Doctor of Philosophy

Efficient Change Detectors for Intelligent Video Analytics

The Graduate School

of the University of Ulsan

Department of Electrical and Computer Engineering

Ajmal Shahbaz

Efficient Change Detectors for Intelligent Video Analytics

Supervisor: Kang-Hyun Jo

A Dissertation

Submitted to

the Graduate School of the University of Ulsan

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

by

Ajmal Shahbaz

Department of Electrical and Computer Engineering

University of Ulsan

December 2020

Efficient Change Detectors for Intelligent Video Analytics

This certifies that the dissertation

of Ajmal Shahbaz is approved:

Committee Chair Prof. Kang-Hee Jun

Committee Member and Supervisor Prof. Kang-Hyun Jo

Committee Member Prof. Young-Soo Suh

Committee Member Prof. Hyun-Deok Kang

Committee Member Prof. Jang-Sik Park

Department of Electrical and Computer Engineering

University of Ulsan, South Korea

December, 2020

"Your Limitation- It's Only Your Imagination."

Tony Robbins

UNIVERSITY OF ULSAN

ABSTRACT

Graduate School of Electrical Engineering

Department of Electrical and Computer Engineering

Doctor of Philosophy

Efficient Change Detectors
for Intelligent Video Analytics

by Ajmal Shahbaz

Detecting an intruder that is trespassing a prohibited area is a critical task of intelligent video analytics. This task requires a change detector to segment an intruder (foreground object) from the background. The task suffers the inherent drawbacks of change detectors due to the dual camera sensors (color/IR), illumination changes, night time, static, and camouflaged foreground objects. This work proposes efficient unsupervised and supervised change detectors to compensate for the aforementioned challenges for intelligent video analytics particularly industrial sterile zone monitoring.

The camera switch detection based on skewness patterns detects a switch between the dual camera sensors (color/IR). The optimal color space selection based on the mean squared error will select tolerant color space (RGB/YCbCr) to illumination changes for modeling the background. Also, the IR camera frames are contrast-enhanced to tackle the camouflaged intruders during the night. The incoming frames are split into respective channels before modeling the background. The background is modeled by Gaussian Mixture Models (GMM). The adaptive background model update scheme is proposed to tackle the various challenges posed by environment and object such as a static foreground object.

Convolutional Neural Network (CNN) based algorithms have shown promise in dealing with the aforementioned challenges. However, they are exclusively focused on accuracy. This work goes on proposing an efficient supervised change detection algorithm based on atrous deep spatial features. The features are extracted using atrous convolution kernels to enlarge the field-of-view (FOV) of a kernel mask, thereby encoding rich context features without increasing the number of parameters. The network further benefits from a residual dense block strategy that mixes the mid and high-level features to retain the foreground information lost in low-resolution high-level features.

The extracted features are expanded using a novel pyramid upsampling network. The feature maps are upsampled using bilinear interpolation and pass through a 3x3 convolutional kernel. The expanded feature maps are concatenated with the corresponding mid and low-level feature maps from an atrous feature extractor to further refine the expanded feature maps. The experiments were performed on three standard change detection and video analytics databases. The proposed algorithms showed better performance than high-ranked unsupervised and supervised change detection counterparts on the three standard databases.

Acknowledgements

I wish to express my utmost gratitude to Prof. Kang-Hyun Jo for his constant support and guidance during my stay as a graduate student at the University of Ulsan. I would also thank the members of the thesis committee for their invaluable comments and suggestion to improve the content of my thesis.

I am grateful to the members of the Intelligent Systems Laboratory for the meaningful discussion and lesson that enhance my knowledge and broaden my understanding of the field of computer vision, artificial intelligence, and machine learning. Important thanks for the Brain Korea (BK21+) scholarship program that supported my monthly stipend and travel expenses to attend international conferences around the globe. It was not possible without the support and prayers of my parents back home. Finally, my grateful praise is for God, the Lord, and the Nourisher of the worlds.

Contents

List of Figures

List of Tables

List of Abbreviations

IVA	**I**ntelligent **V**ideo **A**nalytic
SZM	**S**terile **Z**one **M**onitoring
CD	**C**hange **D**etection
CSD	**C**amera **S**witch **D**etection
FOV	**F**ield **O**f **V**iew
ASFE	**A**trous **S**patial **F**eature **E**xtraction
PUPN	**P**yramid **UP**sampling **N**etwork
AI	**A**rtificial **I**ntelligence
CV	**C**omputer **V**ision
CASIA	**C**hinese **A**cademy of **S**ciences, **I**nstitute of **A**utomation
CCTV	**C**losed **C**ircuit **T**ele**v**ision
CNN	**C**onvolution **N**eural **N**etwork
CUDA	**C**ompute **U**nified **D**evice **A**rchitecture
CPU	**C**entral **P**rocessing **U**nit
FLOP	**FL**oating-point **OP**eration
FPN	**F**eature **P**yramid **N**etwork
FPS	**F**rame **P**er **S**econd
GPU	**G**raphic **P**rocessing **U**nit
MNIST	**M**odified **N**ational **I**nstitute of **S**tandards and **T**echnology
SSD	**S**ingle **S**hot **D**etector
SVM	**S**upport **V**ector **M**achine
RGB	**R**ed **G**reen **B**lue
ROI	**R**egion **O**f **I**nterest
VGA	**V**ideo **G**raphic **A**rray
VGG	**V**isual **G**eometry **G**roup
CDNet	**C**hange **D**etection

List of Symbols

K	number of Gaussian
ω	variance
μ	mean
σ^2	variance
σ	standard deviation
α	learning rate
β	second learning rate
L	adaptive learning rate
T	CSD threshold
F_c	aggragated foreground mask
λ	foreground detection threshold
B	background model
H	height
W	width
t	time
L	loss function
Th	threshold
R	recall
P	precision
F	F-measure

Dedicated to
my family

Chapter 1

Introduction

1.1 Motivation and Background

Intelligent video analytics (IVAs) play an important role in protecting sensitive areas. They are proving helpful not only for detecting anomalies but also for tracking subjects of interest. Smart cities are employing IVAs for public safety. According to statistics, some smart cities have 20-100 cameras per 1000 people (Shahbaz et al., 2017). Earlier surveillance systems require authorized personnel to closely monitor the footage for a possible security breach and required a high level of focus from security personnel. Thus, it is important to automate said surveillance systems.

IVAs are taking over their conventional counterparts. They are autonomous due to the implementation of computer vision algorithms (Bouwmans et al., 2019). They detect anomalies and alert security personnel. Such systems are widely implemented at borders and in industrial complexes to monitor restricted areas. These camera-based surveillance systems implement a change detection algorithm as a building block of high-level tasks, e.g., sterile zone monitoring (Fig. 1.1). Thus, change detection directly affects the overall performance of the system.

Sterile zone monitoring is a crucial task of IVAs to detect intruders/trespassers in a prohibited area. The definition of a sterile zone depends on the application (Shahbaz, Hariyono, and Jo, 2015). It could be a border between countries (Shahbaz, Hoang, and Jo, 2019), a fence of a prison, or a rooftop of a skyscraper (Zhang et al., 2013). Thus, sterile zone monitoring can be utilized in a wide range of prohibited areas.

Sterile zone monitoring employs the change detection algorithm to segment a

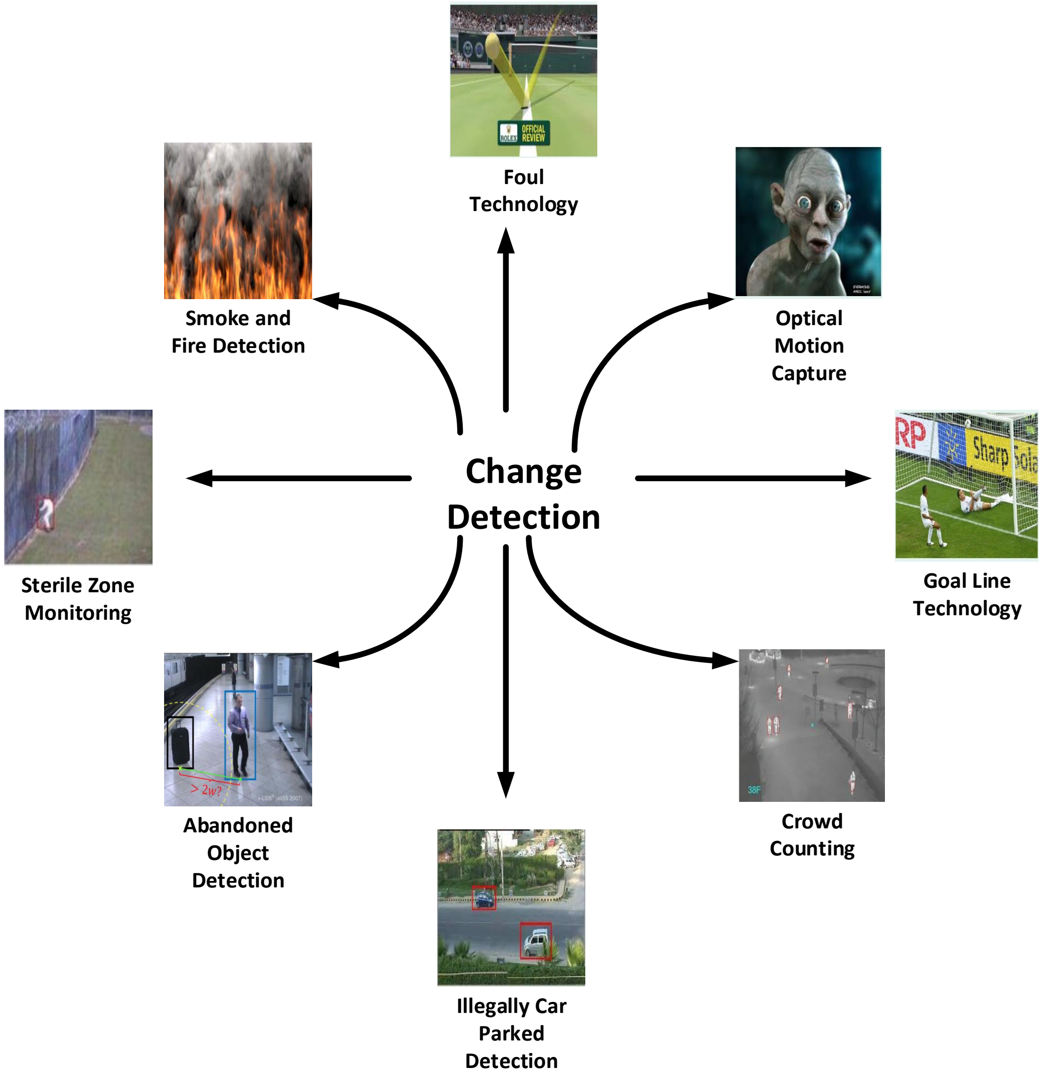


FIGURE 1.1: Change detection as a prepossessing task.

desired foreground object from the background (Fig. 1.2). The definition of a foreground object depends on the application. It could be a human walking in a corridor (Ullah et al., 2019), a car parked illegally on a road (Wahyono and Jo, 2017), a bag abandoned at a bus station (Wahyono, Filonenko, and Jo, 2016), smoke, or a fire in a forest (Filonenko, Hernandez, and Jo, 2018), etc. IVAs are challenged due to inherent and practical drawbacks faced by the change detectors due to dual camera sensors (color/IR), illumination changes, dynamic backgrounds, bad weather, static, and camouflaged foreground objects.

