

**NEW APPROACHES TO IMPROVE  
PERFORMANCE OF BACKGROUND  
SUBTRACTION**

Şahin Işık  
Doktora Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

Mart 2018

**NEW APPROACHES TO IMPROVE PERFORMANCE OF BACKGROUND  
SUBTRACTION**

**Şahin IŞIK**

**Ph.D. Dissertation**

**Computer Engineering Department**

**Supervisor: Assoc. Prof. Dr. Kemal ÖZKAN**

**Eskişehir Anadolu University**

**Graduate School of Sciences**

**March 2018**

## FINAL APPROVAL FOR THESIS

This thesis titled “New Approaches to Improve Performance of Background Subtraction” has been prepared and submitted by Şahin Işık in partial fulfillment of the requirements in “Anadolu University Directive on Graduate Education and Examination” for the Degree of Doctor of Philosophy (PhD) in Computer Engineering Department has been examined and approved on 28/03/2018.

<u>Committee Members</u>	<u>Title Name Surname</u>	<u>Signature</u>
Member (Supervisor)	: Assoc. Prof. Dr. Kemal ÖZKAN	.....
Member	: Prof. Dr. Ömer Nezir GEREK	.....
Member	: Prof. Dr. İdris DAĞ	.....
Member	: Assoc. Prof. Dr. Serkan GÜNAL	.....
Member	: Assist. Prof. Dr. Alper Kürşat UYSAL.....	.....

**Prof. Dr. Ersin YÜCEL**  
**Director of Graduate School of Sciences**

## **ABSTRACT**

### **NEW APPROACHES TO IMPROVE PERFORMANCE OF BACKGROUND SUBTRACTION**

**Şahin IŞIK**

**Department of Computer Engineering**

**Anadolu University, Graduate School of Sciences, March 2018**

**Supervisor: Assoc. Prof. Dr. Kemal ÖZKAN**

Separation of the foreground from background on a processed image, namely background modelling, positively affects performance of certain computer vision applications. It has considered as preprocess for many tasks including moving object recognition, person tracking, traffic monitoring, motion capturing, teleconference and security surveillance systems. Video backgrounds can be considered in two categories as static and dynamic backgrounds. To improve the performance of background subtraction, we have developed four different methods by using different tools in case of distance computation between test and background frame and integrating a feedback mechanism that works beyond dynamic controller parameters. These methods are called as Background Modelling Using Common Vector Approach (BMCVA), Background Modelling Using Common Matrix Approach (BMCMA), Sliding Window-Based Change Detection (SWCD) and Common Vector Approach Based Background Subtraction (CVABS). Various experiments have conducted on different problem types related to dynamic backgrounds over CDnet2014 and Wallflower datasets. Several types of metrics calculated over the results of True-Positive (TP), True-Negative (TN), False-Positive (FP) and False-Negative (FN) counts, have utilized as objective measures and the obtained visual results are judged subjectively. Once the obtained results inspected, it has observed that the proposed methods generate successful results for different challenges.

**Keywords:** Foreground Segmentation, Moving Object Segmentation, Change Detection, Background Subtraction, Background Modelling.

## ÖZET

# ARKA PLAN ÇIKARMA BAŞARIMI İYİLEŞTİRMEK İÇİN YENİ YAKLAŞIMLAR

Şahin IŞIK

**Bilgisayar Mühendisliği Anabilim Dalı**  
**Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Mart 2018**

**Danışman: Doç. Dr. Kemal ÖZKAN**

İşlenen bir görüntüde ön planın arka plandan ayrıştırılması, adıyla arka plan modelleme, bazı bilgisayar görme uygulamalarının performansını olumlu şekilde etkiler. Hareketli cisim tanıma, kişi takibi, trafik izleme, hareket yakalama, telekonferans ve güvenlik gözetim sistemleri de içermek üzere birçok görev için ön işlem olarak düşünülür. Video arka planları statik ve dinamik olarak iki kategoride değerlendirilebilir. Bu çalışmada, arka plandaki çıkarma işleminin performansını artırmak için, test imgesi ve arka plan imgesi arasındaki uzaklığın hesaplanmasında farklı araçlar kullanılarak ve dinamik denetleyici parametrelerinin ötesinde çalışan bir geri bildirim mekanizmasının entegrasyonu ile dört farklı yöntem geliştirilmiştir. Bu yöntemler Ortak Vektör Yaklaşımı Kullanarak Arka Plan Modelleme (BMCVA), Ortak Matris Yaklaşımı Kullanarak Arka Plan Modelleme (BMCMA), Kayan Pencere Tabanlı Hareket Tanıma (SWCD) ve Ortak Vektör Tabanlı Arka Plan Çıkarma (CVABS) olarak adlandırılmıştır. CDnet2014 ve Wallflower veritabanları üzerinde dinamik arka planlarla ilgili alakalı farklı problem türleri üzerinden çeşitli deneyler yapılmıştır. Gerçek-Pozitif (TP), Doğru-Negatif (TN), Yanlış-Pozitif (FP) ve Yanlış-Negatif (FN) sayıları üzerinden hesaplanan metrikler objektif ölçümler olarak kullanılmış ve elde edilen görsel sonuçlar nesnel olarak değerlendirilmiştir. Elde edilen sonuçlar incelendiğinde, önerilen yöntemlerin farklı zorluklar için başarılı sonuçlar verdiğini gözlemlenmiştir.

**Anahtar Kelimeler:** Ön Plan Segmentasyon, Hareketli Nesne Segmentasyonu, Hareket Tespiti, Arka Plan Çıkarma, Arka Plan Modelleme.

## **ACKNOWLEDGEMENTS**

First, I would like to thank my supervisor, Assoc. Dr. Kemal ÖZKAN, for his guidance, advice and valuable knowledge; encouraging and motivating me to develop this work.

Especially, I would like to express my sincere thanks to Prof. Dr. Ömer Nezh Gerek, Prof. Dr. İdris Dağ, Assoc. Dr. Serkan Günal and Dr. Alper Kürşat Uysal for their valuable contributions as well as serving on my committee.

Finally, thanks to my family and all friends for their emotional supports and inspired me to reach my dreams.

Şahin Işık

March 2018

## **STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES**

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Anadolu University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

**Şahin Işık**

## CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>FINAL APPROVAL FOR THESIS</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>ÖZET</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES</b>	<b>vi</b>
<b>CONTENTS</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>1.1. Related Works.....</b>	<b>2</b>
<b>1.2. Problems Related to Background Subtraction .....</b>	<b>6</b>
<b>1.3. Thesis Organization.....</b>	<b>13</b>
<b>2. PROPOSED METHODS ON IMPROVING BACKGROUND SUBTRACTION PERFORMANCE</b>	<b>15</b>
<b>2.1. Method 1: Background Modelling Based on the Common Vector Approach     (BMCVA) .....</b>	<b>17</b>
<b>2.2. Method 2: Background Modelling Based on the Common Matrix     Approach (BMCMA) .....</b>	<b>21</b>
<b>2.3. Method 3: A Sliding Window and Self-Regulated Learning Based     Background Updating (SWCD) .....</b>	<b>24</b>
<b>2.3.1. Foreground detection.....</b>	<b>27</b>
<b>2.3.2. Updating background frames .....</b>	<b>31</b>
<b>2.3.3. Updating internal parameters.....</b>	<b>33</b>
<b>2.3.4. Handling PTZ challenges .....</b>	<b>37</b>
<b>2.3.5. Implementation details .....</b>	<b>38</b>
<b>2.4. Method 4: Moving Object Segmentation with Common Vector Approach     (CVABS).....</b>	<b>39</b>



2.4.1. Application to background modelling.....	42
2.4.2. Common vector versus average vector .....	45
2.4.3. Foreground detection.....	48
<b>3. PERFORMANCE ANALYSIS</b>	<b>51</b>
3.1. Datasets.....	51
3.2. Evaluation Metrics .....	51
3.3. Performance of Method 1: BMCVA .....	52
3.3.1. Subjective evaluation of method 1: BMCVA.....	52
3.3.2. Objective evaluation of method 1: BMCVA.....	54
3.4. Performance of Method 2: BMCMA .....	55
3.4.1. Subjective evaluation of method 2: BMCMA.....	55
3.4.2. Objective evaluation of method 2: BMCMA .....	57
3.5. Performance of Method 3: SWCD.....	58
3.5.1. Subjective evaluation of method 3: SWCD.....	58
3.5.2. Objective evaluation of method 3: SWCD .....	59
3.5.3. Computational issues of method 3: SWCD.....	63
3.6. Performance of Method 4: CVABS .....	63
3.6.1. Subjective evaluation of method 4: CVABS .....	63
3.6.2. Objective evaluation of method 4: CVABS .....	65
3.7. Overall Performance Evaluation of Our Proposed Works .....	67
<b>4. CONCLUSIONS</b>	<b>70</b>
<b>REFERENCES</b>	<b>72</b>
<b>RESUME</b>	<b>79</b>

## LIST OF TABLES

<b>Table 1.1.</b> <i>Pros and cons for each group of background subtraction methods</i> .....	5
<b>Table 2.1.</b> <i>Notations for SWCD algorithm</i> .....	25
<b>Table 3.1.</b> <i>Utilized metrics</i> .....	52
<b>Table 3.2.</b> <i>Objective performance evaluation on the Wallflower dataset</i> .....	54
<b>Table 3.3.</b> <i>Numerical results of CMA on the Wallflower dataset</i> .....	57
<b>Table 3.4.</b> <i>Objective performance comparison with top state of art methods</i> .....	60
<b>Table 3.5.</b> <i>Detailed performance (F-score) evaluation of each category</i> .....	60
<b>Table 3.6.</b> <i>Performance (F-score) evaluation in more details for each category of aforementioned methods</i> .....	66
<b>Table 3.7.</b> <i>F-measure scores of our proposed methods on CDnet</i> .....	68
<b>Table 3.8.</b> <i>F-measure scores of our proposed methods on Wallflower</i> .....	69

## LIST OF FIGURES

<b>Figure 1.1.</b> <i>A simple example to visualize the background subtraction.....</i>	1
<b>Figure 1.2.</b> <i>The videos related to badWeather category.....</i>	7
<b>Figure 1.3.</b> <i>The videos related to lowFramerate category.....</i>	8
<b>Figure 1.4.</b> <i>The videos related to NightVideos category.....</i>	8
<b>Figure 1.5.</b> <i>The videos related to PTZ category.....</i>	9
<b>Figure 1.6.</b> <i>The videos related to Turbulence category.....</i>	10
<b>Figure 1.7.</b> <i>The videos related to Baseline category.....</i>	10
<b>Figure 1.8.</b> <i>The videos related to the DynamicBackground category.....</i>	11
<b>Figure 1.9.</b> <i>The videos related to the CameraJitter category.....</i>	11
<b>Figure 1.10.</b> <i>The videos related to IntermittentObjectMotion category.....</i>	12
<b>Figure 1.11.</b> <i>The videos related to Shadow category.....</i>	13
<b>Figure 1.12.</b> <i>The videos related to Thermal category.....</i>	13
<b>Figure 2.1.</b> <i>Background subtraction process.....</i>	15
<b>Figure 2.2.</b> <i>Overall principles of CVA based background modelling and foreground detection.....</i>	19
<b>Figure 2.3.</b> <i>Overall principles of CMA based background modelling and foreground detection.....</i>	23
<b>Figure 2.4.</b> <i>Visual demonstration of validated foreground maps.....</i>	30
<b>Figure 2.5.</b> <i>An illustration of sliding window based background updating procedure.....</i>	32
<b>Figure 2.6.</b> <i>The visual performance with or without dynamic control parameters and post processing.....</i>	36
<b>Figure 2.7.</b> <i>Scene change detection mechanism for PTZ videos.....</i>	37
<b>Figure 2.8.</b> <i>Performance analysis of different number of samples (N) after conducting experiments on CDnet dataset.....</i>	38
<b>Figure 2.9.</b> <i>The visual demonstration of common frame of backgrounds, discriminative common frame and distance map between them.....</i>	43
<b>Figure 3.1.</b> <i>Subjective results of CVA on the Wallflower dataset.....</i>	53
<b>Figure 3.2.</b> <i>Subjective results of CMA on the Wallflower dataset.....</i>	56
<b>Figure 3.3.</b> <i>Some visual results of SWCD on the CDnet 2014 dataset.....</i>	58
<b>Figure 3.4.</b> <i>Visual performance demonstration on the CDnet dataset.....</i>	64

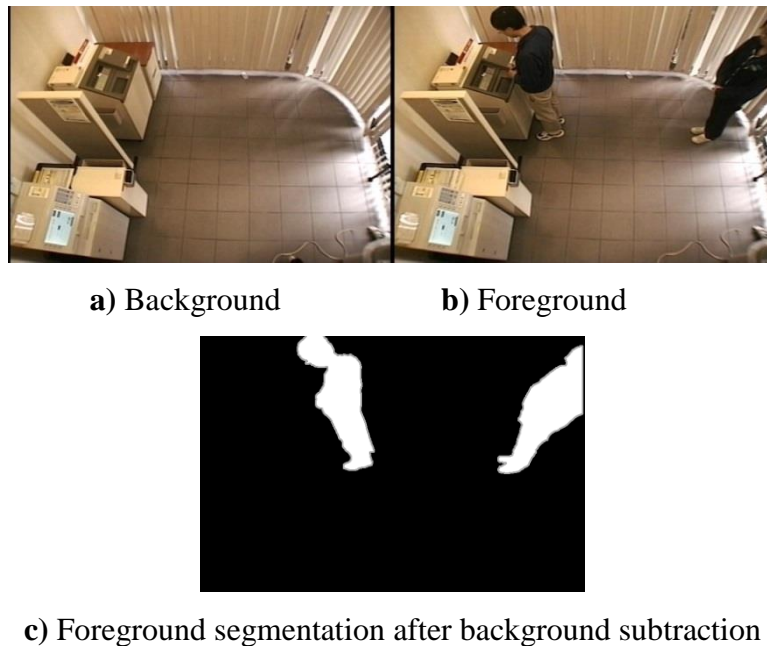
<b>Figure 3.5.</b> <i>MCC and F-score results for top ranked methods given in CDnet and CVABS</i> .....	66
<b>Figure 3.6.</b> <i>Visual outputs returned from proposed methods on CDnet</i> .....	67
<b>Figure 3.7.</b> <i>Visual Outputs returned from proposed methods on Wallflower</i> .....	69

## LIST OF ABBREVIATIONS

PCA	: Principal Component Analysis
INMF	: Non-Negative Matrix Factorization
ICA	: Independent Component Analysis
SG	: Single Gaussian
GMM	: Gaussian Mixture Model
VIBE	: Visual Background Extractor
PBAS	: Pixel-Based Adaptive Segmenter
SuBSENSE	: Self-Balanced SENSitivity SEgmenter
PAWCS	: Pixel-based Adaptive Word Consensus Segmenter
FTSG	: Flux Tensor with Split Gaussian models
EFIC	: Edge based Foreground segmentation with Interior Classification
LTP	: Local Ternary Pattern
MBS	: Multimode Background Subtraction
CNN	: Convolutional Neural Network
VGG	: Very Deep Convolutional Networks for Large-Scale Image Recognition
BMCVA	: Background Modelling Using Common Vector Approach
BMCMA	: Background Modelling Using Common Matrix Approach
SWCD	: Sliding Window-Based Change Detection
CVABS	: Common Vector Approach Based Background Subtraction
LBSP	: Local Binary Similarity Patterns
SL-PCA	: Subspace Learning PCA
SL-ICA	: Subspace Learning ICA
SL-INMF	: Subspace Learning via Incremental Non Negative Matrix Factorization
SL-IRT	: Subspace Learning via Incremental Rank-(R1, R2, R3) Tensor
TP	: True-Positive
TN	: True-Negative
FP	: False-Positive
FN	: False- Negative

## 1. INTRODUCTION

Moving object segmentation can be considered an important and painful procedure in computer vision. Although it has been called by different names ranging from “moving object detection”, “foreground detection” to “change detection” in literature, but it can be aggregated into a unique name, which we known as “background subtraction”. A background subtraction algorithm is concerned about the safety of public and private organizations listed as road surveillance, airplane surveillance, maritime surveillance, boats and store surveillance systems, since these places are carrying the risks about hazards and threats. In order to effectively handle the security of surveillance systems, numerous background subtraction algorithms have been developed and applied to identify threats and reduce the risks.



**Figure 1.1.** A simple example to visualize the background subtraction

The Figure 1.1 shows the background subtraction methodology. The images in Figure 1.1a-11b relates to *copyMachine* video of *Shadow* category, which is downloaded from CDnet2014 dataset [1]. The idea under a typical background subtraction algorithm, is updating the background regions and maintaining the moving object segmentation procedure without collapsing in case of revealing the foreground region. In time domain, the foreground regions can be unearthed by taking L1 norm distance between test frames

and background frame throughout video. After applying a fixed threshold, background regions would be marked with 0 while the foreground regions would be marked with 255, as shown in Figure 1.1c. However, the fixed threshold does not work in real time applications due to dynamic changes including shadow, illumination changes, fountains, waving trees or night videos. For such reasons, there are two convenient solutions as (i) developing a task oriented background subtraction algorithm or (ii) a universal one. While the first choice is applicable and easy to realize, but the issue becomes more illness when selecting the second option. Moreover, the background modelling task can be projected into a “learning” or “classification” procedure, which means that the tool used for background modelling can be relied on either a supervised or an unsupervised strategy.

### **1.1. Related Works**

Until now, various background subtraction algorithms have been proposed along with their advantages and disadvantages. In an example, while the subspace based methods generates valuable results, but they waste a huge rate of memory in case of processing large images. Meanwhile, there is no yet general method for background subtraction that gives accurate results in all scenarios, but it is widely accepted that every method performs well for specific categories. Some comprehensive surveys [2-4] presented by Bouwman to reveal the characteristic, computational and accuracy performance of well-known methods in background literature.

**Simple Methods:** As is evident from the name, some filters and shortsighted techniques were considered for background modelling. These methods can be given as Mean [5], Median [6], Euclidean Distance [7] and Histogram [8]. With an effortless manner, the statistical metrics called mean, median and histogram values of N frames had taken into account in the studies of Mean [5], Median [6] and Histogram [8] in case of initialization stage.

**Clustering Based Methods:** The most known of clustering based methods can be listed as K-Means clustering [9] and Codebook [10].

**Subspace Based Methods:** The idea under the subspace based methods is relied on the assumption that it is possible to represent the background information with Eigen vectors. The Principal Component Analysis (PCA) comes first of main sub-space based

background learning series. Although the PCA is widely preferred for data compression and classification tasks, but it also utilized for background modelling by recovering the data after projected on eigenvectors related to maximum eigenvalues. However, the PCA algorithm consumes much of memory when calculating the eigenvectors, which is known as drawback of sub-space decomposition from large size of matrices.

The discovery of using PCA for background modelling was first realized in the study of Olivier et. al [11]. This also paved the way for raising new methodologies based on the eigenvectors domains [12, 13] to improve the robustness of sub-space methods. To minimize computational cost of eigenvector decomposition, the L1 norm measure was considered instead of using l-2 norm error for eigenvectors extraction. The well-known methods of robust sub-space tracking methods was given as GOSUS [14] and p-ROST [15]. Additionally, some methods were handled data in tensor format including Incremental Rank- (R1, R2, R3) [16] based tensor learning for different types of images as specifically, IRTSA-GBM for background modelling on gray images and IRTSA-CBM for background modelling on color images. Moreover, the Incremental Non-Negative Matrix Factorization (INMF) [17] was proposed for background modelling by deriving the eigenvectors based on l-2 norm. Again, the Independent Component Analysis (ICA) [18] was proposed as alternative sub-space decomposition method for foreground segmentation.

**Gaussian Based Methods:** In the concept of Gaussian methods, the history of pixels was modelled with some Gaussian functions, called Gaussian models, and the similarity between Gaussian model of test and background frame was processed by employing a predefined threshold. A simple version of such methods can be observed as Single Gaussian [19]. However, it was observed that using single Gaussian functions don't provide effective results and the multi-models were revealed as Gaussian Mixture Model (GMM)[20] . The drawback of Gaussian based methods comes into sight when updating the variance and other parameters associated to background model.

**Pixels Based Methods:** Aside from the sub-space methods, the pixels based approaches have been developed for real time background modelling and updated with feedback based mechanisms. When observing their performances on various videos, one can emphasize that the pixel based methods gives effective results for background modelling.



In an attempt to retain the background pixels, the Visual Background Extractor (VIBE) [21] method retains the background pixels with smart random update strategy. Moreover, in the study of Pixel-Based Adaptive Segmenter (PBAS) [22], the decision threshold was updated instantly by introducing dynamic parameters. The Self-Balanced SENSitivity Segmenter (SuBSENSE) [23] performed some modifications on the PBAS by utilizing a distance metric relied on the Local Binary Similarity Patterns (LBSP) and introducing some new rules for updating decision threshold in a pixel-wise approach. Again, the Pixel-based Adaptive Word Consensus Segmenter (PAWCS) [24] method was proposed as an extended version SuBSENSE. In a similar way, background word consensus and the spatiotemporal information analysis was considered for distance computation and adjusting internal feedback driven parameters. Moreover, the concept of Gaussian Mixture Model (GMM) was employed in the SharedModel [25] that is the best matched GMM model chosen with a sharable GMM mechanism. Furthermore, the Flux Tensor with Split Gaussian models (FTSG) method [26] developed a hybrid model for splitting foreground from backgrounds by (i) using the flux tensor for motion detection, (ii) employing the idea of GMM for background modeling and (iii) a fusion step as combination of chamfer matching based validation and consensus of steps (i) and (ii) for deciding a pixel whether foreground or background. Furthermore, the Edge based Foreground background segmentation with Interior Classification (EFIC) [16] method revealed the limitation of Local Ternary Pattern (LTP) features for foreground detection. However, most of FTSG and EFIC performance comes from the validation procedure that relied on chamfer matching of edges. Being motivated from human visual system, the Multimode Background Subtraction (MBS) [27] method utilizes RGB and YCbCr color spaces to handle challenges of dynamic backgrounds rather employing single channel. The final foreground map was obtained after fusing with morphological operations on foreground maps from RGB and YCbCr color spaces. The performance of each method is available on the website of CDnet2014 dataset [1].

