26
- [214] S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Superpixels based manifold structured sparse rpca for moving object detection. *International Workshop on Activity Monitoring by Multiple Distributed Sensing, BMVC*, 2017. 21
- [215] S. Javed, A. Mahmood, T. Bouwmans, and Jung S. Motion-aware graph regularized rpca for background modeling of complex scenes. *International Conference on Pattern Recognition, ICPR*, 2016. 18, 21, 26
- [216] S. Javed, A. Mahmood, T. Bouwmans, and Jung S. Spatiotemporal low-rank modeling for complex scene background initialization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016. 18, 21, 26
- [217] S. Javed, S. Oh, and S. Jung. IPBAS: Improved pixel based adaptive background segmenter for background subtraction. *Human Computer Interaction*, 2014. 20, 25
- [218] S. Javed, A. Sobral, T. Bouwmans, and S. K. Jung. OR-PCA with dynamic feature selection for robust background subtraction. In *ACM Symposium on Applied Computing*, pages 86–91, 2015. 21, 26
- [219] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, and R. Girshick. Caffe: Convolutional architecture for fast feature embedding. In *Arxiv preprint hepht*, 2014. 7

- [220] T. Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning*, pages 200–209, 1999. 27, 29, 81, 93, 94
- [221] Cs JoChang-yeon. Face detection using LBP features. 2008. 100
- [222] P. Jodoin. Motion detection : Unsolved issues and [potential] solutions. *SBMI 2015 in conjunction with ICIAP 2015*, 2015. 43
- [223] P.M Jodoin, L. Maddalena, A. Petrosino, and Y. Wang. Extensive benchmark and survey of modeling methods for scene background initialization. <http://scenebackgroundmodeling.net>, 2017. 26
- [224] R. Johannes and S. Armin. Face recognition with machine learning in OpenCV fusion of the results with the localization data of an acoustic camera for speaker identification. *arXiv*, 2017. 30, 39
- [225] F. Juefei-Xu, K. Luu, and M. Savvides. Spartans : Single-sample periocular-based alignment-robust recognition technique applied to non-frontal scenarios. *IEEE T-IP*, pages 4780–4795, 2015. 36, 41
- [226] Zhang K., Zhang Z., Li Z., and Qiao Y. Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Process. Lett.*, 23(10):1499–1503, 2016. vi, 100, 103, 104, 107
- [227] k. B. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The journal of Machine Learning Research, Advances in Neural Information Processing Systems.*, pages 207–244, 2009. 98, 105
- [228] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. *European Workshop on Advanced Video Based Surveillance Systems (AVSS)*, 2001. vi, 57, 60, 61, 62, 64, 65, 73
- [229] P. Kamencay, M. Zachariasova, and R. Hudec. A novel approach to face recognition using image segmentation based on spca knnn method. *Radio engineering*, pages 92–99, 2013. 37, 41
- [230] M. Karaaba, O. Surinta, M. Schomaker, and M. Wiering. Robust face recognition by computing distances from multiple histograms of oriented gradients. *IEEE Symposium Series on Computational Intelligence*, pages 203–209, 2015. 34, 40
- [231] O. Karadag and O. Erdas. Evaluation of the robustness of deep features on the change detection problem. *IEEE Signal Processing and Communications Applications Conference, SIU*, 2018. 43
- [232] L. J. Karam, T. Zhu, and R. Chellappa. Quality labeled faces in the wild (QLFW): a database for studying face recognition in real-world environments. *Proceedings of SPIE - The International Society for Optical Engineering.*, 2015. 45
- [233] K. Karmann and Brand A. Moving object recognition using an adaptive background memory. *Time-Varying Image Processing and Moving Object Recognition*, 1990. 23, 26

- [234] Y. Kawanishi, I. Mitsugami, and M. Mukunoki. Background image generation keeping lighting condition of outdoor scenes. *International Conference on Security Camera Network, Privacy Protection and Community Safety*, 2009. 20, 26, 45
- [235] B. Kepenekci. Face recognition using gabor wavelet transform. *Ph.D. thesis, The Middle East Technical University*, 2001. 35
- [236] S.A. Khan, Ishtiaq M., Nazir M., and Shaheen M. Face recognition under varying expressions and illumination using particle swarm optimization. *J. Comput. Sci.*, pages 94–100, 2018. 31
- [237] P. Khoi, L.H. Thien, and Viet V.H. Face retrieval based on local binary pattern and its variants : A comprehensive study. *Int. J. Adv. Comput. Sci. Appl.*, pages 249–258, 2016. 33, 40
- [238] C. Kim, Lee J., and Han T. A hybrid framework combining background subtraction and deep neural networks for rapid person detection. *J Big Data*, pages 5–22, 2018. 60, 89, 91, 92, 93, 94
- [239] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *IEEE International Conference on Image Processing, ICIP*, 2004. 20, 25
- [240] K. Kim, M. Franz, and Scholkopf B. Iterative kernel principal component analysis for image modeling. *IEEE T-PAMI*, pages 1351–1366, 2005. 32, 39
- [241] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ArXiv*, 2013. 81
- [242] P. Kingma, Diederik, Danilo J., and Shakir M. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 2014. 28, 81
- [243] T N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017. 81
- [244] B. Kiran and S. Yogamani. Real-time background subtraction using adaptive sampling and cascade of gaussians. *Preprint*, 2017. 20, 25
- [245] R. Kong and Bing Z. A new face recognition method based on fast least squares support vector machine. *Physics Procedia*, pages 616–621, 2011. 30, 39
- [246] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 29
- [247] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 44, 98, 101
- [248] T. Kryjak and M. Gorgon. Real-time implementation of the ViBe foreground object segmentation algorithm. *Federated Conference on Computer Science and Information Systems, FedCSIS*, pages 591–596, 2013. 20, 25
- [249] T. Kryjak, M. Komorkiewicz, and M. Gorgon. Real-time foreground object detection combining the PBAS background modelling algorithm and feedback from scene analysis module. *Int. J. Electron. Telecommun. IJET*, 2014. 20, 25

