Multispectral images-based background subtraction using Codebook and deep learning approaches

Rongrong Liu

UBFC
UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

utbm
université de technologie
Belfort-Montbéliard

THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

École doctorale n°37
Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Automatique

par

RONGRONG LIU

Multispectral Images Based Background Subtraction Using Codebook and Deep Learning Approaches

Thèse présentée et soutenue à Belfort, le 30 Juin 2020

Composition du Jury :

GALLAND STÉPHANE	Professeur à l'Université Bourgogne Franche - Comté	Président et Examinateur
TALEB-AHMED ABDELMALIK	Professeur à l'Université Polytechnique des Hauts de France	Rapporteur
KHOUDOUR LOUAHDI	Directeur de Recherche à CEREMA	Rapporteur
NOYER JEAN-CHARLES	Professeur à l'Université du Littoral Cote d'Opale	Examinateur
MEURIE CYRIL	Chargé de recherche à IFSTTAR	Examinateur
EL BAGDOURI MOHAMMED	Professeur à l'Université Bourgogne Franche - Comté	Directeur de thèse
RUICHEK YASSINE	Professeur à l'Université Bourgogne Franche - Comté	Codirecteur de thèse

N° | X | X | X |

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Multispectral Images Based Background Subtraction Using Codebook and Deep Learning Approaches

Keywords: Background subtraction, Multispectral images, Codebook model, Deep learning

Abstract:

This dissertation aims to investigate the multispectral images in moving objects detection via background subtraction, both with classical and deep learning based methods. As an efficient and representative classical algorithm for background subtraction, the traditional Codebook has first been extended to multispectral case. In order to make the algorithm reliable and robust, a self-adaptive mechanism to select optimal parameters has then been proposed. In this frame, new criteria in the matching process are employed and new techniques to build the background model are designed, including box-based Codebook, dynamic Codebook and fusion strategy. The last attempt is to investigate the potential benefit of using multispectral images via convolutional neural networks. Based on the impressive algorithm FgSegNet_v2, the major contributions of this part lie in two aspects: (1) extracting three channels out of seven in the FluxData FD-1665 multispectral dataset to match the number of input channels of the deep model, and (2) proposing a new convolutional encoder to utilize all the multispectral channels available to further explore the information of multispectral images.

Titre : Multispectral Images Based Background Subtraction Using Codebook and Deep Learning Approaches

Mots-clés : Soustraction d'arrière-plan, Images multispectrales, Modèle de Codebook, Apprentissage profond

Résumé :

Cette thèse vise à étudier les images multispectrales pour la détection d'objets en mouvement par soustraction d'arrière-plan, à la fois avec des méthodes classiques et d'apprentissage profond. En tant qu'algorithme classique efficace et représentatif pour la soustraction de fond, l'algorithme Codebook traditionnel a d'abord été étendu au cas multispectral. Afin de rendre l'algorithme fiable et robuste, un mécanisme auto-adaptatif pour sélectionner les paramètres optimaux a ensuite été proposé. Dans ce cadre, de nouveaux critères dans le processus d'appariement sont employés et de nouvelles techniques pour construire le modèle d'arrière-plan sont conçues, y compris le Codebook de boîtes, le Codebook dynamique et la stratégie de fusion. La dernière tentative est d'étudier les avantages potentiels de l'utilisation d'images multispectrales via des réseaux de neurones convolutifs. Sur la base de l'algorithme impressionnant FgSegNet_v2, les principales contributions de ce travail reposent sur deux aspects: (1) extraire trois canaux sur sept de l'ensemble des données multispectrales du FluxData FD-1665 pour correspondre au nombre de canaux d'entrée du modèle profond, et (2) proposer un nouvel encodeur convolutionnel pour pouvoir utiliser tous les canaux multispectraux disponibles permettant d'explorer davantage les informations des images multispectrales.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France

ACKNOWLEDGEMENTS

I would like to particularly thank my supervisors, Prof. Yassine Ruichek and Prof. Mohammed El Bagdouri. This research could not have been accomplished without their sustaining support, guidance, patience and encouragement all the way.

I would like to acknowledge my thesis committee, Abdelmalik Taleb-Ahmed, Louahdi Khoudour, Jean-Charles Noyer, Cyril Meurie and Stéphane Galland, for accepting and evaluating my thesis work.

I am also grateful to the China Scholarship Council (CSC) for putting faith in my studies and providing the financial support for this research at the Technological University of Belfort Montbéliard (UTBM) in France.

Besides, I would like sincerely to thank a number of colleagues and friends: Jichao, Lichi, Dahuier, Longge, Pauline, Jinjian, Niuniu, Tao, Citlali, Shuang, Liulin, Vincent, Arthur, Jian, Karen, Monzer, Yan and Pik, for providing a welcoming and supportive community through this memorable journey.

In the end, a big big kiss to my beloved family.

Contents

I

CONTEXT AND PROBLEMS

1

INTRODUCTION

This chapter presents an introduction of the background subtraction, describes perspectives and challenges to investigate multispectral images in this computer vision task, and also the objectives of this work. Moreover, an outline of the thesis is included at the end of this chapter.

1.1/ BACKGROUND

Nowadays, we are flooded by digital images and videos. For example, as smart phones become increasingly popular, taking a photo or video and sharing it have never been easier. Besides, just a click on the internet, millions of images and videos can come to you. Computer vision, as a scientific discipline to enable computers to see and understand the content of digital images and videos, helps human process this vast number of digital data publicly available. The subject itself has been arisen to mimic human visual system in the 1960s [Aloimonos, 1990], and advances rapidly in recent years along with cheaper and more capable cameras, affordable processing power and increasingly mature vision algorithms.

Nowadays, computer vision is a very active and booming study field with a wide variety of applications based on many specialized tasks techniques. Some popular and attractive examples of applications are Autonomous Vehicles [Fridman et al., 2019] [Martínez-Díaz et al., 2018] [Litman, 2017], Amazon Go [Polacco et al.,], Google Lens [Achom Nishalakshmi et al., 2018], Face Recognition [Balaban, 2015] [Parkhi et al., 2015]. They rely on the following techniques to understand the digital images and videos, including but not limited to, object classification to group images into different categories, object detection to identify specific objects in an image, image segmentation to partition an image into separate multiple regions, instance segmentation to create masks for each individual object in the image.

Detection of moving objects in video sequences is one of the prerequisite and fundamental techniques in many computer vision systems. Three typical and widely used approaches in moving object detection [Maddalena et al., 2007] are temporal difference computing the difference between two or three consecutive frames [Shuigen et al., 2009] [Lipton et al., 1998], background subtraction distinguishing moving objects by comparing the current frame with a constructed background model [Liang et al., 2002] [Haritaoglu et al., 2000], and optical flow analysis providing all motion information [Thakoor et al., 2004] [Zhang et al., 2006] [Aslani et al., 2013] [Agarwal et al., 2016]. Among these three methods, background subtraction is the most efficient and widely adopted approach to discriminate foreground objects captured by a stationary camera and plays an crucial role due to its potential applications in the tasks discussed below [Bouwmans et al., 2019a].

Visual surveillance of human activities: In several environments, like traffic monitoring, parking lot management, train stations and airports surveillance, background subtraction technique is applied first to identify and track objects of interests.

Visual analysis of human activities: The detection of moving objects by background subtraction has also been used in sport, like soccer and tennis, to make important decisions and analyze athletic performance.

Visual observation of animals and insects: The intelligent visual observation of animals and insects needs to be simple and non-invasive. Thus, a video-based system is suitable to detect and track animals and insects in order to analyze their behavior or keep vision on any unusual activities.

Visual observation of natural environments: Background subtraction can be used to detect foreign objects in natural environments such as forest, ocean and river to protect the biodiversity.

Human-machine interaction: Several multimedia applications require human-machine interaction. For example, the gamer can observe his own image or silhouette composed into a virtual scene, built based on the background subtraction techniques.

Vision-based hand gesture recognition: Hand gesture recognition has been adopted in several applications like robotics, behaviour studies, human-computer interface, sign language interpretation and learning. This technique requires detecting moving hand area first in a video sequence before tracking and recognizing hand gesture.

Content-based video coding: In video coding for transmission applications, such as teleconferencing, digital movies and video phones, only the key frames with the moving objects are transmitted.

Background substitution: Background substitution, also called background cut or video matting, is to extract the foreground from the input video and then combine it with a new background.

1.2/ PERSPECTIVES

Background subtraction is also called foreground detection [Krungkaew et al., 2016] and foreground-background segmentation [Xu et al., 2016]. As the name suggests, it aims to detect foreground regions that are in motion from background of a video sequence. The first task of background subtraction is to construct and maintain a solid background model, which contains the static part of the scene or, more generally speaking, every information that can be considered as background given the characteristics of the observed scene. Then a binary mask, distinguishing pixels as background or foreground pixels, is obtained by performing a subtraction between the current frame and the constructed background model.

According to the Background Subtraction Website [Bouwmans, 2014], which provides exhaustive resources in this research domain, such as references, datasets, codes and also links to demonstration websites, the majority of the current background subtraction algorithms are focused on conventional visible images, mainly Red-Green-Blue (RGB), or transferring it to other color model, like YCbCr [Shah et al., 2015], Lab [Krungkaew et al., 2016] or YUV [Huang et al., 2016]. Commonly, the methods developed for visible light cameras are particularly sensitive to low light conditions and specular reflections.

In recent years, with the rise of different sensors, several background subtraction algorithms are proposed to utilize alternative kinds of data or integrates multiple complementary information to overcome these limitations [Zheng et al., 2019a], such as depth camera [Maddalena et al., 2018] [Camplani et al., 2013], light field sensing [Shimada et al., 2013] [Shimada et al., 2015], infrared camera [Qiu et al., 2019] [Yan et al., 2018]. Besides, thanks to the technological advances in video capture, now it is possible to capture and visualize a scene at various bands of the electromagnetic spectrum. Thus, as one of these alternatives, multispectral imaging has also been gaining in popularity.

The corresponding multispectral sequence, as the name implies, is a collection of several monochrome sequences of the same scene and each band, or channel, is taken with additional receptors sensitive to other frequencies of the visible light or frequencies beyond the visible light like the infrared region of electromagnetic continuum [Bouchech, 2015]. Multispectral imaging is related to hyperspectral imaging in that both provide increased spectral discrimination compared with traditional imaging methods. The difference is primarily in the number of bands employed and the degree of spectral resolution [Ferrato et al., 2013]. Whereas multispectral imaging generally refers to a set of 310 spectral bands, hyperspectral imaging often uses significantly larger numbers of bands [Rüfenacht et al., 2013] [Hagen et al., 2013].

Multispectral video processing is attracting increased interest in recent years, as it offers better spectral resolution, and different bands of multispectral video streams can enhance video analytics capabilities in different ways [Rüfenacht et al., 2013]. One of the early applications of multispectral imaging is to detect or track military targets [Goldberg et al., 2003]. Later, it is also employed in many other vision tasks [Viau et al., 2016] [Salerno et al., 2014] [Li et al., 2017], such as remote sensing [Shaw et al., 2003], weather forecasting [USGS, 2018] [Grolier et al., 1984], food control [Feng et al., 2018], face recognition [Bourlai et al., 2012] [Ice et al., 2012], ancient manuscripts analysis [Easton et al., 2003] [Diem et al., 2007], paintings investigation [Zhao et al., 2008].

Compared with visible images, background subtraction using multispectral images can be more interesting, for the intuitive fact that with more spectral bands, more information could be obtained, particularly for harsh environmental conditions characterized by unfavorable lighting and shadows, or around-the-clock applications, like surveillance and autonomous driving. Thanks to the recent advances in multispectral video acquisition technology, new products such as the FD-1665 Multispectral cameras from FluxData are commercially available to offer the possibility to record multispectral images of more than three spectral channels in the visible and near infrared part of the spectrum simultaneously. In [Benezeth et al., 2014a], Benezeth et el. have built a mutispectral dataset, which introduces novel opportunities and challenges for background subtraction. Based on the preliminary study conducted by the creators of this dataset, multispectral images have demonstrated good potential to be used for background subtraction.

1.3/ Objectives

As the opportunities and challenges always coexist, the main objective of this thesis is how to better use multispectral images for background subtraction based on the pioneering works in this field. To efficiently achieve this goal, the following attempts have been conducted as part of this research effort, with respect to both classical and deep learning based approaches.

As an efficient and representative classical method for background subtraction, Codebook algorithm proposed by Kim [Kim et al., 2005] is widely used with visual images. In order to investigate the advantages of multispectral sequences, the original Codebook algorithm is first adapted from traditional RGB to multispectral case.

In the original Codebook, there are four parameters that need to be tuned experimentally carefully to find the appropriate values and achieve satisfying results for a specific scene, which is always a really cumbersome and tricky task. What is more important for our research objective, when using the multispectral sequences, the parameters also have to be adjusted with different numbers of channels. Therefore, the second attempt in this thesis is to design a self-adaptive mechanism to select optimal parameters automatically.

In the framework of the multispectal self-adaptive Codebook algorithm, two perspectives are available to further improve the multispectral model. The first one is to employ a better feature representation with new robust feature descriptors or different features fusion. The other is to introduce other strategies to build the model to represent the background.

In this decade, deep learning has drawn much attention in the computer vision community and deep features obtained from Convolutional Neural Networks have been shown as powerful and effective image representations for various computer vision tasks. Thus, we'd like to make an attempt to follow the trend of deep learning and apply its concepts to background subtraction using multispectral images.

1.4/ Thesis organization

The rest of this thesis is organized as follows.

Chapter 2: In order to cope with challenges for background subtraction, numerous approaches have been proposed over the past years. This chapter conducts a literature review of the state of arts for background subtraction. Some well-known classical methods and state-of-art deep learning based approaches with RGB images are introduced, together with existing multispectral approaches using the same dataset as the one used in this thesis.

Chapter 3: For better explanation, the original Codebook algorithm is first stated in five important issues, namely background model initialization, matching process, background model updating strategy, background model refining and foreground detection. Then it is adapted to multispectral case to investigate the advantages of multispectral sequences for the task of background subtraction. Besides, a detailed description for the adopted multispectral dataset and evaluation metrics are also presented in this chapter.

Chapter 4: To get rid of the cumbersome and tricky task to select the optimal parameters for various scenes and different numbers of multispectral channels, a self-adaptive mechanism is proposed by calculating iteratively and recording additional statistical information vectors for each codeword. In the multispectal self-adaptive Codebook framework, improvements have been proposed in two aspects. The first one is to introduce a new feature descriptor called spectral information divergence in the matching process, while the latter aspect includes three techniques to build the background model, namely, box-based Codebook, dynamic Codebook and fusion strategy.

Chapter 5: The fashionable deep learning is explored for background subtraction with mutispectral images. Firstly, three channels are extracted from the multispectral images to match the number of input channels of the pretrained deep model, aiming to investigating the possible improvements against RGB. Secondly, a new convolutional encoder was proposed to utilize all the multispectral channels available to further explore the information of multispectral images. To the best of our knowledge, this work is the first attempt to investigate the potential benefits of using multispectral information via deep features learned with convolutional neural networks for the background subtraction task.

Chapter 6: The work of this thesis is summarised and the future works are discussed in the last chapter.

2

State of Art

2.1/ Introduction

Background subtraction is a well studied field in the domain of computer vision and many algorithms have been designed to separate the moving objects (foreground) from the static information (background), as witnessed by several surveys such as [Bouwmans, 2014] [Sobral et al., 2014] [Bouwmans et al., 2017b] [Kalsotra et al., 2019] [Bouwmans et al., 2019b]. A quick search for “background subtraction” on IEEE Xplore returns over 2200 publications in the last ten years (2010-2020).

Fig.2.1 shows several processing modules for the majority of background subtraction algorithms, which are background model initialization, background model maintenance, and foreground detection. As the names suggest, background model initialization regards the initializing step to create the background model [Bouwmans et al., 2017a]; background model maintenance concerns the adaptation of the model to the background changes, and foreground detection is to compare each incoming frame with the built background model to acquire a binary mask distinguishing between foreground pixels and background ones.

To obtain this foreground mask, the simplest and most straightforward technique is an interframe difference between the current frame and the background reference frame with a global static threshold. However, detecting moving objects is not as easy as it may first appear, due to the complexity and challenges of real-world scenes. For example, finding a good empty background reference frame is always impossible in the case of dynamic background, and illumination changes may also make the global static threshold an inferior choice.

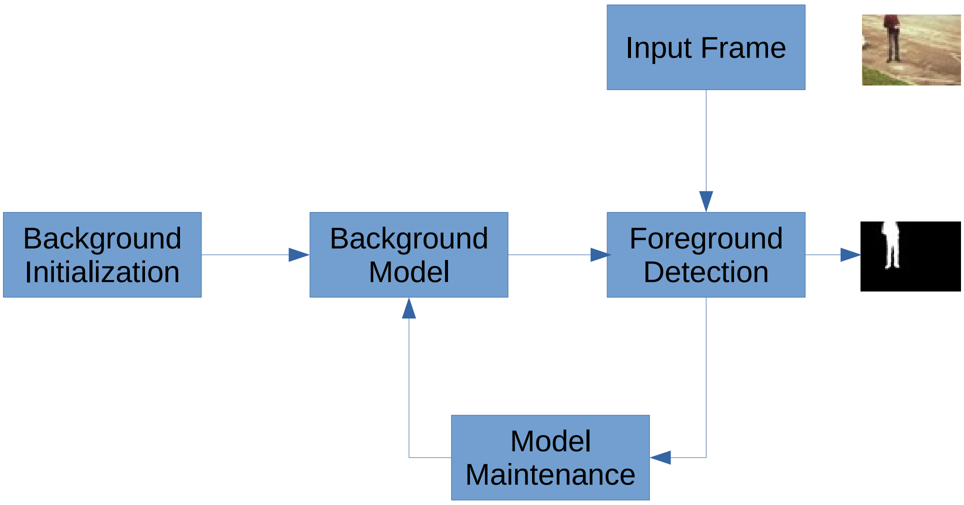

Figure 2.1: Background subtraction process

Table 2.1 summarizes the challenges of background subtraction considering three factors: background, foreground and camera [Kalsotra et al., 2019] [Bouwmans et al., 2019a]. We will make a brief introduction for twofold reasons. First, video sequences in the available datasets are typically structured based on the challenges of background subtraction summarized here, including the multispectral dataset used in this thesis. Second, it can offer a reference for later construction of larger multispectral datasets for exhaustive evaluation of multispectral information for background subtraction.

Illumination changes: The illumination changes affect the pixels in the current video frame and interrupt background model. For example, moving clouds in the sky would lead to gradual illumination changes and switching the light in a room can be a sudden illumination change. These illumination changes would induce false positives.

Dynamic background: The background contains some periodical or irregular movements, such as waving trees, traffic lights, moving escalator, and swaying curtains. As a consequence, parts of the background in the video frame do not overlap with the corresponding parts in the background image. Hence, dynamic background may also cause false positives.

Shadows: The shadows cast by moving objects or fixed background often bring additional difficulty to background subtraction. Overlapping shadows of foreground regions for example hinder their separation and classification and always result into object merging and object distortion.

Table 2.1: Background subtraction challenges

Background	Foreground	Camera
Illumination Changes	Shadows	Video Noise
Dynamic Background	Camouflage	Moving Camera
Shadows	Intermittent Object Motion	Camera Jitter
Challenging Weather	Occlusion	
Bootstrapping		
Moved Background Objects		
Night Videos		

Challenging weather: Videos recorded in challenging weather conditions with low-visibility, such as fog, rain, snow, and air turbulence, would complicate the background model construction and generate false detection.

Night videos: The segmentation of foreground objects and their contours become difficult for lack of sufficient illumination in night scenes. Besides, strong headlights would cause halos and reflections on the street, resulting with false positives.

Bootstrapping: In some environments, a period absent of moving objects is not available for background initialization and thus makes it difficult to compute a representative background image and generates defective background model with some algorithms.

Moved background objects: A background object can be moved. This information needs to be noticed by the background model maintenance strategy so that these objects will not be considered as foreground afterwards.

Camouflage: When some moving objects poorly differ from the appearance of background, for example, in cases where foreground objects and background scene have a similar color, the foreground objects are more likely to be falsely labeled as background.

Intermittent object motion: Foreground objects that are embedded into the background scene and start moving after background initialization are the so-called ghosts. In contrast, foreground objects that stop moving for a certain amount of time, fall into the category of intermittent object motion, such as a sequence of cars that arrive, stop at a street light and then move away.

Occlusion: Occlusion is also a very common challenge, which complicates the computation of background model. For example, a walking person passing behind a tree is partially occluded in several frames.

Video noise: Video signal is generally superimposed by noise. Coping with such degraded signals affected by different types of noise, such as sensor noise or compression artifacts, brings in more difficulties to background subtraction.

Moving camera: Background subtraction can be also used in applications in which cameras are slowly moving, which need more subsequent process to compensate this background motion.

Camera jitter: In some cases, it is possible that the camera itself is frequently in movement due to physical influence such as wind. Similar to the cases of dynamic background and moving camera, the pixel locations between the video and background frame do not overlap anymore.

In order to cope with these challenges illustrated above, numerous approaches have been proposed over the past years for background subtraction. They differentiate in the way they construct the background model, the way the model is updated over time, and how foreground pixels are detected.

In this chapter, we will first review the algorithms for traditional RGB images with both classical and deep learning based methods. Since the objective of this thesis is to investigate multispectral images for the task of background subtraction, we will go on to introduce the existing multispectral approaches. These algorithms have been tested on the same dataset we have adopted, and will be later used for quantitative comparison with the algorithms proposed in this thesis. What needs to be noted is that the objective of this chapter is not to present a comprehensive survey of background subtraction techniques, like the Background Subtraction Website [Bouwmans, 2014], but to present the most important reference methods and recent state-of-the-art improvements in this domain.

To better present the state of art, the remainder of this chapter is arranged as follows. The well-known classical methods and state-of-art deep learning based approaches for background subtraction with RGB images are presented in Section 2.2. The algorithms available in literature dealing with multispectral images are reviewed in Section 2.3. In Section 2.4, we conclude this chapter.

2.2/ Background subtraction methods with RGB images

The majority of the current background subtraction algorithms are focused on conventional RGB images. Detailed overviews are available in [Bouwmans, 2014] [Sobral et al., 2014]. This section will offer a brief review of classical and deep learning based background subtraction methods on RGB images.

2.2.1/ Classical approaches for background subtraction with RGB images

There are numerous different classical background subtraction techniques for visible RGB images in literature. Representative conventional methods include parametric Gaussian Mixture Models (GMM) [Stauffer et al., 2000], nonparametric Kernel Density Estimation (KDE) [Elgammal et al., 2000], Codebook [Kim et al., 2005] and ViBe [Barnich et al., 2010].

In this subsection, two methods, namely GMM and KDE, which are older but frequently cited, are firstly introduced, respectively. As a major motivation of the contribution of this thesis, the Codebook algorithm is going to be presented in detail in Chapter 3. Then, some more recent conventional approaches that perform well on CDnet2014 dataset [Wang et al., 2014], namely, ViBe based estimations [Hofmann et al., 2012] [St-Charles et al., 2014] [St-Charles et al., 2015], In Unity There Is Strength (IUTIS) [Bianco et al., 2017] and Semantic background subtraction [Braham et al., 2017] [Zeng et al., 2019], are discussed later in this chapter.

2.2.1.1/ Gaussian mixture model

Gaussian mixture models (GMM), proposed by Stauffer and Grimson [Grimson et al., 1998] [Stauffer et al., 1999] [Stauffer et al., 2000], are the most popular parametric models used for background subtraction. The authors model each pixel in the scene by a mixture of K Gaussian distributions, where K is a small number from 3 to 5. Different Gaussians are assumed to represent different intensity in the RGB color space. The weight parameters of the mixture represent the time proportion that a certain intensity stays in the scene.

The probability density function $p(\boldsymbol{x}_t)$ of observing the current pixel value is considered given by the following equation in the multidimensional case:

$$p\left(\boldsymbol{x}_t\right) = \sum_{i=1}^{K} \phi_{i,t} \mathcal{N}\left(\boldsymbol{x}_t | \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}\right), \tag{2.1}$$

where $\boldsymbol{x}_t$ is the input at time t; $\phi_{i,t}$ (where $i = 1, \ldots, K$) are the combination weights, which add up to 1 ($\sum_{i=1}^{K} \phi_{i,t} = 1$); and K is the number of Gaussian components. The i^{th} Gaussian density is

$$\mathcal{N}\left(\boldsymbol{x}_t | \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}\right) = \frac{1}{\sqrt{(2\pi)^D \left|\boldsymbol{\Sigma}_{i,t}\right|}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_t - \boldsymbol{\mu}_{i,t}\right)^{\top} \boldsymbol{\Sigma}_{i,t}^{-1} \left(\boldsymbol{x}_t - \boldsymbol{\mu}_{i,t}\right)\right) \tag{2.2}$$

where $\boldsymbol{\mu}_{i,t}$ and $\boldsymbol{\Sigma}_{i,t}$ are the mean vector and the covariance matrix of this component density, respectively.

Based on the fact that static background pixels trend to cluster tightly while moving objects form wider clusters due to different reflecting surfaces during the movement, the probable background intensities are those which stay longer and more static. Thus, the K distributions are ordered based on the fitness value $\phi_{i,t}/\boldsymbol{\Sigma}_{i,t}$ and the first B distributions are used as the background model of the scene, where B is estimated with a threshold. Once the background model is established, foreground pixels are detected if they are more than 2.5 standard deviations away from any of the B distributions.

Several improvements and extensions have been proposed by automatic updating of the GMM component number and learning rate [KaewTraKulPong et al., 2002], [Zivkovic, 2004], using adaptive thresholds [McHugh et al., 2009], or by replacing the Gaussian distribution with more flexible ones [Elguebaly et al., 2014]. These improvements can achieve some automation in adapting the GMM parameters to background dynamics [Boulmerka et al., 2017].

2.2.1.2/ Kernel density estimation

As a representative of nonparametric pixel-level background modeling approaches, Kernel Density Estimation (KDE) [Elgammal et al., 2000] [Elgammal et al., 2002] is also frequently cited in background subtraction domain. Unlike GMM introduced above, it estimates the probability density function from many samples without any prior assumptions.

A simplified review of this algorithm can be summarised as below. Let x_t denote the observed intensity at time t. The probability of this observation can be estimated with the most recent samples $x_1, \ldots, x_i, \ldots, x_N$ for this pixel by Equation 2.3.

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} K_(\sigma)(x_t - x_i) \tag{2.3}$$

where $K_(\sigma)$ is a kernel function with bandwidth σ, which needs to be chosen carefully, as small a bandwidth will lead to a ragged density estimate while too wide a bandwidth will lead to an over-smoothed density estimate [Duda et al., 2012]. Thus, it is better to assign different bandwidths for different pixels. This estimation can also be easily generalized to color space by kernel products. Accordingly, different kernel bandwidths can be chosen for each color channel. The pixel is considered to be a foreground pixel if

$$Pr(x_t) \leq threshold \tag{2.4}$$

Except that KDE does not require the definition of the model's parameters, this model can deal well with changes in background. However, it is very time consuming and memory demanding. Besides, shadows and illumination changes are not well handled using this approach [Boulmerka et al., 2017]. Some improvements have been proposed to overcome these problems [Mittal et al., 2004] [Sheikh et al., 2005] [Zivkovic et al., 2006].

2.2.1.3/ ViBe based estimation

Barnich and Van Droogenbroeck [Barnich et al., 2010] have proposed a method called ViBe, which stands for Visual Background extractor. Unlike the aforementioned GMM and KDE, no estimation of the probability density function of the background pixel is performed in the mechanism of ViBe. The per-pixel background model is made of a collection of N pixel values

$$M = \{v_1, v_2, \cdots, v_N\} \tag{2.5}$$

taken in previous frames.

For foreground-background segmentation of the current pixel value $v(x)$, we compare it to the closest values in the model M by defining a sphere $S_R(v(x))$ of radius R centered on $v(x)$. The pixel $v(x)$ is then classified as background if the cardinality of the set intersection

$$S_R(v(x)) \cap \{v_1, v_2, \cdots, v_N\} \tag{2.6}$$

is not smaller than a given threshold. That is to say, if a pixel in the new input frame is similar to a portion of its recently observed samples, it is considered as background and may be selected to update the background model. This sample consensus approach has motivated various more advanced models for background subtraction in the last decade. Some representatives are listed here.

Hofmann et al. [Hofmann et al., 2012] have proposed the Pixel-Based Adaptive Segmenter (PBAS). The decision threshold for each pixel is not fixed like in ViBe, but can dynamically changes along with dynamic background over time, by recording and updating the minimal decision distance at each observation. Besides, the controller with feedback loop is also designed for the learning parameter in the updating process.

Another improvement of Vibe is the Self-Balanced SENsitivity SEgmenter (SuBSENSE) algorithm [St-Charles et al., 2014] proposed by St-Charles et al., which uses both color and Local Binary Similarity Pattern (LBSP) features to improve the spatial awareness. It also has a pixel-level feedback scheme that dynamically adjusts the decision threshold.

From the same authors, the Pixel-based Adaptive Word Consensus Segmentation (PAWCS) method [St-Charles et al., 2015] is an extension of SubSCENE that implements a real-time internal parameter updating strategy. It also adds a persistence indicator feature to the color-LBSP pixel representation to group a background word and represent the information at the pixel level.

2.2.1.4/ In unity there is strength

Another promising solution for foreground-background segmentation lies in the combination or fusion of different existing detection algorithms. For example, in the CDnet dataset paper [Wang et al., 2014], the authors have applied majority vote strategy on several algorithms and evaluated the performance with different numbers of detection methods used. Instead of this straightforward trial-and-error procedure and the simple majority vote fusion, Bianco et al. [Bianco et al., 2017] have proposed a mechanism which exploits genetic programming for automatical algorithm selection, combination and processing. It is termed as In Unity There Is Strength (IUTIS) and the IUTIS-5, which uses the top five algorithms on the CDnet2014 (until July 2014), currently outperforms all the other unsupervised algorithms on the same dataset.