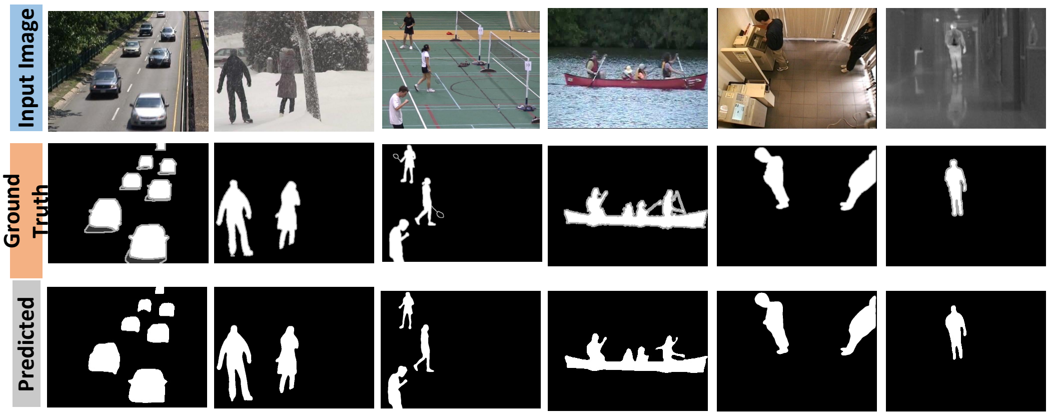


FIGURE 1.2: Change detection.

1.2 Problem Description and Objective

The task of change detection may seem trivial but it is quite challenging due to variations in environmental, object, and camera modalities (Liu and Yan, 2012). Furthermore, real-time performance requirement adds on to the challenges of the change detection task.

The outdoorsy nature of intelligent video analytics (IVAs) severely challenges change detection algorithms by environmental variables such as illumination changes, dynamic backgrounds, shadows, day, and night. Similarly, the foreground object brings more challenges such as variation in speed, variation in size of object, camouflaged object, and static foreground object (Candes et al., 2011).

Variation in speed is challenging for unsupervised change detectors where a learning rate is applied to model temporal variations in the background model. For example, a running foreground object is hard to detect if a low learning rate is employed. Also, a slow-moving foreground object is challenging with a high learning rate. Therefore, a balanced adaptive learning rate scheme is crucial to deal with IVAs.

If a foreground object enters a scene from the far side of the camera view, it would be equally challenging to detect it by unsupervised and supervised change detectors due to its small size. Similarly, the camouflage foreground object is a challenge where foreground pixel intensity is similar to the background. Unsupervised algorithm deploying threshold criterion to segment foreground information is failed.

A static foreground object is a challenge where an object enters a scene and stays static at a particular position. An example of such a challenge would be a suitcase left at an airport. Furthermore, an algorithm designed for particular camera modalities such as a color camera may fail to work well for an IR camera. Thus, the objective of this work is to tackle the aforementioned challenges for the unsupervised and supervised change detectors.

1.3 Contributions

This work contributes to unsupervised and supervised change detectors for the task of industrial sterile zone monitoring. The GMM (KaewTraKulPong and Bowden, 2002) based unsupervised change detector is improved to tackle its inherent drawbacks of dual camera sensors, illumination changes, static, and camouflaged foreground objects. The contributions of this work are as follows:

- A novel camera switch detection (CSD) scheme detects the switch between color and IR sensors.
- A novel optimal color space selection strategy to select illumination change-tolerant color space (RGB/YCbCr) for modeling the background.
- An efficient contrast enhancement scheme for enhancing IR camera frames to tackle camouflaged intruders at night.
- A novel adaptive background model update scheme for updating the background model to tackle the challenges of illumination changes, dynamic backgrounds, moving, and static foreground.

The second part of this work proposes an efficient supervised change detector foreground detection algorithm based on CNN and contributes in four folds:

- A new atrous spatial feature extractor (ASFE) and pyramid upsampling network (PUPN) is proposed. ASFE is designed using standard and atrous convolutional layers to enlarge field-of-view (FOV) without increasing the number of parameters. The mid and high-level features are intermixed via residual-dense blocks strategy (RD) to build a global context and retain the foreground information. PUPN is designed as a sandwich of 3x3 convolutions and bilinear interpolation, which ease the gradient flow during training. The mid and low-level features of ASFE are also propagated to PUPN to improve foreground extraction. SFDNet is trained via a hybrid training strategy.

- The proposed algorithm is tested on three datasets with more than 180,000 image sequences. The choice of network design is further supported by ablation studies to validate the effectiveness of network design.
- A dataset of HD videos in an industrial setting is developed. It is unique as it also provides videos shot with an IR camera at night. The dataset is supported by manually labeled ground-truth images for training the supervised algorithms. The dataset would be made public for the research community.
- IVAs are crucial for securing sensitive areas. The proposed algorithm adds-on with efficiency and efficacy and can be implemented on low-end hardware. However, the scope of the algorithm is not limited to the industrial domain. It can be extended to other applications such as airport ground surveillance, etc.

1.4 Disposition

This part explains the organization of the thesis. The following section is discussing various methods related to the unsupervised and supervised change detectors. Chapter 3 outlines and explains the proposed unsupervised change detector for intelligent video analytics (IVAs). The contributions of the proposed algorithm, for example, camera switch detection, contrast enhancement, adaptive learning scheme are explained in detail.

Chapter 4 explains the proposed supervised change detector for intelligent video analytics (IVAs). It includes a detailed explanation of the contributions outlined in chapter 1. The chapter is concluded with the experimental analysis. Chapter 5 explains the implementation details for the unsupervised and supervised change detectors. The computational complexity of both algorithms is outlined in the end. Finally, Chapter 6 presents the conclusion and the possible future directions of this research work.

Chapter 2

Literature Review

This chapter outlines and briefly discusses the current state-of-the-art in the field of change detection. The change detection algorithms are broadly classified into two categories based on the training process: unsupervised and supervised algorithms (Shahbaz et al., 2017). Unsupervised algorithms are trained online with incoming frames to construct a concrete background model using pixel intensity. The supervised algorithms based on Convolutional Neural Networks (CNN) are trained offline on GPUs with background and foreground information (Bouwmans et al., 2019).

2.1 Unsupervised Change Detection

The unsupervised algorithms mainly consists of three steps: background modeling, foreground detection, and background update/maintenance. Fig. 2.1 shows the overview of the unsupervised algorithm.

2.1.1 Gaussian Mixture Models (GMM)

GMM is the most famous parametric based algorithm. It was proposed by Stauffer and Grimson, 1999. Since then various improvements were published but the ones by Zivkovic (Zivkovic, 2004) and Kawtrakulpong (KaewTraKulPong and Bowden, 2002) are the famous ones. Both methods focus on improving background initialization. Zivkovic improved by automatic selecting number Gaussian K in the mixture and using k-means initialization of the background model. Kawtrakulpong used the EM algorithm for background initialization.

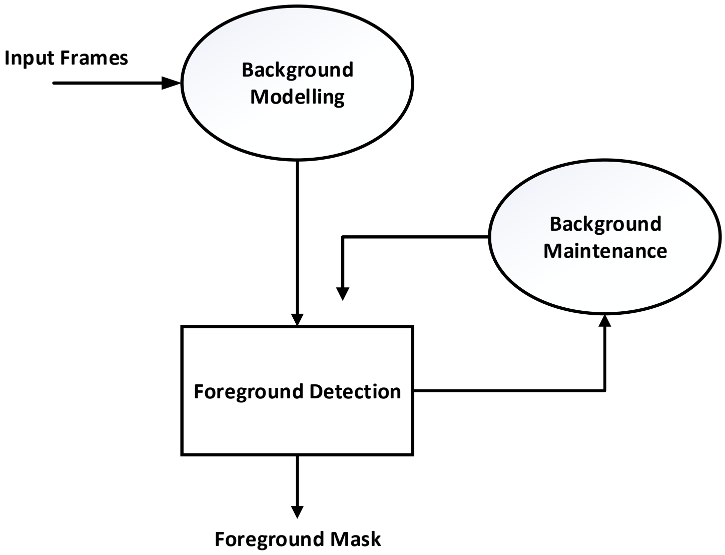


FIGURE 2.1: Overview of an unsupervised algorithm.

Each pixel is modeled as a mixture of Gaussian and classified as the foreground or background based on the mean and variance. The probability of observing a particular pixel value is:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(X_t;\mu_{i,t},\sigma^2_{i,t}), \tag{2.1}$$

where probability density function η is given as,

$$\eta(X_t;\mu_{i,t},\sigma^2_{i,t}) = \frac{1}{\sigma_{i,t}\sqrt{2\pi}}e^{-}(\frac{x-\mu_{i,t}}{2\sigma^2_{i,t}}), \tag{2.2}$$

where K, $\omega_{i},_{t}$, $\mu_{i},_{t}$, $\sigma^2_{i},_{t}$ and are number of Gaussian, estimate of weight, mean, and variance of the ith Gaussian in the mixture at time t. The decision criteria to mark particular pixel at time t as background or foreground is:

$$|X_t - \mu_{i,t}| > \lambda\sigma_{i,t}, \tag{2.3}$$

where λ is a constant threshold equal to 2.5. If a match is found with one of the K Gaussian components, the pixel is classified as background and its parameters are updated as:

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} + \alpha, \tag{2.4}$$

$$\mu_{i,t+1} = (1-\beta)\mu_{i,t+1} + \beta X_{t+1}, \tag{2.5}$$

$$\sigma^2_{i,t+1} = (1-\beta)\sigma^2_{i,t+1} + \beta(X_{t+1} - \mu_{i,t})(X_{t+1} - \mu_{i,t})^T, \tag{2.6}$$

$$\beta = \alpha\eta(X_{t+1}; \mu_{i,t+1}, \sigma^2_{i,t+1}), \tag{2.7}$$

where α and β are constant learning rate and second learning rate respectively. If no match is found with any of the K Gaussian, the pixel is classified as foreground and only weight is updated:

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} \tag{2.8}$$

$$\mu_{i,t+1} = \mu_{i,t} \tag{2.9}$$

$$\sigma^2_{i,t+1} = \sigma^2_{i,t} \tag{2.10}$$

2.1.2 Spatial Gaussian Mixture Models (SGMM)

SGMM (Chen, Wang, and Lu, 2015) is a spatio-temporal framework to specifically handle dynamic backgrounds. This work is the extension of traditional GMM (Stauffer and Grimson, 1999) which includes the spatial information taken from the pixel's neighborhood region. The update equations and maintenance procedures are quite similar to traditional GMM.