- [250] T. Kryjak, M. Komorkiewicz, and M. Gorgon. Real-time implementation of foreground object detection from a moving camera using the ViBe algorithm. *Comput. Sci. Inf. Syst.*, 2014. 20, 25
- [251] T. Kryjak, M. Komorkiewicz, and M. Gorgonn. Hardware implementation of the PBAS foreground detection method in FPGA. *International Conference Mixed Design of Integrated Circuits and Systems*, MIXDES, pages 479–484, 2013. 20, 25
- [252] W. Kusakunniran and R. Krungkaew. Dynamic codebook for foreground segmentation in a video. *ECTI Trans. Comput. Inf. Technol. ECTI-CIT*, 2017. 20, 25
- [253] D. Kuzin. Sparse machine learning methods for autonomous decision making. *PhD Thesis, University of Sheffield*, 2018. 24, 26
- [254] D. Kuzin, O. Isupova, and L. Mihaylova. Compressive sensing approaches for autonomous object detection in video sequences. *CVPR*, 2015. 24, 26
- [255] D. Kuzin, O. Isupova, and L. Mihaylova. Spatio-temporal structured sparse regression with hierarchical gaussian process priors. *Transactions on Signal Processing*, pages 4598–4611, 2018. 24, 26
- [256] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations*, 2017. 81
- [257] A. Lanza, F. Tombari, and L. D. Stefano. Accurate and efficient background subtraction by monotonic second-degree polynomial fitting. *IEEE AVSS*, 2010. 19, 25
- [258] N. Laopracha, K. Sunat, and S. Chiewchanwattana. A novel feature selection in vehicle detection through the selection of dominant patterns of histograms of oriented gradients (dphog). *IEEE Access*, pages 20894–20919, 2019. 29
- [259] B. Laugraud, S. Pierard, and M.V. Droogenbroeck. LaBGen-P-semantic: A first step for leveraging semantic segmentation in background generation. *J. Imaging*, 2018. 24, 26
- [260] I. Laure Kambi Beli and C. Guo. Enhancing face identification using local binary patterns and k-nearest neighbors. *J. Imaging*, 2017. 33, 40
- [261] T. Le and B. Len. Face recognition based on SVM and 2DPCA. *arXiv preprint*, 2011. 31, 39
- [262] B. Lecouat, C. Sheng Foo, H. Zenati, and V. Chandrasekhar. Manifold regularization with gans for semi-supervised learning. *preprint arXiv*, 2018. 81
- [263] Y. LeCun, Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., and Jackel L.D. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, pages 396–404, 1989a. 27, 29
- [264] Y. LeCun, Bengio Y., and Hinton G. Deep learning. *Nature*, pages 436–444, 2015. 53
- [265] B. Lee and M. Hedley. Background estimation for video surveillance. *IVCNZ*, pages 315–320, 2002. 18, 19, 25
- [266] D.-S. Lee. Improved adaptive mixture learning for robust video background modeling. In *Machine Vision for Application (MVA)*, pages 443–446, 2002. 6

- [267] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 827–832, 2005. 20
- [268] S. Lee and D. Kim. Background subtraction using the factored 3-way restricted boltzmann machines. *Preprint*, 2018. 22
- [269] T.S. Lee. Image representation using 2D gabor wavelets. *IEEE T-PAMI*, pages 959–971, 1996. 8
- [270] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, and Kontokostas. DBpedia, a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 2014. 29
- [271] L. Lenc and P. Král. Automatic face recognition system based on the sift features. *Comput. Electr. Eng.*, pages 256–272, 2015. 35, 40
- [272] I. Leonard, A. Alfalou, and C. Brosseau. Face recognition based on composite correlation filters: Analysis of their performances. *Face Recognition*, 2012. 34, 40
- [273] B. Li and K.K. Ma. Fisherface vs. eigenface in the dual-tree complex wavelet domain. *The Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 30–33, 2009. 30, 39
- [274] H. Li, Z. Lin, Shen X., and Brandt J. A convolutional neural network cascade for face detection. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015. 38, 42, 100
- [275] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian. Statistical modeling of complex background for foreground object detection. *IEEE Transactions on Image Processing*, pages 1459–1472, 2004. 26
- [276] W. Li, Fu H., Yu L., Gong P., Feng D., Li C., and Clinton N. Stacked autoencoder-based deep learning for remote-sensing image classification : a case study of african land-cover mapping. *Int. J. Remote Sens.*, pages 5632–5646, 2016. 28, 29
- [277] Y. Li, G. Wang, and X. Lin. Three-level GPU accelerated gaussian mixture model for background subtraction. *Proc. SPIE*, 2012. 20, 25
- [278] D. Liang. AIST-INDOOR dataset. http://ssc-lab.com/~liang/CP3_project/AIST_INDOOR_DATASET.rar, 2013. 25
- [279] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, and X. Zhao. Co-occurrence probability based pixel pairs background model for robust object detection in dynamic scenes. *Pattern Recognition*, pages 1374–1390, 2015. 19, 25
- [280] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, X. Zhao, and Y. Satoh. Co-occurrence-based adaptive background model for robust object detection. *AVSS*, 2013. 19, 25
- [281] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, X. Zhao, and Y. Satoh. Robust object detection in severe imaging conditions using co-occurrence background model. *International Journal of Optomechatronics*, 2014. 19, 25
- [282] K. Lim, W. Jang, and C. Kim. Background subtraction using encoder-decoder structured convolutional neural network. *IEEE AVSS*, 2017. 22, 26

- [283] L. Lim, I. Ang, and H. Keles. Learning multi-scale features for foreground segmentation. *Preprint*, 2018. 44
- [284] L. Lim and H. Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *Preprint*, 2018. 44
- [285] B. Liu, Liu Y., and Zhou K. Image classification for dogs and cats. *Image CF*, 2014. 27, 29
- [286] W. Liu., Y. Wren., Z. Yu., M. Li., B. Raj., and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, pages 212–220, 2017. 38, 42
- [287] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR2015*, pages 3431–3440, 2015. 27, 29
- [288] D.G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, pages 91–110, 2004. 8
- [289] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE T-PAMI*, 2019. 21, 26
- [290] J. Lu, K. Plataniotis, and Venetsanopoulos A. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Netw.*, pages 117–126, 2003. 32
- [291] X. Lu, T. Izumi, T. Takahashi., and L. Wang. Moving vehicle detection based on fuzzy background subtraction. *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, pages 529–532, 2014. 20, 25
- [292] X. Lu, T. Izumi, L. Teng, T. Hori, and L. Wang. A novel background subtraction method for moving vehicle detection. *IEEJ Trans. Fundam. Mater.*, pages 857–863, 2012. 20, 25
- [293] X. Lu, C. Xu, L. Wang, and L. Teng. Improved background subtraction method for detecting moving objects based on GMM. *IEEE Trans. Electr. Electron. Eng.*, 2018. 20, 25
- [294] A. L. Maas, R. E. Daly, P. T. Pham, and D. Huang. Learning word vectors for sentiment analysis. In *ACL*, 2011. 29
- [295] L. Maddalena and Petrosino A. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Comput*, pages 179–186, 2010. vi, 22, 26, 57, 60, 61, 62, 64, 65, 73
- [296] L. Maddalena and A. Petrosino. A self-organizing approach to detection of moving patterns for real-time applications. *Advances in Brain, Vision and Artificial Intelligence.*, pages 181–190, 2007. 22, 26
- [297] L. Maddalena and A. Petrosino. Neural model-based segmentation of image motion. *KES*, pages 57–64, 2008. 22, 26