Genetic programming belongs to the category of evolutionary algorithms, which mimic the biological evolution process that individuals in a population evolve and compete with each other toward a defined goal [Burke et al., 2010] [Pappa et al., 2014]. The inputs of this algorithm are the set of the binary foreground-background masks obtained by existing algorithms $\zeta = \{C_k\}_{k=1}^n$, the candidate solution C_0 is evolved using the functional symbol set, corresponding to the operations performed on the input masks, and terminal symbols set. Given a set of n predefined algorithms, the fitness function $f(C_0)$ of C_0 is defined by a weighted average of three components with Equation 2.7. The optimization process is based on a set of standard performance measure $M = \{m_1, \cdots, m_M\}$. Specifically, the measures are recall, precision, false positive ratio, false negative ratio, percentages of wrong classification and F-measure, as it is used on CDnet2014 dataset [Wang et al., 2014].

$$f(C_0) = \frac{1}{M}\sum_{j=1}^{M}(\omega_0 \cdot rank(C_0; \left\{m_j(C_k(v))\right\}_{k=1}^{n}) + \omega_1 \cdot \sum_{j=1}^{M} P_1^j(C_0) + \omega_2 \cdot P_2(C_0)) \tag{2.7}$$

where $rank(C_0; \cdot)$ represents the rank of the candidate solution C_0 according to the measure m_j; $P_1^j(C_0)$ is the distance between C_0 and the best algorithm respect to the measure m_j in existing algorithm set ζ; the penalty term $P_2(C_0)$ is defined as

$$P_2(C_0) = \frac{number\ of\ algorithms\ selected\ in\ C_0}{number\ of\ algorithms\ in\ \zeta} \tag{2.8}$$

which guides the genetic programming to select a small number of algorithms in ζ. Besides, ω_0, ω_1 and ω_2 in Equation 2.7 are three weights indicating the relative contribution of these three components.

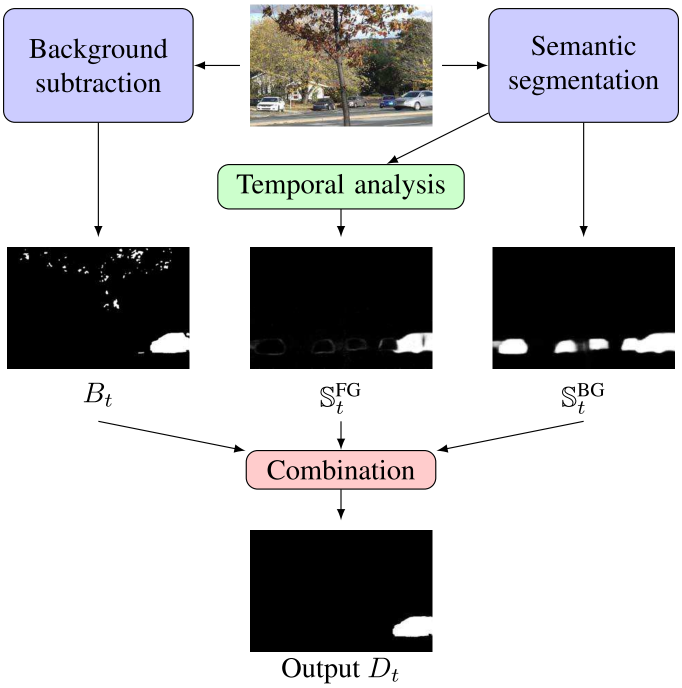

Figure 2.2: Flowchart of semantic background subtraction algorithm [Braham et al., 2017]

2.2.1.5/ Semantic background subtraction

Another promising perspective proposed by Braham et al. [Braham et al., 2017] introduces object-level semantics to the task of background subtraction. The flowchart of this algorithm is illustrated in Fig.2.2.

Let BG and FG denote background and foreground, separately. $B = \{BG, FG\}$ is the pixel-level result from any background subtraction algorithm, while S^{BG} and S^{FG} are the two probability signals derived from the semantics. The decision of the final combination $D = \{BG, FG\}$ is obtained by the Equation 2.9

$$D(x) = \begin{cases} BG \quad if \;\; S^{BG} \leq \tau_{BG} \\ FG \quad if \;\; S^{FG} \geq \tau_{FG} \end{cases} \tag{2.9}$$

with two thresholds τ_{BG} and τ_{FG}. If the two conditions in Equation 2.9 are not met, which means the semantics do not provide enough information to classify the pixel as background or foreground, the final decision will be acquired based on the background subtraction algorithm:

Table 2.2: Top six deep learning based approaches reported on CDnet2014 dataset [Wang et al., 2014]

Approaches	Average F-measure	Network architecture
FgSegNet_v2 [Lim et al., 2019]	0.9847	Convolutional Neural Network
FgSegNet_S [Lim et al., 2018]	0.9804	Convolutional Neural Network
FgSegNet_M [Lim et al.,]	0.9770	Convolutional Neural Network
BSPVGAN [Zheng et al., 2019b]	0.9501	Generative Adversarial Network
BSGAN [Zheng et al., 2018a]	0.9339	Generative Adversarial Network
Cascade CNN [Wang et al., 2017]	0.9209	Convolutional Neural Network

$$D(x) = B(x) \tag{2.10}$$

Based on the work of Braham et al., an improved algorithm called Background Subtraction with real-time Semantic Segmentation (RTSS) [Zeng et al., 2019] has been proposed lately, where the decision of the final combination is used to feedback and guide the background model updating.

2.2.2/ Deep learning based approaches for background subtraction with RGB images

Recently, Convolutional Neural Networks (CNNs or ConvNets) based approaches have been demonstrated to be powerful frameworks for background subtraction in videos acquired by static cameras [Bouwmans et al., 2019b]. Compared with manually selected features such as Scale-Invariant Feature Transform (SIFT) [Lowe, 1999], Histogram of Oriented Gradients (HOG) [Dalal et al., 2005], or Local Binary Pattern (LBP) [Guo et al., 2010], CNNs have the ability to learn features from multiple layers that best fit a given set of data.

These deep learning based algorithms have shown impressive detection results and outperformed the classical methods by large margins. The top six background subtraction algorithms on the well-known large-scale change detection dataset CDnet [Goyette et al., 2012] [Wang et al., 2014] all based on deep neural networks in the category of supervised methods and are presented in Table 2.2.

These leading deep neural networks based background subtraction methods work with RGB images. In this section, we will describe several CNNs that achieve the top perfor-

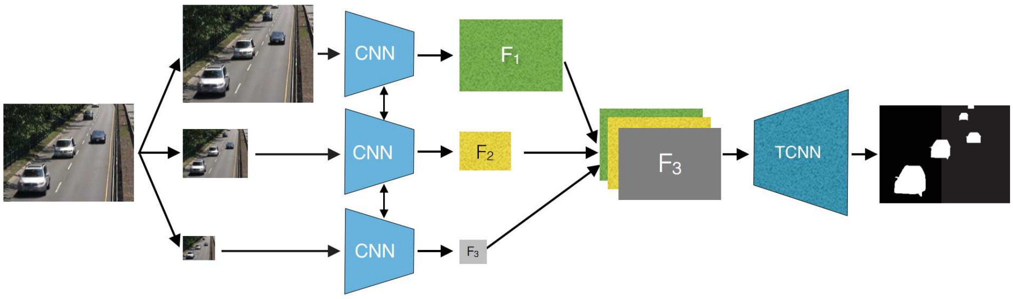

Figure 2.3: FgSegNet_M network architecture [Lim et al., 2018]

mances in the state of the art regarding background subtraction, or change detection.

2.2.2.1/ FgSegNet_M

FgSegNet_M, proposed in [Lim et al., 2018], is a convolutional neural network with encoder-decoder structure and can be trained end-to-end under a triplet framework, as shown in Fig. 2.3, where FgSegNet is short for foreground segmentation networks and M indicates multiple inputs.

The encoder consists of three CNNs that process in parallel with the same input images in different scales to generate multi-scale deep features. Each feature CNN encoder adopts the first four blocks of the pretrained VGG16 network with minor modification. The corresponding architecture is illustrated in Fig. 2.4, where the input is raw RGB images with a size of W × H × 3.

After the triplet encoder, three feature maps, namely F_1, F_2 and F_3, are obtained simultaneously with different sizes. F_2 and F_3 will be firstly upscaled with simple nearest neighbor interpolation to match the size of F_1, then all the three feature maps are concatenated along the depth dimension to be a combined feature map and fed into the attached decoder.

The decoder is novel transposed convolutional neural network to map the features to a foreground probability map pixel by pixel. It is made up with eleven transposed convolutional layers, as shown in Fig. 2.5. The output is a dense probability mask for each pixel, with a value ranging between 0 and 1, indicating the probability of being a foreground pixel. It has the same size as the original input images.

For the last step, thresholding is applied to the probability mask image to obtain a binary

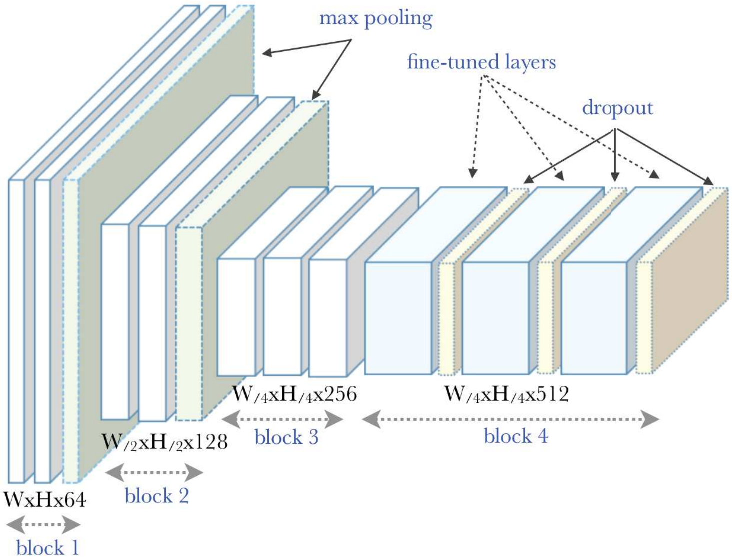

Figure 2.4: Encoder Architecture of each CNN in Triplet Network [Lim et al., 2018]

foreground background segmentation result. In the original paper, a fixed threshold of 0.8 has been chosen based on the classification performance in the experiments with different thresholds [Lim et al., 2018].

2.2.2.2/ FgSegNet_S

In the paper [Lim et al., 2018], Lim et al. have also proposed another multi-scale architecture called FgSegNet_S for foreground background segmentation, where S stands for single input. Instead of downsampling the input pictures to use both higher level and lower level contextual information, the authors have designed a Feature Pooling Module (FPM) to extract multi-scale features from a single-input encoder. In this mechanism, the features obtained from the encoder CNN in Fig. 2.4 first need to be passed through FPM before they are fed to the following decoder network.

Fig. 2.6 illustrates the FPM structure, which has been inspired by the promising results of dilated convolutions for semantic segmentation in [Yu et al., 2015] [Chen et al., 2017] and [Chen et al., 2018]. FPM connects a max-pooling layer and four parallel dilated convolutions with different dilation rates, which operate on same feature maps from encoder layers, followed by a Batch Normalization (BN) layer [Ioffe et al., 2015] and Spatial Dropout (SD) [Tompson et al., 2015].

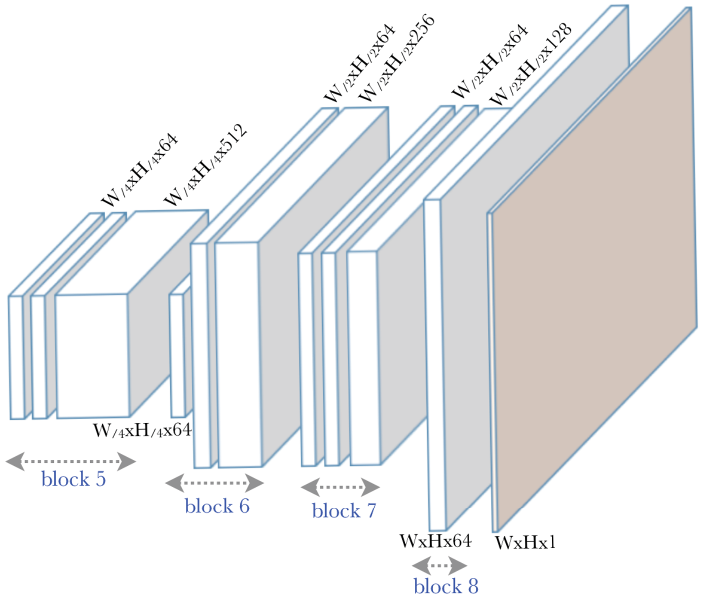

Figure 2.5: Decoder Configuration [Lim et al., 2018]

like the manner in the FgSegNet_M, the five sets of extracted features are subsequently concatenated along the depth axis to be a combined feature map, which is then used to output a pixel-level foreground probability map with the same decoder network introduced for FgSegNet_M. Finally the binary segmentation label for each pixel is obtained via thresholding.

2.2.2.3/ Bayesian generative adversarial networks

The Generative Adversarial Networks (GANs) were invented by Goodfellow et al. [Goodfellow et al., 2014] and were described as ”the coolest idea in machine learning in the last twenty years” by Yann LeCun during a seminar in 2016 [LeCun, 2016]. Given a training set, this technique learns to generate new data with the same statistics as the training set. Though originally proposed as a form of generative model for unsupervised learning, GANs have also been proven useful for semi-supervised learning [Salimans et al., 2016], fully supervised learning [Isola et al., 2017], and reinforcement learning [Ho et al., 2016].

Recently, GANs has shown impressive results in computer vision such as generation of examples for image datasets [Radford et al., 2015], super resolution [Ledig et al., 2017], generation of photographs of human faces [Karras et al., 2017] and so on. Zheng et al.

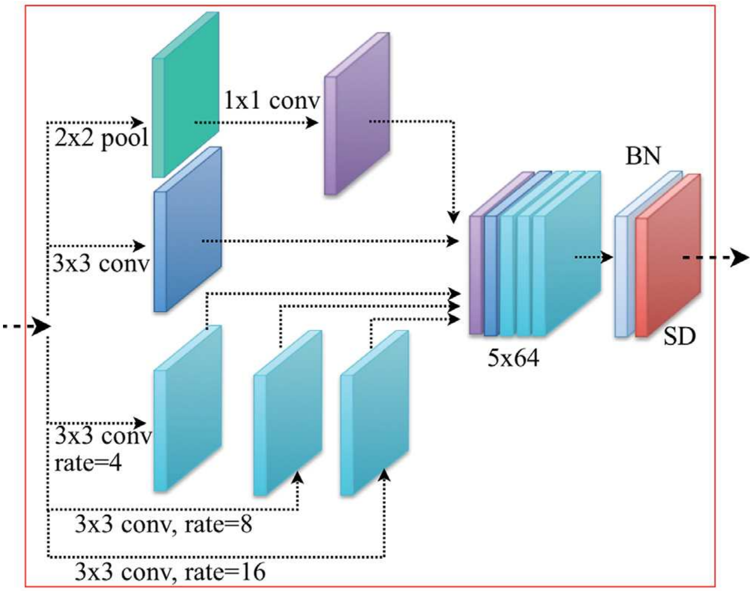

Figure 2.6: Feature Pooling Module [Lim et al., 2018]

have proposed background subtraction algorithms with Bayesian Generative Adversarial Networks in [Zheng et al., 2018a] and [Zheng et al., 2019b], based on the research work of Saatci and Wilson [Saatci et al., 2017].

The background subtraction problem is viewed as a binary classification problem about background and foreground for each image pixel. In their work, parallel vision first constructs virtual foreground-background segmentation images to simulate and represent complex real foreground-background segmentation by using Bayesian GANs, which is trained with the Deep Convolutional Generative Adversarial Networks (BSGAN) [Radford et al., 2015]. Computational experiments are then utilized to train and evaluate a variety of vision algorithms. The pipeline of the algorithm is presented in Fig. 2.7, which includes four steps.

To obtain a gray background image in the first step, the temporal median value is calculated after the RGB frames are converted by Equation 2.11 to gray domain (denoted with Y), which need to be further normalized with respect to the interval between 0 and 1. This simple median filter method works well if each pixel is visible for at least half time during the observing period. Unless, more complex scene background modeling approaches like [Laugraud et al., 2015] will be adopted.

$$Y = 0.229 \times R + 0.587 \times G + 0.114 \times B \tag{2.11}$$

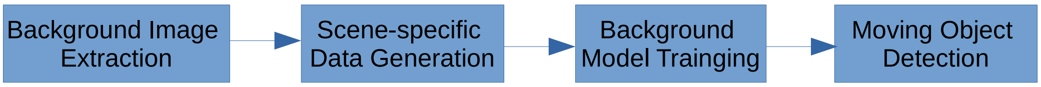
Background Image Extraction
Scene-specific Data Generation
Background Model Trainging
Moving Object Detection

Figure 2.7: BSGAN pipeline

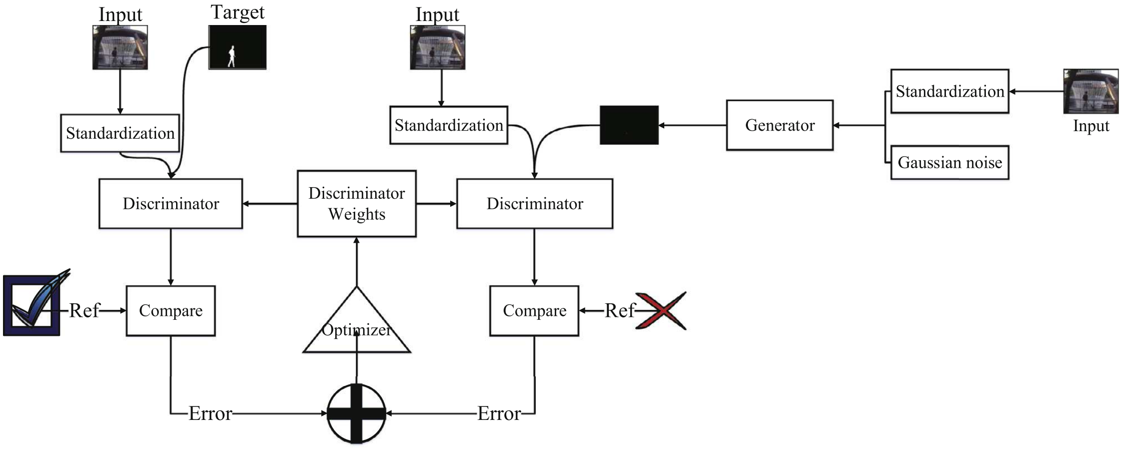

Figure 2.8: Flow Chart of Bayesian GANs [Zheng et al., 2019b]

For the second step, a scene-specific dataset to train the network is established with a $T \times T$ 2-channel images patch, where one channel represents the background patch extracted from the gray median image in the first step, and the other is for the input patch. The corresponding target value is assigned by:

$$t(x) = \begin{cases} 1 & if\ class(p_c) = foreground \\ 0 & if\ class(p_c) = background \end{cases} \tag{2.12}$$

where p_c denotes the central pixel for the patch. T is set as 27 by the authors.

Fig. 2.8 illustrates the flow chart of training process in detail. The inputs of the generator network are the normalized image and Gaussian noise. Then the generated foreground-background segmentation image is fed to the discriminator network, together with the normalized image, to determine 0 or 1 based on the input of groundtruth, which will be then fed back to the generator network. Thus both the generator and discriminator networks are updated iteratively.

2.2.2.4/ CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS

Bakkay et el. proposed another deep background subtraction method based on Conditional Generative Adversarial Networks (cGAN) in [Bakkay et al., 2018]. Following the routine of the GAN design, the proposed model consists of a generator network and a discriminator network, where the former one learns the mapping from the observing image and background to the foreground mask, and the latter optimizes the loss function, which is a combination of a conventional binary cross-entropy loss and an adversarial term, to train this mapping.

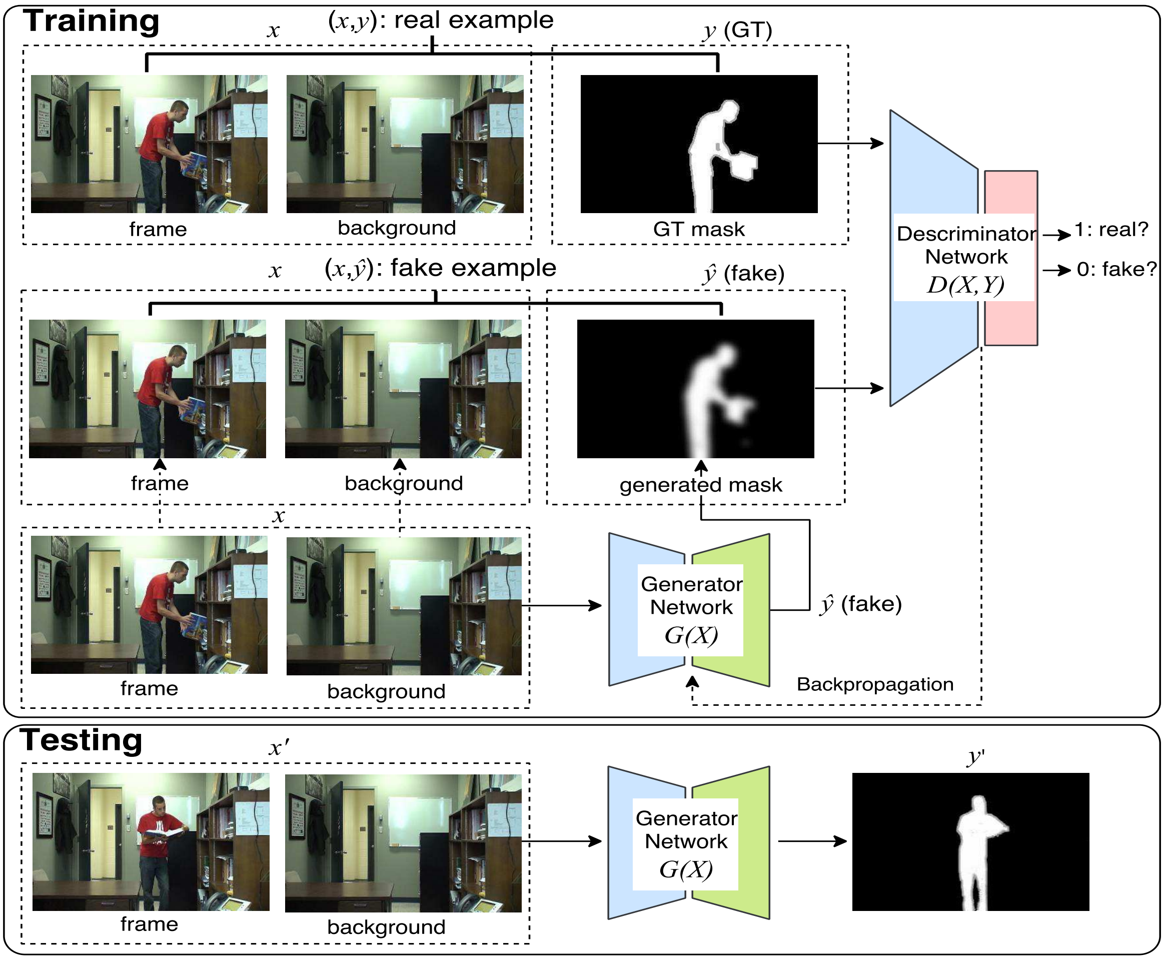

Figure 2.9: Folw Chart of Conditional GAN [Bakkay et al., 2018]

Fig. 2.9. illustrates the flow chart of cGAN. In the training phase, the generater network produces a predicted foreground mask notated as ŷ from the input pair composed of an image and a background. Then the discriminator network compares ground truth and predicted output under condition of the input pair. Backpropagating the two networks leads to generate better masks, which cannot be distinguished from ground truth by the discriminator. In the testing phase, the trained generater network outputs a foreground-background segmentation mask.

Fig. 2.10 shows the architecture of the proposed model. The discriminator network on top includes four convolutional and downsampling layers, and the last convolutional layer is followed by one fully connected layer to transform the features map in a 1 dimensional vector. The generator network below follows the encoder-decoder structure of U-net [Ronneberger et al., 2015]. The encoder is made up of eight convolutional layers as proposed in [Isola et al., 2017], where the middle six layers are six ResNet blocks and adopt the weights trained for ResNet-101 [He et al., 2016] while the parameters are randomly initialized. The decoder is built in a similar way with eight transpose convolutional layers with a reverse layers ordering, where all the weights are trainable.

2.2.2.5/ Cascade CNN

An end-to-end deep model was proposed by Wang et al. in [Wang et al., 2017], based on a multi-resolution CNN. Several CNN configurations and training strategies have been explored. The architecture of the basic CNN model is illustrated in Fig. 2.11. It consists of four convolutional layers, among which the first two come with a 2×2 maxpooling layer, and two fully connected layers.

One of the drawbacks for the basic CNN model lies in the wrong classification of the uniform textureless areas for large moving objects, because of its fixed input patch size. In order to solve this, a multi-scale CNN model has been implemented and illustrated in Fig. 2.12. The input images are first resized with two different scales before they are fed to the same basic CNN model and upscaled back. The final foreground-background mask is obtained with an average pooling.

In order to model the dependencies among adjacent pixels and thus enforce spatial coherence, a cascaded CNN model has been further implemented by the authors. As shown in Fig. 2.13, the first CNN model is used to compute a foreground probability map which is then concatenated with the original frame and fed to a second CNN model to compute refined probability map. The input of the second CNN is thus an image with four channels, namely red, green, blue, and a foreground likelihood probability.

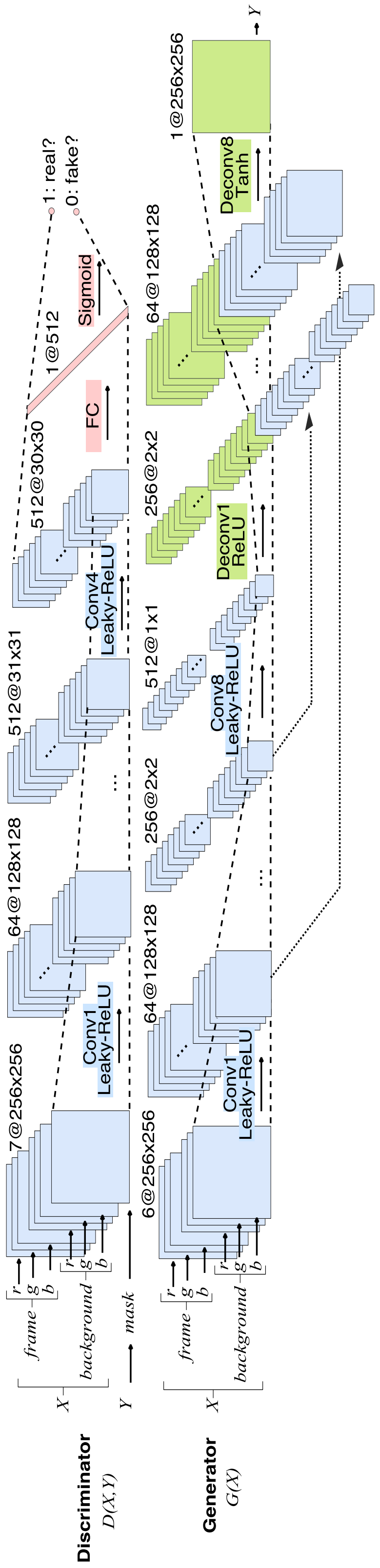

Figure 2.10: Architecture of Conditional GAN model [Bakkay et al., 2018]

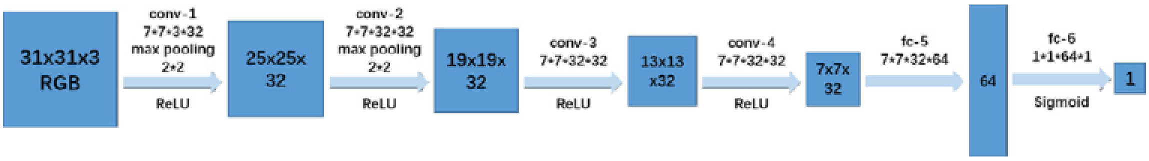

Figure 2.11: Architecture of the Basic CNN Model [Wang et al., 2017]

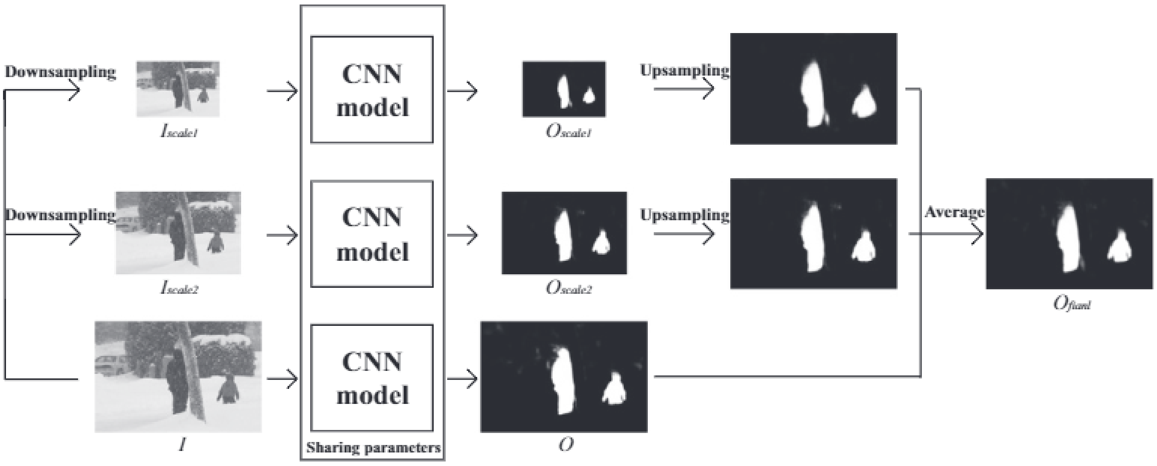

Figure 2.12: Architecture of the Multiscale CNN Model [Wang et al., 2017]

2.3/ Background subtraction methods with mutispectral images

In this section, we will introduce several background subtraction approaches based on multispectral images. They have been tested on the same dataset [Benezeth et al., 2014a] we have adopted in this theis and will be later used for quantitative comparison with the approaches proposed in this paper.

2.3.1/ Mahalanobis distance

The first method we will review here is a straightforward extension of a color-based background subtraction algorithm, which is introduced in the original multispectral dataset paper [Benezeth et al., 2014a]. As it is known, background subtraction can be performed by the Equation 2.13, with the assumption that the observed video sequence $\boldsymbol{I}$ is made of a static background $\boldsymbol{B}$ in front of which moving objects are observed.