**Deep Learning Based Methods:** Since the 2012, the breakthrough of deep neural network algorithms have witnessed as numerous methods developed to maximize strength of machines in terms of learning and processing some tasks.

**Table 1.1.** Pros and cons for each group of background subtraction methods

Method	Pros & Cons
Simple	<ul style="list-style-type: none"> <li>-fast ✓</li> <li>-easy to implement ✓</li> <li>-save memory ✓</li> <li>-save CPU ✓</li> <li>-computationally cheap ✓</li> <li>-low performance ✗</li> <li>-not robust (collapse in all events) ✗</li> </ul>
Clustering	<ul style="list-style-type: none"> <li>-relatively fast ✓</li> <li>-complex to implement ✗</li> <li>-requires high amount of resources in memory ✗</li> <li>-computationally cheap ✓</li> <li>-low performance ✗</li> <li>-robust to illumination changes ✓</li> <li>-not robust to night videos, dynamic, intermittent and PTZ changes ✗</li> </ul>
Gaussian based Methods	<ul style="list-style-type: none"> <li>-relatively fast ✓</li> <li>-easy to implement ✓</li> <li>-requires high amount of resources in memory ✗</li> <li>-computationally cheap ✓</li> <li>-low performance ✗</li> <li>-robust to illumination changes ✓</li> <li>-not robust to night videos, dynamic, intermittent and PTZ ✗</li> </ul>
Subspace based Methods	<ul style="list-style-type: none"> <li>-slow ✗</li> <li>-easy to implement ✓</li> <li>-requires high amount of resources in memory ✗</li> <li>-computationally expensive ✗</li> <li>-low performance ✗</li> <li>-robust to illumination changes ✓</li> <li>-suffer from updating challenge ✗</li> <li>-not robust to night videos, dynamic, intermittent and PTZ ✗</li> </ul>
Pixel Based Methods	<ul style="list-style-type: none"> <li>-fast ✓</li> <li>-complex to implement ✗</li> <li>-requires high amount of resources in memory ✗</li> <li>-computationally cheap ✓</li> <li>-high performance ✓</li> <li>-robust to illumination changes, dynamic motions, intermittent motions ✓</li> <li>-fast updating procedure ✓</li> <li>-not robust to Night Videos and PTZ challenges ✗</li> </ul>
Deep Learning based Methods	<ul style="list-style-type: none"> <li>-slow (requires various convolutions in feed forward) ✗</li> <li>-relatively complex to implement ✗</li> <li>-requires high amount of resources in memory ✗</li> <li>-computationally expensive ✗</li> <li>-high performance ✓</li> <li>-slow training procedure (requires optimization) ✗</li> <li>-robust to every circumstance ✓</li> <li>-requires ground truth prior to training stage ✗</li> </ul>

According to its nature, the architecture of Convolutional Neural Network (CNN) learns through some layers, what can be called as deep layers, and gives superior results when compared with complex algorithms. Motivated from its robustness, the CNN architectures have also been employed to classify a pixel as background or foreground through image sequences of videos. In a study [28], the background frames library was obtained with the aid of SuBSENSE and FLUX tensor algorithm as FLUX tensor used for arranging background frames list and SuBSENSE used to reveal background regions. Once the Background Library obtained, a CNN architecture employed to spot foreground movements through video. The inputs were handled in  $37 \times 37$  patch. In another study [29], only a set of pre-segmented ground truth and RGB samples were utilized to teach a CNN network processing patches with  $31 \times 31$  size in pixel-wise manner. It means that a CNN network performed through an image and a  $31 \times 31$  patch in the neighboring of each pixel processed to determine its label whether foreground or background. The reported results indicate that CNN is an efficient solution to tackle with dynamic and illumination changes. However, the good results were obtained after utilizing CNN on multi-scale inputs. Instead of using a simple CNN model to learn each pixel, a more advanced CNN architecture together with encoder-decoder type network model was developed for moving object segmentation [30]. In referred work, the concept of VGG16 model was employed with multi-scale way (triplet) in case of building network and transposed convolutional network used for transforming features again into image space, which was called as decoder network. The superior results obtained on CDnet 2014 dataset.

The advantages and disadvantages of aforementioned background subtraction methods are summarized with the Table 1.1. One can observe that there is a tradeoff between speed and performance to decide a method as best. However, one can say that deep learning based ones can be utilized to train the background model with convolutional layers after generating the ground truths from a best unsupervised method.

## **1.2. Problems Related to Background Subtraction**

The possible problems related to the background subtraction have been exhibited in CDnet [1] by enveloping the all cases with 11 categories. These conditions have determined by community of *CDnet as BadWeather, LowFramerate, NightVideos, PTZ, Turbulence, Baseline, Dynamic Background, CameraJitter, IntermittentObjectMotion,*

*Shadow* and *Thermal*. In this dataset, there are different types of videos vary from 4 to 6 in size. When it comes to develop a real-time surveillance system with a background subtraction algorithm, then such most likely conditions have to be considered to achieve favorable results. Although using supervised methods like convolutional based deep learning strategies gives superior results in all categories, but prior to background learning stage it requires a massive effort to prepare the ground truth of samples when using the deep learning methods. On the other side, obtaining 90% of accuracy with unsupervised methods can be accepted as sufficient performance in terms of moving object segmentation. The details about problems on background subtraction are summarized with following chapters.

### ***BadWeather Category***

The *badWeather* category includes different videos associated to bad weather conditions which are *blizzard*, *skating*, *snowfall* and *wetSnow*. When excluding the *wetSnow* video, it is possible to obtain well-segmented results for moving objects such as humans and cars on aforementioned videos related to winter conditions.

The Figure 1.2 demonstrates some videos stated on *badWeather* category. While the first row exhibits the color images, the second row denotes their associated ground truth samples. When the videos examined, one can observe that it would be possible to obtain the nice results after employing a functional method.



**Figure 1.2.** *The videos related to badWeather category*

### ***LowFramerate Category***

The *lowFramerate* category contains videos recorded with different frame rates, i.e., including low frame rate per second (fps). The videos are *port\_0\_17fps*, *tramCrossroad\_1fps*, *tunnelExit\_0\_35fps* and *turnpike\_0\_5fps*. The fps information about each video was inserted into their names when recording them.

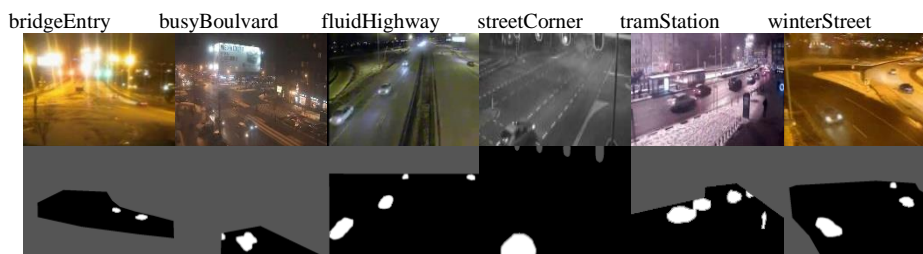


**Figure 1.3.** The videos related to *lowFramerate* category

The Figure 1.3 shows an overview of representative samples belong to each video, as first row denotes original images and second row exhibits related ground truths. The most troublesome video can be noted as *port* video, which includes different conditions of sky and shaking effects as well as small-scaled objects. When observing the performance of each unsupervised method in CDnet website, one can say that obtaining the highest result on *lowFramerate* category requires a great effort. What makes this category is so challenge to obtain valuable scores, can be explained that there are remotely focused images, and so, the objects in captured images have small-scale. Therefore, a smart segmentation procedure is greatly required to reveal the region of such small-scaled objects (refer to *port* video).

### ***NightVideos Category***

As is inferred by its name, the *nightVideos* category includes different videos recorded in night time. These are *bridgeEntry*, *busyBoulevard*, *fluidHighway*, *streetCornerAtNight*, *tramStation* and *winterStreet*. Presented scores on CDnet website indicate that obtaining highest scores for this category is some troublesome.



**Figure 1.4.** The videos related to *NightVideos* category

The samples related to the *NightVideos* category are presented in Figure 1.4, where first and second rows show original and ground images, respectively. Due to brightness effects of traffic lights and headlights of cars, obtaining highest scores for this category

can be considered as not easy. However, using an imaging system that is robust to night effects like SWIR, it could be produced the desirable results.

### ***PTZ Category***

The *PTZ* category includes different videos captured by a Pan Tilt Zoom camera, which covers wider area in a surveillance system. Some special videos of *PTZ* was listed as *continuousPan*, *intermittentPan*, *twoPositionPTZCam* and *zoomInZoomOut* in CDnet. When examining the performance of each method in CDnet, we can observe that the highest F-measure score can be achieved by utilizing a smart validation procedure that works beyond chamfer matching process.



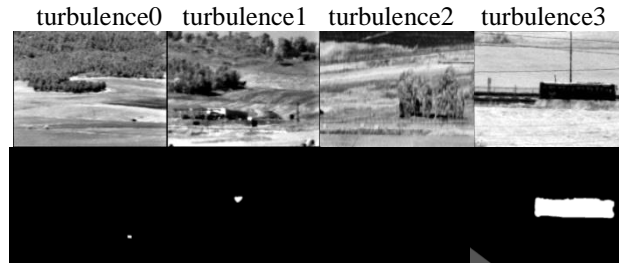
**Figure 1.5.** *The videos related to PTZ category*

The videos of *PTZ* category are given in Figure 1.5 as first and second rows denotes the original and ground truth samples, respectively. Since the camera captures from different angles and perspectives in *PTZ*, unsupervised methods are not able to overcome false positive rates, unless using a smart interference mechanism.

### ***Turbulence Category***

The turbulence category involves turbulence degradation in videos, which are given as *turbulence0*, *turbulence1*, *turbulence2* and *turbulence3*. The obtaining F-measure scores over 90% can be accepted as good performance for this category.

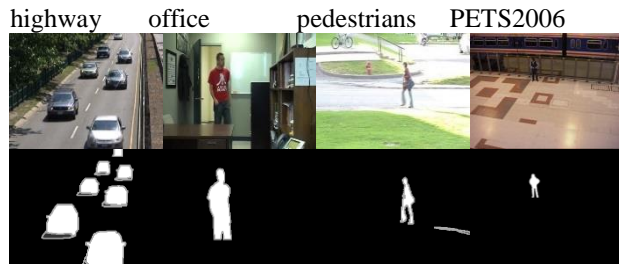
The samples included in *Turbulence* category are visualized in Figure 1.6, as the first and second rows shows the original and ground truth samples, respectively. Also, obtaining highest score for this category have usually foreseen as a cumbersome process.



**Figure 1.6.** *The videos related to Turbulence category*

### ***Baseline Category***

As its name suggests, the videos stated on “Baseline” category are more simple than other ones in terms of segmenting the regions of moving objects. The videos are summarized as *highway*, *office*, *pedestrians* and *PETS2006*.



**Figure 1.7.** *The videos related to Baseline category*

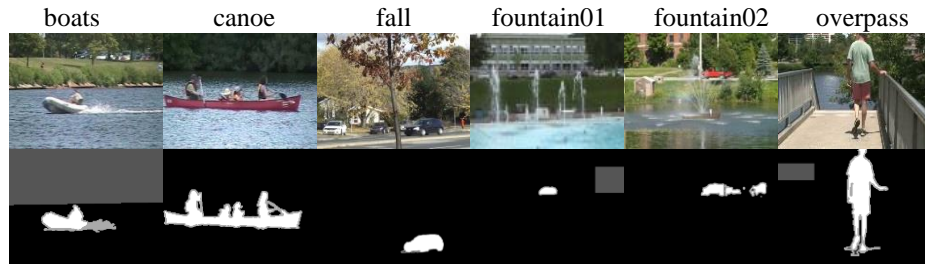
The Figure 1.7 shows samples related to the *Baseline Category* as first rows indicates the original samples and second row associated ground truth samples. After examining the performances presented in CDnet, one can say that acquiring highest F-measure score is possible with an unsupervised method.

### ***DynamicBackground Category***

This category contains dynamic occurrences in real time videos, which are *boats*, *canoe*, *fall*, *fountain01*, *fountain02* and *overpass*. There are some dynamics movements including waves, fountains, shaking of leaves in windy weather, which are reasons of false positives.

Again, representative samples (first row) for each dynamic video are presented in Figure 1.8 with their ground truth segmentations (second row). Among the videos stated on Figure 1.8, the boats video is more troublesome than other ones as it includes different conditions of sky. One can emphasize that a smart feedback is necessary for an

unsupervised method in order to overcome aforementioned dynamic problems in backgrounds.

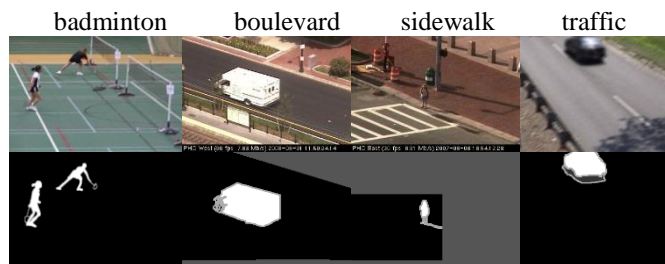


**Figure 1.8.** The videos related to the *DynamicBackground* category

### ***CameraJitter* Category**

This category includes high intense of camera movements for different videos, which are *badminton*, *boulevard*, *sidewalk* and *traffic*. To get highest score for this category, an unsupervised method has to be advocated by either using feedback mechanism or a smart validation operation.

The *CameraJitter* includes *badminton*, *boulevard*, *sidewalk* and *traffic* videos, where the human and car are focused points. Some samples are exhibited in Figure 1.9.



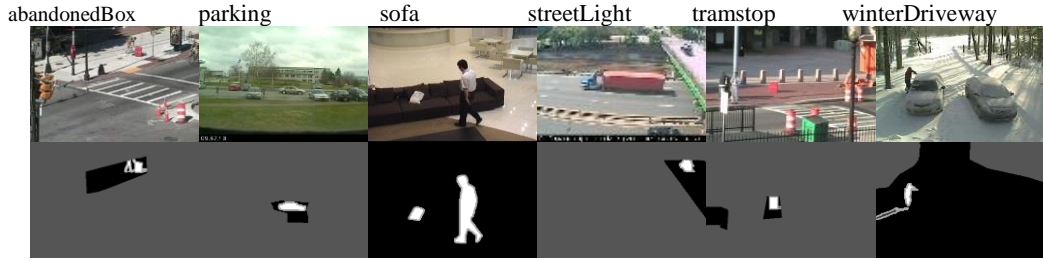
**Figure 1.9.** The videos related to the *CameraJitter* category

Since these videos suffered from effects of camera motion, a precaution is required in case of background modelling as allowing camera motion enters to background whereas revealing the regions of focused objects in foreground mask.

### ***IntermittentObjectMotion* Category**

This category includes the objects that are unwanted to enter the background frame and expected to disappear from background frame if their locations changed in time space, i.e., a parked car.





**Figure 1.10.** The videos related to *IntermittentObjectMotion* category

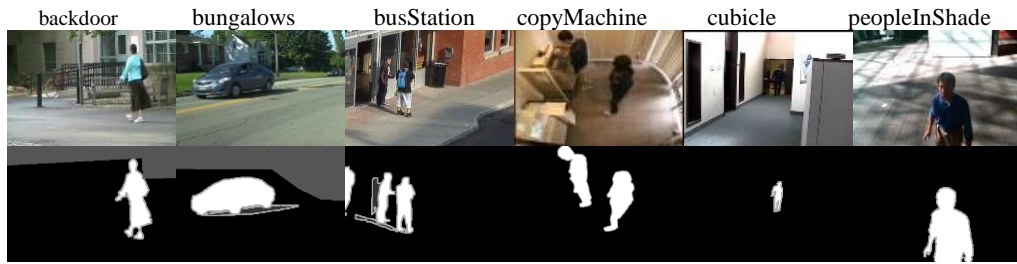
The videos are given as *abandonedBox*, *parking*, *sofa*, *streetLight*, *tramstop* and *winterDriveway*. Again, without some precaution measures, it is impossible to obtain satisfied performance for this category. When analyzing the performances of unsupervised methods, we can observe that employing a validation procedure would be give highest scores on this category.

The Figure 1.10 exhibits some original samples (first) and their manual generated ground truths (second row). Although the intermittent objects can be encountered on all video types, but the particular conditions like “stopped cars”, “left bags” and “stopped tramway” are focused when constituting this category. The backgrounds should be updated with a smart way that the regions of moved objects should be clean in background frame if their locations changed.

### ***Shadow Category***

From its name, one can deduce that the *Shadow* category includes the high intense of illumination changes and shadows. There are six videos listed as *backdoor*, *bungalows*, *busStation*, *copyMachine*, *cubicle* and *peopleInShade*. A scarified algorithm for background subtraction have to capable of adopting itself to sudden illumination changes in time domain. Once the performance of unsupervised methods analyzed, one conclude that it is possible to overcome illumination changes with a good accuracy.

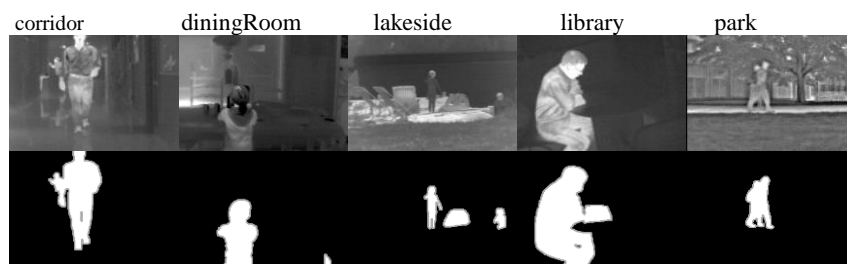
There are some representative samples derived from *Shadow* Category and presented in Figure 1.11 as first and second rows denotes samples and ground truths, respectively. From the Figure 1.11, we can see that there is an illumination change in corridor given of *cubicle* video. By using the fixed threshold, it is impossible to remove the illuminations stated on *Shadow* Category. Therefore, an adaptive thresholding with feedback mechanism is needed to reduce the false positives.



**Figure 1.11.** *The videos related to Shadow category*

### ***Thermal Category***

In this category, the thermal videos related to different scenes are presented with including thermal effects, intermittent objects motions, as well as illumination changes. There are five videos including *corridor*, *diningRoom*, *lakeSide*, *library* and *park*. This category includes some challenges that prevent to achieve nice results.



**Figure 1.12.** *The videos related to Thermal category*

The Figure 1.12 presents the thermal videos with some samples (first row) and their ground truths (second row). When analyzing the library video, achieving superior results for this category requires a great effort since there is a man sitting in the library until 3000 frames, which is called as intermittent object motions.

### **1.3. Thesis Organization**

The background modelling is a fundamental step for several real time computer vision applications including security systems and monitoring. An accurate background model facilitates segmentation of moving objects in a processed video. In this work, we have developed some new methods for moving object segmentation with an accurate process. To plainly explain each proposed method, this study is organized as follows.

In the section 1, we have presented the existing theories for background modelling. The idea under the traditional works are given and the improvements on previous

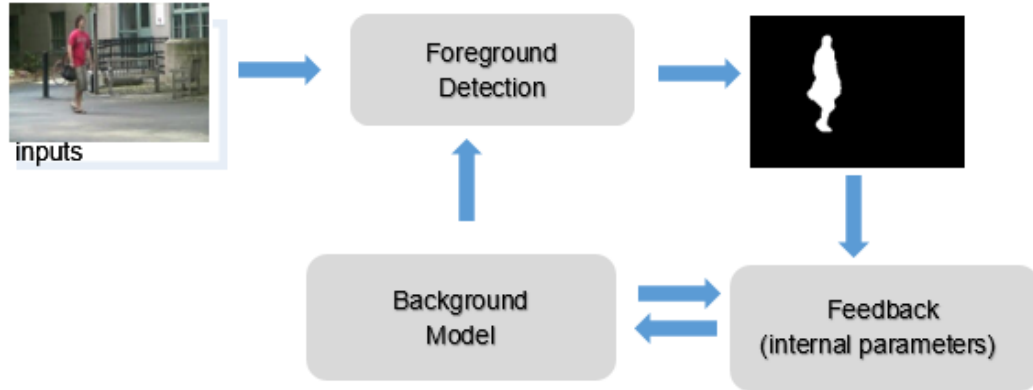
algorithms are touched by displaying the characteristics of recent methods. As a current trend, the studies for background modelling with deep learning strategies are explained together with different structures of deep neural network models.

The section 2 provides proposed methods to improve the background subtraction process. The reasons of developing such methods are also touched with more detail. Moreover, the contributions of utilized tools are expressed by concentrating on their theoretical aspects. Furthermore, the components of each method including background modelling, foreground detection, feedback mechanism and update of internal parameters are explained in the related places.