- [298] L. Maddalena and A. Petrosino. A self organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, pages 1168–1177, 2008. 22, 26
- [299] L. Maddalena and A. Petrosino. A self-organizing neural system for background and foreground modeling. *ICANN*, pages 652–661, 2008. 22, 26
- [300] L. Maddalena and A. Petrosino. 3D neural model-based stopped object detection. *International Conference on Image Analysis and Processing, ICIAP*, pages 585–593, 2009. 22
- [301] L. Maddalena and A. Petrosino. Multivalued background/foreground separation for moving object detection. *International Workshop on Fuzzy Logic and Applications, WILF*, pages 263–270, 2009. 22, 26
- [302] L. Maddalena and A. Petrosino. Self organizing and fuzzy modelling for parked vehicles detection. *Advanced Concepts for Intelligent Vision Systems, ACVIS*, pages 422–433, 2009. 22
- [303] L. Maddalena and A. Petrosino. The SOBS algorithm : What are the limits ? *IEEE Workshop on Change Detection, CVPR*, 2012. 22, 26
- [304] L. Maddalena and A. Petrosino. Stopped object detection by learning foreground model in videos. *IEEE Transactions on Neural Networks and Learning Systems*, pages 723–735, 2013. 22
- [305] L. Maddalena and A. Petrosino. The 3dSOBS+ algorithm for moving object detection. *Computer Vision and Image Understanding, CVIU*, pages 65—73, 2014. 22, 26
- [306] L. Maddalena and A. Petrosino. Towards benchmarking scene background initialization. <http://sbmi2015.na.icar.cnr.it/SBIdataset.html>, 2015. 26
- [307] L. Maddalena and A. Petrosino. Background subtraction for moving object detection in RGB-D data : A survey. *MDPI Journal of Imaging*, 2018. 18
- [308] D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing (IVC)*, pages 143–155, 2004.
- [309] P. K. Mallapragada. Semi-supervised learning. *Some contributions to semi-supervised learning*, 2010. 81
- [310] C. Marghes and T. Bouwmans. Background modeling via incremental maximum margin criterion. *ACCV*, 2010. 18, 21, 26
- [311] C. Marghes, T. Bouwmans, and R. Vasiu. Background modeling and foreground detection via a reconstructive and discriminative subspace learning approach. *IPCV*, 2012. 18, 21, 26
- [312] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *8th Int'l Conf. Computer Vision*, pages 416–423, 2001. 26
- [313] A. Martinez and R. Benavente. The AR face database. *Tech. rep. , Univerzitat Autonoma de Barcelona*, 1998. 35, 39, 40, 41

- [314] I. Martins, P. Carvalho, L. Corte-Real, and J. Alba-Castro. BMOG: Boosted Gaussian Mixture Model with Controlled Complexity. *IbPRIA*, 2017. 20, 25
- [315] N. McFarlane and Schofield C. Segmentation and tracking of piglets in images. *Br. Mach. Vis. Appl.*, pages 187–193, 1995. 57, 61
- [316] S. Messelodi, C. Modena, Segata N., and Zanin M. A kalman filter based background updating algorithm robust to sharp illumination changes. *ICIAP*, pages 163–170, 2005. 18, 23
- [317] A. Mian, M. Bennamoun, and R. Owens. An efficient multimodal 2D-3D hybrid approach to automatic face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1927–1943, 2007. 36, 41
- [318] R. Mieziako. IEEE OTCBVS WS series bench. Terravic Research Infrared Database. 25
- [319] K. Mikolajczyk and Schmid C. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, pages 1615–1630, 2005. 40
- [320] S. Milborrow, J. Morkel, and F. Nicolls. The MUCT landmarked face database. *Pattern Recognition Association of South Africa*, 2010. 29, 41
- [321] D. Miller, E. Brossard, S. Seitz, and KemelmacherShlizerman I. Megaface: A million faces for recognition at scale. *arXiv preprint*, 2015. 42
- [322] Y. Ming, Q. Ruan, and Xueqiao W. Efficient 3D face recognition with gabor patched spectral regression. *Computing and Informatics*, pages 779–803, 2012. 32, 39
- [323] T. Miyato, Maeda S., Koyama M., and Ishii S. Virtual adversarial training : a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 28, 29, 81, 93, 94
- [324] M.Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. *European Conference on Computer Vision.*, pages 720–735, 2014. 100
- [325] J. Mota, Weizman L., N. Deligiannis, Y. Eldar, and M. Rodrigues. Reference-based compressed sensing : A sample complexity approach. *ICASSP*, 2016. 24
- [326] M. Moussa, M. Hmila, and Douik A. A novel face recognition approach based on genetic algorithm optimization. *Stud. Inform. Control*, pages 127–134, 2018. 36, 41
- [327] D. Mukherjee and J. Wu. Real-time video segmentation using student’s t mixture model. *ANT*, pages 153–160, 2012. 19, 25
- [328] O. Munteanu, T. Bouwmans, E. Zahzah, and R. Vasiu. The detection of moving objects in video by background subtraction using dempster-shafer theory. *Transactions on Electronics and Communications*, 2015. 18, 19, 20, 25
- [329] T. Napoléon and A. Alfalou. Local binary patterns preprocessing for face identification/verification using the vanderlugt correlator. *In Optical Pattern Recognition*, 2014. 34, 40

- [330] P. Narayanamurthy and N. Vaswani. A fast and memory-efficient algorithm for robust PCA (MEROP). *IEEE ICASSP*, 2018. 21, 26, 43
- [331] Y. Netzer, T. Wang, A. Coates, and A. Bissacco. Reading digits in natural images with unsupervised feature learning. In *Workshop on deep learning and unsupervised feature learning on NIPS*, 2011. 29
- [332] T. Nguyen, C. Pham, S. Ha, and J. Jeon. Change detection by training a triplet network for motion feature extraction. *IEEE T-CSVT*, 2018. 22
- [333] S. Oh, A. Hoogs, and A. Perera. A large-scale benchmark dataset for event recognition in surveillance video. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011. iii, iv, v, vi, 15, 26, 28, 29, 47, 54, 55, 56, 57, 58, 60, 78, 79, 81, 82, 89, 90, 91, 92, 93, 94, 96
- [334] T. Ojala, M. Petikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: *Proc. IAPR Inter. Conf. Pattern Recognit.*, 1994. 8
- [335] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal.*, pages 831–843, 2000. 18, 20, 26, 45, 57, 61
- [336] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, pages 808–828, 2014. 24
- [337] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, pages 62–66, 1979. 52
- [338] H. Ouanan, M. Ouanan, and Aksasse B. Non-linear dictionary representation of deep features for face recognition from a single sample per person. *Procedia Comput. Sci.*, pages 114–122, 2018. 32
- [339] Y. Ouerhani, M. Jridi, A. Alfalou, and C. Brosseau. Optimized pre-processing input plane GPU implementation of an optical face recognition technique using a segmented phase only composite filter. *Opt. Commun.*, pages 33–44, 2013. 34, 40
- [340] C. O’Donnell and V. Bruce. Familiarisation with faces selectively enhances sensitivity to changes made to the eyes. *Perception*, pages 755–764, 2001. 101
- [341] Rodriguez P. A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling. *IEEE International Conference on Image Processing, ICIP*, 2014. 21, 26
- [342] J. Parris. Face and eye detection on hard datasets. *International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2011. 34
- [343] J. Paruchuri, E. P. Sathiyamoorthy, S.-C.S. Cheung, and C.-H. Chen. Spatially adaptive illumination modeling for background subtraction. In *International Conference on Computer Vision (ICCV)*, pages 1745–1752, 2011. 4
- [344] V. Perlibakas. Face recognition using principal component analysis and log-gabor filters. *ArXiv*, 2006. 31, 39