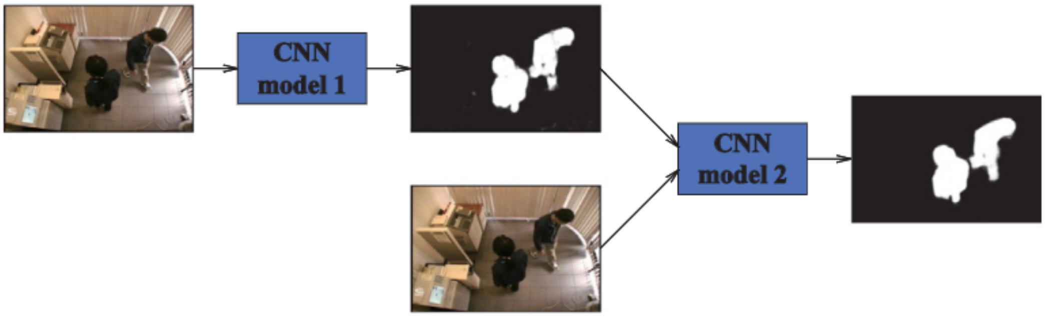

Figure 2.13: Architecture of the cascaded CNN Model [Wang et al., 2017]

$$\chi_t(s) = \begin{cases} 1, & d(\boldsymbol{I}_{s,t}, \boldsymbol{B}_{s,t}) > \tau \\ 0, & \text{otherwise,} \end{cases} \tag{2.13}$$

where τ is a threshold, χ_t is the motion label field at time t, and d is the distance between the pixel value $\boldsymbol{I}_{s,t}$ and the background model $\boldsymbol{B}_{s,t}$ at time t and location s. For a multispectral video sequence, $\boldsymbol{I}_{s,t}$ is a vector defined by $\boldsymbol{I} = [I_1, I_2, \ldots, I_n]$, where n stands for the number of spectral channels of multispectral frames.

If the background $\boldsymbol{B}$ can be determined by a single image free of moving objects, pixels corresponding to foreground moving objects can be detected by thresholding a distance function, such as the Euclidian distance:

$$d = \sqrt{\left(\sum_{i=1}^{k} (I^i_{s,t} - B^i_{s,t})^2\right)}. \tag{2.14}$$

However, it is not the case for real-life scenarios. Modeling the background $\boldsymbol{B}$ with a single image requires a rigorously fixed background void of noise and artefacts. A promising solution is to model each background pixel by a probability density function (PDF) learned over a series of training frames. In this case, the background subtraction problem becomes a PDF-thresholding issue for which a pixel with low probability is likely to correspond to a foreground moving object.

For instance, in order to account for noise, it is possible to model every background pixel with a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ stand for the average background multispectral vector and covariance matrix at pixel s and time t, respectively. In this context, the distance metric can be the Mahalanobis distance:

$$d_M = \sqrt{|\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t}|\boldsymbol{\Sigma}_{s,t}^{-1}|\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t}|^T}. \tag{2.15}$$

Since the covariance matrix contains large values in noisy areas and low values in more stable areas, $\boldsymbol{\Sigma}$ makes the threshold locally dependent on the amount of noise. In other words, the noisier a pixel is, the larger the temporal gradient $|\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t}|$ has to be to get the pixel labeled in motion. This makes the method significantly more flexible and robust.

Besides, In order to adjust the fact that the illumination often changes in time, the mean and covariance of each pixel can also be iteratively updated with:

$$\boldsymbol{\mu}_{s,t+1} = (1-\alpha)\boldsymbol{\mu}_{s,t} + \alpha\boldsymbol{I}_{s,t} \tag{2.16}$$

$$\boldsymbol{\Sigma}_{s,t+1} = (1-\alpha)\boldsymbol{\Sigma}_{s,t} + \alpha(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T \tag{2.17}$$

2.3.2/ Pooling

The second approach introduced by Benezeth et al. in [Benezeth et al., 2014a] is called Pooling. Instead of using all the channels directly, as the algorithm extending Mahalanobis distance does, Pooling combines the background subtraction masks from some spectral channels. The results suggest not all channels are needed to reach the highest accuracy, which indicates only a few number of channels actually define the moving objects.

The definition to perform pooling is as follows,

$$\chi_t(s) = \begin{cases} 1, & \sum_i \chi_t^i(s) > \rho \\ 0, & \text{otherwise,} \end{cases} \tag{2.18}$$

where $\chi_t^i(s)$ is the motion label at time t and location s obtained with the i^{th} channel. $\rho \in [1, n]$ defines the Pooling strategy and n is the whole number of multispectral channels. If ρ is 1, the Pooling is equivalent to logical OR, where the current pixel is detected as foreground as long as the motion mask is foreground for at least one channel. Similarly, the Pooling will become logical AND if ρ is set to n, which means a pixel can be regarded as moving category only when the label is always foreground for all channels.

2.3.3/ Spectral angle

Instead of using a straightforward extension of the color-based Mahalanobis distance, the multispectral dataset paper [Benezeth et al., 2014a] has also proposed another two dedicated ways to measure the similarity or dissimilarity of spectral vectors.

The first one is called spectral angle d_θ, which extracts geometric features by calculating the angle between two spectra [Schowengerdt, 2006] and is defined with the following equation:

$$d_\theta(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{s,t}) = \cos^{-1}\left(\frac{< \boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{s,t} >}{|\boldsymbol{I}_{s,t}||\boldsymbol{\mu}_{s,t}|}\right), \tag{2.19}$$

where $\boldsymbol{I}_{s,t}$ is the multispectral vector of the current image and $\boldsymbol{\mu}_{s,t}$ is that of the background model. As we can see, here spectra are considered as vectors in a k-dimensional space, which indicates this spectral distance measure is suitable for arbitrary number of multispectral channels.

With spectral angle metric, small angles mean similar vectors. Thus, another key advantage of this measure is that it is intensity invariant because the angle between two vectors is independent of the vector length. This property is very interesting for background subtraction problems when there are shadows and illumination variations.

2.3.4/ Spectral information divergence

Another spectral distance measurement utilized in the dataset paper [Benezeth et al., 2014a] is referred to as Spectral Information Divergence (SID) [Chang, 2000], which is also applied to determine the spectral closeness or distance between two multispectral vectors. This measure is relatively recent and is expected to be more effective than spectral angle d_θ in preserving spectral properties.

The spectral information divergence models the spectral channel-to-channel variability as a result of uncertainty caused by randomness, which is based on the Kullback-Leibler divergence to measure the discrepancy of probabilistic behaviours [Chang, 2000]. That is to say, it considers each pixel as a random variable and then defines the desired probability distribution by normalizing its spectral histogram to unity, which is expressed by

$$\begin{cases} P_x(i) = \dfrac{x_t(i)}{\sum_{i=1}^{n} x_t(i)} \\ P_v(i) = \dfrac{v_m(i)}{\sum_{i=1}^{n} v_m(i)} \end{cases} \tag{2.20}$$

where n is the number of channels. Then the spectral information divergence d_{SID} between the current spectral vector $\mathbf{x}_t$ and the background model $\mathbf{v}_m$ can be defined with Equation 2.21.

$$d_{SID}(\mathbf{x}_t, \mathbf{v}_m) = \sum_{i=1}^{n} P_x(i) log \frac{P_x(i)}{P_v(i)} + \sum_{i=1}^{n} P_v(i) log \frac{P_v(i)}{P_x(i)} \tag{2.21}$$

2.3.5/ Online stochastic tensor decomposition

Besides, Sobral et al. [Sobral et al., 2015] have proposed another mechanism based on stochastic decomposition of low-rank and sparse components for background subtraction with multispectral video sequences. In this algorithm, each multispectral image is represented as a three-dimension data cube or tensor, which can be considered as a multidimensional or N-way array.

As reviewed in [Bouwmans et al., 2017b] [Davenport et al., 2016] and [Lin, 2016], low-rank and sparse decomposition methods are based on the assumption that the uncorrupted information lies in a low dimensional subspace, whereas noise is sparse. This assumption holds a particular association to the task of foreground-background segmentation. To be more specific, as it is almost static and highly correlated between frames, the background is assumed to be a low dimensional subspace, where the sparse outliers usually represent the foreground objects.

Based on the pioneering tensor-based decomposition methods [Feng et al., 2013] [Goes et al., 2014], the framework of Online Stochastic Tensor Decomposition (OSTD) has been proposed in [Sobral et al., 2015]. They have extended the online stochastic principal analysis optimization for multispectral images using tensor analysis, where the stochastic optimization is applied on each mode of the tensor and the individual basis is updated iteratively followed by the processing of the current frame.

2.3.6/ Online one-class ensemble for feature selection

In background subtraction, the features characterize a region and can be compared against a known background model to classify it as either foreground or background. As we know, color features, edge features, stereo features, and motion features are commonly used in this field. However, the optimal features for background subtraction may be case by case.

Given the ensemble learning mechanism proposed in [Bolón-Canedo et al., 2014], an algorithm called Online Weighted One-Class Random Subspace (OWOC-RS) has been designed by Silva et al. in [Silva et al., 2016], which is capable to select suitable pixel-based features to separate the foreground regions from the background. Besides, the relative importance of each feature over time is updated with adaptive mechanism.

In order to not only increase the efficiency in terms of time and memory consumption but also the segmentation performance, an improved version has been also proposed by the same authors in [Silva et al., 2017]. The novel method is named as Superpixel-based Online Wagging One-Class Ensemble (Superpixel-OWAOC) as it adopts the superpixel idea and is based on wagging for selecting suitable individual features.

2.4/ Conclusion

In this chapter, we have conducted a study on background subtraction by investigating the state-of-the-art algorithms for both RGB and multispectral images.

The challenges of background subtraction include but are not limited to the following three categories: background (illumination changes, dynamic background, shadows, challenging weather, bootstrapping, moved background objects and night videos), foreground (shadows, camouflage, intermittent object motion and occlusion) and camera (video noise, moving camera and camera jitter). In order to cope with these challenges, numerous algorithms have been designed over the past years for background subtraction. The majority of the current background subtraction algorithms are focused on conventional RGB images, which could offer motivation and perspective for multispectral case.

Firstly, representative classical algorithms with RGB images are reviewed in this chapter. Two methods, namely Gaussian mixture models and Kernel density estimation, which are older but frequently cited, are introduced, respectively. Besides, some more recent conventional approaches that perform well on CDnet2014 dataset, namely, ViBe based estimations (Pixel-Based Adaptive Segmenter, Self-Balanced SENsitivity Segmenter and Pixel-based Adaptive Word Consensus Segmentation), In Unity There Is Strength and Semantic background subtraction, are presented as well.

Recently, deep learning based approaches have shown an impressive detection results and outperformed the classical methods by large margins. Thus, several leading deep models working with RGB images that achieve the top performances are also illustrated in this chapter, including Convoluntional Neural Networks, like FgSegNet_M, FgSegNet_S and Cascade CNN, Bayesian generative adversarial networks and Conditional generative adversarial networks.

Since the objective of this thesis is to investigate multispectral images for the task of background subtraction, we introduced the existing multispectral approaches. In addition to the four algorithms (Mahalanobis distance, Pooling, Spectral angle, and Spectral information divergence) proposed by the creators of the multispectral dataset, Online stochastic tensor decomposition and Online one-class ensemble for feature selection are also presented. These algorithms have been tested on the same dataset we have adopted, and will be later used for quantitative comparison with the algorithms proposed in this thesis.

II

Contribution

3

Multispectral Codebook for background subtraction

3.1/ Introduction

The original Codebook algorithm for background subtraction was inspired by Kohonen [Kohonen, 1995] [Ripley, 2007] and proposed in 2005 by Kim [Kim et al., 2005]. It is a clustering scheme with several important additional elements, which make it robust against moving background, to construct a background model from a sequence of images on a pixel-by-pixel basis, then to compare this background model with every new frame in order to finally obtain a foreground background segmentation mask. The motivation for selecting such a model is that it is fast to run, because it is deterministic; efficient for requiring little memory; adaptive to multiple background situations as it does not assume the potential parametric distributions of the background; and able to handle complex backgrounds with sensitivity [Doshi et al., 2006]. It has been proved to be very efficient in dealing with dynamic backgrounds.

Like many other background subtraction schemes, the Codebook process includes three main steps: background model initialization, background model maintenance and foreground detection [Sobral, 2017], as illustrated in Fig. 3.1.

Accordingly, the original Codebook algorithm will be stated in five important issues for better explanation, namely background model initialization, matching process, background model updating strategy, background model refining and foreground detection. The implementations of the original algorithm is detailed on the basis of these issues. For the

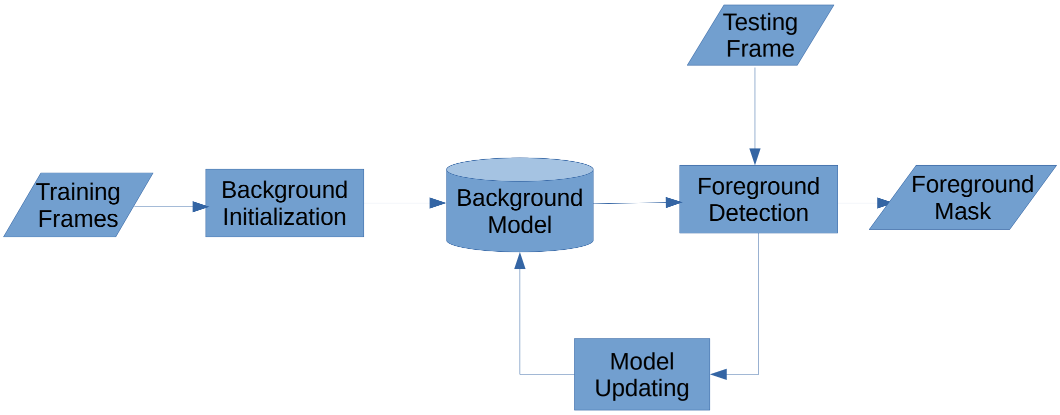

Figure 3.1: Block diagram of the Codebook process

improved versions later in this chapter and next chapter, we will only focus on the modified parts.

During the last decade, many works have been dedicated to improve the original Codebook model. For example, [Zhang et al., 2016] and [Li et al., 2012] have adopted a two-layer model, to handle dynamic background and illumination variation problems. Other modifications like transferring RGB to other color models like YCbCr and Lab, in order to solve the problem of existence of shadows and highlights for foreground detection, can also be found in [Murgia et al., 2014] and [Aung et al., 2017]. In [Zaharescu et al., 2011], a multi-scale multi-feature Codebook model, which integrates intensity, color and texture information across multiple scales, has been presented for challenging environments. A strategy has been proposed in [Noh et al., 2014] which takes advantages of the Codebook and GMM-based techniques to overcome not only non-practical modeling methodologies of the probabilistic approaches, but also inaccurate detection performance of Codebook techniques in moving backgrounds. [Mousse et al., 2014] associates the original Codebook with an edge detection algorithm. In the work of [Tu et al., 2008], the background model is constructed by encoding each pixel into a codebook consisting of codewords based on a box model and it is also appropriate in the Hue-Saturation-Value (HSV) color space. Besides, [Ruidong, 2015] has rewritten the model parameters and then processed the three channels separately to simplify the matching equations. In [Kusakunniran et al., 2016], a dynamic boundary of codebook under the Lab color space has been developed and evaluated using the well-known large-scale dataset CDnet2012 [Goyette et al., 2012] and its extension CDnet2014 [Wang et al., 2014].

The rest of this chapter is organised as follows. Section 3.2 illustrates the original Code-

Table 3.1: The description of the elements in the tuple $\textbf{aux}_m$

$\check{I}, \hat{I}$	the minimum and maximum brightness range assigned to the codeword respectively
f	the frequency of access or the number of times that the codeword is matched
λ	maximum length of time between consecutive accesses
p, q	the first and the last access times of the codeword respectively

book algorithm in detail, which is then adapted to multispectral case in Section 3.3. The multispectral dataset, evaluation metrics and experiments are discussed in Section 3.4, while conclusions and future work are presented in Section 3.5.

3.2/ The original Codebook

The original Codebook is a pixel-based algorithm, which means that the whole process happens at each pixel independently and ignores information observed at other pixels. For each pixel, a codebook $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_L\}$, consisting of several , is constructed. The number of codewords is different according to the pixel's variation. The codeword can be regarded as cluster of samples with similar certain spectral features and defined in a form of super cubes capable of describing these samples [Yongjia et al., 2014]. More precisely, each codeword $\mathbf{c}_m$ is composed of an RGB vector $\mathbf{v}_m = (\overline{R}, \overline{G}, \overline{B})$ and a six-tuple $\textbf{aux}_m = (\check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m)$. The tuple $\textbf{aux}_m$ stores the auxiliary information such as intensity statistics and temporal variables, which are listed and explained in Table 3.1 [Aung et al., 2017], and assist in training and pruning the Codebook background model.

3.2.1/ Codebook model initialization

At the beginning, the codebook is an empty set and the number of the codewords is set to 0. When the first frame comes, the Codebook model is initialized by constructing an associated codeword for each pixel with the corresponding vector $\mathbf{v}$ being set to be the RGB values of that pixel as below:

$$\mathbf{v} = \mathbf{x} = (R, G, B) \tag{3.1}$$

And the corresponding six-tuple vector $\textbf{aux} = (\check{I}, \hat{I}, f, \lambda, p, q)$ is assigned with

$$\mathbf{aux} = (I, I, 1, 0, 1, 1) \tag{3.2}$$

where I represents the intensity or brightness value, which is calculated by

$$I = \sqrt{(R^2 + G^2 + B^2)} \tag{3.3}$$

3.2.2/ Codebook matching process

In the training phase, for a newly incoming frame at time instant t, the current value $\mathbf{x}_t$ of a given pixel is compared to its current codebook to construct the background model. Comparisons are made by both the color distortion measurement and brightness bounds. The color distortion is defined as:

$$\begin{aligned} colordist(\mathbf{x}_t, \mathbf{v}_m) = \delta &= \sqrt{||\mathbf{x}_t||^2 - \frac{<\mathbf{x}_t, \mathbf{v}_m>^2}{||\mathbf{v}_m||^2}} \\ &= \sqrt{(R^2 + G^2 + B^2) - \frac{\overline{R}R + \overline{G}G + \overline{B}B}{\overline{R}^2 + \overline{G}^2 + \overline{B}^2}} \end{aligned} \tag{3.4}$$

where (R,G,B) represents the RGB vector of incoming pixels, whereas ($\overline{R}$, $\overline{G}$, $\overline{B}$) is the average RGB vector of the corresponding codeword. The brightness function is defined as:

$$brightness(I, <\check{I}_m, \hat{I}_m>) = \begin{cases} true & if \;\; I_{low} \leq I \leq I_{hi} \\ false & otherwise \end{cases} \tag{3.5}$$

The lower bound I_{low} and upper bound I_{hi} of the range are defined in Equation 3.6 by $\check{I}_m$ and $\hat{I}_m$, which are the minimum and maximum brightness of all pixels assigned to this codeword, and stored in the 6-tuple $\mathbf{aux}_m$ illustrated in Table 3.1 for the current codeword $\mathbf{c}_m$.

$$\begin{cases} I_{low} = \alpha \hat{I}_m \\ I_{high} = \min\left\{\beta \hat{I}_m, \dfrac{\check{I}_m}{\alpha}\right\} \end{cases} \tag{3.6}$$

where α and β are manually set constant parameters.

If two conditions, namely color distortion and brightness bounds, are both satisfied, that is to say, the colordist value is smaller than a threshold ε_1 and the output of the brightness function is true, as illustrated in Equation 3.7, the current pixel is matched with this codeword $\mathbf{c}_m$, which is then used as the sample's encoding approximation.

$$\begin{cases} colordist(\mathbf{x}_t, \mathbf{v}_m) \leq \varepsilon_1 \\ brightness\left(I, \left\langle \check{I}_m, \hat{I}_m \right\rangle\right) = true \end{cases} \tag{3.7}$$

Otherwise, a new codeword is to be created. The average RGB vector $\mathbf{v}$ is initialized with the current pixel spectral values.

$$\mathbf{v} = \mathbf{x}_t = (R, G, B) \tag{3.8}$$

The corresponding initial six-tuple vector **aux** = $(\check{I}, \hat{I}, f, \lambda, p, q)$ is assigned with

$$\mathbf{aux} = (I, I, 1, t-1, t, t) \tag{3.9}$$

3.2.3/ Codebook model updating

Codewords in the background model are designed to be updated in an online way to account for the environmental changes in video streams. When a new frame is arriving, the color distortion in Equation 3.4 and brightness function in Equation 3.5, defined in the last section, together with the brightness bounds Equation in 3.6 are first calculated to see whether a match can be found based on Equation 3.7.

For the matched codeword, its corresponding average RGB vector $\mathbf{v}_m$ and components in the $\mathbf{aux}_m$ are updated using the new arrived (R,G,B) vector and the intensity value I of the current pixel. Particularly speaking, vector $\mathbf{v}_m$ is updated with Equation 3.10.

$$\mathbf{v}_m \leftarrow \left(\frac{f_m\overline{R}}{f_m + R}, \frac{f_m\overline{G}}{f_m + G}, \frac{f_m\overline{B}}{f_m + B}\right) \tag{3.10}$$

The six elements in the vector $\mathbf{aux}_m$ are updated as what are listed in Table 3.2, where

Table 3.2: The update of the elements in the tuple $\textbf{aux}_m$

minimum brightness range $\check{I}$	$\check{I}_m \leftarrow \min\left\{I, \check{I}_m\right\}$
maximum brightness range $\hat{I}$	$\hat{I}_m \leftarrow \min\left\{I, \hat{I}_m\right\}$
frequency of access f	$f_m \leftarrow f_m + 1$
maximum length of time λ	$\lambda_m \leftarrow \max\{\lambda_m, t - q_m\}$
first access time p	$p_m \leftarrow p_m$
last access time q	$q_m \leftarrow t$

the subscript m indicates the current codeword.

For the codewords that are not matched successfully, the λ component is updated as $\lambda \leftarrow \lambda + 1$ and all other components are kept the same.

When there is no new frame in the training process, the longest time interval between two consecutive accesses λ_m for each codeword $\mathbf{c}_m$ is computed by

$$\lambda_m = \max\{\lambda_m, N - q_m + p_m - 1\} \tag{3.11}$$

where N is the number of training frames.

3.2.4/ Codebook model refining

During the training period, there are not only the background information, but moving foreground objects may also exists in the image sequence. Thus, the codebook model obtained from the previous steps contains all the codewords that represent the scene and can be called fat codebook [Kim et al., 2005]. It is obvious that those redundant codewords that contain moving objects should be eliminated in order to get a refined and true background model.

Thanks to the fact that the values for actual background usually recur, temporal filtering can be adopted to achieve this goal. This is why λ, which is defined as the maximum length of time between consecutive accesses, is recorded in the tuple $\textbf{aux}_m$ for each codeword and then used to discriminate the actual background codewords from the moving foreground codewords that are inactive for a long period of time. To be more specific, codewords having small λ are mostly representing the true background while those with larger λ should be filtered out. Let **C** and **M** denote the fat and refined background model, respectively. The temporal filtering can be described with:

$$\mathbf{M} = \{\mathbf{c}_m \mid \mathbf{c}_m \in \mathbf{C} \ \ and \ \ \lambda_m \leq T\} \tag{3.12}$$

where T is a threshold value. Usually, the threshold is set to the half of the number of training frames [Kim et al., 2005]. That is to say, all the codewords in the final refined background model **M** should recur at least every $N/2$ frames.

3.2.5/ Codebook foreground detection

After constructing the Codebook model, the moving objects can be detected by using background subtraction directly. It simply consists in calculating the difference of the current image from the background model with respect to brightness and color distortion, detailed in Section 3.2.2. Then the pixel is detected as foreground if no acceptable matching codeword exists. Otherwise, it is classified as background and the corresponding matched codeword is updated according to the updating procedure presented in Section 3.2.3. The algorithm is illustrated in Algorithm 1, where ε_2 is color threshold during the detection phase, which can be set with higher value to be more tolerant for noise in the detection phase.

Algorithm 1 Codebook forground detection algorithm

```
$\mathbf{x} = (R, G, B),\ I = \sqrt{(R^2 + G^2 + B^2)}$
search the codeword $\mathbf{c}_m$ in $\mathbf{C}$ matching $\mathbf{x}_t$ if (a) and (b) occur.
(a) $colordist(\mathbf{x}_t, \mathbf{v}_m) \leq \varepsilon_2$
(b) $brightness\left(I, \left\langle \check{I}_m, \hat{I}_m \right\rangle\right) = True$
if there is a match, then
    output the pixel as background and update the current matched codeword
else
    output the pixel as foreground
end if
```

Note that in the construction phase, when there is no appropriate match found, a new codeword is established, while in the detection phase, the pixel is detected as foreground directly and no more extra measure is taken.

3.3/ Multispectral Codebook

Since the object of this research work is to investigate the benefits of multispectral sequences rather than traditional RGB to improve the performance of background subtraction, we would like to adapt the original RGB-based Codebook algorithm to multispectral sequences. Minor modifications need to be performed. Specifically speaking, the definition of brightness in RGB is extended to multispectral case. Besides, unlike color distortion in the original Codebook, we adopt spectral distortion instead, as the term color is always related to RGB, and even for three bands out of multispectral sequences, they are not strictly color.

In the case of multispectral images, each codeword $\mathbf{c}_m$ is also defined by two vectors: the first one contains the average spectral value for each channel or band of the pixel, $\mathbf{v}_m = (V_1, V_2, ..., V_n)$, where n is the number of multispectral channels. The second one is the six-tuple $\mathbf{aux}_m = (\check{I}_m,\ \hat{I}_m,\ f_m,\ \lambda_m,\ p_m,\ q_m)$, as listed in Table 3.1.

3.3.1/ Multispectral Codebook model initialization

At the beginning, the codebook is an empty set and the number of the codewords is set to 0. When the first frame comes, the multispectral Codebook model is initialized by constructing an associated codeword for each pixel with the corresponding vector **v** being set to be the spectral values of that pixel.

$$\mathbf{v} = \mathbf{x} = (X_1, X_2, ..., X_n) \tag{3.13}$$

The auxiliary vector **aux** is set as below:

$$\mathbf{aux} = (I, I, 1, 0, 1, 1) \tag{3.14}$$

As we know, for grayscale images, the grayscale value or the brightness is obtained by $I = |x| = \sqrt{x^2}$. For RGB images, the brightness is calculated by $I = \sqrt{R^2 + G^2 + B^2}$. Accordingly, for multispectral images, given a pixel $\mathbf{x} = (X_1, X_2, ..., X_n)$, the brightness can be measured by

$$I = \sqrt{\sum_{i=1}^{n} X_i^2} \tag{3.15}$$

where i is the index of spectral channel.

3.3.2/ Multispectral Codebook matching process

The matching process for multispectral Codebook algorithm is evaluated by two conditions: brightness bound and spectral distortion.

Like the original Codebook, two criteria need to be satisfied simultaneously to find a match between the current pixel with a codeword $\boldsymbol{c_m}$. Specifically, the brightness of the pixel must lie in the interval $\left[I_{low}, I_{high}\right]$, which is calculated from the min and max brightness $\check{I}_m, \hat{I}_m$ in Equation 3.6.

$$brightness(I, < \check{I}_m, \hat{I}_m >) = true \tag{3.16}$$

Equation 3.17 gives the calculation of spectral distortion, spectral_dist, between a new coming spectral pixel $\mathbf{v}_t = (X_1, X_2, ..., X_n)$ and a codeword $\mathbf{aux}_m$ with the average spectral vector $\mathbf{v}_m = (V_1, V_2, ..., V_n)$

$$spectral\_dist(\mathbf{x}_t, \mathbf{v}_m) = \sqrt{||\mathbf{x}_t||^2 - \frac{\langle \mathbf{x}_t, \mathbf{v}_m \rangle^{2^2}}{||\mathbf{x}_t||^2}} = ||\mathbf{x}_t||^2 - ||\mathbf{x}_t||^2 \cos^2\theta \tag{3.17}$$

The second condition is that the spectral distortion must lie under a given threshold ε_1.

$$spectral\_dist(\mathbf{x}_t, \mathbf{v}_m) \leqslant \varepsilon_1 \tag{3.18}$$

To make it intuitive, the two criteria are visualized in Fig. 3.2. The pixel of a multispectral image is considered as vector in an n-dimensional space and three bands are used as an example. The cylinder model [Zeng et al., 2014] is adopted to cope with illumination changes. In Fig. 3.2, the blue cylinder represents a certain codeword, which is built with the spectral distortion threshold around the mean vector. Thus the bottom radius is the threshold ε_1. Besides, the red and the blue vectors stand for the average spectral $\mathbf{v}_m$

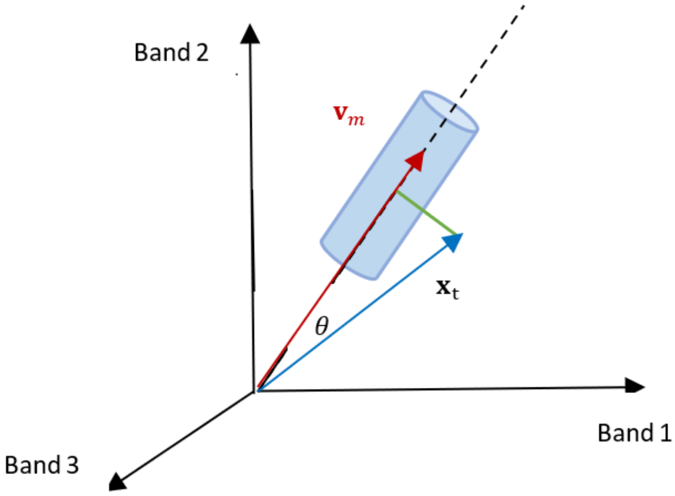

Figure 3.2: Visualization of the judging criteria brightness and spectral distortion

in this codeword and the current pixel $\mathbf{x}_t$, respectively. With Equation 3.17, the spectral distortion can be calculated and illustrated with the green line. As discussed above, a match is found if the brightness of the pixel vector lies between I_{low} and I_{hi}, and the spectral distortion is under a given threshold ε_1. Accordingly, the L2-norm of vector $\mathbf{x}_t$ must be located along the axis in the cylinder and the length of the green line must be smaller than the radius of the cylinder. Obviously, the codeword $\mathbf{c}_m$ and the current $\mathbf{x}_t$ are not matched in the Fig. 3.2, as the criteria of spectral distortion is not satisfied.

3.3.3/ Multispectral Codebook model updating

To construct the background model, the current value $\mathbf{x}_t$ of a given pixel is compared to its current codebook. if there is a match with a codeword $\mathbf{c}_m$, this codeword is used as the sample's encoding approximation. Otherwise, a new codeword is to be created.

Detailedly speaking, the average multispectral vector $\mathbf{v}_m$ is updated as follows:

$$\mathbf{v}_m \leftarrow (\frac{f_m\overline{X}_{m1} + X_1}{f_m + 1}, \frac{f_m\overline{X}_{m2} + X_2}{f_m + 1}, \dots, \frac{f_m\overline{X}_{mn} + X_n}{f_m + 1}) \tag{3.19}$$

As for the six-tuple $\mathbf{aux}_m$ = ($\check{I}_m$, $\hat{I}_m$, f_m, λ_m, p_m, q_m), it is updated exactly the same way as the original Codebook algorithm. At the end of the Codebook construction algorithm, the model also has to clean the codewords that are most probably belonging to foreground objects, based on the λ_m recorded in the $\mathbf{aux}_m$, as illustrated for the original Codebook.

The detailed algorithm of codebook construction is given in Algorithm 2.