The section 3 shows performance of each proposed method by revealing their success on different challenges in terms of foreground segmentation. Moreover, the potential impacts of each method for different video types are exhibited throughout the subjective and objective performance evaluation stages. What makes the employed datasets are feasible to develop a background subtraction algorithm, is explained by exhibiting the included challenges. Additionally, the information about utilized metrics for performance evaluation are given for readers.

Finally, we have finalized the study with a concise conclusion. The computational time of proposed methods are emphasized in conclusion section.

## 2. PROPOSED METHODS ON IMPROVING BACKGROUND SUBTRACTION PERFORMANCE



**Figure 2.1.** Background subtraction process

As shown in the Figure 2.1, one can say that there are four components of a developed background subtraction system. These components can be given as *background modelling*, *foreground detection*, *update of background frames* and *update of internal parameters*. A simple explanation for each term is given as below.

**Step1: Background Modelling:** In case of initialization stage, the background can be modelled with two ways; (i) estimating a single background from a few of samples or (ii) forming a background memory (background list) by choosing N frames related to a video. Selecting the N frames can be done by taking first N frames, random N frames or predetermined N frames among all frames. Besides, the mean or median of N frames can be repeated N times in case of initialization stage. In case of first way (i), the aim is preserving a single background from leaking of foreground regions into background regions and maintaining a clean and meaningful background through video. On the other side, in the second way, it is allowed entering foreground regions into background based on the probability rate. Many studies have considered the second option for background modelling since the using N background frames is more convenient to alleviate the dynamic changes and illumination changes.

**Step2: Foreground Detection:** Foreground detection procedure is related to ways employed for moving object extraction. If one background model is considered, then only the difference between background frame and processed test frame are processed to reveal

the foreground regions. However, if there are  $N$  background frames on memory, then a consensus rule is taking into account. It means that a pixel has to be marked as foreground by the  $N$  frames, otherwise it would be background.

**Step3: Background Updating:** The background updating stage is about in which way the background frame and pixels in background frames should be updated in order to allow entering of illumination and dynamic changes into background. There two policies in background updating process while the first policy is about which background frame should be updated and the other policy deals with how to update pixels of the selected background frame.

**Step4: Updating Internal Parameters:** The other important point for a developed background subtraction framework, is controlling the utilized internal parameters including decision threshold and learning rate (probability rate) of updating process. Usually, a smart feedback mechanisms have utilized to monitor internal parameters.

In this study, we have proposed four systems to improve the performance of background process. In the method 1, the limitation of Common Vector Approach (CVA), which is relied on sub-space decomposition procedure, is evaluated for background modelling, called as BMCVA. In the method 2, again an extended version of CVA method, namely Common Matrix Approach (CMA) is applied for background modelling, called as BMCMA. In the method 3, a new distance metric (gradient transformation) together with feedback mechanism is contributed to background subtraction literature. The method 3 is named as Sliding Window based Change Detection (SWCD) and working in a pixel-wise manner. In the method 4, distance metric of SWCD is improved with CVA and also feedback mechanism of SWCD considered in internal stages, then the method 4 named as Common Vector Approach based Background Subtraction (CVABS). While in the BMCVA and BMCMA, a single background model, namely Common Vector or Common Matrix, was considered in case of the segmentation process through video, but in the SWCD and method CVABS, the first  $N$  background frames are hold in the memory in real time.

## 2.1. Method 1: Background Modelling Based on the Common Vector Approach (BMCVA)

CVA is a popular subspace based classification algorithm as applied for face recognition [31], spam classification [32], image denoising [33] and edge detection [34] tasks. The motivation of CVA is inspired from theory prompted in case of developing the PCA. While in PCA, the data is recovered by using eigenvectors corresponding to largest eigenvalues, but it has been emphasized that using null space of data gives more impressive accuracy in case of classification. Based on this fact, CVA algorithm has been put forward with a purpose of object classification by authors of study in [35]. Specifically, by using CVA algorithm, a frame is represented with two components, which are common and difference as shown in Eq. (2.1). There are two cases in CVA algorithm as sufficient and insufficient data cases. If the number of vectors is less than dimension of a vector, then it is called as insufficient data case, otherwise, it is assumed as sufficient data case. For this study, we have observed that the insufficient data case formed, since number of frames is less than the dimension of a frame. Assuming that we have given 35 frames and each frame is in the form of 256x256 (65536), then a matrix would be extracted as 65536x35 after the frames are converted into vector format and inserted to related columns in the matrix. The obtained matrix indicates that there is an insufficient data case. Thereby, common frame and difference frames can be calculated by using the Gram Schmidt procedure in case of insufficient data case.

In this study, the motivation under the CVA algorithm is adopted for background modelling [36]. The key point of algorithm is encapsulating background information of different frames in order to obtain a single and meaningful background frame. Similar to PCA, each frame is transferred to vector form.

Assuming that we have given  $n$  frames  $(a_1, a_2, \dots, a_n)$  and each frame is in the form of 1-D. With CVA algorithm, it has been validated that a given frame  $a_k$  can be separated into two parts as common and difference frame, which is denoted in Eq. (2.1).

$$a_k = a_{com} + a_{k,diff} , \quad (2.1)$$

where the  $a_{com}$  and  $a_{k,diff}$  refers to common and difference frames, respectively. In order to obtain orthogonal and orthonormal basis, the concept of Gram Schmidt is carried out on given vector set  $(a_1, a_2, \dots, a_n)$ . As a first stage, the selected reference frame is subtracted from remain vectors as shown in Eq. (2.2). In this study, the first frame ( $k = 1$ ) is considered as reference frame for the sake of simplicity. However, there is no restriction to take any frame as reference due to the characteristic of CVA method [37].

$$\begin{aligned} d_1 &= a_2 - a_1 \\ d_2 &= a_3 - a_1 \\ &\dots \\ d_{n-1} &= a_n - a_1 \end{aligned} \quad (2.2)$$

From the combination of difference vectors, a matrix  $M = \{d_1, d_2, \dots, d_{(n-1)}\}$  is obtained. The next stage is computing the orthonormal and orthogonal vectors with the idea of Gram-Schmidt procedure which is shown in Eq. (2.3) and Eq. (2.4).

$$v_1 = d_1 \text{ and } u_1 = \frac{v_1}{|v_1|} \quad (2.3)$$

$$v_k = d_i - \sum_{j=1}^{k-1} \langle d_k, u_j \rangle u_j \text{ and } u_k = \frac{v_k}{|v_k|} \quad k = 1, \dots, n-1, \quad (2.4)$$

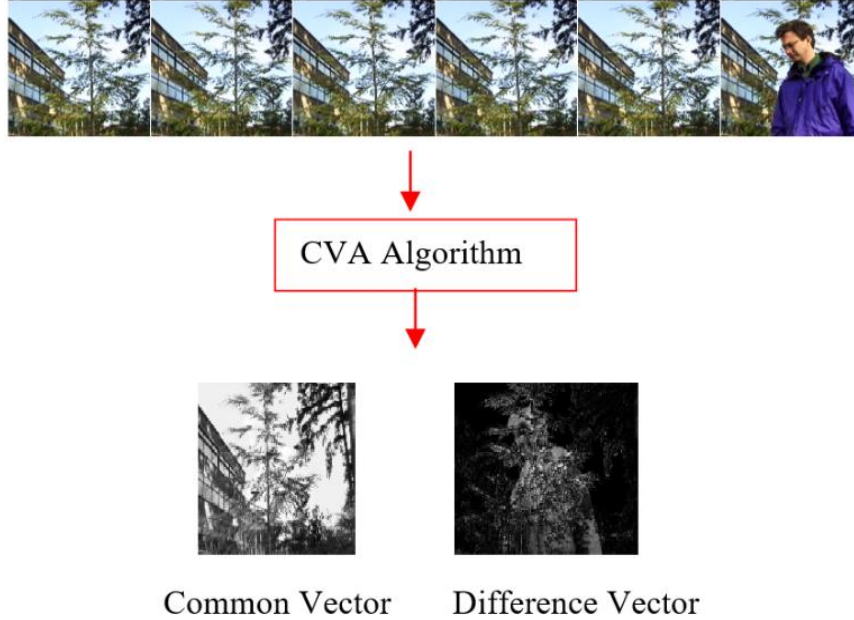
where  $\langle d_k, u_j \rangle$  refers to dot product of two vectors and  $|v_k|$  denotes the l-2 norm of each vector. Each vector is normalized by dividing with their l-2 norm. At the end of Gram-Schmidt orthogonalization procedure the  $(u_1, u_2, \dots, u_{(n-1)})$  orthonormal and orthogonal  $(v_1, v_2, \dots, v_{(n-1)})$  sets are obtained to yield difference frame.

Once the orthonormal sets are obtained, the difference frame is determined as given in the below formula. Specifically, the selected reference frame is projected on orthonormal vectors and summed up to obtain the difference frame. In this study, the first frame is taken as reference, and  $k = 1$ .

$$a_{k,diff} = \langle a_k, u_1 \rangle u_1 + \langle a_k, u_2 \rangle u_2 + \dots + \langle a_k, u_{(n-1)} \rangle u_{(n-1)} \quad (2.5)$$

Finally, the common vector  $a_{com}$  is derived by subtracting the  $a_{k,diff}$  from  $a_k$ .

$$a_{com} = a_k - a_{k,diff} \quad (2.6)$$



**Figure 2.2.** Overall principles of CVA based background modelling and foreground detection

As an improvement on the CVA method, a low noise value between 0-1 is injected to each difference subspace in Eq. 2.2 in terms of making high correlated data as low correlated form. The reason of making data low correlated is explained with idea that if the data is highly correlated then the rank becomes smaller than 2. As a result of small rank value, the obtained common vector does not become meaningful to eye. With this way, a background model with training data set is constructed as common frame refers to background frame and difference frame indicates foreground.

The motivation behind the CVA based background modelling is exhibited in Figure 2.2. Although, the CVA algorithm returns a Common Vector in the means of background model, but we have reshaped it as matrix in case of visualization as shown in Fig. 2.2. Also, though there are the N different difference vectors, but we have only exhibited the difference vector associated with the reference frame. As we can observe from the Figure 2.2, there are two components of a frame as:

- (1) first component provided the common frame of training set, which refers to obtained background model.



- (2) other component denotes the difference frame that exhibits details including moving objects and changes of training set.

From the Figure 2.2, the ability of CVA for change detection can be observed clearly. Therefore, one can deduce that the CVA algorithm can be utilized for background modelling and change detection. In case of foreground extraction, the common vector of processed test frame ( $t$ ) is computed by projecting the test frame onto the orthonormal basis generated by Gram-Schmidt procedure [34]. As a first stage, the difference vector corresponding to the test frame is obtained with Eq. (2.7).

$$t_{diff} = \langle t, u_1 \rangle u_1 + \langle t, u_2 \rangle u_2 + \dots + \langle t, u_{(n-1)} \rangle u_{(n-1)} \quad (2.7)$$

Once the difference vector is subtracted from the test vector, the common vector of processed test frame is determined as shown in Eq. (2.8).

$$t_{com} = t - t_{diff} \quad (2.8)$$

The difference between the two common vectors is considered in terms of observing the foreground regions.

$$\forall (i, j), I(i, j) = \begin{cases} 1 & \text{abs}(t_{com} - a_{com}) > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

As indicated in Eq. (2.9), if the absolute difference is greater than a fixed threshold value, then foreground mask is marked as 1, otherwise marked as 0. However, taking the absolute difference for *Moved Object*, *Light Switch*, *Camouflage videos*, produce a lot of erroneous pixels in foreground mask. To overcome this, only difference of two common vectors is put into the thresholding procedure. The utilized threshold value for each video are predetermined as follows; 0.1 for *Camouflage*, *Bootstrap*, *Light Switch*, *Waving Trees*, 0.2 for *Foreground Aperture* and 0.3 for *Time of Day* and *Moved Object* video, respectively.

After thresholding procedure, it has been observed that some morphological procedure is greatly required to obtain best results. For this purpose, firstly, a 5x5 median filter is applied on the binary foreground mask. The size of 5x5 filter is determined by considering the trade-off between performance and speed. Then, the connected components having size of less than 20, are considered as ghosts and ignored by applying the area open morphological operator. To close the holes in binary region, the morphological closing procedure is performed with disk structural element having size of

5 and binary holes are filled with morphological filling operator. As a last step, morphological opening with disk structural element having size of 5 is performed to mitigate the effect of closing operator.

## 2.2. Method 2: Background Modelling Based on the Common Matrix Approach (BMCMA)

The Common Matrix Approach (CMA) algorithm is an extended form of Common Vector Approach, which is also a subspace based method proposed for classification tasks. However, the ability of CMA for background modelling has not been realized in literature of computer vision. For this purpose, we have employed the CMA algorithm for background modelling [38]. In case of CVA the data is handled in vector format as a 2-D matrix is constituted from frames of training set and matrix decomposition strategy is applied, whereas for CMA, a tensor is generated from 2-D frames. The main idea behind the CMA is combining background information from different frames (matrices) and obtaining a single frame (common matrix), which envelopes cues about background locations.

Assuming that we have given  $n$  sample frames  $(S_1, S_2, \dots, S_n)$  and the each frame is in the 2-D form. In the context of CMA, a frame can be represented with common and difference frame as shown in Eq. (2.10).

$$S_k = S_{com} + S_{k,diff}, \quad (2.10)$$

where the  $S_{com}$  and  $S_{k,diff}$  refers to common and difference frames, respectively. To calculate, Common frame a tensor having 3-D structure is constructed and the concept of Gram Schmidt is applied to derive orthogonal and orthonormal basis. First of all, difference matrices are calculated by a taking a first frame as reference. Instead of first frame, a different frame can be chosen as reference.

$$\begin{aligned} D_1 &= S_2 - S_1 \\ D_2 &= S_3 - S_1 \\ &\dots \\ D_{n-1} &= S_n - S_1 \end{aligned} \quad (2.11)$$

Once a tensor  $T = \{D_1, D_2, \dots, D_{(n-1)}\}$  is obtained, the Gram-Schmidt procedure is activated on elements of T, which is shown in Eq. (2.12) and Eq. (2.13).

$$V_1 = D_1 \text{ and } U_1 = \frac{V_1}{|V_1|} \quad (2.12)$$

$$V_k = D_k - \sum_{j=1}^{k-1} \langle D_k, U_j \rangle U_j \text{ and } U_k = \frac{V_k}{|V_k|} \quad k=1, \dots, n-1, \quad (2.13)$$

where,  $\langle D_k, U_j \rangle$  indicates dot product of two vectors and  $|V_k|$  denotes the Frobenious norm of each matrix. Each of the orthogonal matrices  $V_i$  is divided by their Frobenious norm to make them normalized. After Gram-Schmidt orthogonalization procedure the orthogonal  $(V_1, V_2, \dots, V_{(n-1)})$  and  $(U_1, U_2, \dots, U_{(n-1)})$  orthonormal sets are extracted to compute difference matrix.

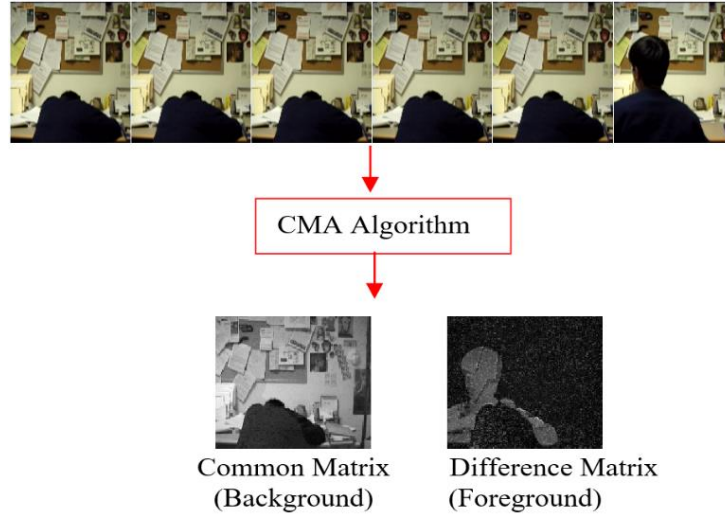
The next stage of CMA based background modelling algorithm is computing the difference and common matrices based upon orthonormal basis as given in the below equation.

$$S_{k,diff} = \langle S_k, U_1 \rangle U_1 + \langle S_k, U_2 \rangle U_2 + \dots + \langle S_k, U_{(n-1)} \rangle U_{(n-1)} \quad (2.14)$$

Finally, subtracting the  $S_{k,diff}$  from  $S_k$  gives difference matrix for class and  $S_{com}$  refers to common matrix of the class.

$$S_{com} = S_k - S_{k,diff} \quad (2.15)$$

With this way, the set of background frames can be represented by a unique 2-D frame, which is named as, common matrix. In other side, all details including noises and outliers of training set are stored in difference matrix  $S_{k,diff}$ .



**Figure 2.3.** Overall principles of CMA based background modelling and foreground detection

To obtain meaningful common matrix a low value of random noise is added to each difference subspaces obtained in Eq. (2.11). Since the rank of data becomes smaller than 2 in case of highly correlated data and results in not meaningful common matrix that is undistinguishable with human eye. To overcome this problem, a low noise value between 0-1 is injected to each difference subspace in Eq. (2.11) in terms of reducing the correlation ratio among the processed images.

From the Figure 2.3, we can observe that the decomposed tensor generates two components:

- (1) first component reserves the common matrix of training set, which is the obtained background model.
- (2) the other component involves the difference matrix that refers to detail features of training set.

By using the CMA, we can see that foreground and changes are observed in difference matrix. Therefore, the strategy behind CMA provides a new way to detect moving and stable objects in a given dataset. In order to reveal the foreground objects, the common matrix of test frame ( $F$ ) is determined from the projection of incoming test frame onto the orthonormal basis returned by Gram-Schmidt procedure [34]. First of all, the difference matrix related to the test frame is calculated as shown in below equation.

$$F_{diff} = \langle F, U_1 \rangle U_1 + \langle F, U_2 \rangle U_2 + \dots + \langle F, U_{(n-1)} \rangle U_{(n-1)} \quad (2.16)$$

Again, the common matrix corresponding to the test frame is computed by subtracting from the difference matrix.

$$F_{com} = F - F_{diff} \quad (2.17)$$

In case of revealing the foreground objects the difference between the common matrix of processed video and common matrix of processed frame is taken into account.

$$\forall (i, j), I(i, j) = \begin{cases} 1 & \text{abs}(F_{com} - S_{com}) > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \quad (2.18)$$

As shown in equation above, the difference of two common matrix presents foreground objects. In case of *Moved Object* and *Camouflage* videos, the difference of two common matrices are considered to find the foreground regions for other ones the absolute difference taken into account. The threshold value for each video are determined as follows; 0.1 for *Camouflage*, *Bootstrap* and *Waving Trees*, 0.2 for *Foreground Aperture*, *Light Switch* and *Moved Object*, and 0.3 for *Time of Day* video, respectively.

To obtain the pleasing visual results, some fixed morphological operations are applied on the foreground mask. Firstly, 5x5 median filter are utilized on the binary image. The connected components with the size of less than 20, are considered as ghost are removed by area open morphological operator. Then, the morphological closing procedure is utilized with disk structural element having size of 5 and binary holes are filled with morphological filling operator. Finally, morphological opening with disk structural element having size of 5 is performed to mitigate the effect of closing operator.

### **2.3. Method 3: A Sliding Window and Self-Regulated Learning Based Background Updating (SWCD)**

Technically, a change detection algorithm relies on two policies; while the first policy is about which background frame should be updated and the other policy deals with how to update pixels of the selected background frame. In the VIBE [21], PBAS [22] and SuBSENSE [23] algorithm, the update process partially relies on randomly selecting a frame among  $N$  background frames. Moreover, in referred studies, pixels of

selected background frame were updated with a neighbor rule where a subsampling factor ( $\phi$ ) was specified to choose the coordinate of a random pixel around the pixel to be updated.

**Table 2.1.** Notations for SWCD algorithm

$x$	denotes the position of image's pixel value
$T$	indicates the map of learning rate. As an accumulator, it counts number of foreground pixels over time. In this respect, it can be considered as histogram of foreground pixels. For a specific pixel, it is called as $T(x)$
$T_{lower}$	The scalar value for lower bound of $T$
$T_{upper}$	The scalar value for upper bound of $T$
$R$	is the decision threshold map, which contains the threshold information for each pixel value in the form of $R(x)$
$R_{lower}$	The scalar value for lower bound of $R$
$R_{upper}$	The scalar value for upper bound of $R$
$R_{scale}$	is used to monitor the decision threshold map ( $R$ )
$P$	indicates a scalar probability rate to update a pixel of background frame.
$B_i$	implies $i^{th}$ background sample.
$F_t$	shows the (binary output) foreground map at time $t$
$d_t$	shows the minimal distance map between current and background frame at time $t$ .
$d_{min,t}$	indicates the mean of minimal distance maps at time $t$
$\hat{d}_{min,t}$	is the normalized version of $d_{min,t}$
$v$	As an accumulator, it involves the histograms of blinking pixels.
$X_t$	is binary output of logical XOR operation between current and previous foreground maps at time $t$ .
$\#min$	is the minimum number to classify a processed pixel of current frame as background or foreground.
$E_r$	is binary edge map of reference frame over time domain.
$E_c$	is binary edge map of current frame over time domain.
$I_t$	is the current frame at time at time $t$ .