Consider the data $X = x_1, x_2, \ldots, x_n$ where is number of data samples. Let i be number data samples in the region and k be number of classes a data sample could belong to. $z_{i,k} = [0, 1]$,be the membership of data sample to a cluster at position i. $\theta = [\omega_{i,k}, \mu_{i,k}, \sigma^2_{ik}]$, be the parameters of model.

$$E_p(z_{i,k}|x_q, \theta^{old})(z_{i,k}) = \sum_{q \epsilon R_i} \gamma_q(z_{i,k}) \tag{2.11}$$

$$\gamma_q(z_{i,k}) = \frac{\omega_{i,k}\eta(x_q|\mu_{i,k}, \sigma^2_{i,k})}{\sum_{q\epsilon R_i}\sum_{j=1}^{K} \omega_j \eta(x_q|\mu_{i,j}, \sigma^2_{i,j})} \tag{2.12}$$

2.1.3 Shared GMM

Shared GMM (Chen, Wang, and Lu, 2015) uses the idea of traditional GMM with a sharable mechanism to exploit the spatial-temporal correlation between the pixels. Pixel values are modeled using Gaussian Mixture Models. The probability of observing certain pixel X at time t can be computed by Eq. 2.1.

The sharable mechanism demonstrates many to one relationship between pixel and models. For each pixel, optimal model is searched from background and foreground models in NxN region. Pixel labeling decision is given as follows:

$$L_B(x^t) = \begin{cases} 1, & |x^t - \mu^t_{B,k}| < 2.5\sigma^t_{B,k} \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

$$L_F(x^t) = \begin{cases} 1 & |x^t - \mu^t_{F,k}| < 2.5\sigma^t_{F,k} \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

A Conservative update strategy with a random sampling mechanism is applied for updating models. By randomly selecting the pixel that matches a model, the background and foreground model is updated. The foreground model is switched to the background if it is kept for a long period (for example, 500 frames). If the model is not used for a long time, then the model is deleted (500 frames for background and 50 frames for foreground).

2.1.4 Universal Background Subtraction System (UBSS)

UBSS (Sajid and Cheung, 2015) proposes universal background subtraction system. Their algorithm selects optimal color space i.e. RGB or YCbCr for the task of background subtraction. The intuition behind the choice of color space is related to how the human eye adjusts different lightening conditions. The human eye uses two different cells, rods and cones. Rods assists in low lighting condition and cones in high lighting conditions. RGB has the same role as rods and YCbCr acts like cones.

Initial frames without foreground objects are used to model background named Background Model Bank (BMB), tune system parameters, and finding optimal color space. BMB consists of single Gaussian models. Initial frames are clustered into N groups based on correlation measures using K-means. An appropriate background model is chosen using a correlation criterion.

Once the BG model is chosen it is passed to background subtraction modules (also known as binary classifiers, BC) along with an input image that produces binary mask D for each color channel of color space. All the BMs produced by BCs are aggregated into a foreground detection mask.

$$FGD_{mask}(X) = \sum_{D}(D_{mask,k}(X)) > t \tag{2.15}$$

If two color spaces are used then the logical AND of both color spaces. In this step, all the BMs are purged using an FDG mask and new BM. The foreground mask is obtained using the equation:

$$FGD_{mask}(X) = \sum_{D}(D^{new}_{mask,k}(X)) > t \tag{2.16}$$

If both color spaces are used, then FG is obtained by the logical OR of the FG mask for each of the color spaces.

2.1.5 Self-Balanced SENsitivity SEgmenter (SuBSENSE)

SuBSENSE (St-Charles, Bilodeau, and Bergevin, 2015) is a non-parametric model-based algorithm that exploits feature space namely Local Binary Similarity Patterns (LBSP). It detects change by comparing a center pixel with neighboring pixels. LBSP can be considered counterparts of Local binary pattern (LBP) and local ternary pattern (LTP). LBSP binary string can be computed using the following equation:

$$LBSP(x) = \sum_{p=0}^{P-1} d(i_n, i_c).2^p, \tag{2.17}$$

where i_n and i_c are neighboring pixels and center pixels respectively. LBSP is threshold in two ways: Absolute T_d and Relative T_r. Both have certain benefits that are explained in the paper.

$$d(i_n, i_c) = \begin{cases} 1, & |i_n - i_c| \leq T_d \\ 0, & \text{otherwise} \end{cases} \tag{2.18}$$

$$d(i_n, i_c) = \begin{cases} 1, & |i_n - i_c| \leq T_r.i_c \\ 0, & \text{otherwise} \end{cases} \tag{2.19}$$

A sample consensus approach is used. It determines if a given observation can be considered as a foreground or background based on its similarity with previously observed samples. First, background model B is formed through a combination of pixels, which each contain a set of N recent background samples.

A pixel is decided to be background if pixel value I is closer than a certain decision threshold R to at least $\sharp_{min}$ of the N background values. Essentially, a sample-based non-parametric statistical model that portrays the background at individual pixel locations (B(x)) using a set of N=50 past representations. When at least ($\sharp_{min} = 2$) samples intersect with representation of x at time t (say ($I_t(x)$), the pixel is labeled as background in the raw segmentation ($S_t(x) = 0$) otherwise it is labeled as foreground, ($S_t(x) = 1$). The way these background samples are updated randomly replaced by local values after the segmentation step, but only when ($S_t(x) = 0$).

2.1.6 Pixel based Adaptive Word Consensus Segmenter (PAWCS)

PAWCS (St-Charles, Bilodeau, and Bergevin, 2016) is another non-parametric method based on SuBSENSE. This method's key advantages lie in its highly persistent and robust dictionary models based color and local binary features as well as its ability to automatically adjust pixel-level segmentation. A word-based approach is implemented for the monitoring of background representations at the pixel level without clustering. These appearances of pixels over time are termed as "background words" in local dictionaries using color and texture information.

If the representation occurs persistently then it is termed as good representations of background. Infrequent representations are discarded and replaced by better alternatives. Similar to SuBSENSE, they use LBSP to compute the neighborhood of centering pixels. However, relative thresholding is applied only, wherein SuBSENSE, absolute and relative thresholding was used. The sensitivity thresholds and learning rate used in segmentation results and model update rules are adjusted dynamically to improve and maintain background representations.

2.1.7 Flux Tensor and Split Gaussian (FTSG)

FTSG (Wang et al., 2014a) is a hybrid foreground detection method that uses motion, change, and appearance information. FTSG uses a split Gaussian method to separately model foreground and background. It consists of three main modules:

Pixel level motion detection module, fusion module, and object-level classification module (Jiang and Lu, 2018).

A multichannel flux tensor is used to detect motion. Flux tensor is convenient to compute motion information directly without expensive eigenvalue decomposition. Flux tensor represents the temporal variation of the optical flow field within the local 3D spatiotemporal volume. Trace of flux tensor matrix can be used to classify moving and non-moving regions.

Split Gaussian is used to model background and foreground. The mixture of Gaussian is used to model background and every new pixel is checked against Gaussian distribution. A pixel is labeled as foreground if it does not match with any of the Gaussian. Single Gaussian is used to model the foreground. The foreground appearance model is used to distinguish static foreground from noise.

The authors claim that the background model can be initialized using the first few frames and the foreground appearance model is initialized to be empty. They follow a blind update mechanism like that of traditional GMM. Foreground mask obtained by flux tensor and split Gaussian are fused using a rule based system to produce improved results. In the object-level classification module, removed and stopped objects are handled. edges of static objects are compared with the edges of the corresponding object in the current image and background model using chamfer matching.

2.1.8 In Unity There Is Strength (IUTIS)

IUTIS (Bianco, Ciocca, and Schettini, 2017) a framework utilizing the ability of genetic programming (GP) to combine several state-of-the-art algorithms. GP algorithm automatically selects the best algorithms, combine them in different ways, perform the most suitable post-processing operations on the output of the algorithms. Unary, binary, and n-ary functions are embedded in the GP framework for combining particular algorithms. In this way, the algorithm once ranked 1st in the change detection dataset.

The authors claim that they have combined 22 algorithms with the subsets of 3, 5, and 7 algorithms. As claimed by authors, the benefit of using genetic programming is threefold:

- Automatic selection of algorithms that gives the best results.
- Automatic deduction of ways to select the algorithms to generate intermediate masks.
- Automatic selection of kind post-processing by using unary, binary, and n-ary functions.

2.2 Supervised Algorithms

Supervised algorithms are trained offline with ground-truth using Convolutional Neural Networks (CNNs) as the feature extractor (Fig. 2.2). The trained model is then used to evaluate the remaining video frames.

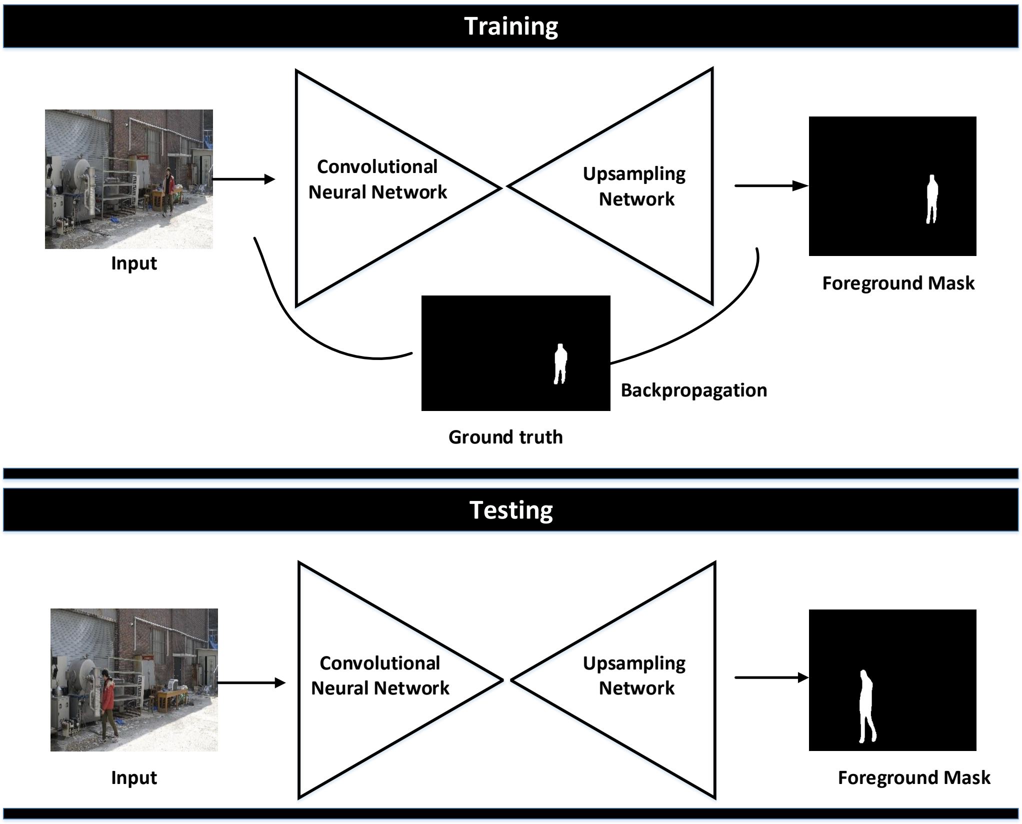


FIGURE 2.2: Overview of a supervised algorithm.