Algorithm 2 Multispectral Codebook construction

```
$L \leftarrow 0$, $\mathbf{C} \leftarrow \phi$
for $t = 1 \rightarrow N$ do
    $\mathbf{x}_t = (X_1, X_1, ..., X_n)$, $I = \sqrt{\sum_{i=1}^{n} X_i^2}$
    Find the codeword $c_m$ in C matching $textbfx_t$ by checking $(a)$ and $(b)$
    $(a)$ $spectral\_dist(\mathbf{x}_t, \mathbf{v}_m) \leqslant \varepsilon_1$
    $(b)$ $brightness\left(I, \left\langle \check{I}_m, \hat{I}_m \right\rangle\right) = True$
    if $\mathbf{C} = \phi$ or there is no match then
        $L \leftarrow L + 1$, create a new codeword $c_L$
        $\mathbf{v}_L = \mathbf{x}_t$
        $\mathbf{aux}_L = \langle I, I, 1, t-1, t, t \rangle$
    else
        update the matched codeword $\mathbf{c}_m$ composed of $\mathbf{v}_m = \left(\bar{X}_{1m}, \bar{X}_{2m}, ..., \bar{X}_{nm}\right)$ and $\mathbf{aux}_m = \left\langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \right\rangle$
        $\mathbf{v}_m \leftarrow \left(\frac{f_m\bar{X}_{1m}+X_1}{f_m+1}, \frac{f_m\bar{X}_{2m}+X_2}{f_m+1}, ..., \frac{f_m\bar{X}_{nm}+X_n}{f_m+1}\right)$
        $\mathbf{aux}_m \leftarrow \left\langle min\left\{I, \check{I}_m\right\}, max\left\{I, \check{I}_m\right\}, f_m + 1, max\left\{\lambda_m, t - q_m\right\}, p_m, t \right\rangle$
    end if
end for
```

Fig. 3.3 offers a working flow for multispectral Codebook model construction with brightness and spectral distortion. When a new frame arrives, the brightness and spectral distortion are first computed based on Equation 3.15 and Equation 3.17. Then if the two criteria illustrated in Equations 3.16 and 3.18 are satisfied simultaneously, a match between the current pixel and codeword is found. The codeword will be updated with the information of this pixel. Unless, a new codeword will be created.

3.3.4/ Multispectral Codebook foreground detection

The foreground detection phase that follows conducts the matching process like that in background model construction phase. The brightness and spectral distortion criteria are evaluated with Equations 3.16 and 3.18 between the testing frame and the background model. The pixel is detected as foreground if no acceptable matching codeword exists, as illustrated in Fig. 3.4. Otherwise, it is classified as background and the corresponding codeword is updated accordingly. During the detection phase, the threshold for spectral distortion is set with higher value to be more tolerant for noise.

We need to note that in the background model construction in Fig. 3.3, when there is no appropriate match found, a new codeword will be established, while in the foreground

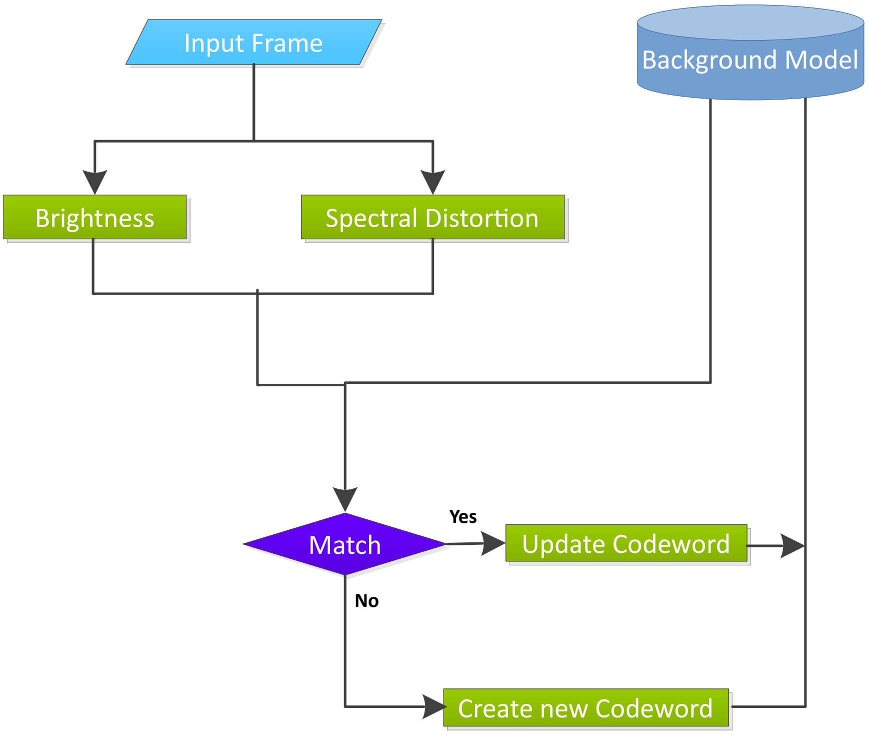

Figure 3.3: Codebook construction with brightness and spectral distortion

detection phase in Fig. 3.4, the pixel is detected as foreground and then the task is accomplished.

3.4/ Experiments

3.4.1/ Evaluation dataset

To evaluate the performance of the proposed approaches for background subtraction, the multispectral dataset presented by Benezeth et al. [Benezeth et al., 2014b] is adopted for testing in this thesis. This dataset was created in order to investigate the use of multispectral videos of more than three bands for background subtraction.

To our knowledge, this dataset is the only public real multispectral image background subtraction dataset available. Most public image datasets built for background subtraction, or change detection, such as the well-known Wallflower dataset [Toyama et al., 1999], the Stuttgart Artificial Background Subtraction (SABS) dataset [Brutzer et al., 2011] and CD-net [Goyette et al., 2012] [Wang et al., 2014], are based on visible spectral images. Some other datasets include still recombined images. For example, the Grayscale-Thermal

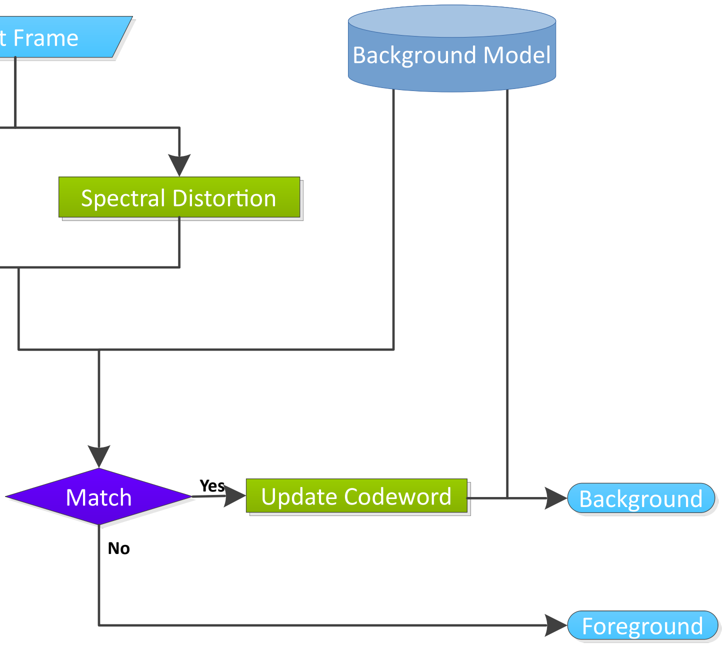

Figure 3.4: Multispectral Codebook foreground detection

Foreground Detection (GTFD) dataset [Li et al., 2016] provides a pair of grayscale and thermal frames captured by two cameras to investigate the fusion methods of thermal and grayscale data for effective foreground detection. Besides, the LITIV 2018 dataset [St-Charles et al., 2019] includes a pair composed of visible and Long Wavelength Infrared (LWIR) spectra.

As [Benezeth et al., 2014b] does not give an official name for the multispectral dataset they have established, some other works that use this dataset like [Silva et al., 2017] [Sobral et al., 2015] call it MSVS as an abbreviation of MultiSpectral Video Sequences. However, in [Kalsotra et al., 2019], which is a survey work of datasets for background subtraction, this multispectral dataset is introduced as FluxData FD-1665 dataset, indicating the camera type for image acquisition. In this thesis, we follow the name in [Kalsotra et al., 2019], with a thought that more multispectral datasets will be built and published in future with a variety of multispectral imaging devices.

The FluxData FD-1665 dataset contains multispectral sequences with seven channels, or bands, captured simultaneously with the commercial camera from FluxData, Inc. (FD-1665-MS camera). Among the seven channels, six are in the visible spectrum and the last one is in the Near-InfraRed (NIR) spectrum. Fig. 3.5 shows an example associated with a single frame of one sequence of the multispectral dataset. In the figure, Images

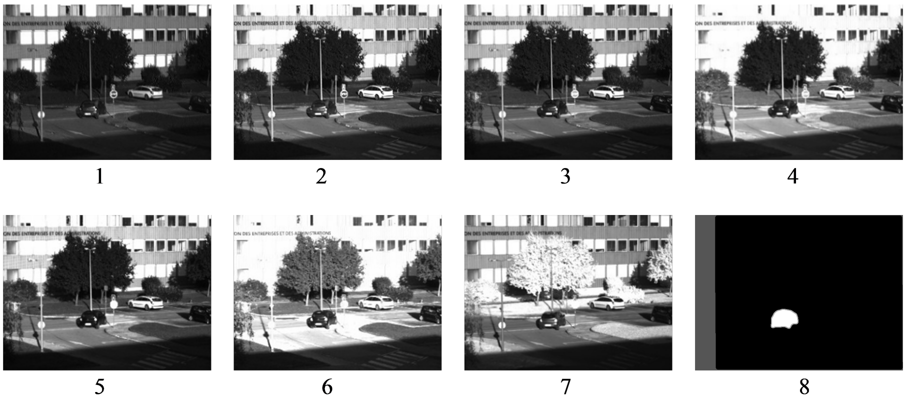

Figure 3.5: Frame of one sequence of the multispectral dataset

1-6 show the 6 visible channels, Image 7 corresponds to the NIR channel, and the last image is the corresponding pixel-based ground truth.

The dataset is composed of five sequences containing frames of size 658×492 for each channel. The dataset represents one indoor video sequence and four outdoor scenes with different challenges such as shadows, intermittent object motion and camouflage effects (color similarity between object and background). The complete details of each video sequence are mentioned in Table 3.3.

The dataset includes as well the corresponding RGB sequences, which are obtained with a linear integration of the seven-channel original multispectral images weighted by three different spectral envelopes [Jacobson et al., 2005], shown in Equation 3.20.

$$R = \sum_{i=1}^{n} r_i X_i, \; G = \sum_{i=1}^{n} G_i X_i, \; B = \sum_{i=1}^{n} b_i X_i \tag{3.20}$$

where r_i, g_i, b_i are the weights on the i^{th} spectral channel, defined based on the characteristics of the specific multispectral camera, and n is the number of channels, which is seven here.

Fig. 3.6 presents examples of RGB frames extracted from the five RGB videos estimated from the five multispectral videos. These sequences are all publicly available. Thus there are three subsets for each scene, namely multispectral image sequence with the size of $658 \times 492 \times 7$, corresponding RGB image sequence of $658 \times 492 \times 3$ and the groundtruth images of $658 \times 492 \times 1$.

Pixel-wise labeled foreground masks, namely ground truth images, are annotated

Table 3.3: Details of each video sequence of the FluxData FD-1665 dataset

Video	Video Scenes	Total Frames	Ground Truth	Challenges
1	Indoor	258	258	Color saturation
2	Outdoor	1658	1658	Presence of shadows and camouflage
3	Outdoor	2213	1103	Dynamic backgrounds, tree shadows and occlusion
4	Outdoor	1351	925	Faraway intermittent object motion
5	Outdoor	3902	2232	Objects with shadows

Figure 3.6: Snapshots of the five scenes in the FluxData FD-1665 dataset

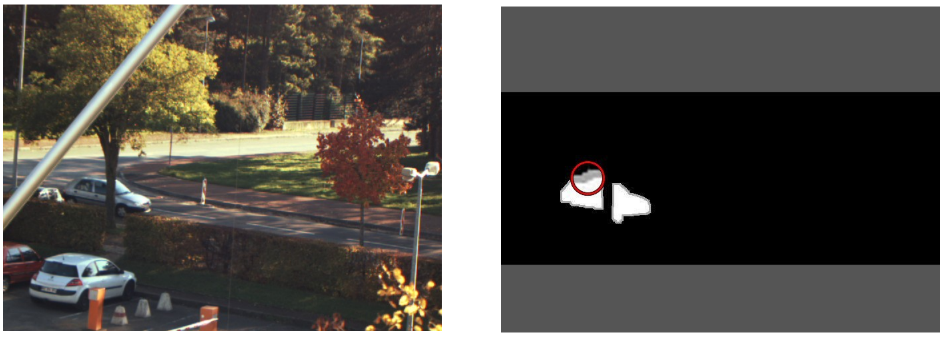

Figure 3.7: Example of groundtruth labels

manually and provided to public for a fair precise validation. With reference to [Goyette et al., 2012], the groundtruth image is not an ideal binary mask, containing only moving pixels with a grayscale value of 1 and static pixels with a grayscale value of 0. Since some areas carry a certain level of uncertainty, it is difficult for a person to reliably classify them as background or foreground pixels, such as those pixels close to moving objects boundaries, as illustrated with the partial magnification in the red circle in Fig. 3.7. To avoid evaluation metrics from being corrupted, these pixels that are corrupted by motion blur, are labeled as unknown and assigned grayscale value of 170.

Besides, the Non-ROI (not in the Region Of Interest) label is adopted to prevent the evaluation metrics being influenced by activities unrelated to the task considered. For example, the scene 3 is cluttered with moving tree, as shown in Fig. 3.7. However, what we care is the performance of an algorithm to detect the moving objects on the street. Thus the top and down parts are labeled as Non-ROI with a grayscale value of 85. The four labels are listed in Table 3.4, where pixels with a grayscale value of 85 and 170 in the ground truth images are ignored in accuracy evaluation.

Table 3.4: Ground truth labels

Label	Grayscale Value	Motion Status
Moving	255	Foreground
Unknown	170	Half-occluded and corrupted by motion blur
Non-ROI	85	Unrelated activities
Static	0	Background

3.4.2/ Evaluation metrics

The binary mask obtained from the foreground detection process is compared with the corresponding ground truth image. As the ground truth is provided at pixel resolution, the performance of background subtraction algorithms can be evaluated more precisely, compared with the ground truth form of bounding box. In order to determine how well the algorithm can correctly identify moving region and conduct a quantitative comparison between different approaches, some basic metrics are defined and usually used in background subtraction[Viau et al., 2016], as listed in Table 3.5.

Table 3.5: Basic evaluation metrics

True positive (TP)	The number of correctly identified foreground
True negative (TN)	The number of correctly identified background
False positive (FP)	The number of incorrectly identified foreground
False negative (FN)	The number of incorrectly identified background

In general, these metrics in Table 3.5 are not very individually useful and are typically quoted together when assessing the performance. From these four basic metrics, several other global performance metrics can be computed to assess the performance of the background subtraction algorithm. Among which, precision and recall are defined with Equations 3.21 and 3.22.

$$Precision = \frac{Correctly\ classified\ foreground\ pixels}{Pixels\ classified\ as\ foreground} = \frac{TP}{TP + FN} \tag{3.21}$$

$$Recall = \frac{Correctly\ classified\ foreground\ pixels}{Ground\ truth\ foreground\ pixels} = \frac{TP}{TP + FP} \tag{3.22}$$

Based on Equations 3.21 and 3.22, F-measure, is defined as a harmonic mean of the precision and recall with Equation 3.23, where the relative contribution of precision and recall to the F-measure are equal. It is also known as balanced F-score or F1 score, which reaches its best value at 1 and worst score at 0. That is to say, the greater the value is, the better the detection quality is. It makes the analysis of results easier and is generally used to assess the performance of background subtraction algorithm or to compare different algorithms against a common dataset. That is why we adopt this measure of accuracy for all the experiments in this thesis.

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision\ +\ Recall} \tag{3.23}$$

3.4.3/ Experiments

Since the traditional codebook algorithm for RGB, is three channels based, we begin with the trials with three bands. The number of combination composed of three bands out of seven is $C_7^3 = 35$. For fair comparison, parameters for RGB and these three-dimensional multispectral sequences are the same for the experiments. The four parameters, which are empiric values determined experimentally and used for the Codebook algorithms are as the following:

$$\alpha = 0.7, \quad \beta = 1.5, \quad \varepsilon_1 = 0.02, \quad \varepsilon_2 = 0.04 \tag{3.24}$$

The experiments are conducted on the five different multispectral sequences and the F-measures are recorded in Table 3.6. The RGB results are also shown in the last row, acting as a reference to illustrate the improvement that can be achieved by using multispectral images. The largest value in each column is in bold. Besides, some visual examples are shown in Fig. 3.8, where the top row is original multispectral sequences. For all sequences, no morphological operation is applied.

The table 3.6 illustrates the performance comparison between the three-dimensional multispectral sequences and RGB, whose result is not the best for all five videos. The av-

Table 3.6: Average F-measure on the five videos (three-channel images case)

	Combination	Video 1	Video 2	Video 3	Video 4	Video 5	mean
1	123	0.6505	0.9422	0.7733	0.8037	0.7211	0.7782
2	124	0.8355	0.9420	0.7516	0.8065	0.7864	0.8244
3	125	0.8342	0.9450	0.7515	0.8148	0.7734	0.8238
4	126	0.7104	0.8950	0.7082	0.8047	0.8154	0.7867
5	127	0.7739	0.9396	0.6558	0.8071	0.7247	0.7802
6	134	0.8421	0.9461	**0.7921**	0.8513	0.7670	**0.8397**
7	135	0.8402	0.9538	0.7838	0.8350	0.7622	0.8350
8	136	0.7017	0.9040	0.7417	0.8381	0.8031	0.7977
9	137	0.7764	0.9463	0.6908	0.8354	0.7113	0.7920
10	145	0.8636	0.9440	0.7689	0.8475	0.7757	**0.8399**
11	146	0.8519	0.8952	0.7296	0.8435	0.8132	0.8267
12	147	0.7932	0.9488	0.6881	0.8084	0.7480	0.7973
13	156	0.8705	0.9038	0.7564	0.8440	0.8091	0.8368
14	157	0.7943	0.9538	0.6908	0.8252	0.7361	0.8000
15	167	0.7839	0.9302	0.6518	0.8105	0.7832	0.7919
16	234	0.8358	0.9434	0.7418	0.8168	0.7810	0.8238
17	235	0.8339	0.9448	0.7345	0.8180	0.7657	0.8194
18	236	0.6959	0.8849	0.6880	0.8104	0.8075	0.7773
19	237	0.7714	0.9459	0.6400	0.8259	0.7248	0.7816
20	245	0.8583	0.9386	0.7099	0.8199	0.7899	0.8233
21	246	0.8437	0.8795	0.6677	0.7901	**0.8241**	0.8010
22	247	0.7900	0.9467	0.6273	0.8083	0.7662	0.7877
23	256	0.8666	0.8831	0.6832	0.8000	0.8179	0.8102
24	257	0.7893	0.9472	0.6262	0.8211	0.7539	0.7875
25	267	0.7838	0.9050	0.5934	0.8221	0.7923	0.7793
26	345	0.8619	0.9423	0.7409	0.8585	0.7746	0.8356
27	346	0.8436	0.8831	0.7088	0.8423	0.8076	0.8171
28	347	0.7904	0.9455	0.6525	0.8116	0.7634	0.7927
29	356	0.8661	0.8904	0.7187	**0.8650**	0.8026	0.8286
30	357	0.7894	**0.9546**	0.6640	0.8298	0.7577	0.7991
31	367	0.7833	0.9169	0.6281	0.8308	0.7773	0.7873
32	456	**0.8718**	0.8799	0.6992	0.8297	0.8131	0.8187
33	457	0.7897	0.9402	0.6435	0.8140	0.7690	0.7913
34	467	0.7854	0.9060	0.6181	0.8071	0.7904	0.7814
35	567	0.7844	0.9098	0.6095	0.8027	0.7869	0.7787
36	RGB	0.8086	0.9431	0.7578	0.7679	0.7789	0.8113

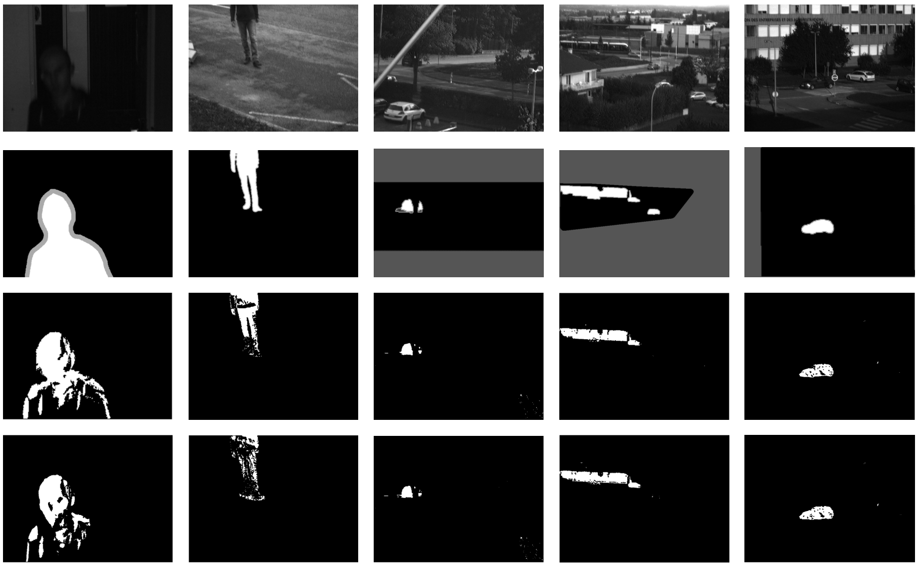

Figure 3.8: Background subtraction results on five videos. The top row is original multispectral sequences, the second row is the corresponding ground truth and the last two rows are the results obtained by the respect best combination of three-channel based multispectral and RGB sequences, respectively.

erage of F-measure on the five videos is calculated and listed in the rightmost column, from which, the accuracy (0.8399) of the best average three-channel combination on five video presents nearly 3% improvement than the result (0.8113) of RGB. As it is shown, the multispectral sequences can represent an alternative to conventional RGB sequences for the background subtraction task.

3.5/ Conclusion and future works

In this chapter, the original Codebook algorithm based on conventional RGB sequences is first stated in detail with five important issues, namely background model initialization, matching process, background model updating strategy, background model refining and foreground detection. Since multispectral sequences can provide more information compared with RGB triplets, the original Codebook algorithm is then adapted to multispectral case to investigate the advantages of multispectral sequences for the task of background subtraction.

In order to achieve this goal, two modifications have been performed comparing with the original RGB Codebook technique. Specifically speaking, the definition of brightness in

RGB is extended to multispectral case, with an arbitrary number of spectral channels. Besides, we replace the color distortion with spectral distortion, as the term of color is not suitable anymore in multispectral sequences. The implementation of multispectral Codebook has also been explained step by step, except for the background model refining, which is the same with that in the original Codebook algorithm.

We have conducted experiments on the FluxData FD-1665 dataset, which contains five scenes with various challenges for multispectral sequences and the corresponding RGB sequences, together with the groundtruth. For fair comparison, the same parameter set is adopted for RGB and three-dimensional multispectral sequences. Both visual and numerical results on five multispectral and RGB sequences are displayed. Experiments have showed that multispectral images may represent an alternative to conventional images in the background subtraction. These encouraging results open a door for future works for applying multispectral images in background subtraction.

However, the four parameters $\alpha, \beta, \varepsilon_1$ and ε_2 are adjusted experimentally carefully according to the empiric values recommended by the pioneering works of Codebook algorithm, which is very time consuming and not robust. When we want to continue the experiments on non three-dimensional multispectral sequences, we find it too inconvenient to obtain the optimal parameters by cumbersome parameter tuning. It becomes a must for automatic selection to further investigate the benefits of multispectral images in the current research work.

4

Multispectral self-adaptive Codebook and its Variants for background subtraction

4.1/ Introduction

In the previous chapter, we have extended the original three dimensional Codebook algorithm to the multispectral field. As a parametric method, the original Codebook needs parameter tuning to find the appropriate values for every scene and meanwhile the detection performance is heavily impacted by the parameters. As we know, the Codebook model devoted in the previous chapter has four basic following key parameters: α, β, ε_1 and ε_2. To be specific, α and β are used to obtain the brightness bounds from the min and max values $\check{I}_m$ and $\hat{I}_m$ in a certain codeword, with Equation 3.6, and, ε_1 and ε_2 are the spectral distortion thresholds used in the background model construction and foreground detection phases, respectively.

The fashionable way to get these parameters is empirical and experimental. The pioneers of this technique have provided the typical range of these parameters, namely, $\alpha \in [0.4, 0.7]$ and $\beta \in [1.1, 1.5]$ [Kim et al., 2005]. However, this is far from adequate, because manual parameters tuning is still required to achieve satisfying results for a specific scene, which is always a really cumbersome and tricky task for researchers. Besides, if the algorithm needs to be run for long periods of time, the parameters should not be static but could be automatically adapted to the environmental changes. What is more important for our research objective, when using the multispectral sequences, the parameters also have to be adjusted with different number of channels. Therefore, there is a need for further research with regard to realizing an automatic selection mechanism for optimal parameters.

Motivated by the work of [Shah et al., 2015], which has proposed the statistical parameter estimation method in YCbCr color space, we first propose a multispectral self-adaptive technique in this chapter for automatically optimal parameter selection to get rid of the tiring searching work. That is to say, those parameters listed above do not need to be obtained by burdensome experiments, but to be estimated from the data themselves statistically, which can help save a lot of efforts and time.

In order to further improve the multispectral model, two aspects can be considered [Ma et al., 2012]. The first one is to employ a better feature representation by discovering a new robust feature descriptor or combining different features together. The other is to introduce other strategies to build the model to represent the background. Thus later in this chapter, we will make attempts to develop the multispectral self-adaptive Codebook along these two directions.

The rest of this chapter is organised as follows. Section 4.2 presents the multispectral self-adaptive mechanism in detail. Based on this framework, Section 4.3 introduces another feature descriptor named spectral information divergence in the matching process. From Section 4.4 to Section 4.6, other approaches to build the multispectral self-adaptive Codebook are investigated, namely box-based Codebook, dynamic Codebook and fusion strategy, respectively. Finally, the experiments are conducted on the FluxData FD-1665 dataset introduced in previous chapter and discussed in Section 4.7, while conclusions and future work are presented in Section 4.8.

4.2/ Multispectral self-adaptive mechanism

A detailed description of the proposed multispectral self-adaptive mechanism is presented in this section. In order of achieve the goal of selecting the optimal parameters automatically, some additional statistical information need to be calculated iteratively and recorded for each codeword during the whole process. In spite of the vector of average spectral values of the codeword $\mathbf{v}_m$ and the six-tuple $\mathbf{aux}_m$, we record another vector named $\mathbf{S}_m$, which represents the set of the variances of the separate spectrums σ_i^2. Besides, for vectors $\mathbf{v}_m$ and $\mathbf{S}_m$, one more dimension is added to record the average of brightness I_m and its variance σ_I^2, respectively. Thus for the multispectral sequences with n channels, the vectors $\mathbf{v}_m$ and $\mathbf{S}_m$ are of $n+1$ dimension. The extra one channel stands for the numerical information of the brightness.

4.2.1/ Multispectral self-adaptive Codebook model initialization

The initialization strategy is kept the same for the six-tuple vector $\mathbf{aux}_m$ with the Codebook algorithm in the previous chapter, while it is modified a little for the average vector $\mathbf{v}_m$, with one more element related with brightness. Specifically speaking, for a new codeword of a given pixel, the initial value of $\mathbf{v}_m$ is:

$$\mathbf{v} = \mathbf{x} = (X_1, X_2, ..., X_n, I) \tag{4.1}$$

Meanwhile, the $\mathbf{S}_m$ of the new codeword is also a vector of dimension $n+1$, which is initialized with the square values of individual spectrum and brightness.

$$\mathbf{S} = (X_1^2, X_2^2, \dots, X_n^2, I^2) \tag{4.2}$$

4.2.2/ Multispectral self-adaptive Codebook matching process

In the multispectral self-adaptive Codebook framework, definitions of brightness and spectral distortion are kept the same with those in the previous chapter, as defined in Equations 3.15 and 3.17. During the matching process, the statistical information calculated and recorded step by step for each codeword is used to estimate both the brightness bounds and spectral distortion threshold. To be specific, the bounds of brightness can be estimated by

$$\begin{cases} I_{low} = \check{I}_m - \sigma_I \\ I_{high} = \hat{I}_m - \sigma_I \end{cases} \tag{4.3}$$

where σ_I is the standard deviation of brightness in the current codeword, whose square is the last element of vector $\mathbf{S}_m$. And the thresholds of the spectral distortion for background model construction phase and foreground detection phase are unified with ε, which is calculated by

$$\varepsilon = max([\sigma_1, \sigma_2, ..., \sigma_n]) \tag{4.4}$$

where σ_i, $i \in [1, n]$ is the standard deviation of the i^{th} channel value in the current codeword, whose square is the corresponding i^{th} element of vector $\mathbf{S}_m$.

With the self-adaptive mechanism, brightness bounds and spectral distortion threshold are obtained automatically and able to adjust themselves with statistical properties of the input sequences. In the phase of background model construction, for each pixel, when a new image arrives, the brightness and spectral distortion are first computed, then the matching process is conducted codeword by codeword. If the brightness of the new pixel lies in the current interval of the brightness bounds, as shown in Equation 4.3, and the spectral distortion is smaller than the current threshold ε of a certain codeword, the new pixel will be modeled as a perturbation on this background codeword. Unless, a new codeword will be seeded.

4.2.3/ Multispectral self-adaptive Codebook model updating

If a match is satisfied in the matching process, the corresponding codeword is updated with the information of the current pixel. Specifically speaking, the average multispectral vector $\mathbf{v}_m$ is updated as follows:

$$\mathbf{v}_m \leftarrow (\frac{f_m\overline{X}_{m1} + X_1}{f_m + 1}, \frac{f_m\overline{X}_{m2} + X_2}{f_m + 1}, \ldots, \frac{f_m\overline{X}_{mn} + X_n}{f_m + 1}, \frac{f_m\overline{I}_{mn} + I_n}{f_m + 1}) \tag{4.5}$$

As for the six-tuple auxiliary vector $\mathbf{aux}_m$ = ($\check{I}_m$, $\hat{I}_m$, f_m, λ_m, p_m, q_m), it is updated exactly the same way with that for the original Codebook algorithm, as listed in Table 3.2.