- [345] F. Perronnin, C. Dance, C.J. Veenman, and A.W.M. Smeulders. Fisher kernels on visual vocabularies for image categorization. *In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007. 9
- [346] F. Perronnin and T. Mensink. Improving the fisher kernel for large-scale image classification. *In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV*, 2010. 9
- [347] A. Petrosino and L. Maddalena. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Trans. Image Process.*, pages 1168–1177, 2008. 57, 61
- [348] V. Pham, P. Vo, H.V. Thanh, and B.L. Hoai. GPU implementation of extended gaussian mixture model for background subtraction. *IEEE International Conference on Computing and Telecommunication Technologies, RIVF*, 2010. 20, 25, 100
- [349] J. Phillips, H. Moon, S. Rizvi, and Rauss P. The FERET evaluation methodology for face-recognition algorithms. *IEEE T-PAM*, pages 1090–1104, 2000. 33, 34, 35, 39, 40, 41
- [350] P.J. Phillips, Flynn P.J., Scruggs T., Bowyer K., Chang J., Hoffman K., Marques J., Min J., and Worek W. Overview of the face recognition grand challenge. *Proc. IEEE Computer Vision and Pattern Recognition*, 2005. 39, 40, 41
- [351] F. Porikli and Wren C. Change detection by frequency decomposition : Wave-back. *International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS*, 2005. 23, 26
- [352] S. Prativadibhayankaram, H. Luong, T. Le, and A. Kaup. Compressive online video background foreground separation using multiple prior information and optical flow. *MDPI*, 2018. 43
- [353] J. Pulgarin-Giraldo, D. Alvarez-Meza, Insuasti-Ceballos D., Bouwmans T., and G. Castellanos-Dominguez. GMM background modeling using divergence-based weight updating. *Conference Ibero American Congresson Pattern Recognition, CIARP*, 2016. 19
- [354] Zhu Q., Yeh M. C., Cheng K. T., and Avidan S. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Computer Conference on Computer Vision and Pattern Recognition.*, pages 1491–1498, 2006. 100
- [355] C. Qiu and N. Vaswani. Real-time robust principal components pursuit. *International Conference on Communication Control and Computing*, 2010. 21, 26
- [356] C. Qiu and N. Vaswani. Support predicted modified-CS for recursive robust principal components' pursuit. *International Conference on Communication Control and Computing*, 2011. 21, 26
- [357] Z. Qu, S. Yu, and M. Fu. Motion background modeling based on context-encoder. *IEEE ICAIPR*, 2016. 20, 22, 26
- [358] J. Quesada and P. Rodriguez. Automatic vehicle counting method based on principal component pursuit background modeling. *IEEE International Conference on Image Processing, ICIP*, 2016. 21, 26

- [359] Wu R., Yan S., Shan Y., Dang Q., and Sun G. Deep image: scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 2015. 101, 102
- [360] A. Radford, Metz L., Chintala S., and Rigoll G. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016. 81, 84, 85, 99, 102
- [361] A. Radford, Jozefowicz R., and Sutskever I. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444, 2017. 28, 29, 82
- [362] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing (TIP)*, pages 294–307, 2005. 6, 19
- [363] G. Ramirez-Alonso and M. Chacon-Murguia. Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recognition*, 2015. 18
- [364] J. Ramirez-Quintana and M. Chacon-Murguia. Self-organizing retinotopic maps applied to background modeling for dynamic object segmentation in video sequences. *IJCNN*, 2013. 18
- [365] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv: 1703.09507*, 2017. 44, 45
- [366] M. Ranzato, V. Mnih, and J.M. Susskind. Modeling natural images using gated MRFs. *IEEE Trans Pattern Anal Mach Intell.*, pages 2206–2222, 2013. 44
- [367] J. Ren, X. Jiang, and J. Yuan. Relaxed local ternary pattern for face recognition. *IEEE ICIP*, pages 3680–3684, 2013. 33, 40
- [368] B. Rezaei and S. Ostadabbas. Background subtraction via fast robust matrix completion. *International Workshop on RSL-CV in conjunction with ICCV*, 2017. 21, 26
- [369] B. Rezaei and S. Ostadabbas. Moving object detection through robust matrix completion augmented with objectness. *IEEE Journal of Selected Topics in Signal Processing*, 2018. 21, 26
- [370] D. Rezende, S. Mohamed, and Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31 st International Conference on Machine Learning*,, 2014. 81
- [371] A. Riddhi, Vyas D., and Shah S. Comparison of PCA and LDA techniques for face recognition feature based extraction with accuracy enhancement. *IRJET*, pages 3332–3336, 2017. 31
- [372] P. Rodriguez. Real-time incremental Principal Component Pursuit for Video Background Modeling. *GPU Technical Conference, GTC*, 2015. 21, 26
- [373] P. Rodriguez and B. Wohlberg. Fast principal component pursuit via alternating minimization. *IEEE International Conference on Image Processing, ICIP*, 2013. 21, 26
- [374] P. Rodriguez and B. Wohlberg. Translational and rotational jitter invariant incremental principal component pursuit for video background modeling. *IEEE International Conference on Image Processing, ICIP*, 2015. 21, 26, 43, 44

- [375] P. Rodriguez and B. Wohlberg. Ghosting suppression for incremental principal component pursuit algorithms. *IEEE Global Conference on Signal and Information Processing, GlobalSIP*, 2016. 21, 26
- [376] P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 2016. 21, 26, 43
- [377] P. Rodriguez and B. Wohlberg. An incremental principal component pursuit algorithm via projections onto the 11 ball. *IEEE International Conference on Electronics, Electrical Engineering and Computing, INTERCON*, pages 1–4, 2017. 21, 26
- [378] J. Rosell-Ortega, G. Andreu, V. Atienza, and F. Lopez-Garcia. Background modeling with motion criterion and multi-modal support. *VISAPP*, 2010. 19, 25
- [379] J. Rosell-Ortega, G. Andreu-Garcia, A. Rodas-Jorda, and V. Atienza-Vanacloig. Background modelling in demanding situations with confidence measure. *IAPR ICPR*, 2008. 19, 25
- [380] C. Rosenberg, Hebert M., and Schneiderman H. Semi-supervised self-training of object detection models. *IEEE Workshops on Application of Computer Vision, WACV/MOTION'05*, 2005. 28, 29, 81
- [381] D. Rout, B. Subudhi, T. Veerakumar, and S. Chaudhury. Spatio-contextual gaussian mixture model for local change detection in underwater video. *Expert Syst. Appl.*, 2017. 20, 25
- [382] S. Roy and A. Ghosh. Real-time adaptive histogram min-max bucket (HMMB) model for background subtraction. *IEEE T-CSVT*, 2017. 18, 25
- [383] E. Rublee, R. Vincent, Kurt K., and Gray B. ORB: an efficient alternative to SIFT or SURF. *In IEEE International Conference on Computer Vision*, 2011. 32
- [384] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, and A. Khosla. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 29
- [385] Yang S., Luo P., Loy C. C., and Tang X. From facial parts responses to face detection: A deep learning approach. *in IEEE International Conference on Computer Vision*, pages 3676–3684, 2015. 100
- [386] R. Saha, D. Bhattacharjee, and Barman S. Comparison of different face recognition method based on PCA. *International Journal of Management Information Technology*, 2014. 30, 39
- [387] T. Salimans, Goodfellow I.J., Zaremba W., Cheung V., and Radford A. Improved techniques for training gans. *CoRR, abs/1606.03498*, 2016. 28, 29, 81, 82, 84, 88, 89
- [388] C. Salvadori, D. Makris, M. Petracca, J. Rincon, and S. Velastin. Gaussian mixture background modelling optimisation for micro-controllers. *International Symposium on Visual Computing, ISVC*, pages 241–251, 2012. 20, 25
- [389] C. Salvadori, M. Petracca, J. Rincon, S. Velastin, and D. Makris. An optimisation of gaussian mixture models for integer processing units. *Real-Time Image Process.*, 2014. 20, 25