**Table 2.1 (Continued).** *Notations for SWCD algorithm*

$I_{gx}$	horizontal gradient magnitude of $I_t$
$I_{gy}$	vertical gradient magnitude of $I_t$
$\bar{B}_t$	implies the mean of background samples at time $t$ .
$\bar{B}_{gx}$	horizontal gradient magnitude of $\bar{B}_t$
$\bar{B}_{gy}$	vertical gradient magnitude of $\bar{B}_t$
$D$	represents a cross projection tensor term. In SWCD, $D_{11}$ , $D_{22}$ , $D_{12}$ denote cross-projection tensor terms.
$\bar{I}_t^m$	is the mean of short-term windowed gradient magnitudes at time $t$ , for PBAS.
$I_{gt,t}^m$	refers to gradient magnitude's transformation of current frame after applying the cross projection tensors at time $t$
$\bar{B}_{gt,t}^m$	refers to gradient magnitude's transformation of mean background frame after applying the cross projection tensors at time $t$

In the VIBE [21], it was clearly emphasized that tuning the parameter to  $\phi = 1$  is better for ghost suppression and that selection coincides with a diffusion rule updating strategy which refers to replacing the value of  $B_k(x)$  with  $I_t(x)$  based on a precomputed ratio. On the contrary, in SuBSENSE, selection of  $\phi = 16$  gives better results. The high value of  $\phi$  reduces the risk of false alarms, but increases the computational cost.

Although the EFIC [39] and FTSG [26] uses chamfer transformation for validation, they have certain drawbacks of foreground validation based on the chamfer matching, such as difficulties in adjusting the threshold value for chamfer matching and obtaining well-localized edge segments for all video types. Particularly, extraction of edges in low contrast images become a challenging bottleneck.

Considering the above-mentioned problems, we have proposed a new process for updating background frames. Contrary to VIBE [21], PBAS [22] and SuBSENSE [23] methods, we have observed that randomly updating the frames corresponding to the intermittent motion and dynamic videos is not a robust and effective way for preserving background frames through sudden and smooth changes. For this purpose, we have updated frames in a consecutive order over time. The introduced strategy relies on

sequential updating background frames, where the updating procedure is applied to the frames in a hierarchical order. Figure 2.5 shows an illustration of sliding window approach for updating the consecutive background frames. On given set, the pixels in the related background frame are updated according to a diffusion rule for the construction of a robust (but dynamic) background frame. Since we have utilized the dynamic update parameter,  $T$ , the foreground objects do not disappear (“eat-up”), thanks to the diffusion rule in updating strategy.

### 2.3.1. Foreground detection

Through this study, we explain foreground detection algorithm on gray-scale images. Therefore, if the image is a color image, first, it is converted to HSV domain, then the V channel is used. Considering the background pixel updating strategy, our segmentation mechanism is similar to the process realized in PBAS [22] and SUBSENSE [23] algorithms. The pixels of incoming test frame are compared with a background frame (from a list of background frames), based on a modified absolute distance (a special form of the L1 norm).

Let’s assume that we have  $N$  recently observed background frames, which are listed as  $B(x) = \{B_1(x), \dots, B_k(x), \dots, B_N(x)\}$ , where  $x$  corresponds to a pixel location. According to the computation/accuracy trade-off recommendations from SUBSENSE [23], we have set the  $N = 35$ . The reason of selecting a window size of 35 will be explained in the section of *implementation details*. At the beginning of the video, the background list is initialized with the first 35 consecutive frames. The L1 norm distance between a pixel of test frame and background frames can be used for categorizing a pixel as foreground or background:

$$F_t(x) = \begin{cases} 1 & \text{if } \# \left\{ \sum_{i=1}^N \text{dist}(I_t(x), B_i(x)) \geq R(x) \right\} > \#min \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

In this equation,  $I_t(x)$  refers to the processed pixel of the incoming test frame at time  $t$ .  $B_i(x)$  indicates a processed background’s pixel and  $R(x)$  shows the threshold value for decision making. The ‘#’ operator is a ‘count of’ operator. For each pixel, the



$R(x)$  is steadily updated to overcome dynamic changes in time domain. Here,  $\#min$  is the minimum count threshold to classify a pixel as foreground. In our algorithm, the threshold is selected according to  $N$  as  $\#min = N-1$ . Therefore, if the statistical count of a pixel is greater than  $N-1$ , then it is marked as foreground, else it is marked as background.

All of the described parameters and comparisons depend on a proper selection and definition of “distance”. For example, the distance measure in SUBSENSE [23] relies on Local Binary Similarity Patterns (LBSP) feature extraction parameters. It defines a threshold ratio  $Tr$ , and constructs a prior segmented map, called  $d_{Tr}$ :

$$d_{Tr}(x) = \begin{cases} 1 & \text{if } |I_t(x) - B_i(x)| \leq Tr \cdot I_t(x) \\ 0 & \text{otherwise} \end{cases} \quad (2.20)$$

The original segmentation in SUBSENSE [23] is constructed as a binary map that gets a value of 1 if the absolute difference between test and background pixels is greater than the threshold ratio ( $Tr$ ) of test pixel. The  $Tr$  value is initialized to 0.1 and increased gradually according to the existence of edges. The maximum value of  $Tr$  was specified as 0.3. Using this binary map, the distance is obtained as the 16-bit integer values LBSP between current and background frames, which is defined as:

$$dist(I_t(x), B_i(x)) = LBSP(I_t(x), B_i(x)) = \sum_{i=0}^{15} d_{Tr}(x) \cdot 2^i \quad (2.21)$$

As an alternative approach, PBAS [22] uses the L1 norm distance between test pixel and background pixel over three color channels. For each channel, the following rule was applied to calculate distance.

$$dist(I_t(x), B_i(x)) = \frac{\alpha}{\bar{I}_t^m(x)} \cdot |I_t^m(x) - B_i^m(x)| + |I_t(x) - B_i(x)|, \quad (2.22)$$

where the short-term averaged gradient,  $\bar{I}_t^m(x)$ , is as explained further paragraphs. Typically, the SOBEL filter was utilized for gradient map estimation. The incorporation of gradients together with the L1 norm help PBAS [22] to cope with shadow and

intermittent object motions. However, this distance metric does not produce satisfactory results in terms of change detection and the selection of  $\alpha$  is deterministic.

With a purpose of eliminating such deficiencies, we have introduced a new gradient distance metric with an inspiration from the work in [40]. The key assumption behind our approach is that the shadow and illuminations can be reduced with a gradient transformation procedure. Particularly, a ‘‘cross-projection tensors’’ based gradient is adopted in distance calculation, which is observed to help solving the ghost problem caused from intermittent object motions.




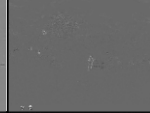


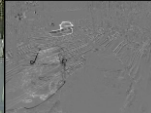




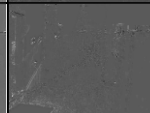


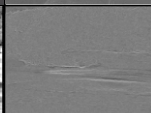
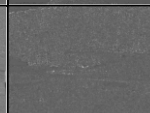
Let us assume that  $I_t$  is the ‘‘value’’ component of incoming HSV test frame and  $\bar{B}_t$  is the mean background derived from background list (i.e. the average of the list  $B = \{B_1, \dots, B_k, \dots, B_N\}$ ). First, the frames are smoothed with a simple 9x9 Gaussian filter ( $\sigma = 1.5$ ). Then, the horizontal ( $gx$ ) and vertical gradient ( $gy$ ) maps of  $I_t$  and  $\bar{B}_t$  for each frame are computed with SOBEL operator. Based on the procedures explained in [40], the cross diffusion tensor terms, called  $D_{11}$ ,  $D_{22}$  and  $D_{12}$  are obtained and applied to gradient magnitudes of  $I_t$  and  $\bar{B}_t$ . Then, gradient transformations of  $I_t$  and  $\bar{B}_t$  are computed separately as given in Eq. (2.23).

$$\begin{aligned}
I_{gx'}(\mathbf{x}) &= D_{11}(\mathbf{x}) \cdot I_{gx}(\mathbf{x}) + D_{12}(\mathbf{x}) \cdot I_{gy}(\mathbf{x}) \\
I_{gy'}(\mathbf{x}) &= D_{12}(\mathbf{x}) \cdot I_{gx}(\mathbf{x}) + D_{22}(\mathbf{x}) \cdot I_{gy}(\mathbf{x}) \\
I_{gt,t}^m(\mathbf{x}) &= \sqrt{I_{gx'}(\mathbf{x})^2 + I_{gy'}(\mathbf{x})^2} \\
\bar{B}_{gx'}(\mathbf{x}) &= D_{11}(\mathbf{x}) \cdot \bar{B}_{gx}(\mathbf{x}) + D_{12}(\mathbf{x}) \cdot \bar{B}_{gy}(\mathbf{x}) \\
\bar{B}_{gy'}(\mathbf{x}) &= D_{12}(\mathbf{x}) \cdot \bar{B}_{gx}(\mathbf{x}) + D_{22}(\mathbf{x}) \cdot \bar{B}_{gy}(\mathbf{x}) \\
\bar{B}_{gt,t}^m(\mathbf{x}) &= \sqrt{\bar{B}_{gx'}(\mathbf{x})^2 + \bar{B}_{gy'}(\mathbf{x})^2}
\end{aligned} \tag{2.23}$$

Finally, a noise free gradient distance map is recovered as  $|I_{gt,t}^m(\mathbf{x}) - \bar{B}_{gt,t}^m(\mathbf{x})|$  which is incorporated into the distance definition as:

$$dist(I_t(x), B_i(x)) = |I_{gt,t}^m(\mathbf{x}) - \bar{B}_{gt,t}^m(\mathbf{x})| + |I_t(x) - B_i(x)|. \tag{2.24}$$

Certain refinements were made in this new distance metric. For instance, in order for the gradient difference term to contribute, a condition on  $|I_t(x) - B_i(x)|$  value was imposed as  $|I_t(x) - B_i(x)|$  must be greater than 5. Hence, high gradient distance responses caused by complex edge regions were suppressed.

	background	input	Grad-Diff SOBEL	Grad-Diff SWCD
parking #0001 #1890				
winterDriveway #0001 #2000				
cubicle #2700 #4774				
turbulence0 #1105 #2728				

**Figure 2.4.** Visual demonstration of validated foreground maps

Some visual gradient results are presented in Figure 2.4. The second and third columns indicate “background” and “input” frames, while the fourth and fifth columns refer to pure gradient difference (Grad-Diff) returned from SOBEL and the proposed gradient difference used in SWCD [41], respectively. Experimental validation results show that the utilized validation procedure gives plausible performance on CDnet 2014 dataset.

Since we have utilized an edge suppression based gradient estimation procedure, edges become recessive in the obtained gradient maps. Consequently, the method successfully overcomes the challenges of intermittent object motion in *winterDriveway* and *parking* videos. Moreover, a clean background model is estimated in case of *parking* and *winterDriveway* videos, even after the parked car and stationary car moves and displace their locations. It is concluded that the validation procedure with cross-projection tensors successfully eliminate shadows, reflections and abrupt illumination changes, which are common problems for all background subtraction methodologies. However, it must be noted that the overall performance of the proposed method does not solely rely on the above distance computation.

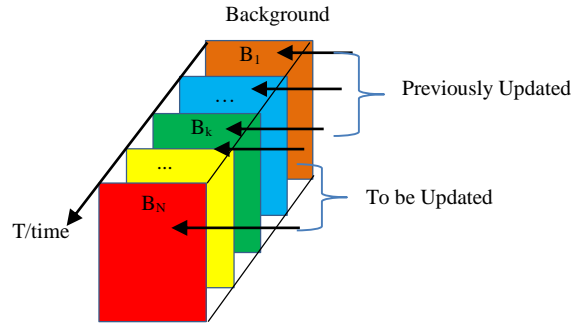
### 2.3.2. Updating background frames

The performance of a background modelling algorithm significantly improves under low or no noise conditions. Therefore, noise effects have to be minimized by carefully setting the parameters. Then, as widely used, updating decision threshold ( $R$ ) and learning rate ( $T$ ) of a background subtraction algorithm are two important parameters for adaptation to gradual changes in test frames. There are two policies related to the background updating procedure over time; selection policy and update policy. The selection policy deals with (i) “which background frame should be selected to update among the  $N$  background frames?” and (ii) “which pixel of selected background frame is needed to update when a processed pixel of current frame is classified as background or foreground?”. Conversely, the update policy seeks the answer to (iii) “which way should the processed pixel of selected background frame be updated?”.

The selection procedure of VIBE [21], PBAS [22] and SuBSENSE [23] works by randomly selecting a frame among  $N$  background frames in case of selecting background frame. Moreover, a random neighboring rule was considered in order to select pixel to be updated. In other words, a random index among the surrounding pixels of processed pixel was selected in case of updating stage. Again, the strategies to update the picked pixel among the surrounding pixels are classified into two policies: (i) *conservative*, and (ii) *blind* [21]. In conservative update policy, only the pixels marked as background in the current frame are considered to be updated. The adverse points of *conservative* policy are the deadlocks and ghost artifacts (which form due to intermittent object changes). Therefore, an alternative solution is devised, known as the *blind* policy. The blind policy is less sensitive to deadlocks and ghosts since the pixel of background frame corresponding to current frame is updated regardless of considering whether its label is foreground or background. The updating strategy of VIBE [21], PBAS [22] and SUBSENSE [23] relies on the *conservative* policy as only the pixels marked as *background* are updated.

Contrary to VIBE [21], PBAS [22] and SuBSENSE [23], we have carried out a sliding window approach while selecting a background frame among the set of background frames. With an iterative selecting rule, background frames are selected in a sequential order. Furthermore, a probabilistic *diffusion* rule is applied where a current frame’s pixel

replaces one of the background images pixel at the same location according to a probability rate of  $p$ . An advantage of *diffusion* rule is that extra efforts aren't required to select a random pixel among neighboring pixel of a processed pixel. On top of these, unlike to VIBE [21], PBAS [22] and SuBSENSE [23], a *blind* update policy is applied in our method in case of updating selected pixel of background frame.



**Figure 2.5.** An illustration of sliding window based background updating procedure

Similar to PBAS [22] and SuBSENSE [23], an accumulator ( $T$ ) is considered to keep the histogram of foreground pixels over time in our algorithm. Then, the updating probability rate (which is approximately  $p \approx 1/T(x)$ ) is calculated for each pixel in case of *blind* updating stage. From the formulation, we can see that the probability rate refers to inverse of histogram as ( $T^{-1}$ ). One can notice that the value of probability rate becomes close to zero in case of static foreground regions (steadily marked as foreground over time – please refer to static scenes of standard videos: *copyMachine*, *library* and *office*). On the other hand, for non-stable foreground regions, the probability rate becomes high. This desirable property helps our method to successfully identify foreground and background regions.

Based on the probability rate,  $p$ , the background frames are updated with a sliding window selecting policy and *blind* update policy as presented in Eq. (2.25)

$$B_i(x) = (1 - p) \cdot B_i(x) + p \cdot I_t(x) \quad (2.25)$$

In Eq. (2.25),  $i$  starts from 1 and gets incremented until  $N^{th}$  frame.  $I_t(x)$  refers to the processed pixel of the incoming test frame at time  $t$ .  $B_i(x)$  indicates a processed background's pixel. In Figure 2.5, the update mechanism of SWCD algorithm is

illustrated as a sliding window process, where background frames are selected and updated consecutively. However, the sliding window indexing is circularly stacked: as the index overshoots  $N$ , the list index goes back to 1.

A disadvantage of the previous *conservative* policy, as already indicated above, is that *only the pixels marked as background are updated* in VIBE [21], PBAS [22] and SuBSENSE [23], and this leads to formation of ghost objects caused by intermittent object motions. In the proposed SWCD algorithm, in order to avoid false alarms, *both foreground and background pixels are updated* regardless of checking the classification of the pixel, using similar dynamic control parameters.

### 2.3.3. Updating internal parameters

#### Updating the Decision Threshold $R(x)$

The performance of background subtraction algorithm immediately depends on a good selection of threshold value. In order to reduce the number of false alarms while maintaining acceptable accuracy rates, an adaptive threshold is required. Therefore, instead of using a user-defined and fixed threshold, many adaptive threshold attempts have been made in the literature to cope with camera jitter, sudden illumination and dynamic changes [21-23, 42, 43]. Many methods prefer a constantly updated threshold map after their foreground segmentation stage. For example, a history of minimal decisions was kept over the time as  $d(x) = \{d_1(x), \dots, d_k(x), \dots, d_N(x)\}$  in the study of PBAS [22] and SUBSENSE [23]. Here, “minimal” decision refers to minimum absolute difference between the pixel of test frame and background frames, computed as:  $d_i(x) = \min\{dist(I_i(x), B_i(x))\}$ . The average value of minimal decision list represents the dynamic changes in time domain and will be called the dynamic’s control parameter,  $d_{min}(x)$ , evaluated around the current time,  $t$ , as  $d_{min,t}(x) = \text{mean}\{d_i(x), \dots, d_{t-n+1}(x)\}$ . For a static region,  $d_{min}(x)$  would be small and for a dynamic region, that value would be high. Using this idea, we have observed that the minimal distance between current and background frames can be simultaneously measured in order to gauge the background dynamics. Once  $d_{min}$  is computed, the threshold map,  $R$ , is increased or decreased based on the rule given PBAS [22] algorithm. In SWCD, for a better performance, only five

historical minimal distance ( $n = 5$ ) maps are held during the change detection procedure and the average of these minimal distances is represented as  $d_{min}(x)$  for each pixel. The decision threshold parameter is instantly updated with respect to  $d_{min}(x)$  in each frame as shown in Eq. (2.26).

$$R(x) = \begin{cases} R(x) + R_{inc/dec} \cdot R(x) & \text{if } R(x) < (d_{min}(x) \cdot R_{scale}) \\ R(x) - R_{inc/dec} \cdot R(x) & \text{otherwise} \end{cases}, \quad (2.26)$$

where  $R_{inc/dec}$  denotes a steering coefficient, which was set to 0.01 in our experiments. The  $R_{scale}$  parameter depends on the complexity of video type and typically attains one of the three values: 0.1, 1, and 2. It was experimentally observed that for static and noise free videos,  $R_{scale}$  should be selected as 0.1 and for complex videos, a value of 1 or 2 should be selected. Specifically, for complex outdoor videos, selecting  $R_{scale} = 2$  causes more effective results. Also, the interval of  $R(x)$  is restricted to  $R_{lower} \leq R(x) \leq R_{upper}$ . Similarly, the upper value of  $R(x)$ ,  $R_{upper}$ , is bounded as  $R_{upper} = \infty$ . Initially,  $R(x)$  is set by the user-defined threshold  $R_{lower}$  (which has a value of 35). The latter  $R(x)$  values are always greater than or equal to  $R_{lower}$ .

### Updating the Learning Rate $T(x)$

The activity of the foreground and background greatly varies according to the selected video; in some cases, the background present static behaviors, while in other cases, it exhibits dynamic movements. For example, the places of some objects in videos do not change for a long time (say, 1000 - 3000 frames), and then these objects just move (even causing an illusion to the human eye). To prevent such objects from entering the background class, it is necessary to increase the corresponding foreground probability values for lower false alarm rates. A parameter that needs updating is the histogram of foreground pixels:  $T$ . Similar to SUBSENSE [23] method, we have updated the  $T$  parameter using  $v$  and  $\hat{d}_{min}$  parameters, which are called as dynamic's controllers, where  $\hat{d}_{min}$  corresponds to the normalized version of  $d_{min}$  (normalized to [0-1] interval):

$$T(x) = \begin{cases} T(x) + \frac{1}{v(x) \cdot \hat{d}_{min}(x) + 1} & \text{if } F_t(x) = 1 \\ T(x) - \frac{v(x) + 0.1}{\hat{d}_{min}(x) + 1} & \text{if } F_t(x) = 0 \end{cases} \quad (2.27)$$

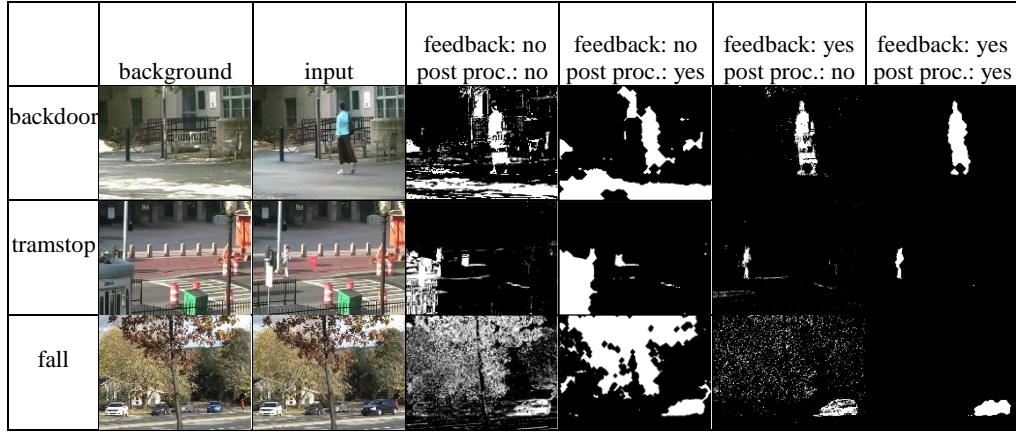
In Eq. (2.27), the  $v(x)$  parameter takes care of pixels that are alternately marked as foreground and background in time (causing a blinking in foreground tagging matrix, which is a binary frame). This  $v(x)$  parameter is, therefore, an accumulator (defined in Eq. (2.28)) to measure the statistical information related to constantly changing pixels. Its value is always positive, and for static regions,  $v(x)$  converges to 0. Since the parameter is in the denominator, in order to avoid division by zero, an offset of 1 is added to denominator in the given equation. In the second condition line of the same equation,  $\left(\frac{v(x) + 0.1}{\hat{d}_{min}(x) + 1}\right)$  is subtracted from current  $T(x)$  to increase the update probability rate of slowly changing pixels. Based on the segmented foreground pixel ( $F_t(x)$ ), the associated threshold value ( $T(x)$ ) is increased or decreased with dynamic control parameters over time.