The vector $\mathbf{S}_m$ storing the variance of each separate spectrum and brightness is updated as follows:

$$\mathbf{S}_m \leftarrow (\frac{f_m\overline{\sigma}_{m1} + (X_1 - \overline{X}_{m1})^2}{f_m + 1}, \quad \frac{f_m\overline{\sigma}_{m2} + (X_2 - \overline{X}_{m2})^2}{f_m + 1}, \ldots, \frac{f_m\overline{\sigma}_{mn} + (X_n - \overline{X}_{mn})^2}{f_m + 1}, \\ \frac{f_m\overline{\sigma}_{Im} + (I - \overline{I}_m)^2}{f_m + 1}) \tag{4.6}$$

The modules for the model refining and the foreground detection thoroughly follow the logic of the original Codebook algorithm. That is to say, the codewords representing the moving object regions will be removed with the help of the information stored in the six-tuple vector $\mathbf{aux}_m$. After the establishment of background model, the matching process is conducted in the detection phase. The new pixel is classified as background if an acceptable matching codeword exists and the vectors in the current codeword will be also updated as illustrated above at the same time. Otherwise, the pixel is detected as foreground.

In this section, we have proposed multispectral self-adaptive mechanism to improve the Codebook algorithm. With this technique, the brightness bounds and spectral distortion thresholds are calculated automatically from the image data themselves statistically, not chosen empirically like the original Codebook, which is helpful for researchers to get rid of the cumbersome task of parameters tuning.

4.3/ SPECTRAL INFORMATION DIVERGENCE

From this section on, we will investigate techniques to improve the multispectral self-adaptive Codebook algorithm proposed in previous section. The first attempt introduced in this section is the utilization of the spectral information divergence to evaluate the spectral distance between the new pixel vector and that in the tested codeword in the matching process.

Based on the algorithms explained above, we can conclude that the main idea of Codebook background model construction is that, if the pixel of the current image is close enough to the average vector of the current codeword in the background model, it will be regarded as a perturbation on that codeword, unless, it will establish a new codeword to be associated with that pixel. However, how to measure this closeness, or in another way of comprehension, distance?

In multispectral Codebook algorithm, two criteria have been adopted to evaluate the distance between two vectors, the brightness and the spectral distortion. Specifically speaking, the brightness is simply the L2-norm of the related channels, defined with Equation 3.15, and the spectral distortion is measured as a function of the brightness-weighted angle between the current and reference spectral vectors, as illustrated in Equation 3.17. We should be aware that the combination of brightness and spectral distortion defined previously is not the only choice of estimation criteria.

Motivated by the spectral information divergence adopted in [Benezeth et al., 2014a], in this section, it is first employed to replace the spectral distortion in the previous multispectral self-adaptive Codebook model to be the matching criteria together with the brightness condition.

The whole diagram keeps the same with that in the previous section. Fig. 4.1 is the flow chart for background model construction. After the initialization of the Codebook for each pixel, the brightness and spectral information divergence are computed based on Equation 3.15 and Equation 2.21 when a new frame arrives. Besides, the same self-adaptive thresholding procedure is adopted for spectral information divergence as for spectral distortion. If a match is found, the current codeword will be updated with the spectral information of the new pixel, unless, a new codeword is created.

To further utilize the spectral information, the three features mentioned here are then employed together, as shown in Fig. 4.2. This step forward opens a door for other possibilities to seek novel kind of feature representation in the construction of the Codebook background model.

4.4/ Multispectral self-adaptive box-based Codebook

In the Codebook scheme, the background model is a set of pixel-wise codebooks consisting of the codewords. The original codeword has cylindrical structure which is visualized in Fig. 3.2 and devised based on an experiment observing how pixel values change over time. However, in essence, the cylindrical codeword has a disadvantage of complex structure and accordingly, it requires high computational cost for matching process.

To remedy the drawback, we are going to propose multispectral box-based Codebook

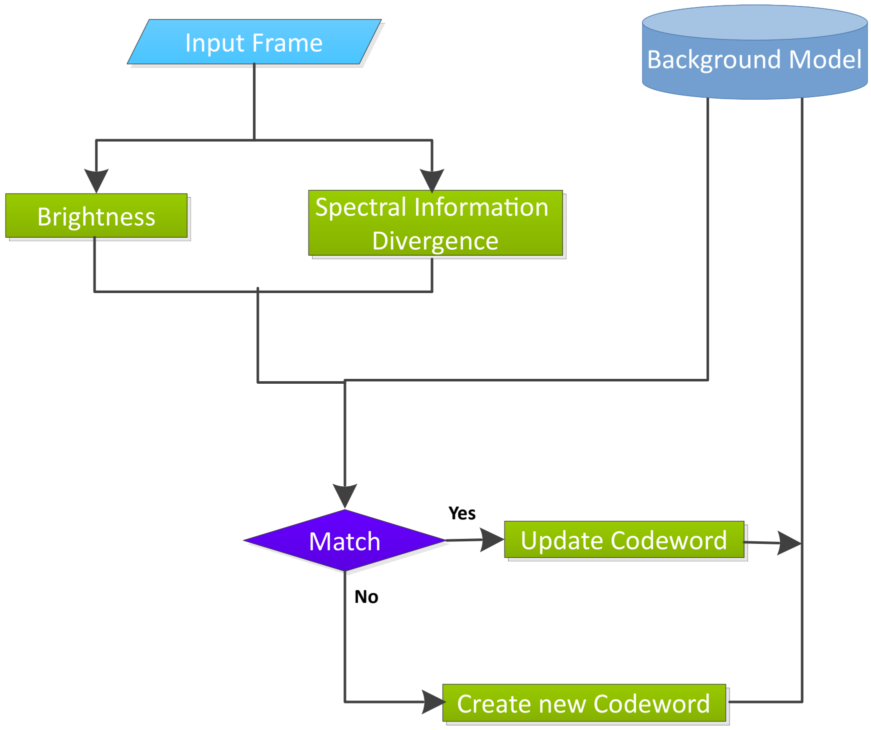

Figure 4.1: Codebook construction with brightness and spectral information divergence

with another codeword structure, which is motivated by the idea of [Tu et al., 2008] and [Noh et al., 2012]. We improve their work by making the boundary self-adaptive and extending the structure to multispectral case.

4.4.1/ Multispectral box-based Codebook model initialization

The codebook for every pixel is an empty set and the number of the codewords is set to 0 at the beginning. When the first multispectral frame comes, the Codebook model is established with the first codeword consisting of **v**, **aux** and S, initialized with the spectral values of all the channels.

$$\mathbf{v} = \mathbf{x} = (X_1, X_1, ..., X_n, I) \tag{4.7}$$

where n is the number of multispectral channels and the brightness I is calculated with Equation 3.15.

The auxiliary information is defined as a four-tuple vector **aux** = (f, λ, p, q), without the information for the minimum and maximum of brightness. The meanings of these four

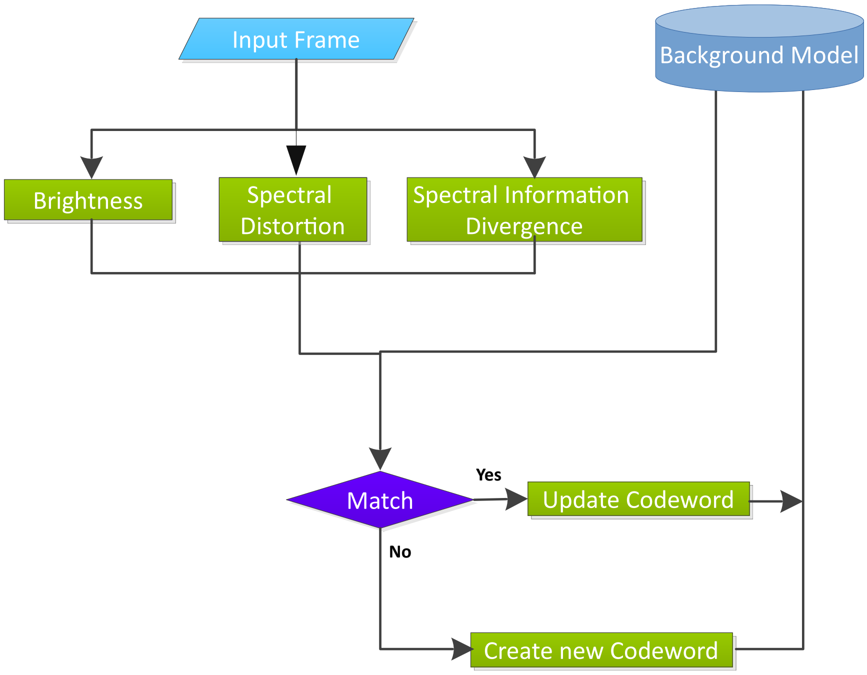

Figure 4.2: Codebook construction with three features

elements keep the same as in Table 3.1 and they are initialized as:

$$\mathbf{aux} = (1, 0, 1, 1) \tag{4.8}$$

S only contains the variance of brightness, which is initialized with:

$$S = I^2 \tag{4.9}$$

4.4.2/ Multispectral box-based Codebook matching process

The main difference lies here. The matching process is much more direct compared with that for the cylindrical Codebook model. For an input pixel at time instant t, with the current pixel value $\mathbf{x}_t = (X_1, X_2, \ldots, X_n)$, a matching codeword $\mathbf{v}_m = (V_1, V_2, \ldots, V_n)$ is found if

$$|X_i - V_i| \leq \sigma_I \tag{4.10}$$

is satisfied for each channel i, where σ_I is the standard deviation of brightness in the

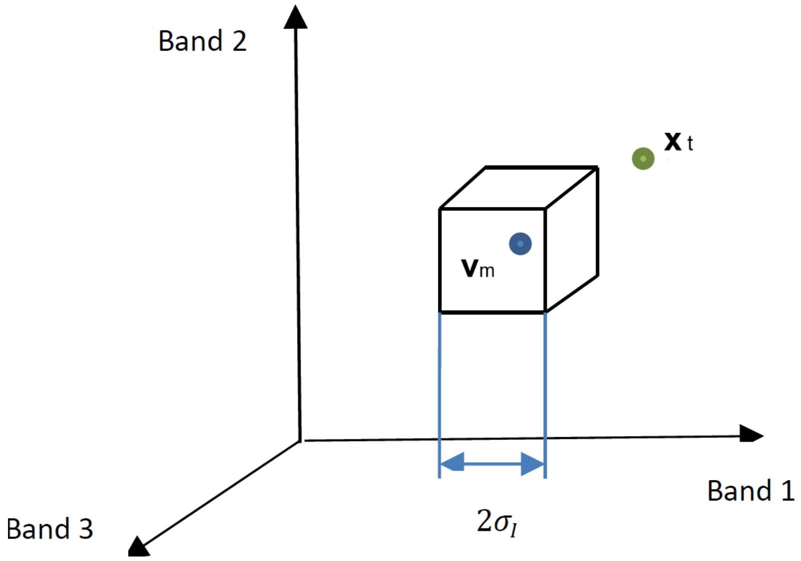

Figure 4.3: Box-based Codebook Model

current codeword, whose square is the element of S.

To make it clear to understand, the box-based Codebook model is represented in Figure 4.3, using three channels for example, whereas, it is very easy to be extended for multispectral images with more than three channels. The blue point in the center of the box denotes the average vector $\mathbf{v}_m$ for the current codeword. According to Equation 4.10, a match is found only when the new coming pixel green point $\mathbf{x}_t$ locates inside the box, whose side length is $2 \times \sigma_I$.

4.4.3/ Multispectral box-based Codebook model updating

Like all algorithms with the framework of Codebook illustrated above, when a new frame arrives, the matching criteria in Equation 4.10 is first evaluated. If it is satisfied, the corresponding codeword is updated with the information of the current pixel.

specifically speaking, the average multispectral vector $\mathbf{v}_m$ is updated with:

$$\mathbf{v}_m \leftarrow (\frac{f_m\overline{X}_{m1} + X_1}{f_m + 1}, \frac{f_m\overline{X}_{m2} + X_2}{f_m + 1}, \ldots, \frac{f_m\overline{X}_{mn} + X_n}{f_m + 1}, \frac{f_m\overline{I}_{mn} + I_n}{f_m + 1}) \tag{4.11}$$

As for the four-tuple $\mathbf{aux}_m$ = (f_m, λ_m, p_m, q_m), it is updated exactly the same way with the original Codebook algorithm, as listed in Table 3.2.

S_m storing the variance of the brightness is updated with:

$$S_m \leftarrow \frac{f_m\overline{\sigma}_{Im} + (I - \overline{I}_m)^2}{f_m + 1} \tag{4.12}$$

The algorithm of the multispectral box-based Codebook construction is summarized in Algorithm 3, where N is the number of frames.

Algorithm 3 Multispectral box-based Codebook construction

```
$L \leftarrow 0$, $\mathbf{C} \leftarrow \phi$
for $t = 1 \rightarrow N$ do
    $\mathbf{x}_t = (X_1, X_1, ..., X_n)$
    Find the matching codeword $\mathbf{c}_m$ to $\mathbf{x}_t$ in $\mathbf{C}$ if the following condition is satisfied for each channel.
    $|X_i - V_i| \leq \sigma_I$
    if $\mathbf{C} = \phi$ or there is no match then
        $L \leftarrow L + 1$, create a new codeword $\mathbf{c}_L$
        $\mathbf{v}_0 = \mathbf{x}_t$
        $\mathbf{aux}_0 = (1, t-1, t, t)$
        $S = I^2$
    else
        update the matched codeword
        $\mathbf{v}_m \leftarrow (\frac{f_m\overline{X}_{m1}+X_1}{f_m+1}, \frac{f_m\overline{X}_{m2}+X_2}{f_m+1}, \ldots, \frac{f_m\overline{X}_{mn}+X_n}{f_m+1}, \frac{f_m\overline{I}_{mn}+I_n}{f_m+1})$
        $\mathbf{aux}_m \leftarrow (f_m + 1, max\lambda_m, t - q_m, p_m, t)$
        $S_m \leftarrow \frac{f_m\overline{\sigma}_{Im}+(I-\overline{I}_m)^2}{f_m+1}$
    end if
end for
```

Subsequently, the modules for the background model refining and the foreground detection thoroughly follow the way of the original Codebook algorithm.

4.5/ Multispectral self-adaptive dynamic Codebook

In this section, we are going to propose a mutispectral dynamic Codebook algorithm, which is motivated by the work of [Ruidong, 2015] and [Kusakunniran et al., 2016], by extending the three-channel based Codebook algorithm to multispectral images with dynamic boundary mechanism for individual channel. It follows the overall workflow of original Codebook model. We will illustrate the details step by step.

4.5.1/ Multispectral dynamic Codebook model initialization

The Codebook model is initialized when the first multispectral frame comes by constructing an associated codeword for each pixel with the corresponding vector **v** being set to be the spectral values of all the channels for that pixel as below:

$$\mathbf{v} = \mathbf{x} = (X_1, X_2, \ldots, X_n) \tag{4.13}$$

where n is the number of channels. The auxiliary information will be defined as a four-tuple $\mathbf{aux}_m$ = (f_m, λ_m, p_m, q_m), which follows the strategy of box-based Codebook, and they are initialized as:

$$\mathbf{aux} = (1, 0, 1, 1) \tag{4.14}$$

In spite of the vector **v** storing the average spectral values of the codeword and the auxiliary tuple **aux**, we also record a third vector named **S**, which represents the set of the variances of the separate spectrums σ_i^2 and is initialized as:

$$\mathbf{S} = (X_1^2, X_2^2, \ldots, X_n^2) \tag{4.15}$$

What's more, another two vectors need to be recorded for the minimum and maximum values for each channel and they are initialized with the spectral values of the first multispectral frame:

$$\mathbf{B}\_min = (X_1, X_2, \ldots, X_n) \tag{4.16}$$

$$\mathbf{B}\_max = (X_1, X_2, \ldots, X_n) \tag{4.17}$$

4.5.2/ Multispectral dynamic Codebook matching process

The matching condition is different from that in the multispectral Codebook algorithms illustrated above. For an input pixel at time instant t, with the current spectral value $\mathbf{x}_t$ =

$(X_1, X_2, \ldots, X_n)$, the matching codeword is found if

$$B\_low_i \leq X_i \leq B\_high_i \tag{4.18}$$

is satisfied for each channel i, $i \in [1, n]$, where B_low and B_high denote the lower and upper boundaries, respectively. The lower boundary for each channel B_low_i can be obtained with the corresponding minimum value stored in $\mathbf{B}\_min$ and the standard deviation in the current codeword by

$$B\_low_i = max(B\_min_i - \sigma_i,\ 0) \tag{4.19}$$

The upper boundary is obtained with the corresponding maximum value stored in $\mathbf{B}\_max$ and the standard deviation by

$$B\_high_i = max(B\_max_i + \sigma_i,\ 255) \tag{4.20}$$

where σ_i is the standard deviation of the i^{th} channel value in the current codeword, whose square is the corresponding i^{th} element of $\mathbf{S}_m$. It is observed that, during this matching process, the boundaries are acquired from the data themselves and no manual parameters tuning is required, which makes the proposed model more practical.

4.5.3/ Multispectral dynamic Codebook model updating

Like the original Codebook algorithm, when a new frame arrives, the matching criteria is first evaluated to see whether it is satisfied. If a match is found, the corresponding codeword is updated with the information of the current pixel.

Specifically speaking, the average multispectral vector $\mathbf{v}_m$ is updated as below.

$$\mathbf{v}_m \leftarrow (\frac{f_m \overline{X}_{m1} + X_1}{f_m + 1}, \frac{f_m \overline{X}_{m2} + X_2}{f_m + 1}, \ldots, \frac{f_m \overline{X}_{mn} + X_n}{f_m + 1}) \tag{4.21}$$

As for the four-tuple vector $\mathbf{aux}_m = (\ f_m,\ \lambda_m,\ p_m,\ q_m\ )$, it is updated exactly the same way as the original Codebook algorithm.

The vector $\mathbf{S}_m$ storing the variance of each separate spectrum is updated with:

$$\mathbf{S}_m \leftarrow (\frac{f_m\overline{\sigma}_{m1} + (X_1) - \overline{X}_{m1}^{\;2}}{f_m + 1}, \frac{f_m\overline{\sigma}_{m2} + (X_2) - \overline{X}_{m2}^{\;2}}{f_m + 1}, \ldots, \frac{f_m\overline{\sigma}_{mn} + (X_n) - \overline{X}_{mn}^{\;2}}{f_m + 1}) \tag{4.22}$$

The two vectors for the minimum and maximum values for each channel, namely,

$$\mathbf{B}\_min = (B\_min_1, B\_min_2, \ldots, B\_min_n) \tag{4.23}$$

$$\mathbf{B}\_max = (B\_max_1, B\_max_2, \ldots, B\_max_n) \tag{4.24}$$

are updated as follows:

$$B\_min_i \leftarrow min(B\_min_i, X_i) \tag{4.25}$$

$$B\_max_i \leftarrow max(B\_max_i, X_i) \tag{4.26}$$

The process of the multispectral self-adaptive dynamic Codebook construction is summarized in Algorithm 4, where N is the number of frames. The modules for the model refining and the foreground detection are the same as those used in the original Codebook algorithm.

Algorithm 4 Multispectral dynamic Codebook construction

```
$L \leftarrow 0$, $\mathbf{C} \leftarrow \phi$
for $t = 1 \rightarrow N$ do
    $\mathbf{x}_t = (X_1, X_1, ..., X_n)$
    Find the matching codeword to $\mathbf{x}_t$ in C if the following condition is satisfied for each channel.
    $B\_low_i \leq X_i \leq B\_high_i$
    if $\mathbf{C} \leftarrow \phi$ or there is no match then
        $L \leftarrow L + 1$, creat a new codeword $\mathbf{c}_L$
        $\mathbf{v}_0 = \mathbf{x}_t$
        $\mathbf{aux}_0 = (1, t - 1, t, t)$
        $\mathbf{S}_0 = (X_1^2, X_2^2, \ldots, X_n^2)$
        $\mathbf{B}\_min = (X_1, X_2, \ldots, X_n)$
        $\mathbf{B}\_max = (X_1, X_2, \ldots, X_n)$
    else
        update the matched codeword
        $\mathbf{v}_m \leftarrow (\frac{f_m\overline{X}_{m1}+X_1}{f_m+1}, \frac{f_m\overline{X}_{m2}+X_2}{f_m+1}, \ldots, \frac{f_m\overline{X}_{mn}+X_n}{f_m+1})$
        $\mathbf{aux}_m \leftarrow (f_m + 1, max\lambda_m, t - q_m, p_m, t)$
        $\mathbf{S}_m \leftarrow (\frac{f_m\overline{\sigma}_{m1}+(X_1)-\overline{X}_{m1}^{\ 2}}{f_m+1}, \frac{f_m\overline{\sigma}_{m2}+(X_2)-\overline{X}_{m2}^{\ 2}}{f_m+1}, \ldots, \frac{f_m\overline{\sigma}_{mn}+(X_n)-\overline{X}_{mn}^{\ 2}}{f_m+1})$
        $B\_min_i \leftarrow min(B\_min_i, X_i)$
        $B\_max_i \leftarrow max(B\_max_i, X_i)$
    end if
end for
```

4.6/ Multispectral self-adaptive fusion strategy

Another step forward to exploit benefits of each spectral channel of multispectral images is to fuse the detection results of the monochromatic channels. The idea is very straight forward after we introduce the mechanism of getting dynamic boundary for individual channel. We first employ the multispectral self-adaptive dynamic Codebook which has been discussed in detail in the previous subsection to each channel separately and obtain multiple foreground background binary masks independently. Then the detection results

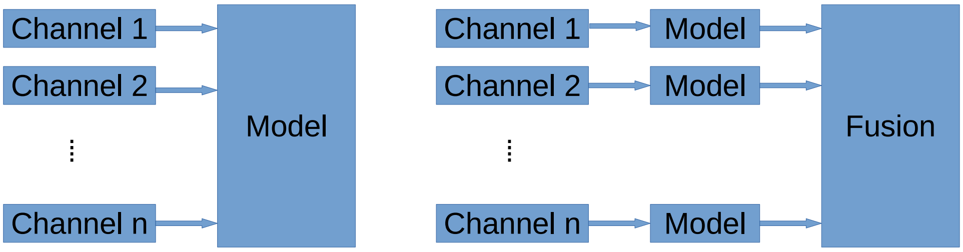

Figure 4.4: Two Different Pipelines

of the monochromatic channels are fused via union, vote or intersection to get the final foreground background segmentation result.

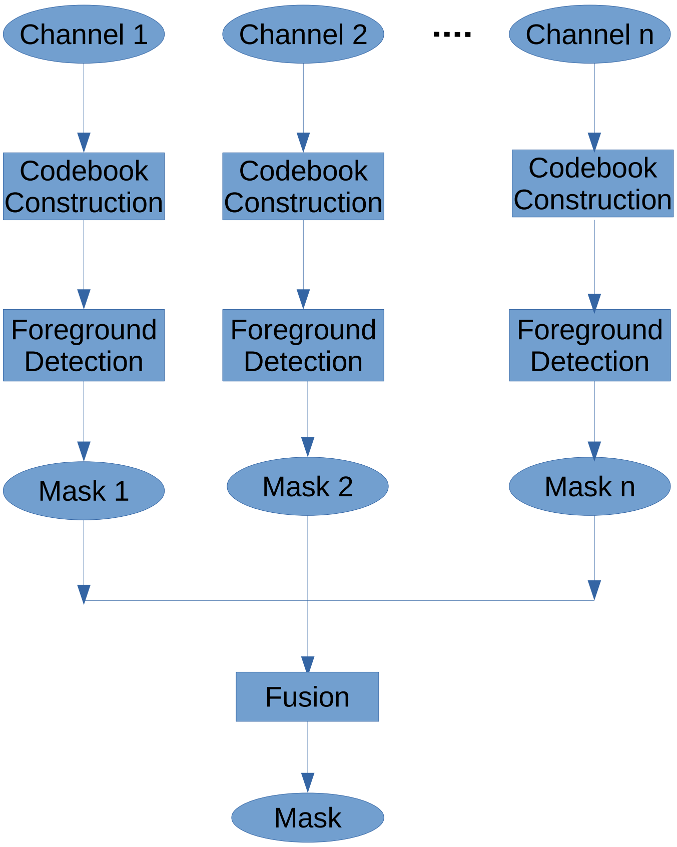

Figure 4.5: Workflow Of Fusion Strategy for Multispectral self-adaptive Codebook

As we know, all the algorithms presented before this section belong to the pipeline on the left in Fig. 4.4, where all the channels used will be fed to the model together. Thus each channel has an influence on whether a match could be found in both background model construction phase and foreground detection phase. That is to say, all the channels shape the model jointly. However, in the fusion pipeline illustrated on the right in Fig. 4.4, each channel decides one model independently, then the individual detection results are fused to obtain the final output. The workflow for multispectral self-adaptive fusion strategy is shown in Fig. 4.5, where each channel defines a background model and obtains a foreground background mask with the self-adaptive dynamic technique proposed in Section 4.5. Then these n masks together produce a final mask with fusion.

4.7/ Experimental results

The experiments have also been conducted on the FluxData FD-1665 dataset and will be presented in two categories for clearer statement. The first part illustrates experiments of multispectral self-adaptive Codebook with cylindrical structure, which consists of the algorithms proposed in Section 4.2 and Section 4.3. The second one evaluates the performance with other multispectral self-adaptive structures, including box-based Codebook, dynamic Codebook and fusion strategy proposed in Section 4.4, Section 4.5 and Section 4.6, respectively.

All the multispectral Codebook algorithms proposed in this chapter are based on the self-adaptive mechanism. That is to say, we don't need to provide the parameters adjusted empirically. The experiments are conducted on n channels on the five multispectral video sequences in the FluxData FD-1665 dataset, where $n \in [3, 7]$, not only for three channels as in the previous chapter.

In the following, the experiments are firstly conducted with different strategies illustrated above on the thirty-five different three-channel-based combinations, thirty-five different four-channel based combinations, twenty-one different five-channel based combinations, seven different six-channel based combinations, and total seven channels, together with the RGB for five videos. Then, only the largest F-measures among different combinations with same number of channels, or bands, are selected and listed in this section.

4.7.1/ Experiments of multispectral self-adaptive Codebook with cylindrical structure

In Table 4.1, brightness (B) and spectral distortion (SD) are used for evaluating the spectral distance between incoming pixel and the average vector in the current codeword. The largest F-measure for each video is in bold and the average F-measures for n bands of

Table 4.1: Best F-measures for multispectral Codebook with B+SD on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands	RGB
1	0.7995	0.8046	**0.8060**	0.8043	0.7983	0.4789
2	0.9615	0.9624	0.9636	**0.9643**	0.9631	0.9535
3	0.9231	**0.9248**	0.9204	0.9051	0.8381	0.9188
4	0.8981	**0.9001**	0.8999	0.8918	0.8856	0.8871
5	0.9171	**0.9198**	0.9190	0.9189	0.9110	0.9130
mean	0.8999	0.9023	0.9018	0.8969	0.8792	0.8303

multispectral sequences on five videos are listed in the last row.

As indicated in Table 4.1, multispectral sequences can always outperform the corresponding RGB sequences. The combination with four channels performs best, with 7.2% higher F-measure than the RGB result on average. The combinations with three, five and six channels also have good performance with slight difference. The fact that the seven-band-based combination performs worst and even the six-channel situation has little accuracy decrease shows that, it is not always a wise decision to adopt as many channels as we have. There may exist information cancellation with more channels. Nevertheless, the combination with seven channels still can be nearly 5% more than the RGB result on average over five videos.

In the following experiments, the spectral information divergence explained in Section 4.3 is employed to replace the spectral distortion in the matching process. As stated above, the experiments for different n-band-based multispectral sequences are conducted and the largest F-measures are selected and listed in Table 4.2. Here, the judging criteria are brightness (B) and spectral information divergence (SID).

From Table 4.2, with this new set of criteria, the multispectral sequences still have better performance than the RGB sequences. Same as in Table 4.1, the four-band-based combination achieves the best results, with a 7.8% higher accuracy on average. The combination of three channels and five channels also perform well with a negligible gap while six-channel-based and seven-channel-based situations have much lower F-measure.

In the third experiment, brightness (B), spectral distortion (SD) and spectral information divergence (SID) are adopted together to determine the distance between two spectral vectors in the matching process, during which, the self-adaptive threshold is shared by spectral distortion and spectral information divergence. From the results on the five videos and each combination, the best F-measures for each column are extracted and listed in

Table 4.2: Best F-measures for multispectral Codebook with B+SID on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands	RGB
1	0.9208	**0.9219**	0.9059	0.8676	0.7883	0.6355
2	**0.9538**	0.9526	0.9504	0.9471	0.9451	0.9479
3	**0.8939**	0.8914	0.8825	0.8766	0.8351	0.8867
4	0.8784	**0.8807**	0.8783	0.8728	0.8558	0.8217
5	0.8765	0.8801	**0.8842**	0.8425	0.7855	0.8447
mean	0.9047	0.9053	0.9003	0.8813	0.8420	0.8273

Table 4.3: Best F-measures for multispectral Codebook with B+SD+SID on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands	RGB
1	0.9144	**0.9147**	0.8971	0.8607	0.7727	0.6555
2	0.9614	0.9623	0.9635	**0.9642**	0.9631	0.9535
3	**0.9213**	0.9180	0.8938	0.8634	0.8045	0.9054
4	0.8968	**0.8979**	0.8972	0.8885	0.8821	0.8867
5	0.8791	0.8800	**0.8948**	0.8459	0.7853	0.8543
mean	0.9146	0.9146	0.9093	0.8845	0.8415	0.8511

Table 4.3.

As we can see from Table 4.3, when all the seven channels are fed to the model, a worse detection result than that of RGB sequence has been produced. However, with the combination of three, four and five channels, we can still have a satisfying 6% higher F-mesaure.

From Table 4.1 to Table 4.3, regardless of different judging criteria used in matching process, multispectral sequences show an attractively better performance than the traditional RGB sequences, only except for the situation of seven channels with all three criteria. Besides, combinations of three, four and five are promising to obtain a more accurate foreground detection results. The six-channel-based situation is less better while that of seven channels together is the worst.

We'd like to conduct a brief summary for the experiment results obtained until now. the best multispectral results not only from Table 4.1 to Table 4.3 above, but also from Table 3.6 in previous chapter are extracted and listed in Table 4.4, together with the corresponding RGB results.