- [390] K Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. *IEEE T-PAMI*, pages 1582–1596, 1998. 8
- [391] A. Savakis and A. Shringarpure. Semantic background estimation in video sequences. *IEEE International Conference on Signal Processing and Integrated Networks, SPIN*, pages 597–601, 2018. 24, 26
- [392] A. Schofield, P. Mehta, and T. Stonham. A system for counting people in video images using neural networks to identify the background scene. *Pattern Recognition*, 1996. 18, 21, 26
- [393] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: a unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 10, 14, 15, 38, 42, 45, 46, 103, 108
- [394] H. Seo and P. Milanfar. Face verification using the lark representation. *IEEE Trans. Inf. Forensics Secur.*, pages 1275–1286, 2011. 29, 39
- [395] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE T-TPAMI*, pages 411–426, 2007. 9
- [396] S. Z. Seyyedsalehi and A. Seyyedsalehi. Simultaneous learning of nonlinear manifolds based on the bottleneck neural network. *Neural Processing Letters*, pages 191–209, 2014. 101, 102
- [397] M. Shafiee, P. Siva, and P. Fieguth. Real-time embedded motion detection via neural response mixture modeling. *Journal of Signal Processing Systems*, 2017. 20, 22
- [398] M. Shafiee, P. Siva, P. Fieguth, and A. Wong. Embedded motion detection via neural response mixture background modeling. *International Conference on Computer Vision and Pattern Recognition*, 2016. 20, 22, 26
- [399] M. Shah, J. Deng, and B. Woodford. Video background modeling: recent approaches, issues and our proposed techniques. *Mach. Vis. Appl.*, pages 1105–1119, 2014. 20, 25
- [400] P.K. Shah and Anand B. Face recognition using SURF features and SVM classifier. *International Journal of Electronics Engineering Research*, pages 1–8, 2016. 35, 40
- [401] S. Shaikh, K. Saeed, and N. Chaki. *Moving Object Detection Using Background Subtraction*. Springer International Publishing, 2014. 2
- [402] H. Shakeri, M. and Deldari, H. Foroughi, and A. Saberi. A novel fuzzy background subtraction method based on cellular automata for urban traffic applications. *International Conference on Signal Processing, ICSP*, pages 899–902, 2008. 20, 25
- [403] M. Shakeri and H. Deldari. Fuzzy-cellular background subtraction technique for urban traffic applications. *World Appl. Sci. J.*, 2008. 20, 25
- [404] S. Shan, Y. Chang, W. Gao, Cao B., and Yang P. Curse of mis-alignment in face recognition: problem and a novel mis-alignment learning solution. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 314–320, 2004. 101
- [405] S.S. Shanbhag, S. Bargi, Manikantan K., and Ramachandran S. Face recognition using wavelet transforms-based feature extraction and spatial differentiation-based pre-processing. *ICSEMR*, pages 1–8, 2014. 31, 39

- [406] R. Sharma and M.S. Patterh. A new pose invariant face recognition system using pca and anfis. *Optik*, pages 3483–3487, 2015. 36, 41
- [407] V. Sharma, N. Nain, and T. Badal. A redefined codebook model for dynamic backgrounds. *International Conference on Computer Vision and Image Processing, CVIP*, 2016. 20, 25
- [408] Y. Bhirud Shruti and Gohokar V.V. Face recognition based on SVM and GABOR filter. *International Journal of Current Engineering and Technology*, 2014. 27, 29
- [409] D. I. Shuman, S. K. Narang, P. Frossard, and A. Ortega. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, pages 83–98, 2013. 24
- [410] M. Sigari. Fuzzy background modeling/subtraction and its application in vehicle detection. *World Congress on Engineering and Computer Science, WCECS*, 2008. 20, 25
- [411] M. Sigari, N. Mozayani, and H. Pourreza. Fuzzy running average and fuzzy background subtraction: Concepts and application. *Int. J. Comput. Sci. Netw. Secur.*, pages 138–143, 2008. 20, 25
- [412] G. Silva and P. Rodriguez. Jitter invariant incremental principal component pursuit for video background modeling on the tk1. *Asilomar Conference on Signals, Systems, and Computers, ACSSC*, 2015. 21, 26, 44
- [413] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2003. 40, 41
- [414] K. Simonyan, O. Parkhi, Vedaldi A., and Zisserman A. Fisher vector faces in the wild. *BMVC*, pages 9–13, 2013. 30, 36, 39, 41
- [415] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Conference on Neural Information Processing Systems (NIPS)*, pages 568–576, 2014. 38, 42
- [416] J. Sing, S. Chowdhury, D. Basu, and Nasipuri M. An improved hybrid approach to face recognition by fusing local and global discriminant features. *Int. J. Biom.*, pages 144–164, 2012. 37, 41
- [417] M. Singh, V. Parameswaran, and V. Ramesh. Order consistent change detection via fast statistical significance testing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 6
- [418] A. Sobral. BGSLibrary: an OpenCV C++ background subtraction library. *IX Workshop de Viso Computacional (WVC'2013)*, 2013. 55, 56, 60, 62, 71, 72
- [419] A. Sobral, T. Bouwmans, E. Zahzah, and Wright J. Double-constrained rpca based on saliency maps for foreground detection in automated maritime surveillance. *AVSS*, 2015. 18, 21, 26
- [420] A. Sobral, S. Javed, S. Jung, T. Bouwmans, and E. Zahzah. Online stochastic tensor decomposition for background subtraction in multispectral video sequences. *ICCV*, 2015. 21, 26