2.2.1 Patch based Deep Background Subtraction (DBS)

DBS (Braham and Droogenbroeck, 2016) labels a patch of an image as a foreground or background. The network is trained with a background image as a ground-truth.

It uses 50% of a video sequence as training and the remainder to test. The algorithm provides promising prospects and a possible solution to the drawbacks inherent to unsupervised algorithms.

2.2.2 Deep Background Subtraction (DeepBS)

DeepBS (Babaee, Dinh, and Rigoll, 2018) trains multiple CNNs with multiscale input images. It trains only one model with 5% of all the frames of the video sequence from the change detection dataset (CDNet). DBS and DeepBS train a CNN with a background model as ground-truth. Thus, such algorithms fail to deal with cluttered scenes. Additionally, CNN weights are learned from scratch, which is time inefficient.

2.2.3 Cascade CNN (CascadeCNN)

CascadeCNN (Wang, Luo, and Jodoin, 2017) takes input images in three different scales and feeds them into three CNNs using foreground/background labels as ground-truth. The feature maps are upsampled using bilinear interpolation. It also trains CNN weights from scratch.

2.2.4 Real-time Background Subtraction (RT-BGS)

RT-BGS (Cioppa, Droogenbroeck, and Braham, 2020) builds probability-based background and foreground models using a pretrained CNN model. The two separate models for background and foreground entities are maintained and updated using difference of probability between two consecutive frames. This procedure helped to address static and intermittent object motion.

2.2.5 Multi-scale Dilated Convolution (M2DC) based Change Detector

Several algorithms tried to learn long-term foreground dependencies using spatio-temporal features. M2DC4 (Hu et al., 2018) proposes a dilated CNN with convolutional long-term short-term memory networks (ConvLSTM). The visual graphical group (VGG-16) based CNN was employed as a feature extractor.

The VGG-16 introduced the concept of a block to explain the effective receptive field (RF) of stacked convolutional filters (2.3). A block in VGG-16 is defined as a stack of two or three convolutional layers. The first two blocks employ a stack of two 3×3 and 3rd-5th blocks stacked three 3×3 convolution filters.

The RF of two 3×3 stacked convolutional filters is effectively equal to the RF of a 5×5 convolutional filter. Such a scheme reduces the number of weights/parameters required by 30%. Similarly, the RF of three 3×3 stacked convolutional filters is effectively equal to the RF of a 7×7 convolutional filter. Rectified Linear Unit (ReLU) is applied after each convolution layer. It gets rid of negative values in the feature map. The input is zero padded to maintain spatial dimensions after the convolutions operation.

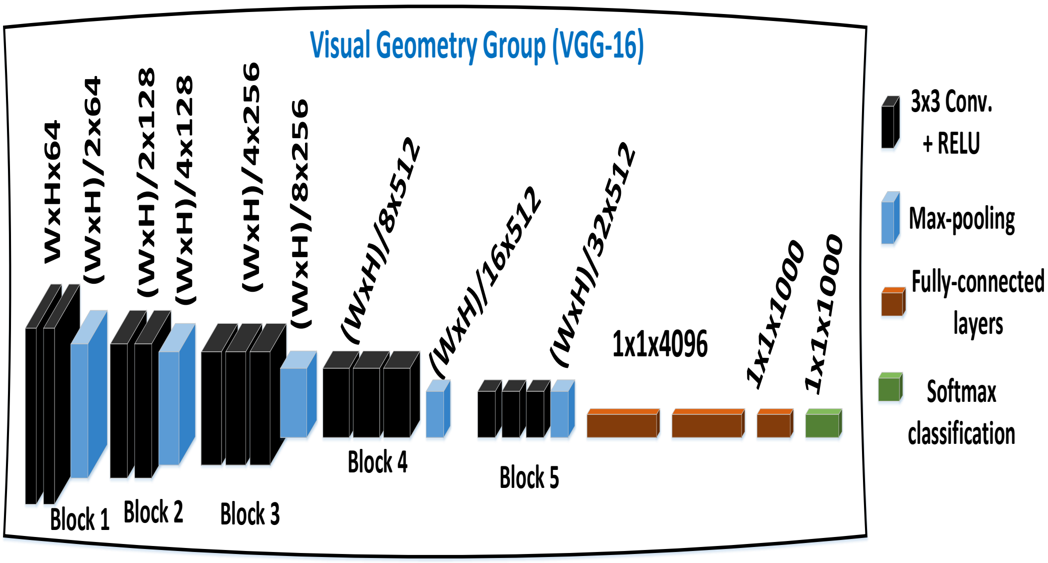


FIGURE 2.3: VGG.

2.2.6 Multi-Scale Spatio-Temporal (MS-ST) based Change Detector

MS-ST (Yang et al., 2019) employed multi-scale spatio-temporal features using a pre-trained CNN and ConvLSTM. Similar to M2DC, MS-ST also employs VGG-16 network without the fully connected layers. Also, only four blocks were applied with three max-pooling layers (Fig. 2.3). This was due to deal with small size foreground object.

2.2.7 Multi-view Receptive Field (MvRF-CNN) based change detector

MvRF-CNN (Akilan, Wu, and Zhang, 2019), similar to M2DC4, employed dilated CNN with ConvLSTM for traffic surveillance applications. While pretrained models might be fast to train, the network might not properly learn the foreground object in a particular video. Their deployment into real-time systems is challenging due to the high computational complexity, a large amount of data needed for training, and a high-end hardware requirement.

2.2.8 3D CNN-LSTM based change detector

3D CNN-LSTM (Akilan et al., 2020) tried to decrease the number of sequential frames to 4 with grayscale input. Pretrained model and 3D convolutional filters were used. The residual Network (ResNet-50) was applied to extract features from input (Fig. 2.4). ResNet solved the problem of network convergence and gradient flow issue in the deeper networks. Still, it takes 80-120 minutes to train a single model. These ConvLSTM based algorithms require sequential input with labeled ground-truths for training (70%). For instance, MS-ST used 14 sequences of images, the most that could fit into NVIDIA 1080Ti GPU memory (Yang et al., 2019).

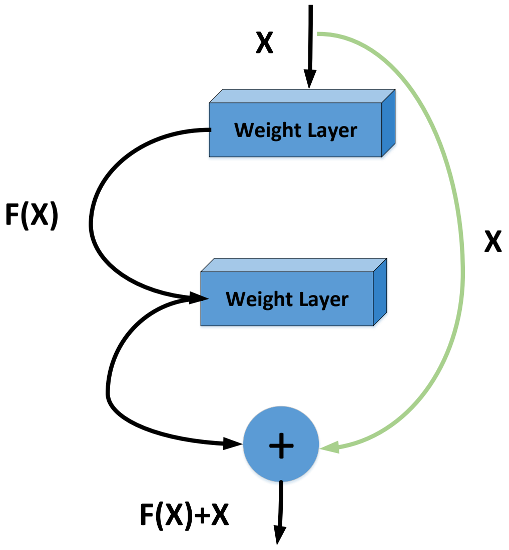


FIGURE 2.4: ResNet.

2.3 Datasets Description

The proposed unsupervised and supervised algorithms are tested on three different datasets: the change detection dataset (Wang et al., 2014b), the i-LIDS dataset (Branch, 2006), and the ISL-ISZM dataset. The datasets pose practical challenges an industrial surveillance system may face such as illumination changes, dynamic backgrounds, shadows, bad weather, thermal camera, moving camera, camera jitter, infra-red (IR) camera, camouflaged foreground object, static foreground object, scale, and speed-variance of an object.

2.3.1 Change Detection Dataset

Chang detection dataset (CDNet) is the largest and most realistic dataset in the field of foreground detection. The dataset consists of 54 videos ($>$ 150,000 frames) consisting of 11 different categories. The videos are being captured by color and thermal cameras. Furthermore, the dataset provides ground truth for each frame. Each frame is annotated as a foreground, background, and shadow. Python and MATLAB code is provided for quantitative analysis. The challenges posed by the dataset are summarized below:

- Baseline: Similar to a baseline of the racing court. This category consists of a mixture of challenges that will come up in later categories of the dataset. These videos cover different scenarios such as small background motion, abandoned object, and slow or static foreground object, etc.

- Dynamic Backgrounds: These videos contain background with repetitive motion such as moving branches of the tree, boats in the water, foreground object moving in front of the water fountain, etc.

- Camera Jitter: These videos contain camera motion due to disturbance such as when the camera is not fixed rigidly, and then there is subtle motion. This jitter varies from video to video.

- Intermittent Object Motion: These videos consist of scenes where foreground object move, then stop for a short while, enough to dissolve it in the background, and then start moving again.

- Shadows: These categories consist of shadows cast by moving or background objects.
- Thermal: These videos are being captured by far-infrared cameras. Due to the use of thermal cameras, there might be a camouflage effect.
- Challenging Weather: outdoor videos captured in challenging winter conditions, blizzard, Snow, fog, etc.
- Low Frame Rate: contains varying frame rate videos between 0.17 fps and 1 fps.
- Night Videos: videos captured at night with difficult light conditions.
- Pan Tilt Zoom (PTZ): videos captured by pan tilt zoom cameras in slow continuous pan mode, intermittent pan mode, 2 position patrol mode PTZ, zooming-in, and zooming out.
- Turbulence: videos showing air turbulence caused by rising heat

2.3.2 i-LIDS Dataset

The Imagery Library for Intelligent Detection Systems (i-LIDS) dataset is the standard benchmark for video surveillance systems. There are 10 videos with 1000 frames each, five for the day, and five for the night. The image size is 390×220. The scenario is an intruder entering a restricted area and trying to bypass the fence.

2.3.3 ISL-ISZM Dataset

Intelligent Systems Laboratory dataset for Industrial Sterile Zone Monitoring (ISL-ISZM) has 15 videos, 10 for the day, and 5 for the night with 1000-2300 frames. The image size is 720×480. The videos were constructed by mimicking the i-LIDS dataset challenges. The scenario consists of an intruder entering a restricted area in an industrial setting. The dataset and training frames can be found at `https://drive.google.com/file/d/1QWCZBa6DIbIK8p0qjsDrqr-lmy0kq7-C/view?usp=sharing`. Table 2.1 shows a detailed description of the i-LIDS and ISL-ISZM datasets with challenges.