In Table 4.4, the first category, using brightness and spectral distortion with static parameter mechanism, records the best multispectral and RGB results of each sequence taken from Table 3.6. In the self-adaptive mechanism, the same items for three different

Table 4.4: Best F-measures with different mechanisms and sets of criteria

Mechanism	criteria	Images	Video 1	Video 2	Video 3	Video 4	Video 5	Mean
Static parameters	B+SD	RGB	0.8086	0.9431	0.7578	0.7679	0.7789	0.8113
		Multi	0.8718	0.9546	0.7921	0.8650	0.8241	0.8615
Self-adaptive mechanism	B+SD	RGB	0.4789	0.9535	0.9188	0.8871	0.9130	0.8303
		Multi	0.8060	**0.9643**	**0.9248**	**0.9001**	**0.9198**	0.9030
	B+SID	RGB	0.6355	0.9479	0.8867	0.8217	0.8447	0.8273
		Multi	**0.9219**	0.9538	0.8939	0.8807	0.8842	0.9069
	B+SD+SID	RGB	0.6555	0.9535	0.9054	0.8867	0.8543	0.8511
		Multi	0.9147	0.9642	0.9213	0.8979	0.8948	**0.9186**

sets of criteria are also extracted from Table 4.1 to Table 4.3. The corresponding average F-measures on the five sequences are calculated and listed in the last column.

From Table 4.4, we can see that on the Videos 2 to 5, which are outdoor scenes, it performs best to adopt multispectral self-adaptive technique using the brightness and spectral distortion as matching criteria. What needs to mention is that, in this process researchers do not have to take time and effort to search for the appropriate parameters. For the indoor Video 1, the utilization of the spectral information divergence does great help. The F-measure shows a great jump when SD is replaced by SID. When the three criteria are used together, the performance drops little from the B+SID combination, but still far better than that of B+SD. If all videos are considered, judging from the mean F-measures, the three-criteria based multispectral self-adaptive Codebook is the most promising choice.

4.7.2/ Experiments of multispectral self-adaptive Codebook with other structures

In the following experiments, we go on evaluating the performance of the multispectral self-adaptive box-based Codebook, dynamic Codebook and fusion strategy proposed in Section 4.4, Section 4.5 and Section 4.6, respectively.

Table 4.5 lists the selected largest F-measures with multispectral self-adaptive box-based Codebook. The combination with three channels performs best and multispectral images with other numbers of channels all can produce a higher F-measure compared with RGB images.

The second experiment in this section is conducted for multispectral self-adaptive dynamic Codebook. The largest F-measures are selected and listed in Table 4.6, from

Table 4.5: Best F-measures with multispectral box-based Codebook on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands	RGB
1	**0.8538**	0.8272	0.8262	0.8270	0.8312	0.4319
2	0.9522	**0.9523**	0.9519	0.9502	0.9462	0.9480
3	0.8853	0.8849	0.8855	0.8840	0.8763	**0.8938**
4	0.8864	0.8874	**0.8884**	0.8876	0.8859	0.8843
5	**0.8890**	0.8801	0.8694	0.8597	0.8482	0.8707
mean	0.8933	0.8864	0.8843	0.8817	0.8776	0.8057

Table 4.6: Best F-measures with multispectral dynamic Codebook on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands	RGB
1	0.8669	0.8750	**0.8757**	0.8749	0.7759	0.8011
2	0.9607	**0.9609**	0.9607	0.9603	0.9307	0.9584
3	0.9456	**0.9464**	0.9460	0.9445	0.8321	0.9414
4	0.8693	**0.8731**	0.8734	0.8728	0.8661	0.8450
5	0.9010	**0.9024**	0.8969	0.8966	0.8719	0.8785
mean	0.9087	0.9116	0.9105	0.9098	0.8553	0.8849

which, combinations with three, four, five and six channels have a similar better performance, while that with seven channels performs worse than RGB images.

The last experiment is for multispectral self-adaptive fusion strategy. Here we will only explore on the thirty-five different three-channel-based combinations, thirty-five different four-channel-based combinations, twenty-one different five-channel-based combinations, seven different six-channel-based combinations, and total seven-channel case, but not the RGB. Because we just want to investigate whether Codebook-based models with other codeword structures will do good to background subtraction task with multispectral images. The best F-measures are selected and shown in Table 4.7, where the F-measure obtained with three, four and five are attractively promising, which is consistent with the results of former experiments.

The results of all the proposed methods are further compared in Table 4.8, by selecting the largest F-measure for each video with the methods illustrated above, in which, the brightness (B), spectral distortion (SD) and spectral information divergence (SID) are adopted to build the cylinder background model. The average F-measures for all the five videos and outdoor sequences are also calculated and listed in the last two rows of this table.

The F-measures with Pooling, the algorithm proposed in the original multispectral dataset paper [Benezeth et al., 2014a], are also listed in the 2^{th} column in Table 4.8 as a refer-

Table 4.7: Best F-measures with fusion strategy on the five videos

Video	3 Bands	4 Bands	5 Bands	6 Bands	7 Bands
1	**0.8967**	0.8836	0.8805	0.8712	0.7796
2	0.9607	**0.9608**	0.9607	0.9604	0.9575
3	**0.9372**	0.9346	0.9277	0.9020	0.8059
4	0.8685	0.8732	0.8741	**0.8742**	0.8709
5	0.8847	**0.8913**	0.8820	0.8646	0.8429
mean	0.9096	0.9087	0.9050	0.8945	0.8514

Table 4.8: Overall evaluation of proposed mechanisms on the five videos

Video	Pooling	B+SD	B+SID	B+SD+SID	Box-based	Dynamic	Fusion
1	0.7984	0.8060	0.9219	0.9147	0.8538	0.8757	0.8967
2	0.8815	0.9643	0.9538	0.9642	0.9523	0.9609	0.9608
3	0.6487	0.9248	0.8939	0.9213	0.8855	0.9464	0.9372
4	0.8392	0.9001	0.8807	0.8979	0.8884	0.8734	0.8742
5	0.7704	0.9198	0.8842	0.8948	0.8890	0.9024	0.8913
Outdoor mean	0.7850	**0.9273**	0.9032	0.9196	0.9038	0.9208	0.9160
Mean	0.7876	0.9030	0.9069	**0.9186**	0.8938	0.9118	0.9121

ence. As reviewed in the second chapter, Pooling also need to select the combination of some spectral channels with the highest accuracy, as we have done for the algorithms in this chapter.

From Tables 4.1, to 4.3, and Tables 4.5 to 4.7, which show the results of six self-adaptive algorithms we have proposed in this chapter, the largest F-measure never appear when all seven channels are used. The combination with four channels has the largest possibility to achieve the best performance. This agrees with the assertion deduced with the Pooling method by Yannick et al. [Benezeth et al., 2014a] that only few channels actually define the moving objects.

From average F-measures for the whole dataset, the approaches of codeword with other structures can produce comparable results to those methods that utilize all the three criteria with cylindrical structure, listed in 5^{th} column in Table 4.8, which proves the effectiveness of methods with other structures. As we can see for the method adopting B and SD, the accuracy for outdoor scenes outperforms in average all the other mechanisms, but it achieves less satisfactory result for the indoor video. The last three techniques proposed with non-cylindrical structures, especially the multispectral self-adaptive dynamic Codebook algorithm can be a compromising solution.

Another advantage is the low complexity of the matching equations, as the multispectral channels are processed separately only utilizing the intensity value of each channel and no correlation between channels need to be considered and calculated in the matching process. Besides, for all the algorithms listed here, it is quite easy to adapt to multispectral images with any number of channels for background subtraction.

4.8/ Conclusion and future works

In this chapter, we have achieved significant improvements by investigating several enhancements on the Codebook framework for background subtraction, using multispectral sequences. The main contributions can be concluded as follows.

Firstly, a multispectral self-adaptive mechanism has been designed to obtain the parameters automatically based on the statistical information extracted from the data themselves, while the parameters in the original version are selected empirically and experimentally. The self-adaptive technique makes the algorithm solid, reliable and robust, as the detection results of Codebook are not heavily impacted by the parameters any more, let alone the time and effort to search for the optimal parameters. This technique is particularly important in the research for multispectral case, as the parameters should be varied with different number of channels used.

Then we have explained in detail the improvements in the framework of multispectral self-adaptive Codebook model that show performance increase in two aspects. The first one is to employ another criteria named spectral information divergence in the matching process. The second aspect is to introduce other strategies to build the model to represent the background.

In the multispectral self-adaptive Codebook framework, we have conducted experiments with the matching criteria as the combination of brightness and spectral distortion, the combination of brightness and spectral information divergence, and the combination of these three distance evaluation together in the matching process. The threshold for spectral information divergence is obtained and updated with the same self-adaptive mechanism of that for spectral distortion. The results clearly show that the multispectral self-adaptive Codebook is more capable of detecting moving object regions than the multispectral Codebook extended directly from the original Codebook devoted in the previous chapter.

Furthermore, we have proposed three techniques to build the background model based on Codebook algorithm for multispectral images: box-based Codebook, dynamic Codebook and fusion strategy, each of which processes the multispectral channels independently. Specifically speaking, only the intensity value for each channel is used to calculate the spectral similarity between the new frame pixel and reference one in current codeword. Besides, like the algorithms illustrated above, the thresholds are not set in advance empirically and fixed for the whole procedure, but obtained based on statistical information extracted from the data and can always adjust themselves to the scene changes. Results demonstrated that we can acquire a comparable accuracy using much simpler matching equations than the aforementioned methods based on cylinder model.

These improvements forward altogether offer new insight for future works for using multispectral sequences for robust detection and motion analysis for moving objects detection. In next chapter, we investigate the use of deep learning technique for background subtraction with multispectral images, since it acquires impressive accuracy and attracting performance in computer vision community.

5

Multispectral background subtraction with deep learning

5.1/ Introduction

In this decade, deep learning based on the work of Yann LeCun et al. in 1989 [LeCun et al., 1989], has revolutionized computer vision, and deep features obtained from Convolutional Neural Networks (ConvNets also called CNNs) have been shown as powerful and effective image representations for various computer vision tasks such as object classification [Krizhevsky et al., 2012] [Simonyan et al., 2014] [Szegedy et al., 2015] [He et al., 2016], object detection [Uijlings et al., 2013] [Girshick et al., 2014] [Girshick, 2015] [Ren et al., 2015] [Redmon et al., 2016] semantic segmentation [Long et al., 2015][Badrinarayanan et al., 2017] [He et al., 2017]. Recently, inspired by the impressive achievement of deep learning, background subtraction based on deep learning shows great success and is now becoming a hot research topic due to its high precision [Lim et al., 2018] [Lim et al., 2019].

The first attempt to apply ConvNets for background subtraction problem was conducted by Braham and Van Droogenbroeck in 2016 [Braham et al., 2016]. Since then, Numerous supervised-based deep learning papers [Babaee et al., 2018] [Lim et al., 2018] [Lim et al., 2017] [Lim et al., 2019] [Zheng et al., 2019b] [Zheng et al., 2018b] [Wang et al., 2017] [Bakkay et al., 2018] have been published in the field of background subtraction. Currently, the top background subtraction methods in the large-scale dataset CDnet2012 [Goyette et al., 2012] and its extension CDnet 2014 [Wang et al., 2014] are based on ConvNets with a large gap of performance in comparison to conventional approaches. Among all these supervised-based deep learning methods, the method called FgSegNet _v2 [Lim et al., 2019] outperforms state-of-the-art approaches.

Moreover, the majority of the current background methods in this research community are focused on visible images or Red-Green-Blue (RGB). With the rise of different sensors, multi-modal foreground detection, which integrates multiple complementary data like visible and thermal infrared sources, has received more and more attention recently [Zheng et al., 2019a]. Compared with visible images, background subtraction using multispectral images can be more interesting because of the better spectral resolution. Thanks to the recent advances in technology, new products such as the FD-1665 Multispectral Cameras from FluxData are commercially available to offer the possibility to record multispectral images of more than three spectral channels in the visible and near infra-red(NIR) part of the spectrum simultaneously [Benezeth et al., 2014a].

To our knowledge, the FluxData FD-1665 dataset is the only public real multispectral image background subtraction dataset. Most public image datasets built for background subtraction, or change detection, such as the well-known Wallflower dataset [Toyama et al., 1999], the Stuttgart Artificial Background Subtraction (SABS) dataset [Brutzer et al., 2011] and CDnet [Goyette et al., 2012] [Wang et al., 2014], are based on visible spectral images or still recombined images. For example, the Grayscale-Thermal Foreground Detection (GTFD) dataset [Li et al., 2016] provides pairs of grayscale and thermal frames to investigate the fusion methods of thermal and grayscale data for effective foreground detection.

According to the above observations and motivated by the impressive accuracy of FgSegNet_v2 [Lim et al., 2019] for foreground segmentation with RGB images, we made an attempt to investigate the potential benefits of using multispectral images via convolutional neural networks for background subtraction in this chapter, based on the FgSegNet_v2 model.

The major contributions of this work lie in two aspects. Firstly, we extracted three channels out of seven of the FluxData FD-1665 multispectral dataset to match the number of input channels of the FgSegNet_v2 deep model, aiming to investigating the possible improvements against RGB. Secondly, a new convolutional encoder was proposed to utilize all the multispectral channels available to further explore the information of multispectral images. To the best of our knowledge, this work is the first attempt to investigate the potential benefits of using multispectral information via deep features learned with convolutional neural networks for the background subtraction task.

The rest parts of this chapter are organized as follows. Section 5.2 briefly discusses the background and related works. The proposed attempts utilising the ConvNets-based multispectral background subtraction methods are explained in Sections 5.3. Sections 5.4 illustrates the experimental evaluation and the obtained results compared with other approaches using all the channels of the same multispectral dataset. In Section 5.5, we conclude our work and provide some future works.

5.2/ Related works

We will discuss some related works regarding deep learning, convolutional neural networks. Specifically, we will focus our task by doing a very brief review of VGG networks and the encoder-decoder architecture, as they are related with the deep model of FgSegNet_v2 we are going to use.

5.2.1/ Deep learning

Recent advances in Artificial Intelligence (AI) and machine learning, especially the emerging field of deep learning, have changed the way we process, analyse and manipulate data. The schematic relationship among these three terminologies are provided with Fig. 5.1. Machine learning is a sub-type of AI, which is inspired by human brain and enables computers to learn from large amounts of data. Besides, Table 5.1 further illustrates the main differences between traditional machine learning and deep learning. With machine learning, the relevant features of an image are manually extracted. With deep learning, the raw images are directly fed into a deep neural network that learns the features automatically. What's more, deep learning often requires hundreds of thousands or millions of images to reach best results. It's also computationally intensive and requires a high-performance computer.

As a subset of machine learning, deep learning has taken off since 2012, which provides advanced analytics and offers great potentials in the era of big data. It has achieved huge success in a variety of domains, not only in classical computer vision tasks, such as

Table 5.1: Comparison between traditional machine learning and deep learning

	traditional machine learning	deep learning
data requirement	performs well with small dataset	needs large dataset
algorithm structure	simple mathematical model	end-to-end high hierarchical nonlinear multi-layer model
feature learning	identified manually	learned automatically
hardware requirement	no need for a powerful hardware	high configurations
model training scheme	each module trained step-by-step	parameters trained jointly
model training time	quick to train a model	computationally intensive
accuracy	accuracy plateaus	excellent performance

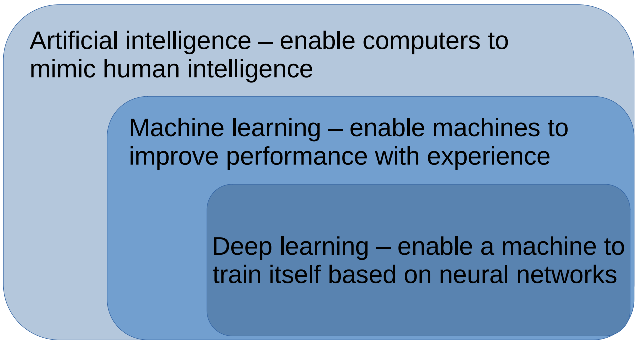
Artificial intelligence – enable computers to mimic human intelligence
Machine learning – enable machines to improve performance with experience
Deep learning – enable a machine to train itself based on neural networks

Figure 5.1: Artificial intelligence, machine learning and deep learning

target detection, visual recognition and robotics, but also in many other practical applications [Nogueira et al., 2017] [Wang et al., 2018] and have attracted great interest in both academia and industrial communities [Zhang, 2018].

Deep models can be referred to as neural networks with deep structures. The concept of neural networks is not something new and can date back to 1940s [Pitts et al., 1947]. The original intention was to simulate the human brain system to solve general learning problems in a principled way. Since then this decades-old scientific discovery started its long journey to innovate the entire academic community.

There are some remarkable technical breakthroughs and significant advances in the design of network structures and training strategies, including but not limited to: [Rumelhart et al., 1985] proposed back-propagation algorithm in 1980s; [Jarrett et al., 2009] came up with a new type of non-linearity, namely the widely used Rectified Linear Unit (ReLU); [Glorot et al., 2010] presented a new weight initialization; [Ciresan et al., 2011] proposed the Max-pooling instead of average sub-sampling; with dropout in [Hinton et al., 2012] and data augmentation, the overfitting problem in training could be relieved; with batch normalization (BN) [Ioffe et al., 2015], the training of very deep neural networks became quite efficient.

It is all these continuous efforts, together with the emergence of large scale annotated training data, such as ImageNet [Deng et al., 2009] and the fast development of high performance parallel computing systems, such as Graphics Processing Units (GPUs) that prosper deep learning nowadays [Zhao et al., 2019].

5.2.2/ Convolutional neural networks

ConvNets are the most representative models of deep learning and are designed to process data that come in the form of multiple arrays, such as a colour image composed of three 2D arrays containing pixel intensities in the three colour channels [LeCun et al., 2015]. ConvNets-based network architectures now dominate the field of computer vision.

Although ConvNets were invented in the 1980s [Fukushima, 1980], the breakthrough was made on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) by Krizhevsky et al. [Krizhevsky et al., 2012], who applied the deep convolutional networks to a dataset of 1.2 million high-resolution images and achieved excellent performance, almost halving the error rates of the best competing approaches. This network is called AlexNet, named after Alex Krizhevsky, the first author of this breakthrough. The results of AlexNet, for the first time, show that a large convolutional neural network is capable of achieving recordbreaking accuracy on a highly challenging dataset using purely supervised learning.

Since AlexNet, even larger and deeper networks have been proposed. These models include the VGG networks [Simonyan et al., 2014], which makes use of a number of repeating blocks of elements; the network in network (NiN) [Lin et al., 2013], which convolves whole neural networks patch-wise over inputs; the GoogLeNet [Szegedy et al., 2015] and its higher versions [Ioffe et al., 2015] [Szegedy et al., 2016] and [Szegedy et al., 2017], which make use of networks with parallel concatenations; residual networks (ResNet) [He et al., 2016] which are currently the most popular go-to architecture today, and densely connected networks (DenseNet) [Huang et al., 2017], which are expensive to compute but have set some recent benchmarks [Zhang et al., 2019]. These architectures are now the base models upon which an enormous amount of research and projects are built and have been applied with great success in the computer vision community.

5.2.3/ VGG networks

The VGG networks are ConvNets designed by Simonyan and Zisserman [Simonyan et al., 2014]. They were originally proposed for the ImageNet Large Scale Visual Recognition Competition (ILSVRC-2014) by the Visual Geometry Group,

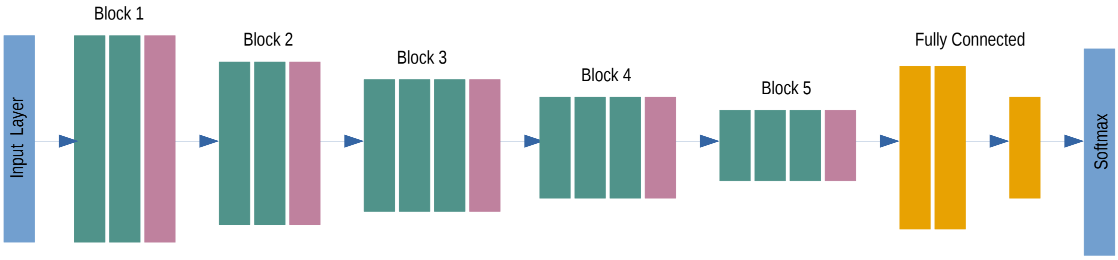

Figure 5.2: Architecture of VGG16

where the name of this set of models comes from.

There are 6 kinds of ConvNet configurations included in the VGG networks. All configurations follow the generic design in architecture and differ only in the depth. As a typical and popular ConvNet architecture, VGG16 has been fine-tuned for many other tasks, such as object detection [Ren et al., 2015], semantic segmentation [Lim et al., 2017] [Dai et al., 2016] and so on.

Fig. 5.2 shows the VGG16 architecture. Like earlier AlexNet [Krizhevsky et al., 2012], VGG16 can be partitioned into two parts: the first one consisting mostly of convolutional and pooling layers and the second one consisting of fully connected layers. The idea of block is used in the first part. The VGG16 has five convolutional blocks, among which the first two have two convolutional layers each and the latter three contain three convolutional layers each.

Table 5.2 illustrates the details of the configuration of VGG16, including the type, the kernel size, the number of channels and the output shape for each hidden layer. Specifically speaking, one VGG block consists of a sequence of convolutional layers, performing 3x3 convolutions with stride 1 and pad 1, followed by a maxpooling layer for spatial downsampling, which performs 2x2 maxpooling with stride 2. Thus the resolution is halved after each block.

We can also see in Table 5.2, the first block has 64 output channels and each subsequent block doubles the number of output channels, until that number reaches 512 [Zhang et al., 2019]. The input size of pictures for VGG16 is 224 $\times$ 224 $\times$ 3 and 3 stands for the three spectral channels of RGB. The output size for each layer is also listed for better understanding the structure of this model.

Table 5.2: VGG16 network configuration

Block	Layer type	kernel size	Number of channels	Output shape
	input	-	-	224 × 224 × 3
1	convolution	3×3	64	224×224×64
	convolution	3×3	64	224×224×64
	maxpooling	2×2		112×112×64
2	convolution	3×3	128	112×112×128
	convolution	3×3	64	112×112×128
	maxpooling	2×2		56×56×128
3	convolution	3×3	256	56×56×256
	convolution	3×3	256	56×56×256
	convolution	3×3	256	56×56×256
	maxpooling	2×2		28×28×256
4	convolution	3×3	512	28×28×512
	convolution	3×3	512	28×28×512
	convolution	3×3	512	28×28×512
	maxpooling	2×2		14×14×512
5	convolution	3×3	512	14×14×512
	convolution	3×3	512	14×14×512
	convolution	3×3	512	14×14×512
	maxpooling	2×2		7×7×512
	FullyConnected		4096	1×1×4096
	FullyConnected		4096	1×1×4096
	FullyConnected		1000	1×1×1000

5.2.4/ ENCODER-DECODER ARCHITECTURE

The encoder-decoder architecture is a neural network design pattern and it is applied in many image classification network, such as AlexNet [Krizhevsky et al., 2012], VGG network [Simonyan et al., 2014], GoogLeNet [Szegedy et al., 2015] and Resnet

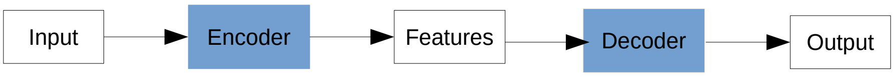

Figure 5.3: Encoder-decoder architecture

[He et al., 2016]. In this architecture, the network is partitioned into two parts, the encoder and the decoder.

As Fig. 5.3 shows, the input, such as an image patch, is fed to the encoder which produces features. The decoder module then converts the features into a prediction results for a specific purpose. Meantime, the error is measured. The encoder and decoder are parameterized functions that are trained to minimize the average error [Ranzato et al., 2007].

The encoder can perform data compression especially in dealing input of high dimensionality by mapping input to a hidden layer [Rumelhart et al., 1986]. The decoder can reconstruct the approximation of input.

An encoder is a network (FC, CNN, RNN (Recurrent neural network), etc) that takes the input, and outputs a feature map/vector/tensor. This feature vector holds the information, the features, that represents the input. The decoder is again a network (usually the same network structure as encoder but in opposite orientation) that takes the feature vector from the encoder, and gives the best closest match to the actual input or intended output.

5.3/ PROPOSED MULTISPECTRAL BACKGROUND SUBTRACTION WITH DEEP LEARNING

In this section, we will present the deep learning approaches we proposed for background subtraction with multispectral images. The proposed algorithms are based on FgSegNet_v2 [Lim et al., 2019], which will be first introduced as it serves as inspiration for multispectral case. Then, the proposed architectures are introduced.

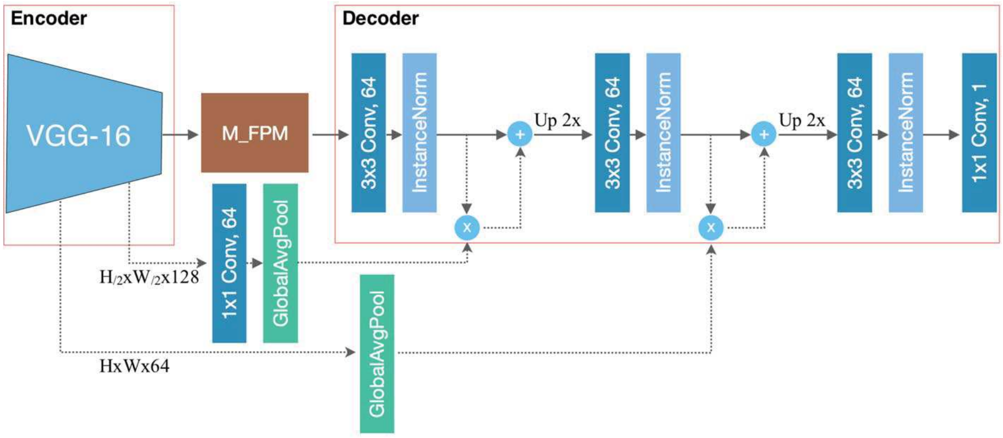

Figure 5.4: FgSegNet_v2 network architecture [Lim et al., 2019]

5.3.1/ FgSegNet_v2

The ranking first algorithm on the large-scale change detection dataset CD-net [Goyette et al., 2012] [Wang et al., 2014] is the method called FgSegNet_v2 [Lim et al., 2019]. FgSegNet_v2 utilizes the concept of the aforementioned encoder-decoder structure and takes advantages of the pretrained VGG16 and the transposed convolutional neural network (TCNN), as shown in Fig.5.4. It is followed by the former version FgSegNet [Lim et al., 2018], which includes two approaches, namely FgSegNet_M and FgSegNet_S.

As reviewed in the second chapter, FgSegNet_M has proposed a triplet of encoders to extract multi-scale features and then used transposed convolution in the decoder to learn a mapping from feature space to image space. FgSegNet_S has adopted a single-input encoder but achieved comparable performance by applying an Feature Pooling Module (FPM) mechanism, which adopts several parallel dilated convolutions with different dilation rates to extract multi-scale features.

Table 5.3 shows the configuration for the encoder part of the FgSegNet_v2. It utilizes the first four blocks of VGG16 with some modifications, i.e. removing the maxpooling layer of the third and forth blocks and inserting a dropout layer after every convolutional layer in the forth block. During the training process, the first three blocks are frozen and only the modified forth block is reinitialized and fine-tuned.

Table 5.3: FgSegNet_v2 encoder configuration

Block	Layer type	kernel size	Number of channels	Output shape
	input	-	-	W × H × 3
1	convolution	3×3	64	W×H×64
	convolution	3×3	64	W×H×64
	maxpooling	2×2		W/2×H/2×64
2	convolution	3×3	128	W/2×H/2×128
	convolution	3×3	64	W/2×112×H/2
	maxpooling	2×2		W/4×H/4×128
3	convolution	3×3	256	W/4×H/4×256
	convolution	3×3	256	W/4×H/4×256
	convolution	3×3	256	W/4×H/4×256
4	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512
	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512
	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512

As a higher version, FgSegNet_v2 is more robust against camera motion, as the original FPM module in FgSegNet_S has been improved to capture multi-scale with wider receptive fields. As shown in Fig. 5.5, it has adopted cascade dilated convolutional layers instead of parallel strategy. Besides, Instance Normalization (IN) [Ulyanov et al., 2016] has been adopted to replace Batch Normalization (BN), as it produces slightly high accuracy by observation.

Furthermore, a novel decoder network with Global Average Pooling (GAP) module has been designed to improve the performance with small cost in computation. As illustrated in Fig. 5.4, the low level feature coefficients vectors in the encoder network are extracted and used to guide the high level features in the decoder part.

In our work, the modified FPM and the decoder with GAP are inherited from FgSegNet_v2.

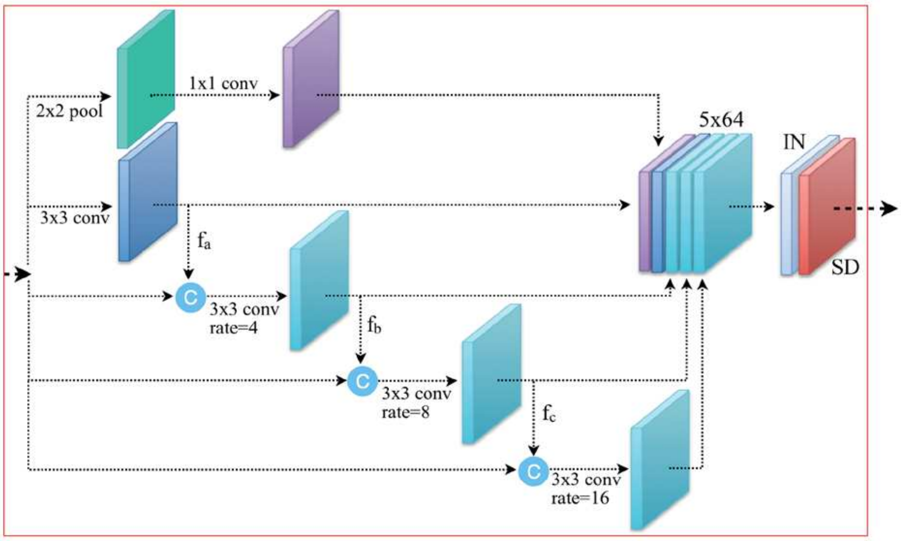

Figure 5.5: Modified FPM structure [Lim et al., 2019]

Since the FPM module can be regarded as preprocessing for decoder network, hereafter, they are merged to the decoder implementation for concise expression.

5.3.2/ Multispectral three-channel based FgSegNet_v2

The original VGG16 deep model is pretrained using conventional RGB images with three channels. Accordingly, the third dimension of the filter for the first convolutional layer is also three. In order to investigate the possible improvement of multispectral images against RGB based on FgSegNet_v2, we first extract three channels out of seven in the multispectral FluxData FD-1665 dataset [Benezeth et al., 2014a], which was introduced in detail earlier in the third chapter. Then, the trials with these extracted three-channel images are conducted using FgSegNet_v2 for each scene.

Fig. 5.6 illustrates the working flow of this proposed mechanism. The multispectral images are first processed through the module of three-channel extraction to produce three-channel based images, which are then fed to the encoder adapted from VGG16 and decoder network, following the same manner of FgSegNet_v2 explained above.