- [421] S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. *IEEE International Conference on Computer Vision*, page 233–240, 2013. 25
- [422] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR, abs/1212.0402*, 2012. 42
- [423] J.T. Springenberg. Unsupervised and semi supervised learning with categorical generative adversarial networks. *BOOK*, 2015. 28, 29, 81, 86, 93, 94
- [424] P. St-Charles, G. Bilodeau, and R. Bergevin. Flexible background subtraction with self-balanced local sensitivity. *IEEE Change Detection Workshop, CDW*, 2014. 19, 25
- [425] P. St-Charles, G. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. *IEEE WACV*, 2015. 19, 25
- [426] P. St-Charles, G. Bilodeau, and R. Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. In *IEEE Transactions on Image Processing (TIP)*, pages 359–373, 2015. 4, 25
- [427] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Computer Vision and Pattern Recognition (CVPR)*, pages 246–252, 1999. vi, 6, 18, 19, 20, 22, 25, 43, 57, 60, 61, 69, 70, 76
- [428] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 747–757, 2000. 6
- [429] J. Stückler and S Behnke. Efficient dense 3D rigid-body motion segmentation in RGBD video. *BMVC*, pages 171–177, 2013. 25
- [430] Z. Sufyanu, F. Mohamad, Yusuf A., and Mamat M. Enhanced face recognition using Discrete Cosine Transform. *Eng. Lett.*, pages 52–61, 2016. 31, 39
- [431] J. Sun, Y. Fu, J. Li, S. and He, C. Xu, and L. Tan. Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors. *J. Sens.*, 2018. 37, 41
- [432] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014. 38, 42, 98
- [433] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2892–2900, 2015. 38, 42, 98
- [434] Y. Sun., Chen. Y., Wang. X., and Tang. X. Deep learning face representation by joint identification-verification. *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems.*, pages 1988–1996, 2008. 38, 42, 104
- [435] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint*, 2015. 98, 99, 103

- [436] C. Szegedy, Liu. W., Y. Jia, and P. Sermanet. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition.*, 1998. 44
- [437] G. Szwoch. Performance evaluation of the parallel codebook algorithm for background subtraction in video stream. *International Conference on Multimedia Communications, Services and Security, MCSS*, 2011. 20, 25
- [438] H. Tabkhi, R. Bushey, and G. Schirner. Algorithm and architecture co-design of mixture of gaussian (mog) background subtraction for embedded vision processor. *Asilomar Conference on Signals, Systems, and Computers*, 2013. 20, 25
- [439] Y. Taigman, M. Yang, M.A. Ranzato, and L. Wolf. Deepface: closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 10, 37, 42, 98
- [440] Y. Tao, P. Palasek, Z. Ling, and I. Patras. Background modelling based on generative Unet. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS*, 2017. 20
- [441] A. Tavakkoli. Foreground-background segmentation in video sequences using neural networks. *Intelligent Systems : Neural Networks and Applications*, 2005. 21, 26
- [442] E. Tejada and P. Rodriguez. Moving object detection in videos using principal component pursuit and convolutional neural networks. *GlobalSIP*, pages 793–797, 2017. 21, 26
- [443] X. Teng, Yan M., Ertugrul A., and Lin Y. Deep into hypersphere: robust and unsupervised anomaly discovery in dynamic networks. *International Joint Conference on Artificial Intelligence, IJCAI*, pages 2724–2730, 2018. 48, 49, 50, 51
- [444] F. Tombari, A. Lanza, L. D. Stefano, and S. Mattoccia. Non-linear parametric bayesian regression for robust background subtraction. *IEEE MOTION*, 2009. 19, 25
- [445] K. Toyama, J. Krumm, Brumiit B., and Meyers B. Wallflower : Principles and practice of background maintenance. *ICCV*, 1999. 18, 23, 25, 26
- [446] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for outdoor foreground/background extraction. In *Asian Conference on Computer Vision (ACCV)*, pages 291–300, 2012. iii, iv, vi, 15, 26, 47, 54, 71, 72, 73, 74, 75, 76, 77, 78
- [447] J.C. van Gemert, J.-M. Geusebroek, C.J. Veenman, and A.W.M.. Smeulders. Kernel codebooks for scene categorization. In: *Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS.*, 2008. 9
- [448] H.D. Vankayalapati and K. Kyamakya. Nonlinear feature extraction approaches with application to face recognition over large databases. *International Workshop on Non-linear Dynamics and Synchronization*, pages 44–48, 2009. 32
- [449] S. Varadarajan, P. Miller, and H. Zhou. Spatial mixture of gaussians for dynamic background modelling. *AVSS*, pages 63–68, 2013. 18
- [450] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurth. Robust subspace learning: Robust pca, robust subspace tracking and robust subspace recovery. *IEEE Signal Process. Mag.*, pages 32–55, 2018. 21, 26, 43

- [451] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy. Robust PCA and robust subspace tracking : A comparative evaluation. *SSP*, 2018. 21, 26
- [452] V. Vijayan, K. W. Bowyer, P. J. Flynn, and D. Huang. Twins 3d face recognition challenge. *Biometrics (IJCB)*, 2011. 42
- [453] A. Vinay, A.S. Cholin, Bhat A.D., Murthy K.B., and S. Natarajan. An efficient ORB based face recognition framework for human-robot interaction. *Procedia Comput. Sci.*, pages 913–923, 2018. 32, 39
- [454] A. Vinay, Hebbbar D., Shekhar V.S., Murthy K.B., and Natarajan S. Two novel detector-descriptor based approaches for face recognition using sift and surf. *Procedia Comput. Sci.*, pages 185–197, 2015. 35, 40
- [455] A. Vinay, S. Shekhar Vinay, Murthy K.N., and Natarajan S. Performance study of LDA and KFA for gabor based face recognition system. *ICRTC*, 2015. 32
- [456] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, pages 3371–3408, 2010. 101, 102
- [457] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision.*, pages 137–154, 2004. 100
- [458] L. Vosters, C. Shan, and T. Gritti. Real-time robust background subtraction under rapidly changing illumination conditions. In *Image and Vision Computing*, pages 1004–1015, 2012. 4
- [459] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010. 9
- [460] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Benezeth Y., and P. Ishwar. Cdn2014: An expanded change detection benchmark dataset,. pages 387–394, 2014. iii, iv, v, vi, vii, 15, 25, 26, 28, 29, 43, 47, 54, 59, 60, 61, 63, 64, 65, 68, 69, 70, 71, 78, 79, 81, 82, 89, 90, 93, 94, 95, 96, 99, 109, 110, 111, 112
- [461] Y. Wang, Z. Luo, and P. Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 2016. 22, 26
- [462] Z. Wang, L. Zhang, and H. Bao. PNN based motion detection with adaptive learning rate. *CIS*, pages 301–306, 2009. 21
- [463] G. Warnell, S. Bhattacharya, R. Chellappa, and T. Basar. Adaptive Rate Compressive Sensing Using Side Information. *ArXiv e-prints*, 2014. 26
- [464] G. Warnell, S. Bhattacharya, R. Chellappa, and T. Basar. Adaptive-rate compressive sensing via side information. *IEEE Transactions on Image Processing*, pages 3846–3857, 2015. 24
- [465] G. Warnell, D. Reddy, and R. Chellappa. Adaptive rate compressive sensing for background subtraction. *IEEE ICASSP*, 2012. 24
- [466] Y. Wen., Zhang. K., Li. Z., and Qiao. Y. A discriminative feature learning approach for deep face recognition. *European Conference on Computer Vision.*, 2016. 38, 42