TABLE 2.1: Datasets Description.

Video name	Duration	Time	Scenario
i-LIDS Dataset			
1	0:30	Day	Walking
2	0:30	Day	Running
3	0:30	Day	Crawling
4	0:30	Day	Walking slowly
5	0:30	Day	Walking fast
6	0:30	Night	Walking away from camera
7	0:30	Night	Walking away from camera slowly
8	0:30	Night	Far from camera
9	0:30	Night	Camouflage intruder
10	0:30	Night	Camouflage intruder
ISL-ISZM Dataset			
11	1:21	Day	Normal Walk
12	0:54	Day	Walking fast
13	0:30	Day	Running
14	0:30	Day	Slow walking
15	1:35	Day	Multiple intruders
16	1:21	Day	Normal Walk
17	0:30	Day	Running
18	0:30	Day	Slow walking
19	0:50	Day	Dynamic background
20	0:58	Day	Dynamic background
21	0:16	Night	Walking fast
22	0:14	Night	Camouflage intruder
23	0:31	Night	Camouflage intruder
24	0:25	Night	Camouflage intruder
25	0:35	Night	Multiple intruders

2.4 Performance Metrics

The quantitative comparison is performed using seven performance metrics defined by change detection dataset. These metrics are recall R, precision P, specificity Sp, F-measure F, false positive rate FPR, false negative rate FNR, and percentage of wrong classification PWC. Let TP= true positive, FP= false negative, TN= true negative, and FN= false negative. The performance metrics can be defined as:

- TP: A pixel is correctly labeled as foreground.
- FP: A pixel is incorrectly labeled as foreground.
- TN: A pixel is correctly labeled as background .
- FN: A pixel is incorrectly labeled as background .

$$\mathrm{R} = \frac{\mathrm{TP}}{(\mathrm{TP+FN})} \tag{2.20}$$

$$\mathrm{P} = \frac{\mathrm{TP}}{(\mathrm{TP+FP})} \tag{2.21}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{(\mathrm{TN+FP})} \tag{2.22}$$

$$\mathrm{F} = \frac{2 \times (\mathrm{P} \times \mathrm{R})}{(\mathrm{P+R})} \tag{2.23}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP+TN})} \tag{2.24}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{(\mathrm{FP+TN})} \tag{2.25}$$

$$\mathrm{PWC} = \frac{100 \times (\mathrm{FP+FN})}{(\mathrm{TP+FN+FP+TN})} \tag{2.26}$$

Chapter 3

Enhanced Unsupervised Change Detector

This chapter focuses on the contribution of this work in the domain of unsupervised change detector. The contributions outlined in chapter 1 are explained in detail. The proposed algorithm consists of two modules, as shown in Fig. 3.1:

1. Enhancement Module: The type of input frames (Color/IR) is detected using camera switching detection. Later, the input is pass through optimal color space selection to select the optimal color space (RGB/YCbCr) to tackle illumination changes. Also, the IR input is contrast-enhanced to distinguish camouflaged intruders from the background.
2. Change Detection Module: The enhanced input is then used to model the background and detect the foreground. The background model is updated automatically during the whole process. The foreground mask is purged to get the final result.

3.1 Enhancement Module

3.1.1 Camera Switching Detection (CSD)

Current Intelligent Video Analytics (IVAs) employ dual camera sensors for the day (color) and night (IR). Changes in sunlight intensity cause a switch between the camera sensors signaling the time of day. Such a scheme while economical comes with

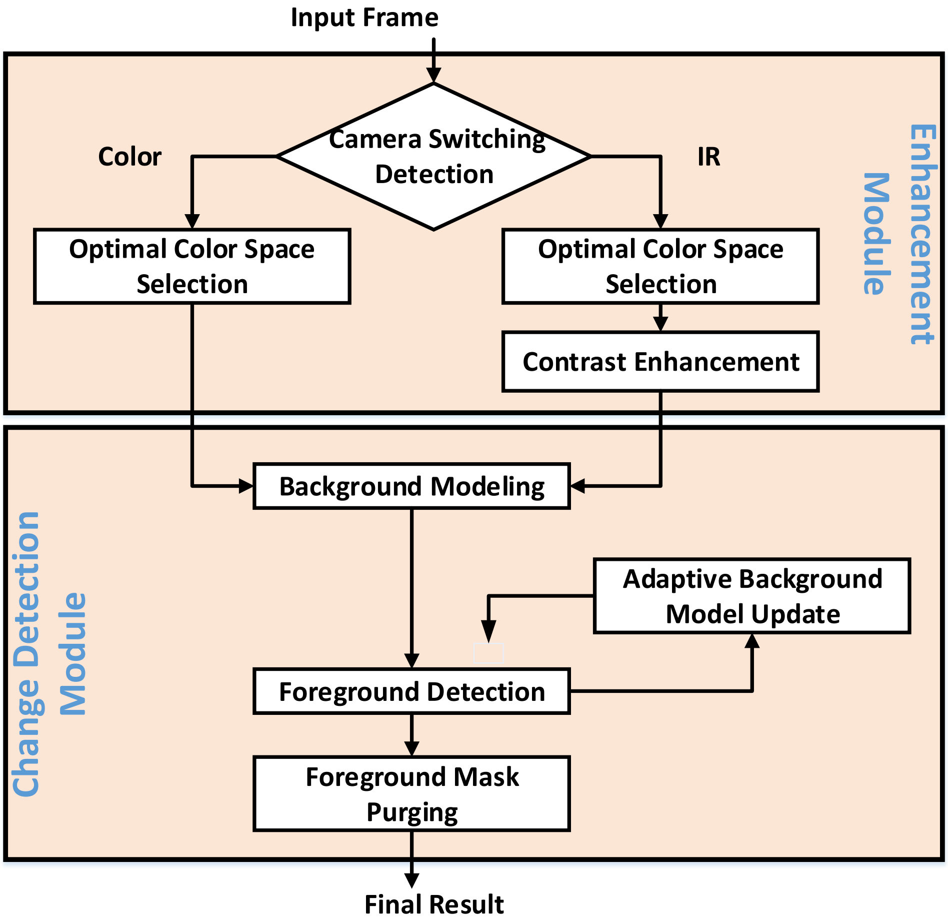


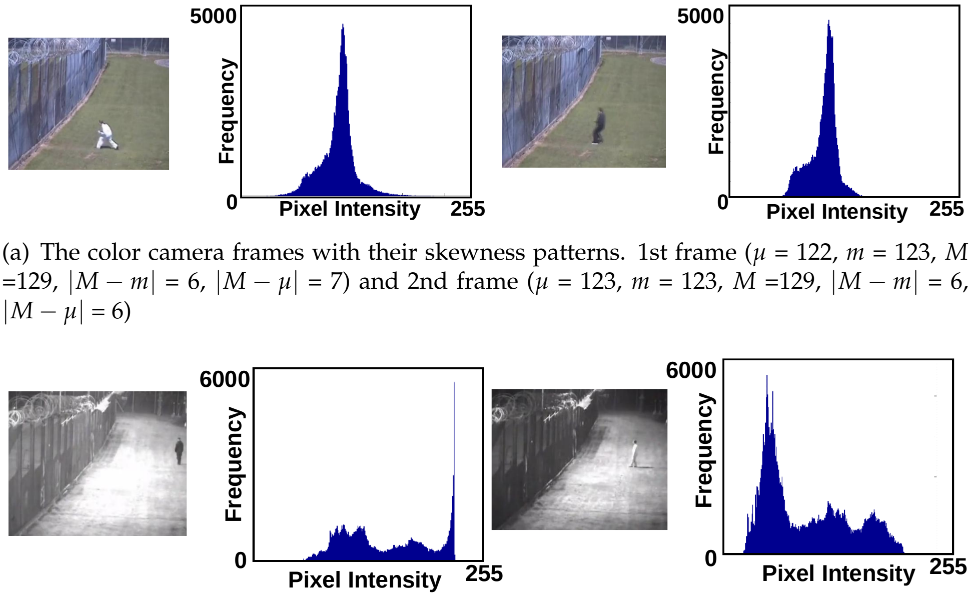
FIGURE 3.1: Overview of the proposed method.

severe drawbacks. The switch between sensors may distort the unsupervised change detector which exploits the background model to segment an intruder from a scene. Such distortion results in false positives. Also, the IR sensor may pose a strong camouflage effect. Due to pixel intensity based background modeling, it could lead to false negatives resulting in IVAs failure.

A novel camera switching detection (CSD) scheme has been proposed to tackle the aforementioned challenges. The premise is derived from the skewness patterns of the color and IR camera. The color camera gives balanced information about a scene and a varied range of intensity. While the IR camera provides information in shades of gray and a congested intensity range. Thus, the skewness patterns of both camera sensors differ remarkably and follow these three patterns:

1. If $\mu = m = M$, it is classified as symmetry.

2. If $\mu > m > M$, it is classified as left-skewed.



(a) The color camera frames with their skewness patterns. 1st frame (μ = 122, m = 123, M =129, $|M - m|$ = 6, $|M - \mu|$ = 7) and 2nd frame (μ = 123, m = 123, M =129, $|M - m|$ = 6, $|M - \mu|$ = 6)

(b) The IR camera frames with their skewness patterns. 1st frame (μ = 166, m = 156, M =254, $|M - m|$ = 98, $|M - \mu|$ = 88) and 2nd frame (μ = 120, m = 102, M =64, $|M - m|$ = 38, $|M - \mu|$ = 56).

FIGURE 3.2: Skewness patterns exhibited by the color and IR camera, where x- axis and y-axis shows pixel intensity and frequency respectively.

3. If $\mu < m < M$, it is classified as right-skewed.

where μ, m, and M are mean, median, and mode of an image respectively. Fig. 3.2 visualizes the skewness patterns in the day (color) and night (IR) frames. The day frames follow a nearly symmetrical pattern (Fig. 3.2 a), whereas the night frames exhibit either left or right skewness (Fig. 3.2 b). Following the symmetrical pattern, the mean μ, median m, and mode M of the day frames were approximately equal (Fig. 3.2 a). However, the night frames showed that the mean μ, median m, and mode M were far apart and followed either a left or a right skewed pattern (Fig. 3.2 b).