In the proposed algorithm, a rule given in Eq. (2.28) is pursued to monitor the  $v(x)$  parameter.

$$v(x) = \begin{cases} v(x) + v_{incr} & \text{if } X_t(x) = 1 \\ v(x) - v_{decr} & \text{if } X_t(x) = 0 \end{cases}, \quad (2.28)$$

where  $X_t(x)$  refers to a flag (i.e. an XOR operation) indicating that the location  $x$  contains alteration for two consecutive (binary) foreground tagging matrices,  $F_t(x)$  and  $F_{t-1}(x)$ . As a final retouch, the areas of  $X_t$  intersected with  $F_t$  are set to 0 in order to avoid leaking of foreground's borders into the background frame. In Eq. (2.28), the increment (punishment) and decrement parameters were specified as  $v_{incr} = 1$  and  $v_{decr} = 0.1$ , respectively, so that camera jitters, waving trees and water waves could be collected in  $v$  map by punishing the related coordinates with 1. The learning rate,  $T$  in Eq. (2.27) is finally restricted as  $T_{lower} \leq T(x) \leq T_{upper}$ .





**Figure 2.6.** The visual performance with or without dynamic control parameters and post processing

The consensus of several studies show that selecting a lower bound as  $T_{lower} = 2$  is sufficient for reasonable performance. Unfortunately, there are different proposals for the upper value,  $T_{upper}$ . While PBAS [22] proposes  $T_{upper} = 200$ , SUBSENSE method [23] makes experiments with  $T_{upper} = 256$ . In this study, we have observed that the upper value should be made as high as possible for real time applications because a wrongly classified foreground pixel should never be let into the background frame – not even after 200 or 256 frames. Consequently, we take  $T_{upper} = \infty$ .

As seen from Figure 2.6 and due to other known examples, it is quite difficult to cope with sudden illumination changes, intermittent object motion and waving trees without using a feedback mechanism based on the dynamic control parameters. The 3<sup>rd</sup> and 4<sup>th</sup> columns in this figure show the results without feedback. As expected, many misclassified pixels inevitably appear, although post-processing somewhat improves the foreground detection. On the other hand, feedback by itself is observed to be greatly improving the foreground segmentation (column 5) except for rapidly changing backgrounds (i.e. waving tree leaves in the last row). Finally, as illustrated in the last column, if feedback and post processing scenarios are engaged together, then remarkable performance is achieved for almost all difficult cases.

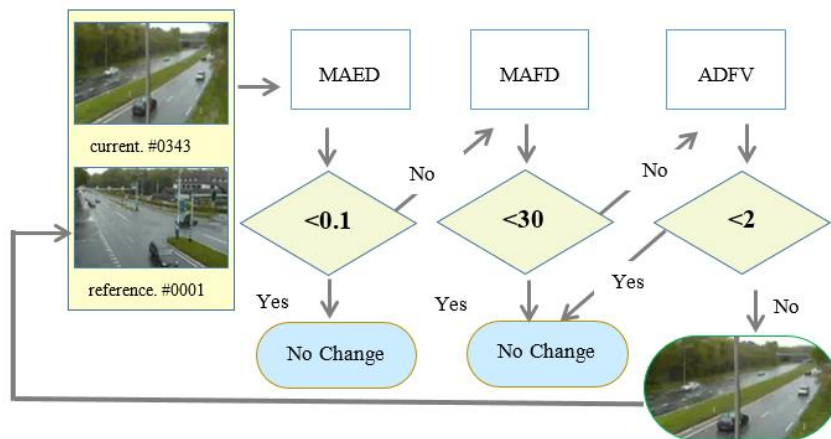
In these experiments, the post processing step consisted of sequentially applying median filter (to remove salt-pepper type noise for isolated chunks smaller than 10

pixels), then opening and closing morphological operations (with diamond-shaped structuring elements having radius size 10).

### 2.3.4. Handling PTZ challenges

PTZ videos have their unique challenges as compared to other video types. Furthermore, foreground detection is mostly applied to these types of videos. Therefore, a precaution is required in order to find the changes in the view points of camera, particularly for *twoPositionPTZCam* and *intermittentPan* videos. For this purpose, we have utilized a fast pixel based scene change detection algorithm that was inspired from the study in [44].

In the referred study on scene change detection, the abrupt scene changes were found with a low computational complexity procedure which depends on statistical measures between processed frames. When a scene change is detected, the related background is totally replaced with the new scene. By combining this scene change detection strategy with the above foreground extraction method, plausible F-score values were achieved in *PTZ* camera videos. The proposed scene change detection depends on serially checking variations between two consecutive frames by Mean Absolute Edge Difference (MAED), Mean Absolute Frame Difference (MAFD) and Absolute Difference Frame Variance (ADFV).



**Figure 2.7.** Scene change detection mechanism for PTZ videos

In Figure 2.7, the reference and current frames are taken as input to the process chain for scene change detection. First, a histogram equalization is applied on the frames

for normalization purposes. Then MAED, MAFD and ADFV measures are quickly and sequentially computed. The algorithm resets the background if ALL of the variation measures agree that a scene change has occurred. Otherwise, a scene change is not decided and the foreground/background detection process continues.

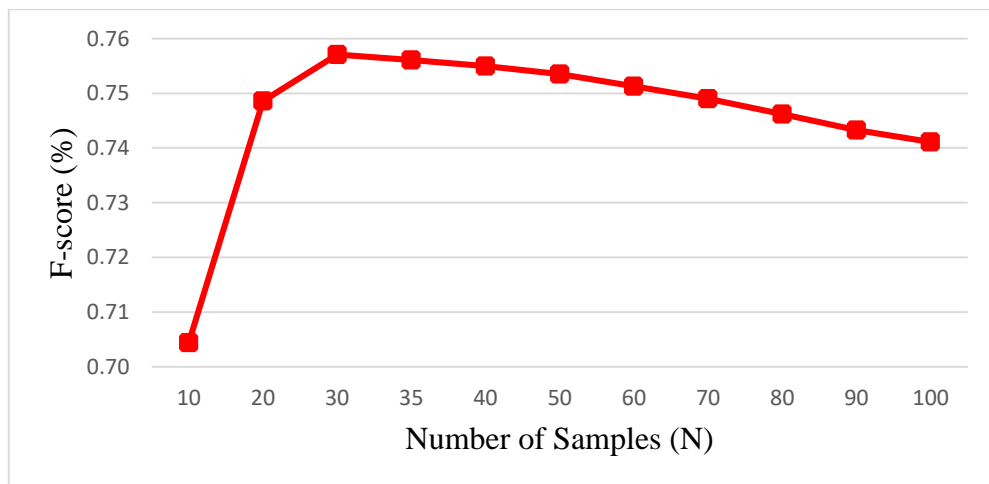
Detailed information about computation of MAFD and ADFV can be found in [44]. However, in this series of checks, MAED is proposed by our study and utilized for the first time in scene change detection, and is defined as

$$MAED = \frac{1}{H \cdot W} \sum_{i=1}^H \sum_{j=1}^W |E_r(x) - E_c(x)|, \quad (2.29)$$

where  $E_r$  and  $E_c$  refer to edge maps obtained from reference and current frames (using a simple Sobel operator), respectively.  $H$  and  $W$  denote frame dimensions.

### 2.3.5. Implementation details

In order to achieve a reasonable background subtraction performance, certain pre- and post-processing steps are typically applied in many studies. For instance, downsampling of video frames improve performance by reducing the noisy regions due to camera jitters, waving trees and reflectance.



**Figure 2.8.** Performance analysis of different number of samples ( $N$ ) after conducting experiments on CDnet dataset

Besides, as emphasized in SuBSENSE [23] algorithm, a downsampling (by a factor of 2) naturally increases the speed. However, such a downsampling was NOT applied in

our tests to see and show the exact performance comparison under real conditions. Instead, a post processing phase (consisting of simple median filtering, opening and closing) is applied for cleaning the foreground and background parts.

The number of background samples was determined as  $N = 35$ . This stack size was decided according to experiments on CDnet 2014 dataset. The value of  $N$  was gradually increased from 10 to 100 and the F-scores were analyzed for each experiment. Figure 2.8 roughly shows results in increments of 5. For instance, an F-score of 0.7521 was obtained at  $N = 30$ . In order to be compatible with the literature and since there is no significant degradation at  $N = 35$ , the value of 35 was chosen.

#### **2.4. Method 4: Moving Object Segmentation with Common Vector Approach (CVABS)**

CVA is a subspace based recognition method that gives satisfactory invoice results [45], face [46], spam e-mail [47], edge detection [48] and classification tasks. Essentially, the derivation of CVA comes from the idea behind the Principal Component Analysis (PCA). While in PCA the projection is taken onto the eigenvector corresponding to the largest eigenvalues, in the CVA method this procedure is carried out in the opposite direction by projecting the data onto the eigenvector associated with the smallest eigenvalues. The advantage of CVA over some subspace based classification methods is the derivation of a solution for an insufficient data case, which occurs when the dimension of vectors is greater than the number of vectors [37]. On the other hand, computing the inverse within the class matrix with Fisher Linear Discriminant Analysis (FLDA) method for an insufficient data case is impossible.

Let suppose that we give the  $k$  samples that correspond to the  $i$ -th class (different sequential views of the same scene,  $\{\mathbf{a}_j^i\} \ j = 1, 2, \dots, k$ ). Now, refer to a particular class and drop the superscript  $i$ . It is possible to represent each  $\mathbf{a}_j$  vector as the sum of  $\mathbf{a}_j = \mathbf{a}_{i,com} + \mathbf{a}_{i,diff}$ . A common vector ( $\mathbf{a}_{i,com}$ ) is what is left when the difference vectors are removed from class members and is invariant throughout the class, whereas  $\mathbf{a}_{i,diff}$  is called the remaining vector, which represents the particular trend of this particular sample. There are two cases in CVA where the number of vectors is either sufficient or

insufficient. In this study, we have focused on the insufficient data case since the frames are handled as a vector format.

For an insufficient data case, it has been stated that the common vector can be obtained based on the covariance matrix and the Gram–Schmidt orthogonalization process. With respect to this concept, the common vector can be derived by obeying the following rules.

Let us suppose that the training set has  $k$  samples  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k)$  corresponding to the  $i$ -th class in  $\mathbf{R}^k$ . Also, each sample has a  $h \times w$  dimension. To find a common vector for any class, we should construct a matrix from these samples. Hence, a column matrix with the  $(h.w) \times k$  dimension is obtained from the given  $k$  samples. Our aim is to project the column matrix into the 1-D space (vector), by preserving the global information. To understand this, the algorithm described below should be followed based on the rules given in the related work on CVA [45].

- Firstly, a random vector, i.e.,  $\mathbf{a}_1$  is taken as a reference, then the difference vector which belongs to processed data is obtained by:

$$\mathbf{d}_{j-1} = \mathbf{a}_j - \mathbf{a}_1 \quad j = 2, 3, \dots, k \quad (2.30)$$

- Once the  $(k-1)$  difference vector is obtained, the difference subspace (**DS**) for  $i$ -th class can be calculated by gathering the difference vectors.

$$\mathbf{DS}_i = \{\mathbf{d}_j, \mathbf{d}_{j+1}, \dots, \mathbf{d}_{k-1}\} \quad j = 1, 2, \dots, k-1 \quad (2.31)$$

- In the next stage, the Gram-Schmidt orthogonalization procedure is applied to obtain the orthonormal basis,  $(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$  which spans the difference subspace and orthogonalizes the difference vectors of the  $i$ -th class. The obtained orthogonal vector is divided by their Frobenious Norm's to make them normalized, called the orthonormal basis.
- In  $\mathbf{R}^{k-1}$ , the orthonormal basis  $(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$  and orthogonal vectors  $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{(k-1)})$  of the plane with  $k$  dimension are computed with the following formulas.

$$\begin{aligned}
\mathbf{v}_1 &= \mathbf{d}_1 \text{ and } \mathbf{z}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \\
\mathbf{v}_2 &= \mathbf{d}_2 - \langle \mathbf{d}_2, \mathbf{z}_2 \rangle \mathbf{z}_2 \text{ and } \mathbf{z}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} \\
\mathbf{v}_3 &= \mathbf{d}_3 - \langle \mathbf{d}_3, \mathbf{z}_2 \rangle \mathbf{z}_2 - \langle \mathbf{d}_3, \mathbf{z}_1 \rangle \mathbf{z}_1 \text{ and } \mathbf{z}_3 = \frac{\mathbf{v}_3}{\|\mathbf{v}_3\|} \\
\mathbf{v}_j &= \mathbf{d}_j - \sum_{j=1}^{j-1} \langle \mathbf{d}_j, \mathbf{z}_{j-1} \rangle \mathbf{z}_{j-1} \text{ and } \mathbf{z}_j = \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|} \quad j = 2, \dots, k
\end{aligned} \tag{2.32}$$

$(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$  refers to an orthonormal basis for  $\mathbf{R}^{k-1}$  and for a given plane, respectively. Also,  $\langle \cdot, \cdot \rangle$  implies the dot product of the given vectors and  $\|\cdot\|$  denotes the Frobenious Norm of the vectors. Again  $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{(k-1)})$  refers to orthogonal vectors for a given plane and for  $\mathbf{R}^{k-1}$ , respectively.

- Once the orthogonal and orthonormal basis are computed, the difference vectors  $\mathbf{a}_{i,\text{diff}}$  can be obtained by the projection of any sample  $\mathbf{a}_j$  from the  $i$ -th class on the difference subspace of a class which is spanned by a orthonormal basis  $(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$ .

$$\mathbf{a}_{i,\text{diff}} = \langle \mathbf{a}_1, \mathbf{z}_2 \rangle \mathbf{z}_2 + \langle \mathbf{a}_2, \mathbf{z}_3 \rangle \mathbf{z}_3 + \dots + \langle \mathbf{a}_k, \mathbf{z}_{k-1} \rangle \mathbf{z}_{k-1} \tag{2.33}$$

- Finally, as shown in Eq. (2.34), subtracting the  $\mathbf{a}_{i,\text{diff}}$  from any vector  $\mathbf{a}_j$ , gives a common vector of the  $i$ -th class. Practically, any sample among the  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k)$  can be used as a reference. By considering the given form of  $\mathbf{a}_j = \mathbf{a}_{i,\text{com}} + \mathbf{a}_{i,\text{diff}}$ , the common vector can be formulized as;

$$\begin{aligned}
\mathbf{a}_{i,\text{com}} &= \mathbf{a}_j - \mathbf{a}_{i,\text{diff}} \\
\mathbf{a}_{i,\text{com}} &= \mathbf{a}_j - (\langle \mathbf{a}_j, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_j, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \mathbf{a}_j, \mathbf{z}_{k-1} \rangle \mathbf{z}_{k-1})
\end{aligned} \tag{2.34}$$

$\mathbf{a}_{i,\text{com}}$  refers to a common matrix of the  $i$ -th class. Thus, a class with several samples can be represented by a unique subspace called a common vector. To summarize, the projection of vectors established from each sample of a class onto an orthonormal basis

gives the difference vectors. If the difference vectors are subtracted from the reference vector, the common vector of the processed class is acquired. In other words, the projection of the matrix in a difference subspace of a processed class, generates the common vector.

#### 2.4.1. Application to background modelling

It can be recalled that our objective is to combine the common characteristic stated in different sequential views with a single view that reserves the rich information about the background. Using the CVA algorithm to represent the different views with a common one is a similar procedure to the work of Oliver et. al, called Subspace Learning based on the PCA algorithm (SL-PCA). When cross-referenced to the study, the eigenspace model generated from the PCA decomposition is utilized by considering the fact that moving objects do not appear in static regions which are the contributions of moving objects to the eigenspace model and are very small and can even be negligible. With this concept, the difference between the mean background ( $\mathbf{u}$ ) and the column representation of each input image ( $\mathbf{I}_t$ ) was projected onto the  $h$  dimensional eigenbackground subspace, ( $\Phi_h$ ), which consists of the eigenvectors associated to the largest eigenvalues of a column representation of the  $k$  frames, denoted with  $\mathbf{B}_t$ . In the following step, the ( $\mathbf{I}_t$ ) has been reconstructed to represent the background model ( $\mathbf{I}_t'$ ) as shown in Eq. (2.35).

$$\mathbf{B}_t = \Phi_h(\mathbf{I}_t - \mathbf{u}) \quad (2.35)$$

$$\mathbf{I}_t' = \Phi_h^T \mathbf{B}_t + \mathbf{u} \quad (2.36)$$

Finally, those foreground pixels related to a moving object are detected by considering the distance between the input ( $\mathbf{I}_t$ ) and the reconstructed background ( $\mathbf{I}_t'$ ) frames regarding the predefined threshold  $T$  as denoted with the rule below;

$$\mathbf{E}_t(\mathbf{i}, \mathbf{j}) = \begin{cases} 1 & \text{if } dist(\mathbf{I}_t(\mathbf{i}, \mathbf{j}), \mathbf{I}_t'(\mathbf{i}, \mathbf{j})) > Threshold \\ 0 & \text{else} \end{cases} \quad (2.37)$$

The procedure stated for the SL-PCA is the inspiration for our study in background subtraction. Specifically, the common vector of a column representation of the  $k$  frames can be obtained either by using the eigenspace model that consists of eigenvectors

corresponding to the smallest eigenvalues, or by obtaining the orthonormal vectors of the processed data with the Gram-Schmid Orthogonalization procedure in the case of an insufficient data case. Since obtaining the eigenvector for a large dimension of data requires an enormous memory, we have concentrated on the Gram-Schmid Orthogonalization procedure instead of deriving the orthonormal vectors.

Although using the CVA algorithm for background modelling gives good results in the case of low correlated data ranked greater than 2, it has been observed, however, that the CVA concept collapses in the case of high correlated data when ranked at about 2. For example, it is favourable to obtain a nice common vector (background model) in the case of ‘*backdoor*’, but it is obvious that the non-meaningful common vector derived from the ‘*copyMachine*’, ‘*office*’ and ‘*library*’ video set in which the sequential frames are too similar to each other. With this, the data range of the common vector becomes different by 0-255. It results in a segmentation problem where difference between test and the common vector does not reveal the accurate foreground regions which can be observed from Figure 2.9.

	<i>backdoor</i>	<i>copyMachine</i>	<i>traffic</i>	<i>highway</i>
test image				
common of first 35 background frames				
discriminative common of test frame				
distance between discriminative common and common				

**Figure 2.9.** The visual demonstration of common frame of backgrounds, discriminative common frame and distance map between them

By taking these adverse effects of pure CVA, we have put together a new CVA methodology to obtain an accurate distance map between the test and background frames. For this purpose, as an extended version of CVA, the Discriminative Common Vector Approach (DCVA) option has been considered in the case of distance computation stage. The DCVA method works based on the CVA method. First of all, the common vector of



the background frames is obtained with the CVA concept and then the discriminative common vector is obtained by taking the projection of the test vector onto the orthonormal vectors generated from the Gram-Schmidt procedure. This process is called DCVA. The L1 norm distance between the discriminative common vector and the common vector gives accurate regions for the foreground motion, which can be observed from Figure 2.9. This distance is then utilized for the foreground segmentation. Detailed information about the foreground segmentation is introduced in the chapter, *Foreground Extraction*.

The discriminative common vector related to the test frame  $\mathbf{a}_t$  can be found based on the following steps.

- At first, the difference vector  $\mathbf{a}_{t,diff}$  is obtained after the projection of the test vector onto the orthonormal basis  $(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$ .

$$\mathbf{a}_{t,diff} = \langle \mathbf{a}_t, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_t, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \mathbf{a}_t, \mathbf{z}_{(k-1)} \rangle \mathbf{z}_{(k-1)} \quad (2.38)$$

- Then, as shown in Eq. (2.39), subtracting the  $\mathbf{a}_{t,diff}$  from the test vector ( $\mathbf{a}_t$ ), gives a discriminative common vector associated to  $\mathbf{a}_t$ .

$$\mathbf{a}_{t,com} = \mathbf{a}_t - \mathbf{a}_{t,diff} \quad (2.39)$$

Figure 2.9 shows some visual results of CVABS in terms of background modelling. The first row presents the test frames. The second row of Figure 2.9 exhibits the common frame (background model) derived from the background frames. On the other side, the third row denotes the visualization of discriminative common frame related test image to be processed for moving object detection.

The last row displays the distance between the discriminative common frame and background model (common frame), which clearly exposes the motions in the test frame. One can obviously observe that using the CVABS promises high valuable results in terms of highlighting the foreground regions. With respect to the idea of the CVABS, it is expected that the common and unvarying characteristic of static regions would be combined with the background model (common frame) while details such as unstable regions including illuminations, reflections and waving trees would be moved to the difference frame.

**Algorithm-1:** Foreground Detection with DCVA

1. *Constructing a background set with column representation of  $k$  frames.*
2. *Subtracting a predefined reference vector from each vectors stated on background set to obtain the difference subspace*
3. *Obtaining orthonormal vectors spanning difference subspace of processed background set by applying Gram–Schmidt orthogonalization process onto the difference subspace*
4. *Computing the difference vector related to reference vector by taking projection of reference vector onto the basis returned from Gram–Schmidt orthogonalization process.*
5. *Computing the common vector, which is the background model of background set, by subtracting the difference vector from reference vector.*
6. *Computing the difference vector related to test frame by taking projection of test vector onto the basis returned from Gram–Schmidt orthogonalization process.*
7. *Computing the discriminative common vector by subtracting the difference vector from test vector.*

When considering the general implementation of CVABS, an overview of the procedure for the foreground detection with the CVA concept can be summarized with the Algorithm-1.