The number of combinations composed of three channels among seven is $C_7^3 = 35$. Thus, for each scene, we have built thirty-five independent background models. After each deep model is trained with the training subset of a certain combination of channels, the testing subset of the same channels is used to evaluate the performance of this background

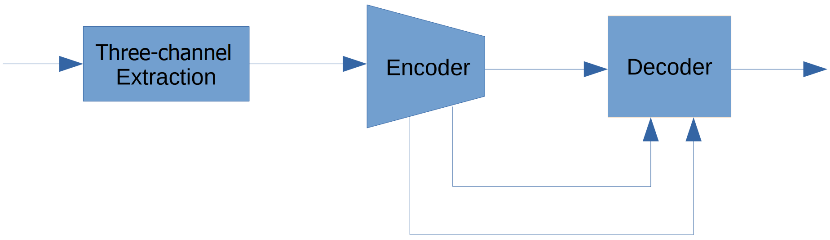

Figure 5.6: Multispectral three-channel based FgSegNet_v2

model.

5.3.3/ Proposed convolutional encoder for all multispectral channels

As we mentioned in the last section, the third dimension of first filter of the original VGG16 is three, as it has been pretrained with three-channel based RGB images. If we simultaneously feed the multispectral images with more than three channels to the deep model of FgSegNet_v2, which adopts the first four blocks of VGG16 as the encoder, only the first three channels are really processed, while others are ignored. Therefore, we can not utilize the FgSegNet_v2 model directly for multispectral images with more than three channels.

In order to further explore the benefits of multispectral images, we have proposed a new convolutional encoder for extracting the relevant deep features from the given multispectral-groundtruth pair with images consisting any arbitrary number of channels. Table 5.4 illustrates the configuration of the proposed encoder for multispectral images, where the number of input is seven, as the FluxData FD-1665 dataset we use contains seven channels. Including the trainable Block 4 from the FgSegNet_v2, the proposed encoder consists of two more convolutional layers, both of which are followed by a max-pooling layer in order to match the size of the inputs for the decoder.

Fig.5.7 shows the architecture of background subtraction for multispectral images with the proposed convolutional encoder. Following the idea of FgSegNet_v2 that feeds different levels of features to the decoder, the low level feature coefficients vectors after each convolutional layer in the encoder network are extracted and used to guide the high level features in the decoder part.

As it is known, the parameters in the VGG16 network have been pretrained with a vast

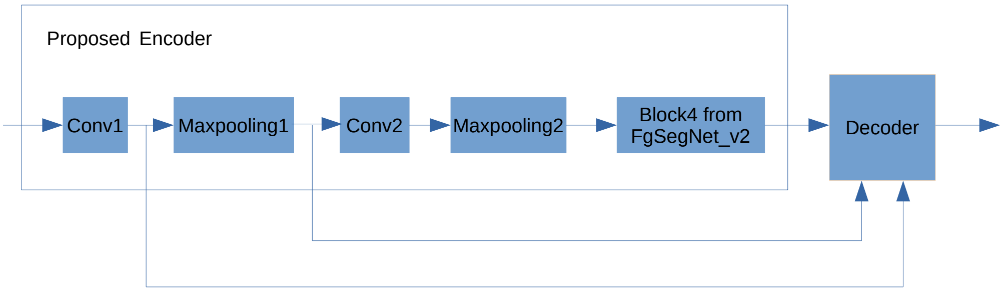

Figure 5.7: Proposed multispectral background subtraction architecture

number of RGB images, which are not available for multispectral case. We can see in the proposed convolutional encoder illustrated in Fig.5.7, there is no pretrained deep model adopted. Thus, all the parameters in this proposed mechanism are trainable.

What is needed to be stated is that, the filter channel of the first convolutional layer is not fixed in the proposed encoder, and will only be assigned when the images for training arrive. That is to say, for multispectral case, the size of the filter will be 3×3×7 (7 stands for the number of available channels), while the size is set to 3×3×3 if RGB images are fed. This property allows us to conduct a fair comparison for performance in background subtraction between multispectral images based model and RGB images based model.

Table 5.4: Proposed multispectral encoder configuration

Block	Layer type	Kernal size	Number of channels	Output shape
	input	-	-	W×H×7
1	convolution	3×3	64	W×H×64
	maxpooling	2×2		W/2×H/2×64
2	convolution	3×3	128	W/2×H/2×128
	maxpooling	2×2		W/4×H/4×128
3 (Block 4 from FgSegNet_v2)	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512
	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512
	convolution	3×3	512	W/4×H/4×512
	dropout			W/4×H/4×512

5.4/ Experimental results

5.4.1/ Multispectral three-channel based results

The first experiments are conducted for thirty-five three-channel based combinations on the five different videos in the FluxData FD-1665 dataset. Given the groundtruth data, the performance of foreground detection is evaluated at a pixel level by F-measure, which is a harmonic value of precision and recall and is widely used in the domain of background subtraction. What is needed to be noted is that only the pixels with the label of moving or static are taken into consideration in the evaluation process.

Table 5.5 shows the three-channel based F-measures on five videos, together with the RGB results in the last row, acting as a reference. The highest accuracy for each scene is in bold. It is clear that the best three-channel combination on one video is not always the same on the others, which is reasonable, since the spectral property and the ability to represent scenes vary with different channel combinations. This is quite obvious for Video 1, which is the only indoor scene, where the largest F-measure is 0.9703 (with the combination of Channels 4, 5 and 7), while the smallest is 0.9425 (with Channels 2, 4 and 5). As it is known, Channel 7 corresponds to the NIR spectrum and it offers good complementary information in this scene.

In order to be more clear in comparing the performance of RGB and other three-channel combinations extracted from the original seven-channel multispectral images, the largest F-measures for each scene in Table 5.5 are selected, together with the RGB for five videos and listed in Table 5.6. As Table 5.6 indicates, there are always three-channel based combinations that have better spectral discrimination and outperform the RGB images with a certain degree, especially for the Video 5, where there exists objects with shadows. With the combination of Channels 1, 4 and 7, a higher F-measure of 0.9840 is obtained compared to 0.9714 for RGB images. This is quite interesting, because in this part we have followed the same mechanism of FgSegNet_v2 for the fine-tuning strategy, that is, only the parameters in the forth block are reinitialized and fine-tuned by the new data, while the first three blocks are frozen and the parameters are adopted directly from VGG16 deep model. As VGG16 is pretrained with conventional RGB images, one can expect better segmentation results with RGB images than the extracted three-channel based multispectral images. We think that the combination of channels (1, 4 and 7) per-

Table 5.5: Three-channel F-measures on five videos

Combination	channels	Video 1	Video 2	Video 3	Video 4	Video 5
1	123	0.9655	0.9983	0.9787	0.9666	0.9833
2	124	0.9618	0.9984	0.9804	0.9676	0.9799
3	125	0.9657	0.9983	0.9795	0.9688	0.9817
4	126	0.9667	0.9983	0.9785	0.9673	0.9830
5	127	0.9645	0.9983	0.9781	0.9695	0.9832
6	134	0.9675	0.9983	0.9790	0.9684	0.9828
7	135	0.9657	0.9983	0.9789	0.9693	0.9815
8	136	0.9675	0.9983	0.9781	0.9623	0.9832
9	137	0.9656	0.9983	0.9788	0.9669	0.9829
10	145	0.9629	0.9984	0.9780	0.9686	0.9828
11	146	0.9661	0.9982	0.9800	0.9659	0.9828
12	147	0.9653	0.9983	0.9783	0.9668	**0.9840**
13	156	0.9658	0.9983	0.9777	0.9645	0.9838
14	157	0.9638	0.9984	0.9790	0.9686	0.9818
15	167	0.9653	0.9984	0.9790	0.9682	0.9829
16	234	0.9480	0.9980	0.9818	0.9726	0.9765
17	235	0.9449	0.9981	0.9819	0.9706	0.9772
18	236	0.9445	0.9980	0.9809	0.9735	0.9795
19	237	0.9455	0.9982	**0.9835**	0.9729	0.9803
20	245	0.9425	0.9981	0.9820	0.9732	0.9800
21	246	0.9467	0.9982	0.9813	0.9715	0.9802
22	247	0.9451	0.9982	0.9817	0.9711	0.9818
23	256	0.9473	0.9979	0.9826	0.9709	0.9798
24	257	0.9455	0.9982	0.9813	0.9715	0.9753
25	267	0.9503	0.9982	0.9818	0.9757	0.9798
26	345	0.9584	**0.9986**	0.9829	0.9730	0.9759
27	346	0.9613	0.9986	0.9819	0.9731	0.9758
28	347	0.9611	0.9986	0.9833	0.9707	0.9787
29	356	0.9572	0.9985	0.9822	0.9730	0.9780
30	357	0.9624	0.9986	0.9819	0.9734	0.9758
31	367	0.9601	0.9986	0.9809	0.9735	0.9762
32	456	0.9669	0.9985	0.9797	0.9770	0.9786
33	457	**0.9703**	0.9984	0.9788	0.9774	0.9801
34	467	0.9692	0.9986	0.9797	**0.9788**	0.9785
35	567	0.9690	0.9986	0.9801	0.9754	0.9762
	RGB	0.9683	0.9982	0.9808	0.9715	0.9714

formed better than RGB thanks, once again, to the NIR complementary spectral property of Channel 7.

Table 5.6: Best three-channel F-measures on five videos

Video	Best MUL	RGB
1	0.9703	0.9683
2	0.9986	0.9982
3	0.9835	0.9808
4	0.9788	0.9715
5	0.9840	0.9714

5.4.2/ Proposed convolutional encoder results

Followingly, the experiments with the new convolutional encoder are conducted for both multispectral images and RGB images. As there is no pretrained deep model, all the parameters are trained from scratch. The only difference for multispectral images and the RGB case lies at the filter size of the first convolutional layer, namely, 3×3×7 for the former and 3×3×3 for the latter.

Table 5.7 illustrates the F-measures obtained with multispectral images based model and RGB images based model on the five videos in the FluxData FD-1665 dataset. As it is shown, multispectral images based model generally performs better than the RGB based one.

Specifically speaking, the proposed convolutional approach with multispectral images performs quite well on Video 2, where a very high F-measure can already be obtained by RGB images and there is no obvious accuracy difference between the two kinds of methods. However, for other videos, we could get considerable higher accuracy with multispectral images based model, especially for Video 3, where more than three percentages improvement is obtained via the utilization of multispectral images, which is very impressive. The results show that multispectral images based model could be a promising alternative to conventional RGB images based one in background subtraction.

Besides, Fig.5.8 shows some visual results. The top two rows are multispectral and RGB

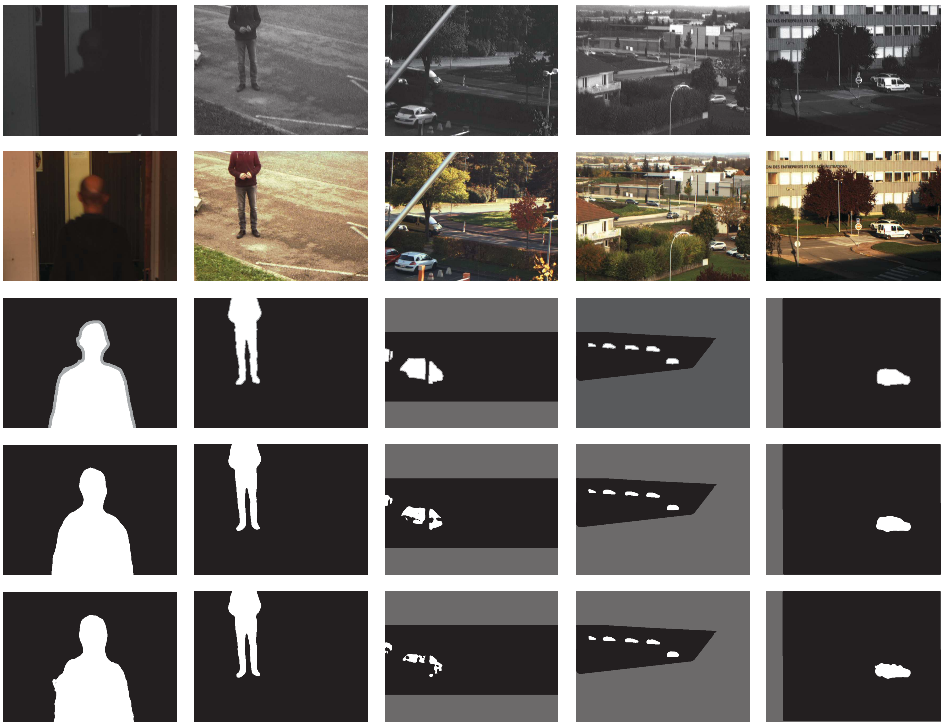

Figure 5.8: Background subtraction results on the five videos

images, respectively. The third one is the corresponding groundtruth images. The forth and fifth rows are the background subtraction masks obtained by multispectral and RGB images, respectively. Since the pixels out of ROI or unknown are not considered in the training process, the detection results of these corresponding areas are random. In order to make the visual results tidy and easy to read, we assign the same gray value for these pixels in the mask images as they are originally in the groundtruth images.

Table 5.7: F-measures with new convolutional encoder on five videos

Video	MUL	RGB
1	0.9786	0.9540
2	0.9982	0.9942
3	0.9430	0.9109
4	0.9619	0.9558
5	0.9603	0.9383

We further compare the multispectral results obtained by the proposed convolutional encoder with the classical approaches in [Benezeth et al., 2014a] [Sobral et al., 2015] [Silva et al., 2016] and [Silva et al., 2017] using the same dataset. They have been explained in the chapter of state of art. The F-measures of these different methods on the five videos of the FluxData FD-1665 dataset are collected and listed in Table 5.8.

Table 5.8: F-measures of different approaches on five videos

Video	MD	SA	SID	OSTD	OWOC -RS	Superpixel -OWAOC	Proposed
1	0.8105	0.9042	0.9022	0.9365	0.9008	0.9135	**0.9786**
2	0.8900	0.9562	0.9686	0.9517	0.8727	0.9591	**0.9982**
3	0.6889	0.8970	0.8958	0.9064	**0.9635**	0.9376	0.9430
4	0.8327	0.6733	0.6878	0.8929	0.8997	0.8827	**0.9619**
5	0.7724	0.7422	0.7574	0.9266	0.8400	0.8693	**0.9603**
mean	0.7989	0.8346	0.7427	0.9228	0.8953	0.9124	0.9684

MD = Mahalanobis Distance [Benezeth et al., 2014a]

SA = Spectral Angle [Benezeth et al., 2014a]

SID = Spectral Information Divergence [Benezeth et al., 2014a]

OSTD = Online Stochastic Tensor Decomposition [Sobral et al., 2015]

OWOC-RS = Online Weighted One-Class Random Subspace [Silva et al., 2016]

Superpixel-OWAOC = Superpixel-based Online Wagging One-Class Ensemble [Silva et al., 2017]

The proposed convolutional approach outperforms the classical methods with a considerable gap on average, with a mean F-measure of 0.9684, which is nearly five percent higher than the ranking first classical algorithm OSTD in the fifth column. This shows the impressive ability of deep features learned with the proposed ConvNet in the task of background subtraction.

However, we need to be aware that the conventional feature selection methods can also obtain great accuracy with a carefully designed mechanism. As we can see, the OWOC-RS proposed by [Silva et al., 2016] has achieved the highest F-measure for the Video 3, where it exists dynamic backgrounds, shadows and occlusion. Since it is a challenging scene, we think the proposed multispectral images based model needs more training images to better learn and represent the background. This supposal coincides with the fact that deep learning based methods always rely on big amount of data to achieve better performance. That is also part of reason why there are still researchers devoting themselves in classical methods or other new approaches, while deep leaning is changing the domain of computer vision nowadays.

5.4.3/ Prediction time

Computational complexity is also observed during our experiments. The deep learning algorithm was implemented using Keras framework with TensorFlow backend, on a single NVIDIA GeForce GTX1080Ti GPU with a memory of 11GB and an Intel Core i7K-8700K CPU (6 cores and 12 threads) with a RAM of 32 GB. The time for prediction with our proposed approach is 0.134s for each multispectral image with a resolution of 492×658.

To be best of our knowledge, our work is the first attempt to utilize multispectral images [Benezeth et al., 2014a] for background subtraction task with deep learning based method. Although the authors of [Sobral et al., 2015] and [Silva et al., 2017] have provided the computational cost of their classical algorithms with Matlab installed in laptop, we can not conduct relevant comparison of prediction time with different platforms and languages adopted.

5.5/ Conclusion

In this work, we followed the trend of deep learning and apply its concepts to background subtraction using multispectral images. Based on the ranking first algorithm FgSegNet_v2 on the large-scale change detection dataset CDnet, we first extracted three channels out of the seven channels on five videos in the FluxData FD-1665 dataset to match the number of input channels for the pretrained VGG16 deep model with RGB images. The results were interesting, as some combinations of three-channel based multispectral images performed better than the conventional RGB images.

In order to further explore the benefits of multispectral images, we have proposed a new convolutional encoder for extracting the relevant deep features from the given multispectral-groundtruth pair with images consisting of any arbitrary number of channels. The modified VGG16 has been pretrained with large-scale RGB ImageNet dataset, which are not available for multispectral case. Thus, there is no pretrained deep model adopted in the proposed encoder and all the parameters are trainable. The accuracy of the proposed convolutional approach is quite appealing when compared with other approaches using the same multispectral dataset. As the filter channel of the first convolutional layer is not fixed and can also be used for RGB images, the results show that the multispectral images based model outperforms the RGB images based one.

Our work can be seen as an attempt to investigate the potential advantages of using multispectral information via deep features learned with ConvNets for the background subtraction task. Some future research directions using multispectral images in background subtraction are as follows. Larger multispectral datasets including other challenges, like night videos, should be investigated for exhaustive evaluation of multispectral information for background subtraction. Besides, as the filter channel of the first convolutional layer is arbitrary in the proposed encoder, the performance of foreground-background segmentation with other sizes (4, 5, 6) of multispectral images combination can be further studied. Another interesting direction lies in the combination of the proposed encoder and the pretrained deep models like VGG16 to take use of their powerful generic feature encoding properties. Last but not least, other data sources like depth, could also be explored and exploited for future research as they can offer important complementary information for background subtraction task.

III

Conclusion and Future Works

6

CONCLUSIONS AND FUTURE WORKS

6.1/ CONCLUSIONS

Background subtraction is a crucial task in many computer vision applications. In this thesis, we have proposed several background subtraction methods using multispectral image sequences. The following contributions have been established as part of this research effort.

First of all, the original RGB based Codebook algorithm has been adapted to multispectral case. In order to achieve this goal, two modifications have been conducted comparing with the original Codebook technique. Specifically speaking, the definition of brightness in RGB color space is extended to multispectral case, with an arbitrary number of spectral channels. Besides, we replace the color distortion with spectral distortion, as the term of color is no longer suitable in multispectral sequences. The experiments have been conducted on all the five scenes in the FluxData FD-1665 dataset, for both RGB images and multispectral three-channel based images, respectively. Same set of parameters is adopted for fair comparison. Experiments have shown that multispectral images may represent an alternative to conventional images in the background subtraction. These encouraging results open a door for future works for applying multispectral images in background subtraction.

Followingly, a self-adaptive mechanism is designed to select the optimal parameters based on the statistical information extracted from the data themselves, by calculating iteratively and recording additional statistical information vectors for each codeword. With the self-adaptive mechanism, brightness bounds and spectral distortion threshold are obtained automatically and able to adjust themselves with statistical properties of the input sequences. Comparing with the original Codebook algorithm, where the parameters are carefully selected by manual tuning based on the typical range provided by the pioneers of this technique, this step forward is necessary and significant for three-fold considerations.

1) Manual tuning for optimal parameters is time consuming and not robust, as the detection results of Codebook are heavily impacted by the parameters.

2) If the algorithm needs to be run for long periods of time, the parameters should not be static but could be automatically adapted to the environmental changes.

3) When using the multispectral sequences, the parameters also have to be adjusted with different number of channels used. This is a big issue for our research objective.

Furthermore, in the framework of multispectral self-adaptive Codebook model, improvements have been proposed in two aspects. The first one is to introduce a new feature descriptor called spectral information divergence in the matching process to evaluate the spectral distance between the new pixel vector and that in the tested codeword. It has been first employed to replace the spectral distortion in the previous multispectral self-adaptive Codebook model to be the matching criteria together with the brightness condition. To further utilize the spectral information, the three features mentioned here have been then adopted together. In both mechanisms, the strategy to acquire the self-adaptive threshold for spectral information divergence is the same with that for spectral distortion. According to the experimental results, the utilization of the spectral information divergence does great help for the indoor scene. The three-criteria based multispectral self-adaptive Codebook is the most promising choice across all the five scenes. This attempt opens a door for other possibilities to seek novel kind of feature representation in the construction of the Codebook background model.

The latter aspect includes three techniques to build the background model, namely, box-based Codebook, dynamic Codebook and fusion strategy, each of which processes the multispectral channels independently and only the intensity value for each channel is used to calculate the spectral similarity between the new frame pixel and reference one in current codeword. From average F-measures for the whole dataset, the approaches of codeword with these new non-cylindrical structures can produce comparable results to those methods that utilize all the three criteria with aforementioned cylindrical structure. Besides, for all the algorithms listed here, it is quite easy to adapt to multispectral images with any number of channels for background subtraction.

Last but not least, we have followed the trend of deep learning and applied its concepts to background subtraction using multispectral images. Based on the ranking first algorithm FgSegNet_v2 for RGB images, we first extracted three channels out of the seven channels to match the number of input channels for the pretrained VGG16 deep model with RGB images. Besides, in order to use more channels of the multispectral images, a new convolutional encoder has been proposed for extracting the relevant deep features from the given multispectral-groundtruth pair with images consisting of any arbitrary number of channels. The proposed convolutional approach outperforms the classical methods with a considerable gap on average, which shows the impressive ability of deep features learned with the proposed ConvNet in the task of background subtraction.

6.2/ FUTURE WORKS

Based on the algorithms proposed and results obtained in this thesis, we believe the following perspectives could help further develop the research conducted in this thesis.

1) In the framework of multispectral self-adaptive Codebook model, we have investigated spectral information divergence to measure the spectral distance. There is another feature description named Local Binary Patterns (LBP) 1994 [Ojala et al., 1994] [Ojala et al., 1996], which can also be a promising choice. As a very powerful feature for texture classification, LBP has been proven to be useful in moving object detection [Wang et al., 2009] [Heikkila et al., 2006]. The accuracy could improve with a well designed mechanism to integrate LBP with the spectral features adopted in our work.

2) In order to utilise all the seven multispectral channels in the deep learning framework, we have proposed an new encoder with an arbitrary number of input for first filter. We could also further study the performance of foreground-background segmentation with other sizes (four, five or six) of multispectral images combination.

Another promising direction lies in the combination of the proposed encoder and the pretrained deep models like VGG16 to take use of their powerful generic feature encoding properties.

Some other interesting future research directions are as follows.

1) We sincerely appreciate the creators of FluxData FD-1665 dataset, which is the only public real multispectral image dataset for background subtraction, to the best of our knowledge. However, it only consists of five videos and all are in daytime with good illumination condition. Thus, larger multispectral datasets including other challenges, as illustrated in Chapter 2, especially night videos, should be investigated for exhaustive evaluation of multispectral based algorithms for background subtraction task.

2) The leading background subtraction algorithms work with RGB images. It would be interesting if we introduce more diverse sources of information for this task. The multispectral images adopted in this thesis is one choice. Nowadays, some researchers are also working on exploiting and integrating depth data, as they can offer important complementary information, to resolve the challenges of background subtraction like [Camplani et al., 2014] [Camplani et al., 2017] [Moyà-Alcover et al., 2017] [Fernandez-Sanchez et al., 2013] with attractive and promising performance.

3) Most of the background subtraction approaches available are specifically designed for static cameras. As we know, a typical procedure for the traditional background subtraction algorithms first builds a statistical background model and then extracts moving objects by detecting foreground regions, which do not share similar characteristics with the static background. The model would probably be invalid when the camera moves. It would be also very interesting if the background subtraction methods illustrated in the state of arts can be modified and extended to the case of moving cameras, like the efforts done in [Kim et al., 2013] [Viswanath et al., 2015] [Gong et al., 2017].

4) As we can see on CDnet website [Wang et al., 2014], the ranking first approaches using the large-scale CDnet dataset for background subtraction, are mainly based on supervised learning with convolutional networks. Although they have output really impressive performance, unsupervised and semi-supervised methods [Saatci et al., 2017] are still needed to be investigated and explored to use of as little labeled data as possible, since they are not always available or consuming to obtain [Sultana et al., 2018].

Bibliography

[Achom Nishalakshmi et al., 2018] Achom Nishalakshmi, D., et Gaurav (2018). **Reviews on augmented reality: Google lens**. *International Journal of Computer Trends and Technology*.

[Agarwal et al., 2016] Agarwal, A., Gupta, S., et Singh, D. K. (2016). **Review of optical flow technique for moving object detection**. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 409–413. IEEE.

[Aloimonos, 1990] Aloimonos, J. (1990). **Purposive and qualitative active vision**. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 1, pages 346–360. IEEE.

[Aslani et al., 2013] Aslani, S., et Mahdavi-Nasab, H. (2013). **Optical flow based moving object detection and tracking for traffic surveillance**. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 7(9):1252–1256.

[Aung et al., 2017] Aung, S. S., et Kyu, Z. M. (2017). **Modified codebook algorithm with kalman filter for foreground segmentation in video sequences**. In *2017 International Conference on Signal Processing and Communication (ICSPC)*, pages 332–336. IEEE.

[Babaee et al., 2018] Babaee, M., Dinh, D. T., et Rigoll, G. (2018). **A deep convolutional neural network for video sequence background subtraction**. *Pattern Recognition*, 76:635–649.

[Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., et Cipolla, R. (2017). **Segnet: A deep convolutional encoder-decoder architecture for image segmentation**. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

[Bakkay et al., 2018] Bakkay, M. C., Rashwan, H. A., Salmane, H., Khoudour, L., Puigtt, D., et Ruichek, Y. (2018). **Bscgan: deep background subtraction with conditional**

generative adversarial networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 4018–4022. IEEE.

[Balaban, 2015] Balaban, S. (2015). **Deep learning and face recognition: The state of the art**. In *Biometric and Surveillance Technology for Human and Activity Identification XII*, volume 9457, page 94570B. International Society for Optics and Photonics.

[Barnich et al., 2010] Barnich, O., et Van Droogenbroeck, M. (2010). **Vibe: A universal background subtraction algorithm for video sequences**. *IEEE Transactions on Image processing*, 20(6):1709–1724.

[Benezeth et al., 2014a] Benezeth, Y., Sidibé, D., et Thomas, J.-B. (2014a). **Background subtraction with multispectral video sequences**. In *IEEE International Conference on Robotics and Automation workshop on Non-classical Cameras, Camera Networks and Omnidirectional Vision (OMNIVIS)*, pages 6–p.

[Benezeth et al., 2014b] Benezeth, Y., Sidibé, D., et Thomas, J.-B. (2014b). **Background subtraction with multispectral video sequences**. In *IEEE International Conference on Robotics and Automation workshop on Non-classical Cameras, Camera Networks and Omnidirectional Vision (OMNIVIS)*, pages 1–6.

[Bianco et al., 2017] Bianco, S., Ciocca, G., et Schettini, R. (2017). **Combination of video change detection algorithms by genetic programming**. *IEEE Transactions on Evolutionary Computation*, 21(6):914–928.

[Bolón-Canedo et al., 2014] Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M., et Herrera, F. (2014). **A review of microarray datasets and applied feature selection methods**. *Information Sciences*, 282:111–135.

[Bouchech, 2015] Bouchech, H. (2015). **Selection of optimal narrowband multispectral images for face recognition**. PhD thesis.

[Boulmerka et al., 2017] Boulmerka, A., et Allili, M. S. (2017). **Foreground segmentation in videos combining general gaussian mixture modeling and spatial information**. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1330–1345.

[Bourlai et al., 2012] Bourlai, T., et Cukic, B. (2012). **Multi-spectral face recognition: Identification of people in difficult environments**. In *2012 IEEE International Conference on Intelligence and Security Informatics*, pages 196–201. IEEE.

[Bouwmans, 2014] Bouwmans, T. (2014). **Traditional and recent approaches in background modeling for foreground detection: An overview**. *Computer Science Review*, 11-12:31 – 66.

[Bouwmans et al., 2019a] Bouwmans, T., et Garcia-Garcia, B. (2019a). **Background subtraction in real applications: Challenges, current models and future directions**. *arXiv preprint arXiv:1901.03577*.

[Bouwmans et al., 2019b] Bouwmans, T., Javed, S., Sultana, M., et Jung, S. K. (2019b). **Deep neural network concepts for background subtraction: A systematic review and comparative evaluation**. *Neural Networks*.

[Bouwmans et al., 2017a] Bouwmans, T., Maddalena, L., et Petrosino, A. (2017a). **Scene background initialization: A taxonomy**. *Pattern Recognition Letters*, 96:3–11.

[Bouwmans et al., 2017b] Bouwmans, T., Sobral, A., Javed, S., Jung, S. K., et Zahzah, E.-H. (2017b). **Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset**. *Computer Science Review*, 23:1–71.

[Braham et al., 2017] Braham, M., Piérard, S., et Van Droogenbroeck, M. (2017). **Semantic background subtraction**. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4552–4556. IEEE.

[Braham et al., 2016] Braham, M., et Van Droogenbroeck, M. (2016). **Deep background subtraction with scene-specific convolutional neural networks**. In *2016 international conference on systems, signals and image processing (IWSSIP)*, pages 1–4. IEEE.

[Brutzer et al., 2011] Brutzer, S., Höferlin, B., et Heidemann, G. (2011). **Evaluation of background subtraction techniques for video surveillance**. In *CVPR 2011*, pages 1937–1944. IEEE.

[Burke et al., 2010] Burke, E. K., Hyde, M., Kendall, G., et Woodward, J. (2010). **A genetic programming hyper-heuristic approach for evolving 2-d strip packing heuristics**. *IEEE Transactions on Evolutionary Computation*, 14(6):942–958.

[Camplani et al., 2017] Camplani, M., Maddalena, L., Alcover, G. M., Petrosino, A., et Salgado, L. (2017). **A benchmarking framework for background subtraction in**

rgbd videos. In *International Conference on Image Analysis and Processing*, pages 219–229. Springer.

[Camplani et al., 2013] Camplani, M., Mantecon, T., et Salgado, L. (2013). **Depth-color fusion strategy for 3-d scene modeling with kinect**. *IEEE transactions on cybernetics*, 43(6):1560–1571.