The CSD criterion is formulated from three skewness patterns to detect the switch from a color to an IR camera sensor. The criterion can be written as:

$$CSD = \begin{cases} IR, & |M - m| \geq T \vee |M - \mu| \geq T \\ Color, & otherwise \end{cases} \tag{3.1}$$

where T is the CSD threshold selected heuristically. The mean μ, median m, and mode M are scalar entities averaged over the three image channels. For example,

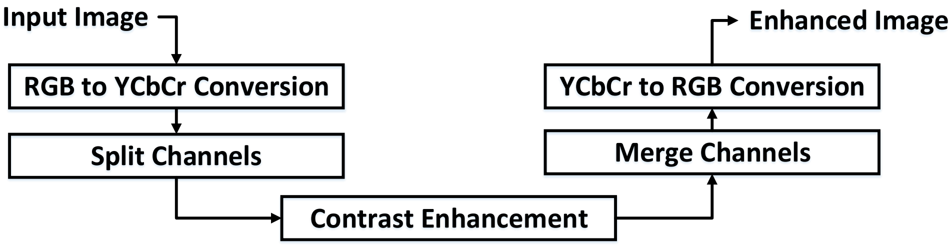


FIGURE 3.3: Cost-efficient contrast enhancement (CE) scheme for the IR frames.

the mean μ is the sum of all the pixels divided by the total number of pixels in an image averaged over three channels. If there is a switch between the camera sensors then the CSD signals to initialize the background modeling again. Also, if the IR camera is detected, the incoming frames are contrast-enhanced before modeling the background. The criterion is simple yet powerful to detect the left and right skewed incoming IR images.

3.1.2 Optimal Color Space Selection (OCSS)

Sterile zone monitoring is an outdoor task. There will be a time at which the IVA faces sudden or variable illumination changes. This may result in false positives. The optimal color space selection (OCSS) aims at tackling the illumination changes by selecting tolerant color space (RGB/YCbCr) to model the background. The effectiveness of both color spaces for illumination changes has been documented in the literature. Several works have proposed the application of multiple color spaces to tackle illumination changes. Such algorithms are cost ineffective as they maintain multiple background models. The OCSS selects the optimal color space to model the background which is a cost-efficient solution.

The premise of OCSS is derived from the working principle of the human eye. The human eye has two different cells called rods and cones. They supplement each other according to illumination changes. Rods are effective in general conditions while cones are designed to work in a variable or sudden illumination changes. The color spaces RGB and YCbCr are analogous to rods and cones respectively. Following this premise, the OCSS was proposed which aids in deciding the optimal color space tolerant to the illumination changes.

The OCSS exploits mean squared error μ_{se} to select the optimal color space. It is a measure of image similarity between consecutive frames and sensitive to illumination changes. The initial frames (say 100) without foreground information were used to calculate μ_{se} for both color spaces is given as:

$$\mu_{se} = \frac{1}{mn} \sum_{i=0}^{m} \sum_{j=0}^{n} [I(i,j) - G(i,j)]^2, \tag{3.2}$$

where $I(i,j)$ and $G(i,j)$ are input image and ground-truth image. m and n are the number of pixels in respective frames. The first frame of the input sequence without foreground information is selected as the ground truth $G(i,j)$. It is possible to get such a frame as IVA has the liberty to record input sequences without foreground information. Also, initial frames (100-200 frames) of IVA benchmark (e.g., i-LIDS datasets) are recorded without foreground information.

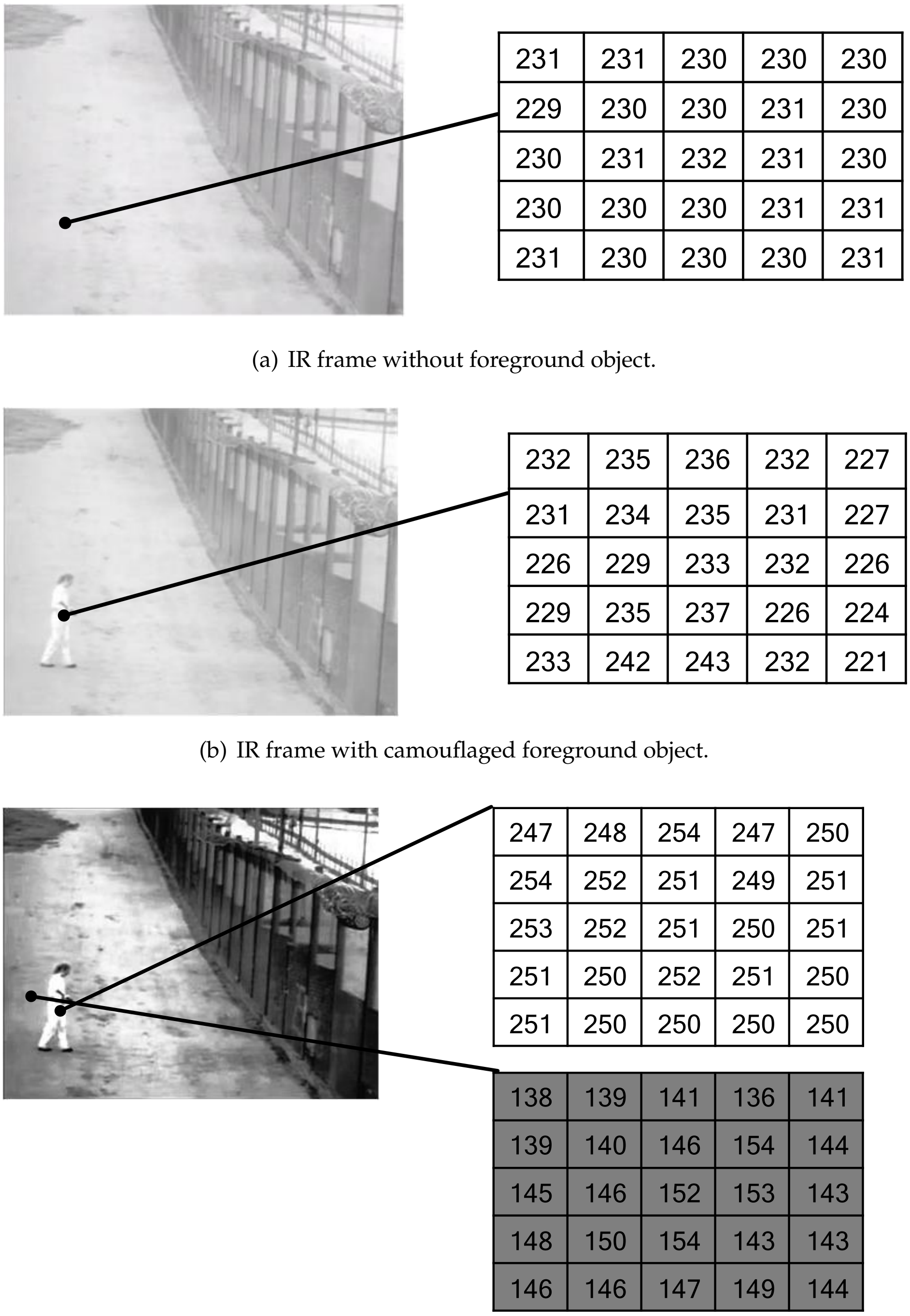
The optimal color space is selected as the one satisfying the following criterion:

$$\mu_{se}^{avg} \leq 5\mu_{se}^{1}, \tag{3.3}$$

where μ_{se}^{avg} is the average mean squared error of consecutive frames. μ_{se}^{1} is the mean squared error between the first frame and ground-truth. Here the first frame is the one after the selected ground-truth. Such criterion is inferred from the foreground detection rule for unsupervised change detectors, which allows a deviation of pixel intensities from ± 5 incorporated as the background. If both color spaces satisfy the condition, RGB color space is selected.

3.1.3 Contrast Enhancement (CE)

The IR input frames may pose a strong camouflage effect, i.e., the foreground object and background have similar pixel intensity. The cost-efficient contrast enhancement (CE) schema is proposed to tackle the camouflaged intruder at night as shown in Fig. 3.3. The incoming IR frames are converted to YCbCr color space if required. If the OCSS selects RGB as optimal color space, it will be converted to YCbCr for applying CE and then converted back to RGB for further processing. As CE is an intensity stretching operation, the ideal color space would be the one showing intensity values instead of color, i.e., YCbCr.



231	231	230	230	230
229	230	230	231	230
230	231	232	231	230
230	230	230	231	231
231	230	230	230	231

(a) IR frame without foreground object.

232	235	236	232	227
231	234	235	231	227
226	229	233	232	226
229	235	237	226	224
233	242	243	232	221

(b) IR frame with camouflaged foreground object.

247	248	254	247	250
254	252	251	249	251
253	252	251	250	251
251	250	252	251	250
251	250	250	250	250

138	139	141	136	141
139	140	146	154	144
145	146	152	153	143
148	150	154	143	143
146	146	147	149	144

(c) Contrast-enhanced IR frame of (b).

FIGURE 3.4: Visualizing the contrast enhancement (CE).

If CE is directly applied to RGB color space, it may cause color imbalance leading to false positives due to a noisy video. The input frames are split into their respective channels. The probability mass function (PMF) and cumulative density function (CDF) are computed and mapped to the intensity range. Later, the channels are merged and color space is converted back to RGB, if necessary.

Fig. 3.4 shows the effectiveness of the CE in differentiating the camouflaged intruder in the IR input frames. Fig. 3.4 a shows the IR input without camouflaged intruder (only background) with pixel intensities in a 5×5 region. Similarly, Fig. 3.4 b shows the IR input with the camouflaged intruder. The pixel intensities of both images (Fig. 3.4 a and b) in the specified 5×5 regions are similar. Such small differences are hard to detect by change detectors due to pixel intensity based background modeling (Bouwmans et al., 2019). Fig. 3.4 c shows the contrast-enhanced version of Fig. 3.4 b. It can be seen that the contrast has been increased between the background and the camouflaged intruder. This helps to detect the intruder effectively by the change detection module.

3.2 Change Detection Module

3.2.1 Background Modeling

The background is modeled from the initial frames (say 100) without foreground information. Each frame is split into their respective color channels (e.g, R, G, B). Each pixel in its respective channel is modeled using GMM (KaewTraKulPong and Bowden, 2002). The probability P of a pixel X at time t being background is formulated as:

$$P_{X_t} = \sum_{i=1}^{G} \omega_{i,t} \eta(X_t; \mu_{i,t}, \sigma_{i,t}^2), \tag{3.4}$$

where G, $\omega_{i,t}$, $\mu_{i,t}$, and $\sigma_{i,t}^2$ are number of Gaussian, estimate of weight, mean, and variance of the ith Gaussian in the mixture at time t. Since only Y channel carries information while Cb and Cr are useless without actual colors. Thus, the Y channel is used to model the background in the YCbCr-based IR camera frame.

3.2.2 Foreground Detection

The background model is compared with an incoming frame with the foreground information. The foreground detection rule to mark particular pixels at time t as the foreground is:

$$|X_t - \mu_{i,t}| > \lambda \sigma_{i,t}, \tag{3.5}$$

where λ = 2.5 is the foreground detection threshold inferred from the 68-95-99.7 standard deviation σ rule in (Shahbaz et al., 2017). 1 σ, 2 σ, and 3 σ covers 68%, 95%, and 99.7% of pixel values within a Gaussian. Thus, a pixel value located at more than 2.5 σ (99%) away from the estimated mean component of a Gaussian is labeled as foreground.