#### 2.4.2. Common vector versus average vector

As aforementioned, the common vector is obtained by taking the difference between the sum of the projection average vector onto the orthonormal basis and average vector. With respect to this, one can note that the common vector refers to the weighting of the average vector with a constant. To explain the contribution of common vector against average, a mathematical proof is illustrated and the difference in performance is compared with a simple numerical example and a visual demonstration.

Presuming that we give  $m$  vectors and  $\mathbf{a}_i$ ,  $\mathbf{a}_{\text{com}}$  and  $\mathbf{z}_i$  refers to a training vector, the common vector and orthonormal basis returned from Gram-Schmidt Orthogonalization, respectively. Hence, all the vectors in the training set can be written with following forms:

$$\begin{aligned}\mathbf{a}_1 &= \mathbf{a}_{\text{com}} + \langle \mathbf{a}_1, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_1, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \mathbf{a}_1, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1} \\ \mathbf{a}_2 &= \mathbf{a}_{\text{com}} + \langle \mathbf{a}_2, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_2, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \mathbf{a}_2, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1} \\ &\dots \\ \mathbf{a}_m &= \mathbf{a}_{\text{com}} + \langle \mathbf{a}_m, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_m, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \mathbf{a}_m, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1}\end{aligned}\tag{2.40}$$

If we summarize both sides of the above equation side by side, we would like to obtain:

$$\begin{aligned}\sum_{i=1}^m \mathbf{a}_i &= m \mathbf{a}_{\text{com}} + \langle \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1} \\ \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i &= \mathbf{a}_{\text{com}} + \langle \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \langle \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1} \\ \mathbf{a}_{\text{com}} &= \mathbf{a}_{\text{ave}} - \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_1 \rangle \mathbf{z}_1 - \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_2 \rangle \mathbf{z}_2 - \dots - \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1}\end{aligned}\quad (2.41)$$

As we can see, the common vector is obtained by subtracting the average vector from its projection onto all of the orthonormal basis. To analyze the behaviors of an average and a common vector on three simple training vectors, a numerical example is given. Let suppose that we have been given 3 vectors, such as  $\mathbf{a}_1 = [\mathbf{1} \ \mathbf{1} \ \mathbf{1}]^T$ ,  $\mathbf{a}_2 = [\mathbf{1} \ \mathbf{1} \ -\mathbf{1}]^T$  and  $\mathbf{a}_3 = [\mathbf{1} \ \mathbf{5} \ \mathbf{5}]^T$ , then assume that the closest vectors ( $\mathbf{a}_1$  and  $\mathbf{a}_2$ ) refer to the background and the distinct vector ( $\mathbf{a}_3$ ) refers to the foreground. Hence, the common vector can be obtained by the training set that consisted of these vectors. First of all, we should compute absolute difference vectors ( $\mathbf{b}_1$  and  $\mathbf{b}_2$ ) by taking  $\mathbf{a}_1$  as a reference.

$$\mathbf{b}_1 = [\mathbf{0} \ \mathbf{0} \ -\mathbf{2}]^T \text{ and } \mathbf{b}_2 = [\mathbf{0} \ \mathbf{4} \ \mathbf{4}]^T \quad (2.42)$$

To go on, by using the Gram-Schmidt Orthogonalization, the common vector of the training set is acquired by the following rules:

$$\mathbf{d}_1 = \mathbf{b}_1, \quad \mathbf{z}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|} = [\mathbf{0} \ \mathbf{0} \ -\mathbf{1}]^T \quad (2.43)$$

$$\mathbf{d}_2 = \mathbf{b}_2 - \langle \mathbf{b}_2, \mathbf{z}_1 \rangle \mathbf{z}_1 = [\mathbf{0} \ \mathbf{4} \ \mathbf{0}]^T, \quad \mathbf{z}_2 = \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|} = [\mathbf{0} \ \mathbf{1} \ \mathbf{0}]^T \quad (2.44)$$

$(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{(k-1)})$  shows the orthonormal basis and  $(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{(k-1)})$  indicates the orthogonal vectors. Hence, the summation of the projections of  $\mathbf{a}_1$  onto the orthonormal basis of the difference subspace  $\mathbf{B}$ , which is denoted with  $\mathbf{a}_{\text{sum}}$ , can be obtained by as follows:

$$\mathbf{a}_{\text{sum}} = \langle \mathbf{a}_1, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_1, \mathbf{z}_2 \rangle \mathbf{z}_2 = [\mathbf{0} \ \mathbf{1} \ \mathbf{1}]^T \quad (2.45)$$

Finally, the common vector can be obtained by subtracting the  $\mathbf{a}_{\text{sum}}$  from either the reference vector ( $\mathbf{a}_1$ ) or the average vector. By using the reference vector, the common vector can be obtained with Eq. (2.46).

$$\mathbf{a}_{\text{com}} = \mathbf{a}_1 - \mathbf{a}_{\text{sum}} = [\mathbf{1} \ \mathbf{0} \ \mathbf{0}]^T \quad (2.46)$$

Also, by using the average vector, the common vector can be obtained with Eq. (2.49).

Let us define the average vector as:

$$\mathbf{a}_{\text{ave}} = \sum_{i=1}^m \mathbf{a}_i = [\mathbf{1} \ \mathbf{2.33} \ \mathbf{1.67}]^T \quad (2.47)$$

$$\begin{aligned} \mathbf{a}_{\text{ave,sum}} = & \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_2 \rangle \mathbf{z}_2 + \dots + \\ & \langle \mathbf{a}_{\text{ave}}, \mathbf{z}_{m-1} \rangle \mathbf{z}_{m-1} = [\mathbf{0} \ \mathbf{2.33} \ \mathbf{1.67}]^T \end{aligned} \quad (2.48)$$

Hence, the common vector would be:

$$\mathbf{a}_{\text{com}} = \mathbf{a}_{\text{ave}} - \mathbf{a}_{\text{ave,sum}} = [\mathbf{1} \ \mathbf{0} \ \mathbf{0}]^T \quad (2.49)$$

If the absolute between the training vectors, average and common vector is computed with the rules in Eq. (2.50), we will observe the different results that stand out.

$$\begin{aligned} F_1^{\text{ave}} &= \|\mathbf{a}_1 - \mathbf{a}_{\text{ave}}\|^2 = 1.4907, \quad F_1^{\text{com}} = \|\mathbf{a}_1 - \mathbf{a}_{\text{com}}\|^2 = 1.4142 \\ F_2^{\text{ave}} &= \|\mathbf{a}_2 - \mathbf{a}_{\text{ave}}\|^2 = 2.9814, \quad F_2^{\text{com}} = \|\mathbf{a}_2 - \mathbf{a}_{\text{com}}\|^2 = 1.4142 \\ F_3^{\text{ave}} &= \|\mathbf{a}_3 - \mathbf{a}_{\text{ave}}\|^2 = 4.2687, \quad F_3^{\text{com}} = \|\mathbf{a}_3 - \mathbf{a}_{\text{com}}\|^2 = 7.0711 \end{aligned} \quad (2.50)$$

To compare the performance of an average and a common vector, the obtained  $F$  values are taken as a reference. For the best performance, we would like to expect that the obtained results are close to the background model ( $\mathbf{a}_1$  and  $\mathbf{a}_2$ ) and the difference between the foreground vector ( $\mathbf{a}_3$ ) should be as high as possible. By inspecting the  $F$  values, we can easily make an inference that the common vector outperforms the average vector when the difference of the foreground vector is considered, which is reported as 4.2687 and 7.0711 for average and common vectors, respectively. The objective results imply that the CVA can separate the foreground from the background with a high-performance rate when compared with an average vector.

### 2.4.3. Foreground detection

As is outlined in the sections above, the common frame associated with background frames is obtained for background modelling. The main objective is to derive a robust distance between the test frames and the background frames with respect to a CVABS based motion segmentation procedure. In the case of initialization, the first  $N$  frames are considered as a background list called  $B(x) = \{B_1(x), \dots, B_k(x), \dots, B_N(x)\}$ , where  $x$  corresponds to a pixel location. According to a common idea, using the first  $N = 35$  frames for initialization of the background bank is suitable by considering the speed and performance. The Eq. (2.51) represents the foreground detection module utilized to determine the foreground map between the test and the background frames.

$$F_t(x) = \begin{cases} 1 & \text{if } \# \left\{ \sum_{i=1}^N \text{dist}(I_t(x), B_i(x)) \geq R(x) \right\} > \#min \\ 0 & \text{otherwise} \end{cases} \quad (2.51)$$

In Eq. (2.51), the ‘#’ operator is used as a counter of the foreground pixels. The  $\#min$  is a decision threshold to assign the label of a pixel as a foreground or a background. It is specified as  $\#min = N - 1$ . For example, the label of a pixel is considered as a foreground (1) if it is marked as 1 in all binary output maps ( $F_1(x), F_2(x), \dots, F_N(x)$ ) otherwise it is assigned as a background (0). With this strict decision threshold process, it is ensured that the algorithm becomes more robust to the noisy pixels. Again, the  $R(x)$  is a gray level threshold to generate binary output maps.

Fundamentally, the performance of all algorithms depends on the utilized distance metric. In this study, we performed a hybrid distance metrics where the three distance metrics are considered in the case of foreground detection. The first distance metric is  $\ell_1$  norm distance, which is usually known as the traditional distance in the literature of background subtraction. However, the  $\ell_1$  distance is sensitive to sudden illumination changes. Therefore, as a second distance metric, the Gradient information is taken into account to bottle with the sudden illumination changes and shadows. The last distance is determined based on the common vector approach concept.

$$\text{dist}_{\ell_1}(I_t(x), B_i(x)) = |I_t(x) - B_i(x)| \quad (2.52)$$

As shown in Eq. (2.52), the  $\ell_1$  distance is performed by comparing the gray value of each pixel. Simply, the absolute value of grey values is considered to compute  $\ell_1$  distance.

$$\begin{aligned}
I_{gx'}(\mathbf{x}) &= D_{11}(\mathbf{x}) \cdot I_{gx}(\mathbf{x}) + D_{12}(\mathbf{x}) \cdot I_{gy}(\mathbf{x}) \\
I_{gy'}(\mathbf{x}) &= D_{12}(\mathbf{x}) \cdot I_{gx}(\mathbf{x}) + D_{22}(\mathbf{x}) \cdot I_{gy}(\mathbf{x}) \\
I_{gt,t}^m(\mathbf{x}) &= \sqrt{I_{gx'}(\mathbf{x})^2 + I_{gy'}(\mathbf{x})^2} \\
\bar{B}_{gx'}(\mathbf{x}) &= D_{11}(\mathbf{x}) \cdot \bar{B}_{gx}(\mathbf{x}) + D_{12}(\mathbf{x}) \cdot \bar{B}_{gy}(\mathbf{x}) \\
\bar{B}_{gy'}(\mathbf{x}) &= D_{12}(\mathbf{x}) \cdot \bar{B}_{gx}(\mathbf{x}) + D_{22}(\mathbf{x}) \cdot \bar{B}_{gy}(\mathbf{x}) \\
\bar{B}_{gt,t}^m(\mathbf{x}) &= \sqrt{\bar{B}_{gx'}(\mathbf{x})^2 + \bar{B}_{gy'}(\mathbf{x})^2}
\end{aligned} \tag{2.53}$$

$$dist_{Gmag}(I_t(x), B_i(x)) = |I_{gt,t}^m(\mathbf{x}) - \bar{B}_{gt,t}^m(\mathbf{x})| \tag{2.54}$$

In case of the second distance metric, the gradient information is activated between the test and the background frames. As is known, the illuminations and shadows distribute in a homogeneous way. Therefore, using the gradient information between test and background frames improves the performance by diminishing the sudden illumination effect. Owing to this, the homogeneous regions are suppressed by taking the derivation procedure. In this study, a new gradient distance metric is computed to reduce the ghost problem caused by sudden changes and intermittent object motion problems. The utilized gradient distance metric is calculated with an edge suppression based gradient transformation approach [49], shown in Eq. (2.53) and Eq. (2.54). Firstly, the horizontal gradient map,  $I_{gx}(\mathbf{x})$ , and vertical gradient map  $I_{gy}(\mathbf{x})$  of the test frame ( $I_t$ ) and the mean background frame ( $\bar{B}_t$ ) are computed with the Sobel operator. Later, the cross-diffusion tensor terms, called  $D_{11}$ ,  $D_{22}$  and  $D_{12}$ , are determined with respect to the rules given in the referred study. After applying the cross-diffusion tensor terms to  $I_t$  and  $\bar{B}_t$ , then the gradient transformed versions,  $I_{gt,t}^m(\mathbf{x})$  and  $\bar{B}_{gt,t}^m(\mathbf{x})$ , are acquired to compute gradient distance. The absolute distance between these two new robust gradient maps,  $I_{gt,t}^m(\mathbf{x})$  and  $\bar{B}_{gt,t}^m(\mathbf{x})$ , gives accurate and noise free foreground localization.

$$dist_{cva}(I_t(x), B_i(x)) = |I_{d,com}(x) - B_{com}(x)| \tag{2.55}$$

The third and main distance metric is expressed with a CVABS based distance computation. As emphasized in the chapters above, firstly, the common frames related to the background frames are determined with respect to the CVA method. Moreover, the discriminative common frame of the test frame is computed by taking projection of the test frame onto the orthonormal vectors associated with the background list. As exhibited in Eq. (2.55), the absolute difference between the common frame and the discriminative common frame gives us a novel distance metric for foreground detection.

$$\begin{aligned}
dist(I_t(x), B_i(x)) &= \\
dist_{\ell_1}(I_t(x), B_i(x)) &+ dist_{Gmag}(I_t(x), B_i(x)) + dist_{com}(I_t(x), B_i(x)) \quad (2.56) \\
&= |I_t(x) - B_i(x)| + |I_{gt,t}^m(x) - \bar{B}_{gt,t}^m(x)| + |I_{d,com}(x) - B_{com}(x)|
\end{aligned}$$

Eq. (2.56) demonstrates the final distance metric as a combination of the three distance metrics. As a hybrid distance metric; the pure grey level distance, gradient distance and CVA distance are combined to obtain a robust and weighted distance term. The condition of  $dist_{\ell_1}(I_t(x), B_i(x)) > 1$  is carried out in the final distance metric to refuse the noisy regions caused by dynamic scenes. By using this final distance metric, those pixel values related to foreground regions take higher values and the value of noisy and unwanted pixels becomes lower. Later, the segmentation process is applied to the final distance map to determine the label of each pixel. Moreover, the gradient transformation enables us to wipe out the ghosts and illuminations. The CVA distance together with the traditional and gradient distance generates more valuable results in challenging videos.

While the updating procedure for background frames is same as explained in the *Section 2.3.2*, the updating procedure for inner parameters is same as explained in the *Section 2.3.3*. Also, the procedure for handling PTZ challenges parameters is same as explained in the *Section 2.3*.

### 3. PERFORMANCE ANALYSIS

In this study, we have analyzed the performance of four studies, which are mentioned above as Method 1: Background Modelling Using Common Vector Approach (BMCVA), Method 2: Background Modelling Using Common Matrix Approach (BMCMA), Method 3: Sliding Window based Change Detection (SWCD) and Method 4: Common Vector Approach based Background Subtraction (CVABS). In below sections, we have provided some information about utilized datasets and evaluation metrics.

#### 3.1. Datasets

In the experimental stage, two well-known background subtraction datasets, namely Change Detection (CDnet) [1] and Wallflower [50], are utilized in order to analyze the performance of proposed methods.

The CDnet 2014 dataset contains different types of backgrounds related to real world events including *illumination variation* and *dynamic changes*. The dataset contains 53 videos (corresponding to nearly 160,000 frames) in 11 categories: *badWeather*, *baseline*, *cameraJitter*, *dynamic Background*, *intermittentObjectMotion*, *lowFramerate*, *nightVideos*, *PTZ*, *shadow*, *thermal* and *turbulence*.

Technically Wallflower dataset provides different classes of about dynamic backgrounds which are *Moved Object (MO)*, *Time of Day (TD)*, *Light Switch (LS)*, *Waving Trees (WT)*, *Camouflage (C)*, *Bootstrapping (B)* and *Foreground Aperture (FA)*. Until now, various methods have been made experimental on this dataset. The specified training and test images with their ground truth are utilized to obtain subjective and objective results.

#### 3.2. Evaluation Metrics

In this section, some objective metrics are provided to evaluate the results of each proposed method. As noted by CDnet2014 [1], possible metrics to measure the performance of background subtraction techniques can be listed as number of True Positives (TP), number of True Negatives (TN), number of False Negatives (FN) and number of False Positives (FP), whose definitions are compactly presented in Table 3.1. For CVABS and SWCD, we have considered two main metrics as the MCC and F-score



metrics to compare the performance of proposed methods with other state of art methods in an objective way. The F-score is computed based on the precision,  $TP / (TP + FP)$ , and recall,  $TP / (TP + FN)$ , values, and the value of the F-score is in the interval [0-100] in terms of percentage.

**Table 3.1.** Utilized metrics

<i>Metrics</i>			
Matthew Correlation Coefficient (MCC)			
$(TP \cdot TN) - (FP \cdot FN)$			
$(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)$			
<i>Recall (Re)</i>	<i>Specificity (Sp)</i>	<i>False Positive Rate</i>	<i>False Negative Rate (FNR)</i>
$TP / (TP + FN)$	$TN / (TN + FP)$	<i>(FPR) <math>FP / (FP + TN)</math></i>	$FN / (TN + FP)$
<i>Precision (Pre)</i>	<i>F-score</i>	<i>Percentage of Wrong Classifications (PWC)</i>	
$TP / (TP + FP)$	$2 * Pr * Re / (Pr + Re)$	$100( FN + FP ) / ( TP + FN + FP + TN )$	

The MCC is a balanced metric that reveals the correlation between the two binary samples and its value between -1 and 1. In this study, the average MCC value is computed after calculating the MCC value for each video. For the best performance, the MCC value is equal to 1.

### 3.3. Performance of Method 1: BMCVA

#### 3.3.1. Subjective evaluation of method 1: BMCVA

In order to comment the obtained results, we have compared the produced results with other ones. For this purpose, the subjective outputs are presented on Figure 3.1. Specifically, the visual results that are presented in the study of Bouwman [2] are considered as reference in case of performance comparison. For a benchmark comparison, the obtained visual results are compared with state of popular subspace and other methods, which are given as Single Gaussian (SG) [51], Mixture of Gaussian (MOG) [52], Kernel Density Estimation (KDE) [53], Subspace Learning PCA (SL-PCA) [11], Subspace Learning ICA (SL-ICA) [54], Subspace Learning via Incremental Non Negative Matrix Factorization (SL-INMF) [17] and Subspace Learning via Incremental Rank-(R1, R2, R3) Tensor (SL-IRT) [16].

The all of visual results are exhibited in Figure 3.1. The first column indicates the method's name and the rest of columns show the performance of each aforementioned

method. Also, the first row denotes the processed image, second row indicates the ground truth related to given image and other rows show visual result generated by each method.

Method	Moved Objects	Time of Day	Light Switch	Waving Trees	Camou-flage	Boot-strap	Foreg. Aperture
Test image							
Ground truth							
<b>SG</b> <i>Wren et al.</i>							
<b>MOG</b> <i>Stauffer et al.</i>							
<b>KDE</b> <i>Elgammal et al.</i>							
<b>SL-PCA</b> <i>Oliver et al.</i>							
<b>SL-ICA</b> <i>Tsai and Lai</i>							
<b>SL-INMF</b> <i>Bucak et al.</i>							
<b>SL-IRT</b> <i>Li et al.</i>							
<b>BMCVA</b> <i>Proposed</i>							

**Figure 3.1.** Subjective results of CVA on the Wallflower dataset

At a first glance, we can observe that similar outputs are obtained from each method. Upon inspecting results, one can emphasize that probabilistic based methods including MOG and KDE produce similar results in terms of foreground region detection. The results of KDE and MOG are superior than SG, since background modelling with single Gaussian is a short-side in term of complex background. Again, we can emphasize that SG, MOG and KDE are sensitive illumination changes because of working on historical probability of each pixel.

On the other side, the subspace based methods are more robust to illumination and complex background changes. By examining results of PCA, ICA, INMF and IRT, it can be seen that the visuals result of IRT are not converged to ground truth as some objects are disappeared in foreground mask. Moreover, although the PCA method exhibits good results in case of *Time of Day*, *Light Switch*, *Waving Trees*, *Camouflage*, *Foreground Aperture*, but some erroneous pixels are obtained for *Moved Object* and *Bootstrap* videos.

Furthermore, visual outputs of ICA and INMF are similar to each other, however, the performance of ICA is more dominant for Camouflage and Bootstrap videos.

Finally, we can observe that CVA and PCA generate closest results, however, the PCA method fails in case of indoor crowded scene (bootstrap). Also, one can note that the proposed method can perfectly model the clean background in case of illumination changes, crowded scenes and other complex backgrounds. As a result, good foreground masks are determined for all videos.