[Camplani et al., 2014] Camplani, M., et Salgado, L. (2014). **Background foreground segmentation with rgb-d kinect data: An efficient combination of classifiers**. *Journal of Visual Communication and Image Representation*, 25(1):122–136.

[Chang, 2000] Chang, C.-I. (2000). **An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis**. *IEEE Transactions on information theory*, 46(5):1927–1932.

[Chen et al., 2017] Chen, L.-C., Papandreou, G., Schroff, F., et Adam, H. (2017). **Rethinking atrous convolution for semantic image segmentation**. *arXiv preprint arXiv:1706.05587*.

[Chen et al., 2018] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., et Adam, H. (2018). **Encoder-decoder with atrous separable convolution for semantic image segmentation**. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818.

[Ciresan et al., 2011] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., et Schmidhuber, J. (2011). **Flexible, high performance convolutional neural networks for image classification**. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[Dai et al., 2016] Dai, J., He, K., et Sun, J. (2016). **Instance-aware semantic segmentation via multi-task network cascades**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158.

[Dalal et al., 2005] Dalal, N., et Triggs, B. (2005). **Histograms of oriented gradients for human detection**. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

[Davenport et al., 2016] Davenport, M. A., et Romberg, J. (2016). **An overview of low-rank matrix recovery from incomplete observations**. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., et Fei-Fei, L. (2009). **Imagenet: A large-scale hierarchical image database**. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Diem et al., 2007] Diem, M., Lettner, M., et Sablatnig, R. (2007). **Registration of multi-spectral manuscript images.** In *VAST*, pages 133–140.

[Doshi et al., 2006] Doshi, A., et Trivedi, M. (2006). **" hybrid cone-cylinder" codebook model for foreground detection with shadow and highlight suppression**. In *2006 IEEE International Conference on Video and Signal Based Surveillance*, pages 19–19. IEEE.

[Duda et al., 2012] Duda, R. O., Hart, P. E., et Stork, D. G. (2012). **Pattern classification**. John Wiley & Sons.

[Easton et al., 2003] Easton, R. L., Knox, K. T., et Christens-Barry, W. A. (2003). **Multi-spectral imaging of the archimedes palimpsest**. In *32nd Applied Imagery Pattern Recognition Workshop, 2003. Proceedings.*, pages 111–116. IEEE.

[Elgammal et al., 2002] Elgammal, A., Duraiswami, R., Harwood, D., et Davis, L. S. (2002). **Background and foreground modeling using nonparametric kernel density estimation for visual surveillance**. *Proceedings of the IEEE*, 90(7):1151–1163.

[Elgammal et al., 2000] Elgammal, A., Harwood, D., et Davis, L. (2000). **Non-parametric model for background subtraction**. In *European conference on computer vision*, pages 751–767. Springer.

[Elguebaly et al., 2014] Elguebaly, T., et Bouguila, N. (2014). **Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection**. *Machine vision and applications*, 25(5):1145–1162.

[Feng et al., 2018] Feng, C.-H., Makino, Y., Oshita, S., et Martín, J. F. G. (2018). **Hyper-spectral imaging and multispectral imaging as the novel techniques for detecting defects in raw and processed meat products: Current state-of-the-art research advances**. *Food Control*, 84:165–176.

[Feng et al., 2013] Feng, J., Xu, H., et Yan, S. (2013). **Online robust pca via stochastic optimization**. In *Advances in Neural Information Processing Systems*, pages 404–412.

[Fernandez-Sanchez et al., 2013] Fernandez-Sanchez, E. J., Diaz, J., et Ros, E. (2013). **Background subtraction based on color and depth using active sensors**. *Sensors*, 13(7):8895–8915.

[Ferrato et al., 2013] Ferrato, L.-J., et Forsythe, K. W. (2013). **Comparing hyperspectral and multispectral imagery for land classification of the lower don river, toronto**. *Journal of Geography and Geology*, 5(1):92–107.

[Fridman et al., 2019] Fridman, L., Brown, D. E., Glazer, M., Angell, W., Dodd, S., Jenik, B., Terwilliger, J., Patsekin, A., Kindelsberger, J., Ding, L., et others (2019). **Mit advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation**. *IEEE Access*, 7:102021–102038.

[Fukushima, 1980] Fukushima, K. (1980). **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**. *Biological cybernetics*, 36(4):193–202.

[Girshick, 2015] Girshick, R. (2015). **Fast r-cnn**. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., et Malik, J. (2014). **Rich feature hierarchies for accurate object detection and semantic segmentation**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[Glorot et al., 2010] Glorot, X., et Bengio, Y. (2010). **Understanding the difficulty of training deep feedforward neural networks**. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[Goes et al., 2014] Goes, J., Zhang, T., Arora, R., et Lerman, G. (2014). **Robust stochastic principal component analysis**. In *Artificial Intelligence and Statistics*, pages 266–274.

[Goldberg et al., 2003] Goldberg, A., Stann, B., et Gupta, N. (2003). Multispectral, hyperspectral, and three-dimensional imaging research at the us army research laboratory. Technical Report, ARMY RESEARCH LAB ADELPHI MD.

[Gong et al., 2017] Gong, L., Yu, M., et Gordon, T. (2017). Online codebook modeling based background subtraction with a moving camera. In *2017 3rd International Conference on Frontiers of Signal Processing (ICFSP)*, pages 136–140. IEEE.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., et Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Goyette et al., 2012] Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., et Ishwar, P. (2012). Changedetection. net: A new change detection benchmark dataset. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 1–8. IEEE.

[Grimson et al., 1998] Grimson, W. E. L., Stauffer, C., Romano, R., et Lee, L. (1998). Using adaptive tracking to classify and monitor activities in a site. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 22–29. IEEE.

[Grolier et al., 1984] Grolier, M. J., Tibbitts, G. C., Ibrahim, M. M., et others (1984). A qualitative appraisal of the hydrology of the Yemen Arab Republic from Landsat images. US Government Printing Office.

[Guo et al., 2010] Guo, Z., Zhang, L., et Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, 19(6):1657–1663.

[Hagen et al., 2013] Hagen, N. A., et Kudenov, M. W. (2013). Review of snapshot spectral imaging technologies. *Optical Engineering*, 52(9):090901.

[Haritaoglu et al., 2000] Haritaoglu, I., Harwood, D., et Davis, L. S. (2000). W/sup 4: real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., et Girshick, R. (2017). **Mask r-cnn**. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[He et al., 2016] He, K., Zhang, X., Ren, S., et Sun, J. (2016). **Deep residual learning for image recognition**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Heikkila et al., 2006] Heikkila, M., et Pietikainen, M. (2006). **A texture-based method for modeling the background and detecting moving objects**. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662.

[Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., et Salakhutdinov, R. R. (2012). **Improving neural networks by preventing co-adaptation of feature detectors**. *arXiv preprint arXiv:1207.0580*.

[Ho et al., 2016] Ho, J., et Ermon, S. (2016). **Generative adversarial imitation learning**. In *Advances in neural information processing systems*, pages 4565–4573.

[Hofmann et al., 2012] Hofmann, M., Tiefenbacher, P., et Rigoll, G. (2012). **Background segmentation with feedback: The pixel-based adaptive segmenter**. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 38–43. IEEE.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., et Weinberger, K. Q. (2017). **Densely connected convolutional networks**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Huang et al., 2016] Huang, J., Jin, W., Zhao, D., Qin, N., et Li, Q. (2016). **Double-trapezium cylinder codebook model based on yuv color model for foreground detection with shadow and highlight suppression**. *Journal of Signal Processing Systems*, 85(2):221–233.

[Ice et al., 2012] Ice, J., Narang, N., Whitelam, C., Kalka, N., Hornak, L., Dawson, J., et Bourlai, T. (2012). **Swir imaging for facial image capture through tinted materials**. In *Infrared Technology and Applications XXXVIII*, volume 8353, page 83530S. International Society for Optics and Photonics.

[Ioffe et al., 2015] Ioffe, S., et Szegedy, C. (2015). **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. *arXiv preprint arXiv:1502.03167*.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., et Efros, A. A. (2017). **Image-to-image translation with conditional adversarial networks**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

[Jacobson et al., 2005] Jacobson, N. P., et Gupta, M. R. (2005). **Design goals and solutions for display of hyperspectral images**. *IEEE Transactions on Geoscience and Remote Sensing*, 43(11):2684–2692.

[Jarrett et al., 2009] Jarrett, K., Kavukcuoglu, K., Ranzato, M., et LeCun, Y. (2009). **What is the best multi-stage architecture for object recognition?** In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE.

[KaewTraKulPong et al., 2002] KaewTraKulPong, P., et Bowden, R. (2002). **An improved adaptive background mixture model for real-time tracking with shadow detection**. In *Video-based surveillance systems*, pages 135–144. Springer.

[Kalsotra et al., 2019] Kalsotra, R., et Arora, S. (2019). **A comprehensive survey of video datasets for background subtraction**. *IEEE Access*, 7:59143–59171.

[Karras et al., 2017] Karras, T., Aila, T., Laine, S., et Lehtinen, J. (2017). **Progressive growing of gans for improved quality, stability, and variation**. *arXiv preprint arXiv:1710.10196*.

[Kim et al., 2005] Kim, K., Chalidabhongse, T. H., Harwood, D., et Davis, L. (2005). **Real-time foreground-background segmentation using codebook model**. *Real-Time Imaging*, 11(3):172–185.

[Kim et al., 2013] Kim, S. W., Yun, K., Yi, K. M., Kim, S. J., et Choi, J. Y. (2013). **Detection of moving objects with a moving camera using non-panoramic background model**. *Machine vision and applications*, 24(5):1015–1028.

[Kohonen, 1995] Kohonen, T. (1995). **Learning vector quantization**. In *Self-organizing maps*, pages 175–189. Springer.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., et Hinton, G. E. (2012). **Imagenet classification with deep convolutional neural networks**. In *Advances in neural information processing systems*, pages 1097–1105.

[Krungkaew et al., 2016] Krungkaew, R., et Kusakunniran, W. (2016). **Foreground segmentation in a video by using a novel dynamic codebook**. In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6. IEEE.

[Kusakunniran et al., 2016] Kusakunniran, W., et Krungkaew, R. (2016). **Dynamic codebook for foreground segmentation in a video**. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, 10(2):144–155.

[Laugraud et al., 2015] Laugraud, B., Piérard, S., Braham, M., et Van Droogenbroeck, M. (2015). **Simple median-based method for stationary background generation using background subtraction algorithms**. In *International Conference on Image Analysis and Processing*, pages 477–484. Springer.

[LeCun, 2016] LeCun, Y. (2016). **RI seminar: The next frontier in ai: Unsupervised learning**. Technical Report.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., et Hinton, G. (2015). **Deep learning**. *nature*, 521(7553):436–444.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et Jackel, L. D. (1989). **Backpropagation applied to handwritten zip code recognition**. *Neural computation*, 1(4):541–551.

[Ledig et al., 2017] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et others (2017). **Photo-realistic single image super-resolution using a generative adversarial network**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

[Li et al., 2016] Li, C., Wang, X., Zhang, L., Tang, J., Wu, H., et Lin, L. (2016). **Weighted low-rank decomposition for robust grayscale-thermal foreground detection**. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):725–738.

[Li et al., 2012] Li, F., et Zhou, H. (2012). **A two-layers background modeling method based on codebook and texture**. *J. Univ. Sci. Technol. China*, 2:4.

[Li et al., 2017] Li, H., Sudusinghe, K., Liu, Y., Yoon, J., Van Der Schaar, M., Blasch, E., et Bhattacharyya, S. S. (2017). **Dynamic, data-driven processing of multispectral video streams**. *IEEE Aerospace and Electronic Systems Magazine*, 32(7):50–57.

[Liang et al., 2002] Liang, W., et Niu, H. W. M. T. (2002). **A survey of visual analysis of human motion [j]**. *Chinese Journal of Computers*, 3:225–237.

[Lim et al., 2017] Lim, K., Jang, W.-D., et Kim, C.-S. (2017). **Background subtraction using encoder-decoder structured convolutional neural network**. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE.

[Lim et al.,] Lim, L., et Keles, H. **Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. arxiv 2018**. *arXiv preprint arXiv:1801.02225*.

[Lim et al., 2018] Lim, L. A., et Keles, H. (2018). **Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding**. *Pattern Recognition Letters*, 112.

[Lim et al., 2019] Lim, L. A., et Keles, H. Y. (2019). **Learning multi-scale features for foreground segmentation**. *Pattern Analysis and Applications*.

[Lin et al., 2013] Lin, M., Chen, Q., et Yan, S. (2013). **Network in network**. *arXiv preprint arXiv:1312.4400*.

[Lin, 2016] Lin, Z. (2016). **A review on low-rank models in data analysis**. *Big Data and Information Analytics*, 1(2/3):139–161.

[Lipton et al., 1998] Lipton, A. J., Fujiyoshi, H., et Patil, R. S. (1998). **Moving target classification and tracking from real-time video**. In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No. 98EX201)*, pages 8–14. IEEE.

[Litman, 2017] Litman, T. (2017). **Autonomous vehicle implementation predictions**. Victoria Transport Policy Institute Victoria, Canada.

[Long et al., 2015] Long, J., Shelhamer, E., et Darrell, T. (2015). **Fully convolutional networks for semantic segmentation**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

[Lowe, 1999] Lowe, D. G. (1999). **Object recognition from local scale-invariant features**. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

[Ma et al., 2012] Ma, F., et Sang, N. (2012). **Background subtraction based on multi-channel siltp**. In *Asian Conference on Computer Vision*, pages 73–84. Springer.

[Maddalena et al., 2007] Maddalena, L., et Petrosino, A. (2007). **Moving object detection for real-time applications**. In *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, pages 542–547. IEEE.

[Maddalena et al., 2018] Maddalena, L., et Petrosino, A. (2018). **Background subtraction for moving object detection in rgbd data: A survey**. *Journal of Imaging*, 4(5):71.

[Martínez-Díaz et al., 2018] Martínez-Díaz, M., et Soriguera, F. (2018). **Autonomous vehicles: theoretical and practical challenges**. *Transportation Research Procedia*, 33:275–282.

[McHugh et al., 2009] McHugh, J. M., Konrad, J., Saligrama, V., et Jodoin, P.-M. (2009). **Foreground-adaptive background subtraction**. *IEEE Signal Processing Letters*, 16(5):390–393.

[Mittal et al., 2004] Mittal, A., et Paragios, N. (2004). **Motion-based background subtraction using adaptive kernel density estimation**. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. Ieee.

[Mousse et al., 2014] Mousse, M. A., Ezin, E. C., et Motamed, C. (2014). **Foreground-background segmentation based on codebook and edge detector**. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, pages 119–124. IEEE.

[Moyà-Alcover et al., 2017] Moyà-Alcover, G., Elgammal, A., Jaume-i Capó, A., et Varona, J. (2017). **Modeling depth for nonparametric foreground segmentation using rgbd devices**. *Pattern Recognition Letters*, 96:76–85.

[Murgia et al., 2014] Murgia, J., Meurie, C., et Ruichek, Y. (2014). **All-day moving objects detection for security at level crossing**.

[Nogueira et al., 2017] Nogueira, K., Penatti, O. A., et dos Santos, J. A. (2017). **Towards better exploiting convolutional neural networks for remote sensing scene classification**. *Pattern Recognition*, 61:539–556.

[Noh et al., 2012] Noh, S., et Jeon, M. (2012). **A new framework for background subtraction using multiple cues**. In *Asian Conference on Computer Vision*, pages 493–506. Springer.

[Noh et al., 2014] Noh, S., Shim, D., et Jeon, M. (2014). **Background subtraction method using codebook-gmm model**. In *The 2014 International Conference on Control, Automation and Information Sciences (ICCAIS 2014)*, pages 117–120. IEEE.

[Ojala et al., 1994] Ojala, T., Pietikainen, M., et Harwood, D. (1994). **Performance evaluation of texture measures with classification based on kullback discrimination of distributions**. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585. IEEE.

[Ojala et al., 1996] Ojala, T., Pietikäinen, M., et Harwood, D. (1996). **A comparative study of texture measures with classification based on featured distributions**. *Pattern recognition*, 29(1):51–59.

[Pappa et al., 2014] Pappa, G. L., Ochoa, G., Hyde, M. R., Freitas, A. A., Woodward, J., et Swan, J. (2014). **Contrasting meta-learning and hyper-heuristic research: the role of evolutionary algorithms**. *Genetic Programming and Evolvable Machines*, 15(1):3–35.

[Parkhi et al., 2015] Parkhi, O. M., Vedaldi, A., Zisserman, A., et others (2015). **Deep face recognition.** In *bmvc*, volume 1, page 6.

[Pitts et al., 1947] Pitts, W., et McCulloch, W. S. (1947). **How we know universals the perception of auditory and visual forms**. *The Bulletin of mathematical biophysics*, 9(3):127–147.

[Polacco et al.,] Polacco, A., et Backes, K. **The amazon go concept: Implications, applications, and sustainability**.

[Qiu et al., 2019] Qiu, S., Tang, Y., Du, Y., et Yang, S. (2019). **The infrared moving target extraction and fast video reconstruction algorithm**. *Infrared Physics & Technology*, 97:85–92.

[Radford et al., 2015] Radford, A., Metz, L., et Chintala, S. (2015). **Unsupervised representation learning with deep convolutional generative adversarial networks**. *arXiv preprint arXiv:1511.06434*.

[Ranzato et al., 2007] Ranzato, M., Huang, F. J., Boureau, Y.-L., et LeCun, Y. (2007). **Unsupervised learning of invariant feature hierarchies with applications to object recognition**. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., et Farhadi, A. (2016). **You only look once: Unified, real-time object detection**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., et Sun, J. (2015). **Faster r-cnn: Towards real-time object detection with region proposal networks**. In *Advances in neural information processing systems*, pages 91–99.

[Ripley, 2007] Ripley, B. D. (2007). **Pattern recognition and neural networks**. Cambridge university press.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., et Brox, T. (2015). **U-net: Convolutional networks for biomedical image segmentation**. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Rüfenacht et al., 2013] Rüfenacht, D., Fredembach, C., et Süsstrunk, S. (2013). **Automatic and accurate shadow detection using near-infrared information**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1672–1678.

[Ruidong, 2015] Ruidong, G. (2015). **Moving object detection based on improved codebook model**. In *2nd International Conference on Modelling, Identification and Control*. Atlantis Press.

[Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., et Williams, R. J. (1985). **Learning internal representations by error propagation**. Technical Report, California Univ San Diego La Jolla Inst for Cognitive Science.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., et Williams, R. J. (1986). **Learning representations by back-propagating errors**. *nature*, 323(6088):533–536.

[Saatci et al., 2017] Saatci, Y., et Wilson, A. G. (2017). **Bayesian gan**. In *Advances in neural information processing systems*, pages 3622–3631.

[Salerno et al., 2014] Salerno, E., Tonazzini, A., Grifoni, E., Lorenzetti, G., Legnaioli, S., Lezzerini, M., Marras, L., Pagnotta, S., et Palleschi, V. (2014). **Analysis of multispectral images in cultural heritage and archaeology**.

[Salimans et al., 2016] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., et Chen, X. (2016). **Improved techniques for training gans**. In *Advances in neural information processing systems*, pages 2234–2242.

[Schowengerdt, 2006] Schowengerdt, R. A. (2006). **Remote sensing: models and methods for image processing**. Elsevier.

[Shah et al., 2015] Shah, M., Deng, J. D., et Woodford, B. J. (2015). **A self-adaptive codebook (sacb) model for real-time background subtraction**. *Image and Vision Computing*, 38:52–64.

[Shaw et al., 2003] Shaw, G. A., et Burke, H. K. (2003). **Spectral imaging for remote sensing**. *Lincoln laboratory journal*, 14(1):3–28.

[Sheikh et al., 2005] Sheikh, Y., et Shah, M. (2005). **Bayesian modeling of dynamic scenes for object detection**. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1778–1792.

[Shimada et al., 2013] Shimada, A., Nagahara, H., et Taniguchi, R.-i. (2013). **Object detection based on spatio-temporal light field sensing**. *Information and Media Technologies*, 8(4):1115–1119.

[Shimada et al., 2015] Shimada, A., Nagahara, H., et Taniguchi, R.-i. (2015). **Change detection on light field for active video surveillance**. In *2015 12th IEEE International*

Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE.

[Shuigen et al., 2009] Shuigen, W., Zhen, C., et Hua, D. (2009). **Motion detection based on temporal difference method and optical flow field**. In *2009 Second International Symposium on Electronic Commerce and Security*, volume 2, pages 85–88. IEEE.

[Silva et al., 2016] Silva, C., Bouwmans, T., et Frélicot, C. (2016). **Online weighted one-class ensemble for feature selection in background/foreground separation**. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2216–2221. IEEE.

[Silva et al., 2017] Silva, C., Bouwmans, T., et Frelicot, C. (2017). **Superpixel-based online wagging one-class ensemble for feature selection in foreground/background separation**. *Pattern Recognition Letters*, 100:144–151.

[Simonyan et al., 2014] Simonyan, K., et Zisserman, A. (2014). **Very deep convolutional networks for large-scale image recognition**. *arXiv preprint arXiv:1409.1556*.

[Sobral et al., 2015] Sobral, A., Javed, S., Ki Jung, S., Bouwmans, T., et Zahzah, E.-h. (2015). **Online stochastic tensor decomposition for background subtraction in multispectral video sequences**. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 106–113.

[Sobral et al., 2014] Sobral, A., et Vacavant, A. (2014). **A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos**. *Computer Vision and Image Understanding*, 122:4 – 21.

[Sobral, 2017] Sobral, A. C. (2017). **Robust low-rank and sparse decomposition for moving object detection: from matrices to tensors**. PhD thesis.

[St-Charles et al., 2014] St-Charles, P.-L., Bilodeau, G.-A., et Bergevin, R. (2014). **Subsense: A universal change detection method with local adaptive sensitivity**. *IEEE Transactions on Image Processing*, 24(1):359–373.

[St-Charles et al., 2015] St-Charles, P.-L., Bilodeau, G.-A., et Bergevin, R. (2015). **A self-adjusting approach to change detection based on background word consensus**. In *2015 IEEE winter conference on applications of computer vision*, pages 990–997. IEEE.

[St-Charles et al., 2019] St-Charles, P.-L., Bilodeau, G.-A., et Bergevin, R. (2019). **Online mutual foreground segmentation for multispectral stereo videos**. *International Journal of Computer Vision*, pages 1–19.

[Stauffer et al., 1999] Stauffer, C., et Grimson, W. E. L. (1999). **Adaptive background mixture models for real-time tracking**. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE.

[Stauffer et al., 2000] Stauffer, C., et Grimson, W. E. L. (2000). **Learning patterns of activity using real-time tracking**. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):747–757.

[Sultana et al., 2018] Sultana, M., Mahmood, A., Javed, S., et Jung, S. K. (2018). **Unsupervised deep context prediction for background foreground separation**.

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., et Alemi, A. A. (2017). **Inception-v4, inception-resnet and the impact of residual connections on learning**. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., et Rabinovich, A. (2015). **Going deeper with convolutions**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., et Wojna, Z. (2016). **Rethinking the inception architecture for computer vision**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[Thakoor et al., 2004] Thakoor, N., Gao, J., et Chen, H. (2004). **Automatic object detection in video sequences with camera in motion**. In *Proceedings of Advanced Concepts for Intelligent Vision Systems*. Citeseer.

[Tompson et al., 2015] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., et Bregler, C. (2015). **Efficient object localization using convolutional networks**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656.

[Toyama et al., 1999] Toyama, K., Krumm, J., Brumitt, B., et Meyers, B. (1999). **Wallflower: Principles and practice of background maintenance**. In *Proceedings* *of the seventh IEEE international conference on computer vision*, volume 1, pages 255–261. IEEE.

[Tu et al., 2008] Tu, Q., Xu, Y., et Zhou, M. (2008). **Box-based codebook model for real-time objects detection**. In *2008 7th World Congress on Intelligent Control and Automation*, pages 7621–7625. IEEE.

[Uijlings et al., 2013] Uijlings, J. R., Van De Sande, K. E., Gevers, T., et Smeulders, A. W. (2013). **Selective search for object recognition**. *International journal of computer vision*, 104(2):154–171.

[Ulyanov et al., 2016] Ulyanov, D., Vedaldi, A., et Lempitsky, V. (2016). **Instance normalization: The missing ingredient for fast stylization**. *arXiv preprint arXiv:1607.08022*.

[USGS, 2018] USGS, U. (2018). **What are the band designations for the landsat satellites?** *línea]. Available: http://landsat. usgs. gov/band_designations_landsat_satellites. php*.

[Viau et al., 2016] Viau, C., Payeur, P., et Cretu, A.-M. (2016). **Multispectral image analysis for object recognition and classification**. In *Automatic Target Recognition XXVI*, volume 9844, page 98440N. International Society for Optics and Photonics.

[Viswanath et al., 2015] Viswanath, A., Behera, R. K., Senthamilarasu, V., et Kutty, K. (2015). **Background modelling from a moving camera**. *Procedia Computer Science*, 58:289–296.

[Wang et al., 2018] Wang, J., Ma, Y., Zhang, L., Gao, R. X., et Wu, D. (2018). **Deep learning for smart manufacturing: Methods and applications**. *Journal of Manufacturing Systems*, 48:144–156.

[Wang et al., 2009] Wang, X., Han, T. X., et Yan, S. (2009). **An hog-lbp human detector with partial occlusion handling**. In *2009 IEEE 12th international conference on computer vision*, pages 32–39. IEEE.

[Wang et al., 2014] Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., et Ishwar, P. (2014). **Cdnet 2014: an expanded change detection benchmark dataset**. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 387–394.

[Wang et al., 2017] Wang, Y., Luo, Z., et Jodoin, P.-M. (2017). **Interactive deep learning method for segmenting moving objects**. *Pattern Recognition Letters*, 96:66–75.

[Xu et al., 2016] Xu, Y., Dong, J., Zhang, B., et Xu, D. (2016). **Background modeling methods in video analysis: A review and comparative evaluation**. *CAAI Transactions on Intelligence Technology*, 1(1):43–60.

[Yan et al., 2018] Yan, Y., Ren, J., Zhao, H., Sun, G., Wang, Z., Zheng, J., Marshall, S., et Soraghan, J. (2018). **Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos**. *Cognitive Computation*, 10(1):94–104.

[Yongjia et al., 2014] Yongjia, Z., et Weihua, L. (2014). **Fast robust foreground-background segmentation based on variable rate codebook method in bayesian framework for detecting objects of interest**. In *2014 7th International Congress on Image and Signal Processing*, pages 55–59. IEEE.

[Yu et al., 2015] Yu, F., et Koltun, V. (2015). **Multi-scale context aggregation by dilated convolutions**. *arXiv preprint arXiv:1511.07122*.

[Zaharescu et al., 2011] Zaharescu, A., et Jamieson, M. (2011). **Multi-scale multi-feature codebook-based background subtraction**. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1753–1760. IEEE.

[Zeng et al., 2019] Zeng, D., Chen, X., Zhu, M., Goesele, M., et Kuijper, A. (2019). **Background subtraction with real-time semantic segmentation**. *IEEE Access*, 7:153869–153884.

[Zeng et al., 2014] Zeng, Z., et Jia, J. (2014). **Arbitrary cylinder color model for the codebook based background subtraction**. *Optics express*, 22(18):21577–21588.

[Zhang et al., 2019] Zhang, A., Lipton, Z. C., Li, M., et Smola, A. J. (2019). **Dive into Deep Learning**. http://www.d2l.ai.

[Zhang, 2018] Zhang, C. (2018). **Deep learning for land cover and land use classification**. PhD thesis, Lancaster University.

[Zhang et al., 2006] Zhang, Y., Kiselewich, S. J., Bauson, W. A., et Hammoud, R. (2006). **Robust moving object detection at distance in the visible spectrum and beyond**

using a moving camera. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 131–131. IEEE.

[Zhang et al., 2016] Zhang, Y.-T., Bae, J.-Y., et Kim, W.-Y. (2016). **Multi-layer multi-feature background subtraction using codebook model framework**. *World Acad. Sci., Eng. Technol., Int. J. Comput. Inf. Eng*, 3(1).

[Zhao et al., 2008] Zhao, Y., Berns, R. S., Taplin, L. A., et Coddington, J. (2008). **An investigation of multispectral imaging for the mapping of pigments in paintings**. In *Computer image analysis in the study of art*, volume 6810, page 681007. International Society for Optics and Photonics.

[Zhao et al., 2019] Zhao, Z.-Q., Zheng, P., Xu, S.-t., et Wu, X. (2019). **Object detection with deep learning: A review**. *IEEE transactions on neural networks and learning systems*.

[Zheng et al., 2019a] Zheng, A., Ye, N., Li, C., Wang, X., et Tang, J. (2019a). **Multi-modal foreground detection via inter-and intra-modality-consistent low-rank separation**. *Neurocomputing*.

[Zheng et al., 2018a] Zheng, W., et Wang, K. (2018a). **Background subtraction algorithm with bayesian generative adversarial networks**. *Zidonghua Xuebao/Acta Automatica Sinica*, 44.

[Zheng et al., 2018b] Zheng, W., Wang, K., et Wang, F. (2018b). **Background subtraction algorithm based on bayesian generative adversarial networks**. *Acta Automatica Sinica*, 44(5):878–890.

[Zheng et al., 2019b] Zheng, W., Wang, K., et Wang, F.-Y. (2019b). **A novel background subtraction algorithm based on parallel vision and bayesian gans**. *Neurocomputing*.

[Zivkovic, 2004] Zivkovic, Z. (2004). **Improved adaptive gaussian mixture model for background subtraction**. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE.

[Zivkovic et al., 2006] Zivkovic, Z., et Van Der Heijden, F. (2006). **Efficient adaptive density estimation per image pixel for the task of background subtraction**. *Pattern recognition letters*, 27(7):773–780.

LIST OF FIGURES

List of Tables

A

PUBLICATIONS

A.1/ CONFERENCES

Liu, R., Ruichek, Y., & El Bagdouri, M. **Background Subtraction with Multispectral Images using Codebook Algorithm**. In International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, 2017.

Liu, R., Ruichek, Y., & El Bagdouri, M. **Enhanced Codebook Model and Fusion for Object Detection with Multispectral Images**. In International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, 2018.

Liu, R., Ruichek, Y., & El Bagdouri, M. **Multispectral Dynamic Codebook and Fusion Strategy for Moving Objects Detection**. In International Conference on Image and Signal Processing. Springer, 2020.

A.2/ JOUNALS

Liu, R., Ruichek, Y., & El Bagdouri, M. **Extended Codebook with Multispectral Sequences for Background Subtraction**. Sensors 19.3 (2019): 703.

Liu, R., Ruichek, Y., & El Bagdouri, M. **Multispectral Background Subtraction with Deep Learning**. Submitted to Signal Processing: Image Communication.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the `tex-upmethodology` package suite — http://www.arakhne.org/tex-upmethodology/