3.2.3 Adaptive Background Model Update

The new background and foreground values need to be updated in the background model after foreground detection. The general scheme (Wahyono and Jo, 2017-Shahbaz et al., 2017) to update the current pixel value in the new background model is as weighted sum of the pixel value in the current frame and pixel value in the previous background model:

$$B_t = \alpha I_t + (1 - \alpha) B_{t-1}, \tag{3.6}$$

where B_t, I_t, B_{t-1}, and α is the new background model, current pixel value, previous background model, and learning rate respectively.

The learning rate α is a crucial parameter to decides how long a certain pixel classified as foreground, will stay as a foreground. A fixed α value ranging between 0 to 1 is usually utilized to update a background model (Wahyono, Filonenko, and Jo, 2016). However, a fast-changing scene needs a high α value such as illumination changes, dynamic backgrounds, and moving foregrounds. For example, leaves moving on a tree (dynamic backgrounds) may be labeled as foreground and should promptly be labeled as background.

A slowly-changing scene requires a low α. For example, a static foreground object (SFO) is a challenge when a foreground object enters a scene and stays static at a certain position for a long time. SFO would be diffused into the background over time due to the background model update. Hence, an adaptive background model update scheme is required for an ISS to tackle the aforementioned challenges.

Several works addressed the fixed α problem by modeling and updating the background model with multiple learning rates. Wahyono and Jo, 2017 proposed a dual-learning rate scheme to model and update the background models separately. The scheme is only focused on extracting SFO by subtracting two foreground masks. Lin, Chuang, and Liu, 2011 proposed four learning rates to deal with the illumination changes, dynamic backgrounds, and moving foreground objects. The extracted foreground masks were aggregated to obtain the final foreground mask (Lin, Chuang, and Liu, 2011). Such schemes are computationally inefficient due to multiple background model maintenance.

A novel adaptive background model update scheme is proposed based on the measure of change of foreground pixels f_r in the scene. The innovation lies in its ability to track the changes in a scene based on the f_r using a single background model. Depending on f_r, four optimal learning rates are automatically switched in the background model update process. The rate of change of foreground pixels (Sajid and Cheung, 2015) is written as:

$$f_r = \frac{f_n^t \text{ - } f_{avg}^t}{f_{avg}^t}, \tag{3.7}$$

where f_n^t and f_{avg}^t is the number of foreground pixels at time t and average of foreground pixels at time t in a scene. The background model update process is initialized with a minimum value of f_r and is translated into four optimal learning rates, defined in the literature (Wahyono and Jo, 2017-KaewTraKulPong and Bowden, 2002). The criterion to assign different learning rate $\alpha \to L$ is defined as:

$$L = \begin{cases} 0.1, & f_r \geq 1 \\ 0.01, & 1.0 > f_r \geq 0.5 \\ 0.001, & 0.5 > f_r \geq 0.1 \\ 0.0001, & 0.1 > f_r \geq 0.01 \end{cases} \tag{3.8}$$

where L and f_r is the adaptive learning rate and rate of change of foreground pixels. A high f_r corresponds to a fast-changing scene or a moving foreground object. As a fast-changing scene requires a high α. Thus, the background model is updated with α = 0.1. As the foreground object stays in a particular position for a long time, f_r will start decreasing to a minimum where a foreground object becomes static (SFO). Thus, the background model is updated with a low α according to the f_r. Such

TABLE 3.1: Parameter Setting.

Parameter Name	Symbol	Value
CSD Threshold	T	20
Number of Gaussian	G	3
Foreground Detection Threshold	λ	2.5
Aggregated Foreground Mask	F_c	≥ 2

correspondence between f_r and four widely adopted learning rates α helps to tackle the challenges of illumination changes, dynamic backgrounds, moving, and static foreground.

3.2.4 Aggregating and Purging Foreground Mask

Later the foreground masks obtained by the respective color spaces are aggregated as follows:

$$f = \sum_{c=1}^{C} F_c \geq 2, \tag{3.9}$$

whereas Y channel is the final foreground mask in the YCbCr based IR frame. The aggregated foreground mask might have some isolated noise and cavities in the foreground object. Morphological opening and closing are applied to eliminate isolated noise and fill the cavities in the foreground object. The kernel size for opening (3×3) and closing (5×5) were kept as small as possible to keep the foreground object intact.

3.3 Experimental Results and Analysis

The proposed algorithm is compared with the top-ranked unsupervised change detection algorithms such as GMM (KaewTraKulPong and Bowden, 2002), SuBSENSE (St-Charles, Bilodeau, and Bergevin, 2015), PAWCS (St-Charles, Bilodeau, and Bergevin, 2016), WeSamBE (Jiang and Lu, 2018), and ML-RPCA (Zuluaga et al., 2017). Supervised change detection algorithms are not included in the comparison as they are trained offline with foreground and background information. This consensus has been reached by the wider change detection research community (http://www.changedetection.net/).

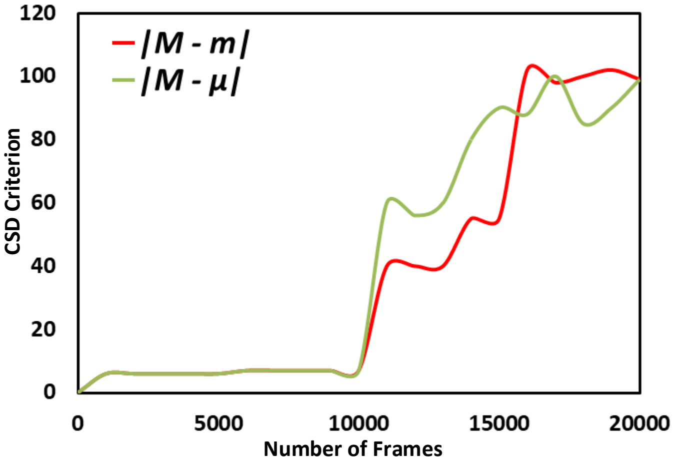


FIGURE 3.5: The CSD criterion, where x-axis and y-axis shows the number of frames and CSD criterion.

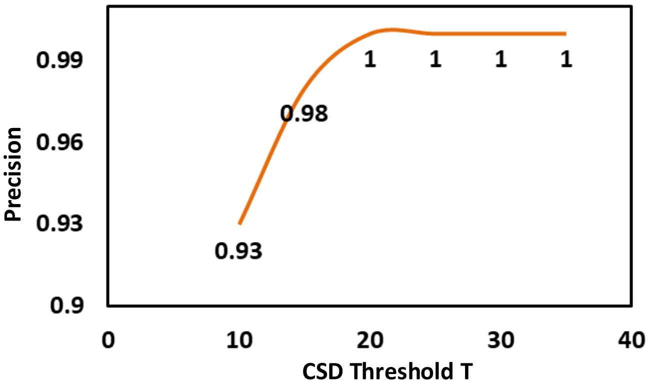


FIGURE 3.6: The ablation study of CSD threshold T, where x-axis and y-axis shows the CSD threshold T and precision.

3.3.1 Parameter Setting

Table 3.1 shows the parameter setting used in the proposed algorithm along with the definition. All the video sequences were tested using the same parameter setting. The optimal values were chosen through extensive experiments. Also, optimal parameters were kept for GMM and its improvements. Similarly, SuBSENSE, PAWCS, WeSamBE, and ML-RPCA were applied with the original setting. The proposed algorithm employs 4 parameters only, fewer than the comparative algorithms. For example, SuBSENSE and its improvements have more than 10 parameters to tweak.

Fig. 3.5 shows the variation of CSD criterion, $|M\text{-}m|$ (green line) and$|M\text{-}\ \mu|$ (red line), in the day (color) and night frames (IR). The day and night frames (10,000 each) from the i-LIDS and ISL-ISZM datasets were used to get the optimal value of the CSD threshold T. The frames come from six different background settings.

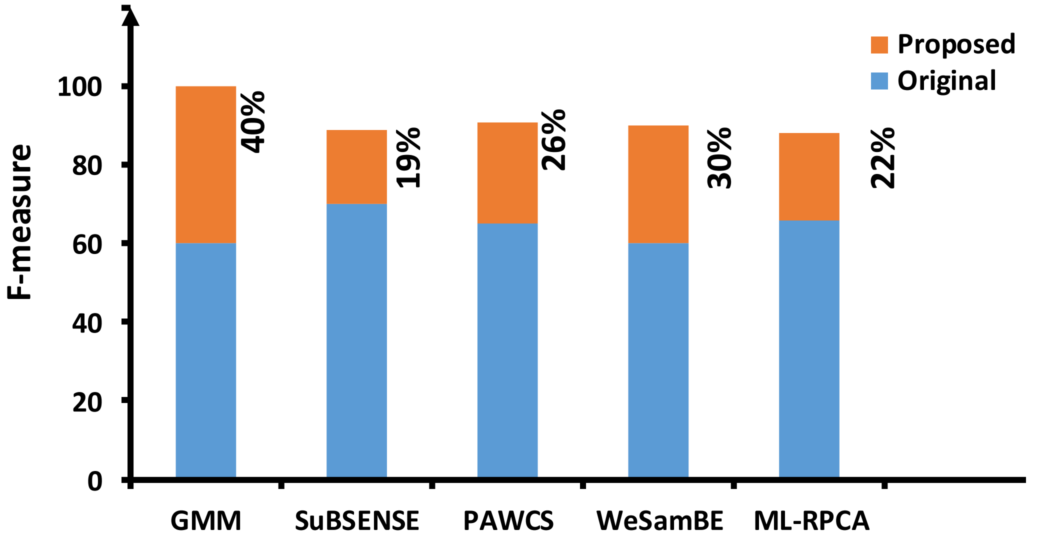


FIGURE 3.7: Quantitative comparison of the proposed algorithm with the comparative algorithms.

TABLE 3.2: Quantitative analysis on the CDNet.

Algorithm	R	Sp	FPR	FNR	PWC	F	P
GMM	0.7334	0.9928	0.0071	0.2660	1.9973	0.7164	0.7663
Proposed+GMM	**0.7897**	0.9946	0.0054	0.2123	1.4748	**0.8028**	**0.8242**
SuBSENSE	0.8616	0.9958	0.0041	0.1383	0.4855	0.8691	0.8895
Proposed+SuBSENSE	**0.8861**	0.9966	0.0033	0.1138	0.7916	**0.8988**	**0.9133**

The frames were arranged as day sequences followed by night sequences. The difference of $|M\text{-}m|$ and $|M\text{-}\ \mu|$ for day sequences was small, i.e., 4-9. This difference jumped above 40 and fluctuates between 40-90 for the night sequences. The variation of $|M\text{-}m|$ and $|M\text{-}\ \mu|$ between day (color) and night (IR) frames helps in deciding the CSD threshold T, as shown in Table 3.1. CSD is powerful to detect either a left or right skewed IR frames due to its dual condition, i.e., $|M\text{-}m|$ and $|M\text{-}\ \mu|$.