### 3.3.2. Objective evaluation of method 1: BMCVA

**Table 3.2.** *Objective performance evaluation on the Wallflower dataset*

Method	Error	Problem Type							Total Errors	TE without LS	TE without C
		MO	ToD	LS	WT	C	B	FA			
SG	FN	0	949	1857	3110	4101	2215	3464	35133	18153	28992
	FP	0	535	15123	357	2040	92	1290			
MOG	FN	0	1008	1633	1323	398	1874	2442	27053	11251	23557
	FP	0	20	14169	341	3098	217	530			
KDE	FN	0	1298	760	170	238	1755	2413	26450	11537	22175
	FP	0	125	14153	589	3392	933	624			
SL-PCA	FN	0	879	962	1027	350	304	2441	17677	16353	15779
	FP	1065	16	362	2057	1548	6129	537			
SL-ICA	FN	0	1199	1557	3372	3054	2560	2721	15308	13541	12211
	FP	0	0	210	148	43	16	428			
SL-INMF	FN	0	724	1593	3317	6626	1401	3412	19098	17202	12238
	FP	0	481	303	652	234	190	165			
SL-IRT	FN	0	1282	2822	4525	1491	1734	2438	17053	13842	15448
	FP	0	159	389	7	114	2080	12			
BMCVA	FN	0	1012	946	766	708	982	2537	7891	6625	7175
	FP	0	0	320	20	8	130	482			

In addition to subjective evaluation, the objective results for each method is determined with respect to statistical metrics, called false positive (FP) and false negative (FN). While the FP indicates the pixel marked as foreground in processed image but it is background in ground truth image, conversely the FN refers to the pixel marked as background in processed image but it is foreground in ground truth image.

If a pixel is marked as 1 in processed image, but it is 0 in ground truth image, then the count of FP is incremented by 1. Similarly, if a pixel is marked as 0 in processed image, but it is 1 in ground truth image, then the count of FN is incremented by 1. By

combining these error values, the Total Error (TE) metric is computed as a sum of FP and FN. The lower value of error value denotes the best performance in the concept of foreground segmentation. Also, the Total Errors without light switch (TE without LS) and Total Errors without Camouflage switch (TE without Camouflage) are presented on the last columns of Table 3.2.

The Table 3.2 summarizes all of the objective results for aforementioned background modelling methods. As we can see that the performance MOG and KDE are close to each other and show better performance than SG method. The performance of MOG and KDE are better when the light switch video excluded, but worse in case of TE metric. Comparing the PCA, ICA, INMF and IRT, one can observe that the performance of ICA is dominant in case of all metrics. On the other side, we can find that the CVA method combining with the basic post processing procedure show favorable results in terms of all metrics.

### **3.4. Performance of Method 2: BMCMA**

#### **3.4.1. Subjective evaluation of method 2: BMCMA**

In the present work, a simple thresholding methodology is realized in case of revealing the binary skeleton of objects. Since the difference of two common matrix gives changes, a fixed thresholding is carried over the absolute difference. The obtained visual results are demonstrated on Figure 3.2. To subjectively judge performance of both methods, the obtained visual results are compared with state of popular subspace and other methods, which are given as Single Gaussian (SG) [19], Mixture of Gaussian (MOG) [20], Kernel Density Estimation (KDE) [53], Subspace Learning PCA (SL-PCA) [11], Subspace Learning ICA (SL-ICA) [54], Subspace Learning via Incremental Non Negative Matrix Factorization (SL-INMF) [17] and Subspace Learning via Incremental Rank-(R1, R2, R3) Tensor (SL-IRT) [16]. For this purpose, the visual results determined in the work of Bouwman [2] are taken as ground on in case of performance comparison.

In Figure 3.2, the first column denotes method's name, the other columns show video's name, respectively. Again, the first row and second row exhibit test image and related ground truth, and other rows demonstrates visual results returned from each method.

Sequence	Moved Objects	Time of Day	Light Switch	Waving Trees	Camou-flage	Boot-strap	Foreg. Aperture
Test image							
Ground truth							
<b>SG</b> <i>Wren et al.</i>							
<b>MOG</b> <i>Stauffer et al.</i>							
<b>KDE</b> <i>Elgammal et al.</i>							
<b>SL-PCA</b> <i>Oliver et al.</i>							
<b>SL-ICA</b> <i>Tsai and Lai</i>							
<b>SL-INMF</b> <i>Bucak et al.</i>							
<b>SL-IRT</b> <i>Li et al.</i>							
<b>BMCMA</b> <i>Proposed</i>							

**Figure 3.2.** Subjective results of CMA on the Wallflower dataset

From the exhibited results, we can observe that each method presents similar foreground objects in the meaning of obtained foreground skeleton. By analyzing results, one can note that results of MOG and KDE are closes to each other and are dominant than SG method. The performance of SG, MOG and KDE are weakness to illumination changes due to work on historical probability of pixels.

To continue, we can see that subspace based method are more robust to light changes. By comparing the PCA, ICA, INMF and IRT, we can emphasize that the result of IRT is the worst one in terms of preserving foreground skeleton. While the INMF shows good results in case of bootstrap video, but the same performance has not maintained in case of camouflage video. Moreover, the results of PCA are similar to CVA method, however, the PCA method fails in case of indoor crowded scene (bootstrap). Furthermore, the proposed method not only robust to dynamic structures but also resistance to illumination change in case of foreground detection.

### 3.4.2. Objective evaluation of method 2: BMCMA

In addition to subjective results, the statistical results are obtained by considering the false positive (FP) and false negative (FN) pixels. With this aim, the ground truth images and acquired foreground object are compared to find the number of erroneous pixels by counting the number of FP and FN. If a pixel marked as foreground in processed image, but marked as background in ground truth, then it is considered as FP.

**Table 3.3.** Numerical results of CMA on the Wallflower dataset

Method	Error								Total Errors	TE without LS	TE without C
		MO	ToD	LS	WT	C	B	FA			
SG	FN	0	949	1857	3110	4101	2215	3464	35133	18153	28992
	FP	0	535	15123	357	2040	92	1290			
MOG	FN	0	1008	1633	1323	398	1874	2442	27053	11251	23557
	FP	0	20	14169	341	3098	217	530			
KDE	FN	0	1298	760	170	238	1755	2413	26450	11537	22175
	FP	0	125	14153	589	3392	933	624			
SL-PCA	FN	0	879	962	1027	350	304	2441	17677	16353	15779
	FP	1065	16	362	2057	1548	6129	537			
SL-ICA	FN	0	1199	1557	3372	3054	2560	2721	15308	13541	12211
	FP	0	0	210	148	43	16	428			
SL-INMF	FN	0	724	1593	3317	6626	1401	3412	19098	17202	12238
	FP	0	481	303	652	234	190	165			
SL-IRT	FN	0	1282	2822	4525	1491	1734	2438	17053	13842	15448
	FP	0	159	389	7	114	2080	12			
BMCMA	FN	0	1017	882	26	172	929	2534	8218	7016	7430
	FP	0	0	320	1106	616	157	485			

For opposite case, if a pixel marked as foreground by ground truth, but marked as background in processed image, then it is considered as FN. The sum of FP and FN denotes the error measure in terms of comparing the objective results. Specifically, the Total Errors, Total Errors without light switch (TE without LS) and Total Errors without Camouflage switch (TE without Camouflage) are demonstrated on the last columns of Table 3.3. The less error value indicates the best performance in terms of foreground segmentation. The obtained statistical results are presented in Table 3.3. From the Table, one can derive that a superior performance is obtained by the proposed method, called CMA. In conjunction with visual results, the performance SG, MOG and KDE similar to each other. However, when the light switch video is excluded in case of performance evaluation, we can observe that the MOG and KDE generate better results than almost of

all algorithms except CMA. These results are attributed to characteristic of probability based foreground and change detection property. Moreover, when the camouflage video is not considered, the worst performance is produced by probabilistic based background subtraction methods. Also, comparing the subspace based methods including SL-PCA, SL-ICA and SL-INMF, one can note that the performance of SL-ICA is favorable against SL-PCA and SL-INMF. The performance of SL-PCA and SL-IRT are closes to each other, but difference bears in case of removing the light switch

### 3.5. Performance of Method 3: SWCD

#### 3.5.1. Subjective evaluation of method 3: SWCD



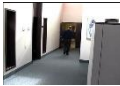




























































Method	sidewalk #1013	fountain01 #0715	cubicle #2024	winterDriv. #2025	skating #1895	continPan #844	turbulence0 #2973
Test							
GT							
Proposed							
Shared Model							
PAWCS							
SuBSENSE							
C-EFIC							
MBS							
FTSG							

Figure 3.3. Some visual results of SWCD on the CDnet 2014 dataset

In order to test the effectiveness of the proposed method, the results are compared with the results of state of the art change detection approaches as stated on CDnet website.

The ground truth of each sample is subjectively constructed. Results are summarized in Figure 3.3. By inspecting Figure 3.3, one can say that the results of the proposed method are in competition with best methods. Particularly, in *sidewalk*, *fountain01*, *cubicle*, *winterDriveway*, *skating*, *continuousPan*, *turbulence0* videos, the performance of the proposed method is fairly plausible when compared with state of art methods. Especially, in case of *winterDriveway* which includes the intense effects of *intermittent object motion*, the proposed method outperforms the rest. For the *cubicle* video case, it is observed that our method is robust against sudden illumination changes, which is widely accepted as a challenging problem for real time image processing tasks. Our background updating procedure is also observed to work well in case of the *continuousPan* video. In the same set of experiments, the proposed method provides competitive results in case of *fountain01* and *skating* videos, which are assumed as real representations of *dynamic background* and *bad weather* categories. Furthermore, the proposed updating strategy is able to handle turbulence effects as demonstrated in *turbulence0* video.

The proposed method also accurately reduces the effects of ghosts, which are caused by intermittent object motions. As a result, it was concluded that the proposed method performs at least as good as the state of the art methods, and gains a favorable edge on truly challenging cases in dynamic background, shadow and PTZ categories. Another important conclusion is that the proposed SWCD method seems to be an “on the average fair” method for all scenarios. This property is not observed in other methods, which usually focus to a specific class of video scenarios and ignore the others. This observation is also in accordance with the objective F-score values presented in the next subsection.

### 3.5.2. Objective evaluation of method 3: SWCD

Using these metrics, extensive experimental evaluations are made and their CDnet averages are presented in Table 3.4. Again, the proposed method attains very comparable or superior results as compared to other state of the art techniques. Particularly, the F-score value of the proposed method (0.7510) exceeds the nearest competitor. The false alarm rates (FPR and FNR) perfectly cope with the competitors. We can also observe that the proposed method produces lower wrong classified rates, when compared with SharedModel, SuBSENSE, PAWCS and FTSG. The SWCD method also gives better



performance than PAWCS in terms of specificity metric. Finally, the precision and recall values of the proposed method are compatible with popular methods.

**Table 3.4.** Objective performance comparison with top state of art methods

Overall Averages (%)							
Method	Re	Sp	FPR	FNR	PWC	F-score	Pre
<b>SWCD</b>	0.7695	0.9934	0.0066	0.2305	1.3444	<b>0.7510</b>	0.7667
<b>SharedModel</b>	0.8098	0.9912	0.0088	0.1902	1.4996	0.7474	0.7503
<b>SuBSENSE</b>	0.8124	0.9904	0.0096	0.1876	1.6780	0.7408	0.7509
<b>PAWCS</b>	0.7718	0.9949	0.0051	0.2282	1.1992	0.7403	0.7857
<b>C-EFIC</b>	0.7976	0.9782	0.0218	0.2024	2.6316	0.7307	0.7543
<b>MBS</b>	0.7389	0.9927	0.0073	0.2611	1.2614	0.7288	0.7382
<b>FTSG</b>	0.7657	0.9922	0.0078	0.2343	1.3763	0.7283	0.7696

Table 3.5 shows the categorically separated performances according to F-scores. The proposed method gives particularly superior results in challenging cases of Thermal (Th) and Turbulence (Tu) categories, while maintaining very reasonable results in all other categories. Unlike several methods which aim a very specific class while compromising the performance from others, our method always keeps on par with the best of each class, making it a very reasonable alternative in real life and general background subtraction applications.

**Table 3.5.** Detailed performance (F-score) evaluation of each category

Method	F-scores of each Category (%)										
	PTZ	BW	BA	CJ	DB	IOM	LF	NV	Sh	Th	Tu
SWCD	0.4561	0.8395	0.9225	0.7451	0.8480	0.6630	<b>0.7096</b>	0.5200	<b>0.8692</b>	<b>0.8533</b>	<b>0.8348</b>
SharedModel	0.3860	0.8480	<b>0.9522</b>	0.8141	0.8222	0.6727	<b>0.7286</b>	0.5419	0.8455	0.8319	0.7339
SuBSENSE	0.3476	<b>0.8619</b>	0.9503	0.8152	0.8177	0.6569	0.6445	0.5599	0.8646	0.8171	0.7792
PAWCS	0.4615	0.8152	0.9397	0.8137	<b>0.8938</b>	0.7764	0.6588	0.4152	<b>0.8710</b>	0.8324	0.6450
C-EFIC	<b>0.6207</b>	0.7867	0.9309	0.8248	0.5627	0.6229	0.6806	<b>0.6677</b>	0.8453	0.8349	0.6275
MBS	0.5520	0.7980	0.9287	<b>0.8367</b>	0.7915	0.7568	0.6350	0.5158	0.8262	0.8194	0.5858
FTSG	0.3241	0.8228	0.9330	0.7513	0.8792	<b>0.7891</b>	0.6259	0.5130	0.8535	0.7768	0.7127

The categorical advantages of the proposed SWCD method are listed as follows.

- *PTZ*: This category includes well known PTZ challenges such as slow continuous pan mode, intermittent pan mode, 2-position patrol-mode PTZ and zooming-in/zooming-out. Contrary to other categories, the scenes behind these videos are not stable. As can be seen from Table 3, most of the methods perform below a rate of 50% in terms of F-score. In this category, C-EFIC seems to perform best. However, C-EFIC fails

badly in BW, DB, IOM and Tu categories. Our proposed SWCD algorithm stands at 46% (top-4) using fast statistical measurements to catch sudden scene changes with an adaptation mechanism for the background set. Besides, the proposed method is observed to always stand among the group of better algorithms (even being the best for some cases) for each of the presented categories.

- *badWeather (BW)*: This category contains challenging weather conditions including *blizzard*, *skating*, *snowFall* and *wetSnow*, which generally occur in winter. The segmentation challenges include illumination changes in *blizzard* and vertical sliding behavior in *wetSnow* videos. The results indicate that the SWCD can overcome the *badWeather* difficulties and achieves an acceptable level of performance with 84% F-score and takes the 3<sup>rd</sup> rank after SuBSENSE (85%) and SharedModel (86%). It must be noted that SuBSENSE and SharedModel perform quite bad for PTZ at F-scores of 35% and 39%, respectively.
- *baseline (BA)*: This category encapsulates basic videos with static backgrounds. Therefore, foreground regions can be usually detected with an effortless manner. The F-scores in this category are all over 90% for all methods.
- *cameraJitter (CJ)*: This category contains movements caused by vibrating unstable camera. The SWCD method has an F-score of 75% (similar to FTSG), which is slightly less than other methods. For example, MBS method reaches to 83%. However, again, MBS goes down to a poor level of 64% for LF category. The lack of high performance with SWCD in CJ category is mostly attributed to the *sidewalk video*, according to the total count of TPs and FNs.
- *dynamicBackground (DB)*: This category consists of complex dynamic movements such as tree branches and water waves, fountains. One can infer from the Table 3 that the feedback mechanism of SWCD provides very competitive results, closely

following the top two: PAWCS (89%) and FTSG (88%). On the other hand, PAWCS performs quite bad in NV and Tu categories and FTSG cannot cope with the PTZ category.

- *intermittentObjectMotion (IOM)*: The videos in this category mostly causes ‘ghost’ artifacts. The F-score of SWCD is an average performance in this category with FTSG being the best. Yet, as indicated above, FTSG completely ignores PTZ cases.
- *lowFramerate (LF)*: This category includes videos captured at various low frame rates. In some videos (such as the “port” video), some critical objects appear too small and there are sudden illumination changes. In this tough category, SWCD reaches an F-score of 71%, which puts it at a compatible level of the best, which is SharedModel (at 73% F-score). However, SharedModel performs way below SWCD at PTZ and Tu categories.
- *nightVideos (NV)*: This category involves complex background dynamics due to darker moving objects and bright background objects, such as light bulbs. Most methods yield an F-score around 50-60% and SWCD performs similarly. The best in this category is C-EFIC, however that method fails at DB and Tu categories.
- *Shadow (Sh), Thermal (Th) and Turbulence (Tu)*: These are challenging SWCD shines. Many of the top-tiers of other categories perform well below SWCD in these categories. This observation supports that use of gradient transformation combined with adjustments of dynamic’s controller parameters gives reasonable segmentation results. As proposed in Section 2.6 (*updating learning*), setting the upper value for learning rate to infinity ( $T_{upper} = \infty$ ) seems to be a reasonable precaution to prevent integration of foreground regions into background and avoiding the leak of foreground objects into background at pixel coordinates where a foreground object

may stand for a long time (referred to *library* video). One can also see that SWCD has potential to cope with air turbulence. The sliding window based background updating procedure together with feedbacks clearly improves performance in this category.

Although the proposed method does not always outperform the best method (which is dedicated to the corresponding category) of each scenario, it was observed to stand well and never “totally” fail in “any” of the categories. This property cannot be argued for other state of the art methods, which usually focus a class of scenarios and ignore others. Therefore, it is argued that SWCD is a robust, useful and effective tool for general purposes.

### **3.5.3. Computational issues of method 3: SWCD**

A key concern in foreground-background separation is the run-time (hence the complexity) of the algorithm. We have investigated the run-time of our method on a modest PC (Intel core i7-6700HQ with 2.60 GHz CPU and 8 GB memory) and a generic MATLAB interpreting environment. Even under these low-end conditions, the experimental runs show that each frame of test videos (frame size = 240x320, video lengths = 2000 frames/video) takes about 0.1 sec for processing. After a quick conversion to a compiled environment (C++) with OpenCV library, the runtime immediately became real time with over 30 frames/sec process speed. The memory utilization was also not above a level of keeping few image frames in RAM. It is concluded that, the proposed method is a plausible alternative to real-time foreground/background separation application with high efficiency.

## **3.6. Performance of Method 4: CVABS**

### **3.6.1. Subjective evaluation of method 4: CVABS**

To give a general insight into the performance of the CVABS method, the visual outputs of several state of art methods are demonstrated and compared in Figure 3.4. As

a common evaluation method, the judgements are expressed in terms of visual inspection of foreground masks, that is the obtained segmentation results are compared with ground truth (GT) images given in Figure 3.4. By evaluating the results, one can observe that the CVABS method gives well segmented and satisfactory results to cope with the illuminations and dynamics changes.

	<i>badWeather</i>	<i>dynamic Background</i>	<i>intermittent ObjectMotion</i>	<i>nightVideos</i>	<i>shadow</i>	<i>thermal</i>	<i>turbulence</i>
	<i>blizzard</i> #3109	<i>fall</i> #1889	<i>winterDriveway</i> #2025	<i>streetCorner</i> <i>AtNight</i> #0902	<i>cubicle</i> #2788	<i>library</i> #3608	<i>turbulence0</i> #2854
<i>Test</i>							
<i>GT</i>							
<i>CVABS</i>							
<i>SharedModel</i>							
<i>WeSamBE</i>							
<i>PAWCS</i>							
<i>SubSENSE</i>							
<i>MBS</i>							
<i>FTSG</i>							

**Figure 3.4.** Visual performance demonstration on the CDnet dataset

The CVABS advocates the reduction of unstable light effects especially for *shadow* videos like *cubicle*. Again, from the *fall* and *turbulence0* videos that include the intense

effects of dynamic scenes, we can observe that the utilized dynamic controller parameters present a great performance in terms of accurate update of the backgrounds as well as reducing the false alarms.

To reveal the strength of CVABS to sustain the performance in case of turbulence degradations, the foreground segmentation results of each method is compared by taking the *turbulence0* video as reference. At first glance, we can say that the CVABS gives clean results for the given sample of the *turbulence0* videos whereas the SharedModel, PAWCS and MBS methods show relative performance impairment for the *turbulence0* video. However, WeSamBE, SuBSENSE and FTSG generate similar outputs with the ground truth for the *turbulence0* video, but we cannot say the same for the SharedModel, PAWCS and MBS methods, where the performance degradation of such methods is caused by the utilized distance computation and updating techniques. Again, we can observe that almost all methods (except MBS) achieve nice segmentation results for *streetCornerAtNight* video which belongs to the *night* category. Furthermore, when the *winterDriveway* video is considered, we can observe that the CVABS gradually updates background frames with utilized feedback parameters to regulate changes in frames. Thus, ghosts are alleviated based on the utilized self-regulation and self-learning procedure in terms of the background frames updating. When analyzing the results of other methods for the *winterDriveway* video, one can note that the performance of FTSG outperforms SharedModel, WeSamBE, PAWCS and SuBSENSE and MBS.