- [467] B. Wohlberg. Endogenous convolutional sparse representations for translation invariant image subspace models. *IEEE International Conference on Image Processing, ICIP*, 2014. 44
- [468] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–534, 2011. iv, vii, 15, 42, 99, 109, 110, 114, 115, 116, 117, 119
- [469] Y. Wong, S. Chen, S. Mau, Sanderson C., and Lovell B.C. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR)*, pages 81–88, 2011. iv, vii, 15, 99, 109, 110, 114, 115, 116, 117, 118, 119
- [470] C. Wren and Azarbayejani A. Pfunder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 780–785, 1997. vi, 19, 23, 25, 26, 57, 60, 61, 62, 64, 65, 73
- [471] Z. Wu, Y. Huang, L. Wang, and T. Tan. Group encoding of local features in image classification. In: *Proc. IAPR Inter. Conf. Pattern Recognit.*, 2012. 9
- [472] Liu X., Zhao G., Yao J., and Qi C. Deep 3D face identification. In *Proceedings of the IEEE International Joint Conference on Biometrics*, pages 133–142, 2017. 38
- [473] Zhu X. and Ramanan D. Face detection, pose estimation, and landmark localization in the wild. in *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 2879–2886, 2012. 100
- [474] M. Xi, M. Chen, Polajnar D., and Tong W. Local binary pattern network : A deep learning approach for face recognition. *IEEE ICIP*, pages 3224–3228, 2016. 33, 40
- [475] H. Xiao, Y. Liu, and M. Zhang. Fast l1-minimization algorithm for robust background subtraction. *EURASIP Journal on Image and Video Processing*, 2016. 24
- [476] J. Xie. Face recognition based on curvelet transform ls-svm. *International Symposium on Information Processing (ISIP'09)*, 2009. 40
- [477] Q. Xie, Z. Dai, E. Hovy, M. Luong, and Q. V. Le. Unsupervised data augmentation for consistency training. *arXiv preprint*, 2019. 81
- [478] L. Xu, Y. Li, Y. Wang, and E. Chen. Temporally adaptive restricted boltzmann machine for background modeling. *AAAI*, 2015. 22, 26
- [479] P. Xu, M. Ye, X. Li, Q. Liu, Y. Yang, and J. Ding. Dynamic background learning through deep auto-encoder networks. In *International Conference on Multimedia*, pages 107–116, 2014. 22, 26
- [480] P. Xu, M. Ye, Q. Liu, X. Li, L. Pei, and J. Ding. Motion detection via a couple of auto-encoder networks. *International Conference on Multimedia and Expo, ICME*, 2014. 21, 22, 26
- [481] LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E, Hubbard W., and Jackel L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, pages 541–551, 1989. 29, 99

- [482] Qian Y., Gong M., and Cheng L. Stocs: An efficient self-tuning multiclass classification approach. *In Proceedings of the Canadian Conference on Artificial Intelligence*, pages 291–306, 2015. 37
- [483] Xu Y., Li X., Yang J., and Zhang D. Integrate the original face image and its mirror image for face recognition. *Neurocomputing*, pages 191–199, 2014. 7, 101
- [484] M. Yamazaki, G. Xu, and Y. Chen. Detection of moving objects by independent component analysis. *Asian Conference on Computer Vision, ACCV*, pages 467–478, 2006. 20, 26, 45
- [485] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for finegrained categorization and verification. *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3973–3981, 2015. 29
- [486] W.J. Yang, Y.C. Chen, Chung P.C., and Yang J.F. Multi-feature shape regression for face alignment. *J. Adv.Signal Process.*, 2018. 32, 39
- [487] J. Yao and J. Marc Odobez. Multi-layer background subtraction based on color and texture. *IEEE Computer Vision and Pattern Recognition Conf. (CVPR)*, pages 1–8, 2007. 57, 61
- [488] B. Yeo, W. Lim, and H. Lim. Scalable-width temporal edge detection for recursive background recovery in adaptive background modeling. *Appl. Soft Comput.*, 2013. 20, 25
- [489] B. Yeo, W. Lim, H. Lim, and W. Won. Extended fuzzy background modeling for moving vehicle detection using infrared vision. *IEICE Electron. Express*, pages 340–345, 2011. 20, 25
- [490] D. Yi, Z. Lei, S. Liao, and S.Z. Li. Shared representation learning for heterogeneous face recognition. *International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–15, 2015. 35, 40
- [491] L. Yin, X. Wei, Y. Sun, and J. Wang. A 3d facial expression database for facial behavior research. *In 7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216, 2006. 42
- [492] Pei Z., Xu H., Zhang Y., Guo M., and Yang Y. Face recognition via deep learning using data augmentation based on orthogonal experiments. *Electronics*, 2019. 106, 111, 112, 113, 114, 115
- [493] N. Zaghden, Mullot R., and Alimi M. Characterization of ancient document images composed by arabic and latin scripts. *Innovations in Information Technology (IIT)*, pages 124–127, 2011. 27, 29
- [494] N. Zaghden, Mullot R., and Alimi M. Categorizing ancient documents. *International Journal of Computer Science Issues (IJCSI)*, 2013. 27, 29
- [495] D. Zeng, X. Chen, M. Zhu, M. Goesele, and A. Kuijper. Background subtraction with real-time semantic segmentation. *IEEE Access*, 2019. 24, 26, 43
- [496] D. Zeng and M. Zhu. Combining background subtraction algorithms with convolutional neural network. *Preprint*, 2018. 22