### **3.6.2. Objective evaluation of method 4: CVABS**

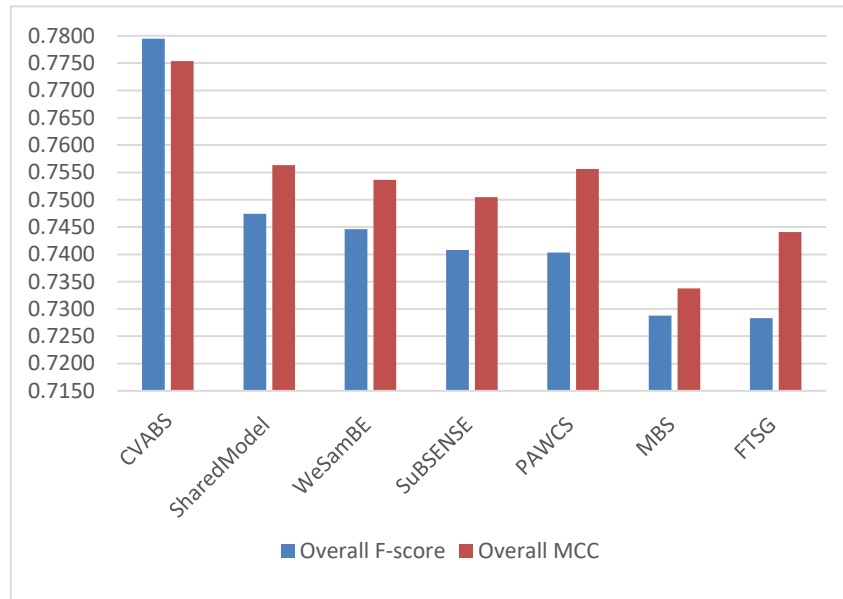
Table 3.6 summarizes the detailed version of F-score values for each method on each category stated at the CDnet 2014 dataset. By examining the results, one can clearly note that the CVABS method gives superior results in the case of *BadWeather*, *LowFramerate*, *NightVideos*, *Thermal* and *Turbulence* categories. In the case of the *PTZ* category, the high F-score value is obtained with the MBS method. Again, among all the methods, the FTSG method is more robust to the ghost problem caused by *intermittent object motions*. Moreover, the PAWCS and SuBSENSE gives good efforts to overcome noise problems observed in the categories of the *dynamicBackgrounds* and *BadWeather*,

respectively. Although the CVABS presents valuable outputs for the *BadWeather* category, but the WeSamBE and SuBSENSE gives competitive results.

**Table 3.6.** Performance (*F*-score) evaluation in more details for each category of aforementioned methods

Method	F-measure Scores of each Category (%)										
	PTZ	BW	BA	CJ	DB	IOM	LF	NV	Sh	Th	Tu
CVABS	0.5082	<b>0.8693</b>	0.9147	0.7837	0.8618	0.6586	<b>0.7755</b>	<b>0.6295</b>	0.8757	<b>0.8567</b>	<b>0.8403</b>
SharedModel	0.3860	0.8480	<b>0.9522</b>	0.8141	0.8222	0.6727	0.7286	0.5419	0.8455	0.8319	0.7339
WeSamBE	0.3844	0.8608	0.9413	0.7976	0.7440	0.7392	0.6602	0.5929	<b>0.8999</b>	0.7962	0.7737
SuBSENSE	0.3476	0.8619	0.9503	0.8152	0.8177	0.6569	0.6445	0.5599	0.8646	0.8171	0.7792
PAWCS	0.4615	0.8152	0.9397	0.8137	<b>0.8938</b>	0.7764	0.6588	0.4152	0.8710	0.8324	0.6450
MBS	<b>0.5520</b>	0.7980	0.9287	<b>0.8367</b>	0.7915	0.7568	0.6350	0.5158	0.8262	0.8194	0.5858
FTSG	0.3241	0.8228	0.9330	0.7513	0.8792	<b>0.7891</b>	0.6259	0.5130	0.8535	0.7768	0.7127

Furthermore, the MBS method is the best one to alleviate *Camera Jitter* problems. Again, we can note for the *Shadow* category that while WeSamBE outperforms all methods with high F-scores, CVABS and PAWCS take the second and third rank in terms of robustness to illumination changes, respectively.



**Figure 3.5.** MCC and F-score results for top ranked methods given in CDnet and CVABS

Figure 3.5 demonstrates the MCC and F-score for the top ranked methods listed in CDnet and the CVABS algorithm. By examining the MCC coefficients in Figure 3.5, we can highlight the superior performance of CVABS and achieved a 77.95% F-score and 77.54% MCC value when compared with others. An interesting point is that it can be seen that degradation in the F-score also results in low MCC values, with the exception of

PAWCS and FTSG. Comparing PAWCS with other methods, one can observe that while the F-measure of PWACS is lower than the SharedModel, PAWCS generates a higher value of the MCC coefficient.

Again, FTSG produces a higher MCC value than MBS while MBS is dominant in the case of the F-score. Moreover, one can say that the performances of the CVABS and SharedModel are in competition with each other, but *CVABS* outperforms all methods exhibited in the CDnet in terms of F-scores and MCC values. When focusing on the F-scores of other methods, it can be noted that the F-score PAWCS and SuBSENSE are similar to each other. As a harmonic mean of precision and recall, the good F-score indicates the precision of each method in terms of performance for moving object segmentation.

### 3.7. Overall Performance Evaluation of Our Proposed Works









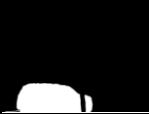

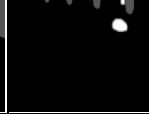


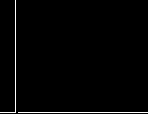


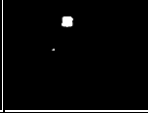










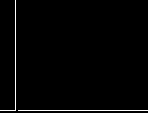










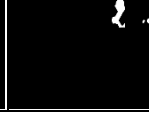



	<i>badWeather</i>	<i>dynamic Background</i>	<i>intermittent ObjectMotion</i>	<i>nightVideos</i>	<i>shadow</i>	<i>thermal</i>	<i>turbulence</i>
	<i>blizzard</i> #3109	<i>fall</i> #1889	<i>winterDriveway</i> #2025	<i>streetCorner AtNight</i> #0902	<i>cubicle</i> #2788	<i>library</i> #3608	<i>turbulence0</i> #2854
<i>Test</i>							
<i>GT</i>							
<i>CVABS</i>							
<i>SWCD</i>							
<i>BMCVA</i>							
<i>BMCMA</i>							

Figure 3.6. Visual outputs returned from proposed methods on CDnet

To compare the performance of each proposed method, the objective and subjective results have obtained on two well-known datasets, namely CDnet and Wallflower. In case



of objective evaluation, the F-measure scores have considered, whereas the segmentation quality have determined based on the human eyes as a subjective evaluation.















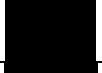



























The Figure. 3.6 exhibits the visual outputs obtained on CDnet. While the first row shows *Test* images and second row indicates the ground truths related to expected segmentation results for test images. As one can clearly observe that employing the feedback mechanism in CVABS and SWCD gives superior results nearly almost in all categories. For blizzard video, where the dynamic changes are not too much, the both of four methods capable of generating the valuable results.

For *fall* video, only SWCD and CVABS are able to struggle with waving trees. Again, one can note that it is impossible to produce desirable results in *winterDriveway* video, unless performing a smart feed-back mechanism, which is considered SWCD and CVABS. Moreover, in case of night effect (refer to *streetCornerAtNight* video), the both of four methods presented good segmentation accuracy. Furthermore, we can say that without using the feedback mechanism, it would be troublesome to combat with illumination changes. For *turbulence* effect, the BMCVA and BMCMA collapsed whereas the CVABS and SWCD generates desirable results. Again, for thermal video (refer to *library* video), the both of four methods produced the well-segmented foreground regions.

**Table 3.7.** *F-measure scores of our proposed methods on CDnet*

F-measure Scores of each Category (%) on CDnet											
Method	PTZ	BW	BA	CJ	DB	IOM	LF	NV	Sh	Th	Tu
CVABS	<b>0.5082</b>	<b>0.8693</b>	<b>0.9147</b>	<b>0.7837</b>	<b>0.8618</b>	0.6586	<b>0.7755</b>	<b>0.6295</b>	<b>0.8757</b>	<b>0.8567</b>	<b>0.8403</b>
SWCD	0.4561	0.8395	0.9225	0.7451	0.8480	<b>0.6630</b>	0.7096	0.5200	0.8692	0.8533	0.8348
BMCVA	0.0343	0.5790	0.9027	0.4777	0.3620	0.5168	0.5817	0.4477	0.6546	0.6809	0.3827
BMCMA	0.0343	0.5798	0.9026	0.4796	0.3617	0.5170	0.5820	0.4471	0.5359	0.6811	0.3819

In the Table 3.7, the categorical F-measure score of each method exhibited in order to evaluate the performance objectively. As we can see that SWCD and CVABS keep pace with each other in terms of F-measure scores. Also, the overall F-measure scores of methods are determined as CVABS 77.95%, SWCD 75.11%, BMCVA 51.09% and BMCMA 50.03%. From the results shown in Table 3.7, we can note that the CVABS dominates the others.

Method	MO #0985	ToD #1850	LS #1865	WT #0247	C #0251	B #0299	FA #0489
Test image							
Ground truth							
CVABS							
SWCD							
BMCVA							
BMCMA							

**Figure 3.7.** Visual Outputs returned from proposed methods on Wallflower

The Figure 3.7 shows the results obtained on Wallflower dataset, which includes seven different videos. For this category, the four methods produced successful results, except for *Light Switch* video (LS). In case of sudden illumination changes like Light Switch, the SWCD regulates itself after some frames, therefore the SWCD gives worst segmentation result for sample numbered as #1865.

**Table 3.8.** F-measure scores of our proposed methods on Wallflower

F-measure Scores of each Category (%) on Wallflower							
Method	B	C	FA	LS	MO	ToD	WT
CVABS	0.7377	0.9268	0.6153	0.5619	0.0000	0.9182	0.7661
SWCD	0.7650	0.9793	0.6357	0.1720	0.0000	0.8557	0.9037
BMCVA	0.8177	0.9771	0.6241	0.7625	0.0000	0.3611	0.9386
BMCMA	0.8230	0.9703	0.6242	0.7772	0.0000	0.3559	0.9136

The Table 3.8 presents categorical F-measure scores of each method obtained on Wallflower dataset. The four methods produce good F-measure scores. Once results scrutinized, CVABS is superior in 5 of 7 videos, SWCD is superior in 4 of 7 videos, BMCVA is superior in 3 of 7 videos and BMCMA is superior in 2 of 7 videos. When the overall F-measure scores are sorted, we can observe the results achieved as CVABS is 76.61%, SWCD is 90.31%, BMCVA is 64.01% and BMCMA is 63.77%. By considering the overall F-measure scores, one can emphasize that the CVABS is more dominant. If the Wallflower considered as a category in addition to 11 categories of CDnet, then we can say that CVABS achieved superior results for 10 of 12 videos and SWCD gives superior results only for Intermittent Object Motions (IOM) and Wallflower categories.

## 4. CONCLUSIONS

In this study, we have developed four foreground detection methods based on different ideas for background modelling and distance computation. During the background modelling, we have considered two ways as (i) using single background model and (ii) using multi backgrounds in memory for real time foreground segmentation in videos. The utilized methods are given as BMCVA, BMCMA, SWCD and CVABS. The SWCD and CVABS are relied on feedback mechanism that is background frames are updated with respect to internal dynamic controller parameters.

The SWCD and CVABS, which relied on sliding window based background model updating procedure, are pixel-wise change detection algorithms and incorporate a smart feedback system in which the dynamic internal parameters are self-regulated and updated. The methods also incorporate a fast and robust estimate of distance map by using cross-projection tensor based gradient transformation. Experimental results on static and dynamic test videos show that the proposed methods successfully tackle challenging problems of background modeling, in terms of both subjective quality and objective measures using ground truth data. The proposed methods (SWCD and CVABS) are not only capable of handling variations in dynamic views, but also efficiently alleviates problems caused by shadows, illumination variation, and thermal and turbulence effects. Experimental results show that SWCD and CVABS stands a top-half position in all categories and outperforms most methods in certain categories. Since several methods focus on certain video types while compromising their performances for the rest of the scenarios, this property renders the proposed algorithms a very reasonable alternative in background modeling and foreground detection. Also, the proposed CVABS method presents valuable performance for overall results of F-measure score and MCC on CDnet dataset.

However, we have observed that using a single background frame through time domain in video does not produce favorable results in terms of foreground segmentation as carried out in our proposed BMCVA and BMCMA methods. The reasons to why the BMCVA and BMCMA methods collapsed can be attributed to encountered dynamics including waving trees, illumination changes due to weather conditions, water waves and motions of camera. The other challenge is revealed as updating the basis of subspace for

such methods when it comes to regularly refresh the background model. Therefore, one can emphasize that considering N frames as background model is better than using a single background model in case of initialization stage.

Moreover, the execution time of proposed algorithms (SWCD and CVABS) is reported as approximately 0.1 second per each image having size 240x320 pixels. The execution time includes processes of reading and writing of images, distance computation, segmentation, post processing, update of internal parameters and update of backgrounds. It should be noted that all the experiments have carried out on the same hardware with a software implemented on the Matlab. The elapsed running time is sufficient for a real time application if it is implemented in OpenCV framework with C++ programming language.

## REFERENCES

- [1] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, CDnet 2014: an expanded change detection benchmark dataset, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2014, pp. 387-394.
- [2] T. Bouwmans, Subspace learning for background modeling: A survey, Recent Patents on Computer Science, 2 (2009) 223-234.
- [3] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Computer Science Review, 11 (2014) 31-66.
- [4] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of gaussians for foreground detection-a survey, Recent Patents on Computer Science, 1 (2008) 219-237.
- [5] B. Lee, M. Hedley, Background estimation for video surveillance, (2002).
- [6] N.J. McFarlane, C.P. Schofield, Segmentation and tracking of piglets in images, Machine vision and applications, 8 (1995) 187-193.
- [7] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Comparative study of background subtraction algorithms, Journal of Electronic Imaging, 19 (2010) 033003-033003-033012.
- [8] J. Zheng, Y. Wang, N. Nihan, M. Hallenbeck, Extracting roadway background image: Mode-based approach, Transportation Research Record: Journal of the Transportation Research Board, (2006) 82-88.
- [9] D.E. Butler, V.M. Bove, S. Sridharan, Real-time adaptive foreground/background segmentation, EURASIP Journal on Advances in Signal Processing, 2005 (2005) 841926.

- [10] K. Kim, T.H. Chalidabhongse, D. Harwood, L. Davis, Background modeling and subtraction by codebook construction, Image Processing, 2004. ICIP'04. 2004 International Conference on, IEEE2004, pp. 3061-3064.
- [11] N.M. Oliver, B. Rosario, A.P. Pentland, A Bayesian computer vision system for modeling human interactions, IEEE transactions on pattern analysis and machine intelligence, 22 (2000) 831-843.
- [12] F. De La Torre, M.J. Black, A framework for robust subspace learning, International Journal of Computer Vision, 54 (2003) 117-142.
- [13] J. Rymel, J. Renno, D. Greenhill, J. Orwell, G.A. Jones, Adaptive eigen-backgrounds for object detection, Image Processing, 2004. ICIP'04 2004 International Conference on, IEEE2004, pp. 1847-1850.
- [14] J. Xu, V.K. Ithapu, L. Mukherjee, J.M. Rehg, V. Singh, GOSUS: Grassmannian online subspace updates with structured-sparsity, Proceedings of the IEEE International Conference on Computer Vision 2013, pp. 3376-3383.
- [15] C. Hage, M. Kleinsteuber, Robust PCA and subspace tracking from incomplete observations using  $\ell_0$ -surrogates, Computational Statistics, 29 (2014) 467-487.
- [16] X. Li, W. Hu, Z. Zhang, X. Zhang, Robust foreground segmentation based on two effective background models, Proceedings of the 1st ACM international conference on Multimedia information retrieval, ACM2008, pp. 223-228.
- [17] S.S. Bucak, B. Günsel, O. Gursoy, Incremental Non-negative Matrix Factorization for Dynamic Background Modelling, PRIS2007, pp. 107-116.
- [18] M. Yamazaki, G. Xu, Y.-W. Chen, Detection of moving objects by independent component analysis, Computer Vision-ACCV 2006, (2006), 467-478.

- [19] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: Real-time tracking of the human body, *IEEE Transactions on pattern analysis and machine intelligence*, 19 (1997) 780-785.
- [20] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, 1999. *IEEE Computer Society Conference on*, IEEE1999, pp. 246-252.
- [21] O. Barnich, M. Van Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, *IEEE Transactions on Image processing*, 20 (2011) 1709-1724.
- [22] M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: The pixel-based adaptive segmenter, *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012 *IEEE Computer Society Conference on*, IEEE2012, pp. 38-43.
- [23] P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin, Subsense: A universal change detection method with local adaptive sensitivity, *IEEE Transactions on Image Processing*, 24 (2015) 359-373.
- [24] P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin, A self-adjusting approach to change detection based on background word consensus, *Applications of Computer Vision (WACV)*, 2015 *IEEE Winter Conference on*, IEEE 2015, pp. 990-997.
- [25] Y. Chen, J. Wang, H. Lu, Learning sharable models for robust background subtraction, *Multimedia and Expo (ICME)*, 2015 *IEEE International Conference on*, IEEE2015, pp. 1-6.
- [26] R. Wang, F. Bunyak, G. Seetharaman, K. Palaniappan, Static and moving object detection using flux tensor with split gaussian models, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2014*, pp. 414-418.

- [27] H. Sajid, S.-C.S. Cheung, Universal multimode background subtraction, *IEEE Transactions on Image Processing*, 26 (2017) 3249-3260.
- [28] M. Babae, D.T. Dinh, G. Rigoll, A deep convolutional neural network for background subtraction, *arXiv preprint arXiv:1702.01731*, (2017).
- [29] Y. Wang, Z. Luo, P.-M. Jodoin, Interactive deep learning method for segmenting moving objects, *Pattern Recognition Letters*, (2016).
- [30] L.A. Lim, H.Y. Keles, Foreground Segmentation Using a Triplet Convolutional Neural Network for Multiscale Feature Encoding, *arXiv:1801.02225*, (2018).
- [31] S. Ergin, M.B. Gulmezoglu, A novel framework for partition-based face recognition, *International Journal of Innovative Computing Information and Control*, 9 (2013) 1819-1834.
- [32] S. Günal, S. Ergin, M.B. Gülmezoğlu, Ö.N. Gerek, On feature extraction for spam e-mail detection, *International Workshop on Multimedia Content Representation, Classification and Security, Springer2006*, pp. 635-642.
- [33] K. Özkan, E. Seke, Image denoising using common vector approach, *IET Image Processing*, 9 (2015) 709-715.
- [34] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, *IEEE Transactions on pattern analysis and machine intelligence*, 27 (2005) 4-13.
- [35] M. Gulmezoglu, V. Dzhafarov, A. Barkana, The common vector approach and its relation to principal component analysis, *IEEE Transactions on Speech and Audio Processing*, 9 (2001) 655-662.



- [36] Ş. Işık, K. Özkan, Ö.N. Gerek, M. Doğan, A new subspace based solution to background modelling and change detection, *International Journal of Intelligent Systems and Applications in Engineering*, 4 (2016) 82-86.
- [37] M.B. Gülmezoğlu, V. Dzhabarov, R. Edizkan, A. Barkana, The common vector approach and its comparison with other subspace methods in case of sufficient data, *Computer Speech & Language*, 21 (2007) 266-281.
- [38] Ş. Işık, K. Özkan, M. Doğan, Ö.N. Gerek, A Note on Background Subtraction by Utilizing a New Tensor Approach, *International Journal of Intelligent Systems and Applications in Engineering*, 4 (2016) 87-91.
- [39] G. Allebosch, F. Deboeverie, P. Veelaert, W. Philips, EFIC: edge based foreground background segmentation and interior classification for dynamic camera viewpoints, *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer2015, pp. 130-141.
- [40] A. Agrawal, R. Raskar, R. Chellappa, Edge suppression by gradient field transformation using cross-projection tensors, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), IEEE 2006, pp. 2301-2308.
- [41] Ş. Işık, K. Özkan, S. Günal, Ö.N. Gerek, SWCD: a sliding window and self-regulated learning-based background updating method for change detection in videos, *Journal of Electronic Imaging*, 27 (2018) 023002.
- [42] D. Raburn, E. Ratner, Fractal-based analysis for foreground detection, *Signals, Systems and Computers*, 2015 49th Asilomar Conference on, IEEE 2015, pp. 1393-1397.
- [43] M.E. Farmer, Robust pre-attentive attention direction using chaos theory for video surveillance, *Applied Mathematics*, 4 (2013) 43.

- [44] X. Yi, N. Ling, Fast pixel-based video scene change detection, *Circuits and Systems*, 2005. ISCAS 2005. IEEE International Symposium on, IEEE2005, pp. 3443-3446.
- [45] M. Gulmezoglu, V. Dzhafarov, A. Barkana, The common vector approach and its relation to principal component analysis, *Speech and Audio Processing*, IEEE Transactions on, 9 (2001) 655-662.
- [46] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 27 (2005) 4-13.
- [47] S. Günal, S. Ergin, Ö.N. Gerek, Spam E-mail Recognition by Subspace Analysis, *INISTA–International Symposium on Innovations in Intelligent Systems and Applications 2005*, pp. 307-310.
- [48] K. Özkan, Ş. Işık, A novel multi-scale and multi-expert edge detector based on common vector approach, *AEU-International Journal of Electronics and Communications*, 69 (2015) 1272-1281.
- [49] A. Agrawal, R. Raskar, R. Chellappa, Edge suppression by gradient field transformation using cross-projection tensors, *Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on, IEEE2006, pp. 2301-2308.
- [50] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: Principles and practice of background maintenance, *Computer Vision*, 1999. The Proceedings of the Seventh IEEE International Conference on, IEEE 1999, pp. 255-261.
- [51] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, *P nder: Real-time tracking of the human body*. Media Lab 353, MIT 1995.

- [52] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, 1999. IEEE Computer Society Conference on., IEEE 1999.
- [53] A. Elgammal, D. Harwood, L. Davis, Non-parametric model for background subtraction, *European conference on computer vision*, Springer2000, pp. 751-767.
- [54] D.-M. Tsai, S.-C. Lai, Independent component analysis-based background subtraction for indoor surveillance, *IEEE Transactions on image processing*, 18 (2009) 158-167.