- [497] H. Zhang and D. Xu. Fusing color and texture features for background model. *Fuzzy Systems and Knowledge Discovery, FSKD*, 2006. 19, 25, 57, 61
- [498] R. Zhang, X. Liao, and J. Xu. A background subtraction algorithm robust to intensity flicker based on ip camera. *J. Multimedia*, 2014. 20, 25
- [499] R. Zhang, X. Liu, J. Hu, K. Chang, and K. Liu. A fast method for moving object detection in video surveillance image. *Signal Image Video Process.*, 2017. 20, 25
- [500] Y. Zhang, X. Li, Z. Zhang, and F. Wu. Deep learning driven blockwise moving object detection with binary scene modeling. *Neurocomputing*, 2015. 20, 22, 25, 26
- [501] J. Zhangjian and W. Weiqiang. Detect foreground objects via adaptive fusing model in a hybrid feature space. *Pattern Recognition (PR)*, pages 2952–2961, 2014. 2, 4
- [502] C. Zhao, T. Cham, X. Ren, J. Cai, and H. Zhu. Background subtraction based on deep pixel distribution learning. In *IEEE international conference on multimedia and expo, ICME*, pages 1–6, 2018. vi, 63, 64, 65, 69, 70
- [503] C. Zhao, X. Wang, W.-K Cham, and T. Basar. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011. 26
- [504] L. Zhao, Tong Q., and Wang H. Study on moving-object-detection arithmetic based on w4 theory. *IEEE AIMSEC*, pages 4387–4390, 2011. 18
- [505] X. Zhao, Y. Chen, M. Tang, and J. Wang. Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network. *Preprint*, 2017. 22, 26
- [506] Y. Zhao and W. Liu. Fast robust foreground-background segmentation based on variable rate codebook method in bayesian framework for detecting objects of interest. *International Congress on Image and Signal Processing*, 2014. 20, 25
- [507] Z. Zhao, T. Bouwmans, and Zhang X. A fuzzy background modeling approach for motion detection in dynamic backgrounds. *Int. Conf. Communications in Computer and Information Science, CMSP*, pages 177–185, 2012. 57, 61
- [508] X. Zhou, X. Liu, A. Jiang, B. Yan, and C. Yang. Improving video segmentation by fusing depth cues and the vibe algorithm. *MDPI Sens.*, 2014. 20, 25
- [509] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In: *Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV*, pages 141–154, 2010. 9
- [510] J. Zhu, S. Liao, Z. Lei, D. Yi, and S. Li. Pedestrian attribute classification in surveillance : Database and evaluation. *International Conference on Computer Vision Workshops*, pages 631–338, 2013. 27, 29
- [511] X. Zhu. Semi-supervised learning literature survey. *Technical report 1530, Computer Sciences, University of Wisconsin-Madison*, 2005. 81
- [512] X. Zhu, S. Liao, Z. Lei, R. Liu, and S. Li. Feature correlation filter for face recognition. *International Conference on Biometrics*, pages 77–86, 2007. 34, 40

- [513] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonical-view faces in the wild with deep neural networks. *ArXiv*, pages 5325–5334, 2014. 38, 42
- [514] T. Zin, P. Tin, Toriu T., and Hama H. A new background subtraction method using bivariate poisson process. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 419–422, 2014. 19, 25
- [515] Z. Zivkovic and F. Heijden. Recursive unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 651–656, 2004. 20
- [516] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, pages 773–780, 2006. 19, 25, 57, 61
- [517] O. Zou, L. Ni, T. Zhang, and Q. Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.*, pages 2321–2325, 2015. 28, 29

Un système de récupération et de classification d'images extraites des caméras de vidéo-surveillance.

Résumé :

Dans cette thèse, nous présentons un descripteur robuste pour la soustraction d'arrière-plan fondé sur un algorithme de détection d'anomalies non-supervisé, appelé DeepSphere, capable de détecter les objets en mouvement dans les séquences vidéos. Contrairement aux algorithmes de séparation arrière-avant plan conventionnels, ce descripteur est tolérant aux variations d'illumination, robuste face aux bruits et aux régions d'arrière-plan dynamiques et détecte les objets de premier-plan sans utiliser de traitement d'image supplémentaire. En outre, ce descripteur exploite à la fois les autoencodeurs profonds et les méthodes d'apprentissage en hypersphère, ayant la capacité de capturer les dépendances spatio-temporelles entre les composants et à travers les pas de temps, d'apprendre de manière flexible une représentation non-linéaire des caractéristiques et de reconstruire les comportements normaux à partir des données d'entrée potentiellement anormales. Les représentations non linéaires de haute qualité apprises par l'autoencodeur aident l'hypersphère à mieux distinguer les cas anormaux en apprenant une frontière compacte séparant les données normaux et anormales. En adaptant cet algorithme à la tâche de soustraction d'arrière plan, les objets de premier plan sont bien capturés par DeepSphere et la qualité de la détection de ces objets est améliorée. Une fois que ces objets sont détectés (personnes/voitures...), une approche est proposée pour les classer en utilisant le réseau discriminatoire du DCGAN de manière semi-supervisée. Le discriminatoire est transformé en un classificateur multi-classes qui utilise à la fois un grand nombre de données non étiquetées et un très petit nombre de données étiquetées pour compenser la limite de manque de données et le coût élevé de collecte des données supplémentaires ou d'étiqueter toutes les données. Enfin, nous avons adopté une approche basée sur le model FaceNet pour la reconnaissance faciale des personnes extraites. De plus, nous avons étendu notre proposition par une méthode d'augmentation des données basée sur DCGANs au lieu d'utiliser les méthodes standard d'augmentation des données. Cela augmente non seulement la précision du modèle, mais réduit aussi de près de moitié le temps d'exécution et le temps d'apprentissage du réseau neuronal profond.

Mots clés : détection d'objets mobiles, soustraction d'arrière-plan, détection d'anomalies, DeepSphere, classification semi-supervisée, DCGANs, reconnaissance faciale.

A System For Retrieving and Classifying Images Extracted From Video Surveillance Cameras.

Summary:

In this thesis, we present a robust descriptor for background subtraction based on an unsupervised anomaly detection algorithm, called DeepSphere which is able to detect moving objects from video sequences. Unlike conventional background-foreground separation algorithms, this descriptor is less sensitive to noise and detects foreground objects without additional image processing. In addition, our proposal exploits both deep autoencoders and hypersphere learning methods, having the ability to capture spatio-temporal dependencies between components and through "timesteps", to flexibly learn a non-linear feature representation and reconstruct normal behaviors from potentially anomalous input data. The high quality non-linear representations learned by the autoencoder helps the hypersphere to better distinguish anomalous cases by learning a compact boundary separating normal and anomalous data. By adapting this algorithm to the background subtraction task, foreground objects are well captured by DeepSphere and the quality of detection of these objects is improved. Once these objects are detected (people / cars ...), an approach is proposed to classify them using a DCGAN discriminator network in a semi-supervised manner. The discriminator is transformed into a multi-class classifier that uses both a large number of unlabeled data and a very small number of labeled data to compensate the lack of data and the high cost of collecting additional data or labeling all the data. Finally, we have adopted an approach based on FaceNet model to recognize the extracted people through their faces. In addition, we extended our proposal with a data augmentation method based on DCGANs instead of using standard data augmentation methods. This not only increases the accuracy of the model, but also reduces the execution time and the deep neural network learning time by almost half.

Keywords: moving objects detection, background/foreground separation, anomaly detection, DeepSphere, semi-supervised classification, DCGANs, face recognition.



Laboratoire MIA - Mathématiques, Image et Applications
Faculté des Sciences et Technologies, Université de La
Rochelle, Avenue Michel Crépeau

Laboratoire MIRACL - Multimedia, InfoRmation systems and
Advanced Computing Laboratory
Faculté des Sciences Économiques et de Gestion de Sfax ,
Université de Sfax, Route de l'aéroport

17042 La Rochelle - Cedex 01 - France 3029 Sfax - Tunisie