The ablation study of the CSD threshold T is shown in Fig. 3.6. Six values of T=$\{10, 15, 20, 25, 30, 35\}$ were evaluated. 10,000 frames from the change detection dataset (4,000), i-LIDS dataset (3,000), and ISL-ISZM dataset (3,000) were employed with 5,000 frames from each day (color) and night (IR). The frames were different from the ones evaluated in Fig. 3.5 and comes with eight different background settings. The threshold value (T=10) close to the CSD color camera range (4-9) gives a precision of 0.93. However, T=$\{20, 25, 30, 35\}$ achieved 100% precision. This is due to the big gap of $|M\text{-}m|$ or $|M\text{-}\ \mu|$ for color and IR frames which helps to decide the optimal T (Fig. 3.5).

3.3.2 Quantitative Analysis

The quantitative analysis on i-LIDS and ISL-ISZM dataset using F- measure F are shown in Fig 3.7. The blue column shows a comparative algorithm while the orange column shows the proposed algorithm integrated with the corresponding algorithm. The success criterion is defined by the i-LIDS dataset for the ISS evaluation. The intruder (true positive) should be detected for at least 75% of a particular video sequence. The analysis is shown in Fig. 3.7 is the average value over the 20 videos from both datasets. Each video contributes 5% of the overall F-measure.

GMM and its improvements were successful in 12 sequences of both datasets. Hence, it has a 60% F-measure. SuBSENSE was able to detect and track an intruder in 14 sequences, PAWCS in 13, and WeSamBE in 12. ML-RPCA was able to detect an intruder in all sequences. However, these algorithms gave false positives. These false positives resulted in a decrease in their overall F-measures.

The proposed algorithm with GMM showed impressive performance by tackling the camouflaged intruder in the night. It was able to detect and track intruders in all sequences without false positives. The proposed algorithm was also integrated with other comparative algorithms to show its generalization and effectiveness. It improved the performance of the comparative algorithms from 19-40%.

Table 3.2 shows the quantitative analysis of the 5 categories of CDNet such as baseline, dynamic backgrounds, bad weather, thermal, and shadows. The table shows the average value of 7 performance metrics namely recall R, specificity Sp, false positive rate FPR, false negative rate FNR, percentage of wrong classifications PWC, F-measure F, and precision P. The performance metrics are calculated by pixel-wise comparison between foreground mask and ground-truth using the software provided by the CDNet team.

The quantitative results of the comparative algorithms are available on the CDNet website. The proposed algorithm improved GMM 4-6% in performance metrics such as R, P, and F. The proposed algorithm showed better precision, which is crucial for the ISS. Similarly, the proposed algorithm was integrated with SuBSENSE. It also improved the SuBSENSE by the 2-3% in terms of R, P, and F.

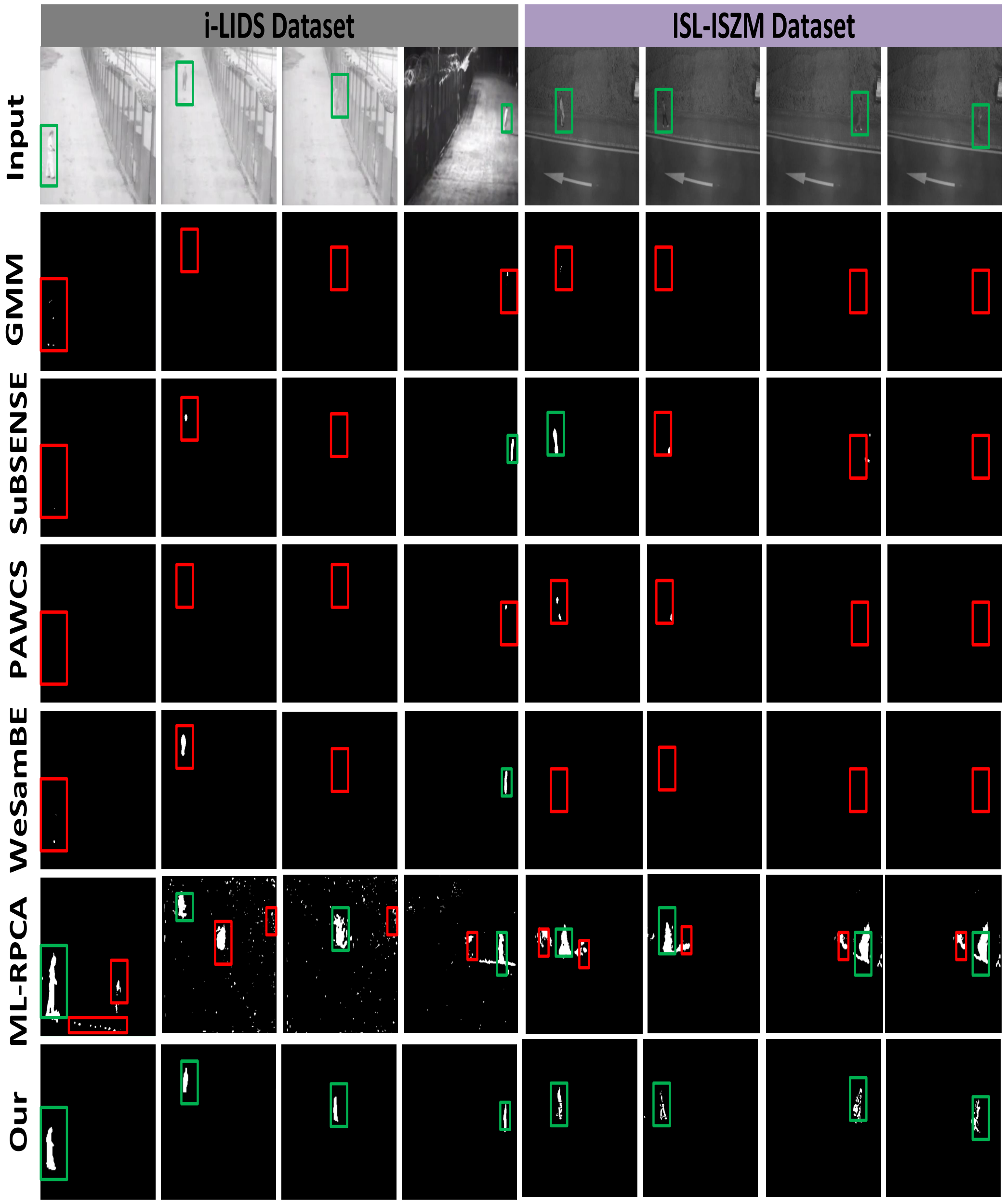


FIGURE 3.8: Qualitative comparison of the proposed algorithm with the comparative algorithms.

3.3.3 Qualitative Analysis

Fig. 3.8 shows the qualitative analysis of the proposed algorithm with the top-ranked change detection algorithms. The general scenario of the video sequences is the intruder entering the prohibited area. The night time sequences are shown to support the superior performance of the proposed algorithm.

The GMM and its improvements (2nd row) failed to detect the intruders properly. They segmented intruders partially. For example, the head and the shoes of the

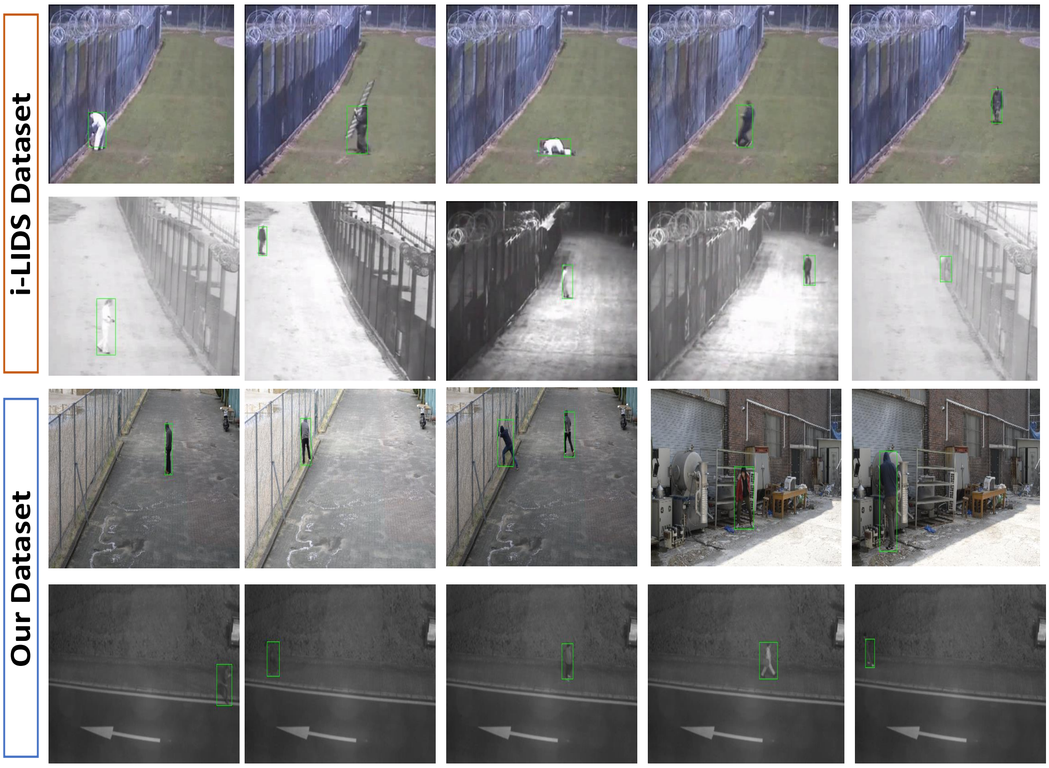


FIGURE 3.9: Final detection results of the proposed algorithm on all the video sequences of i-LIDS and ISL-ISZM dataset.

camouflaged intruder were different from the background (1st image). Also, GMM segmented only the head of the intruder which was different from the background (4th image). However, GMM failed all the challenges in ISL-ISZM dataset.

SuBSENSE (3rd row) segmented the intruder in one sequence of the i-LIDS dataset, as the intruder was significantly different from the background (4th image). It segmented the intruders partially in some sequences of i-LIDS and ISL-ISZM dataset, for which the part of the intruders was significantly different from the background (4th and 5th image).

PAWCS (4th row), similar to GMM and SuBSENSE, also segmented the intruders partially in both datasets. WeSamBE (5th row) had better performance than SuBSENSE and PAWCS on the i-LIDS dataset. It segmented the intruders in the two sequences (the 2nd and 4th images). However, it failed all the sequences of the ISL-ISZM dataset. It is evident from Fig. 3.8 that SuBSENSE, PAWCS, and WeSamBE failed to cope with the ISL-ISZM dataset.

ML-RPCA (6th row) was able to segment the intruder in all the sequences of both datasets. However, it labeled large portions of the background as foreground (false positives). The proposed algorithm (7th row) was able to detect the precise geometry of the camouflaged intruder in all the video sequences. It was able to cope with strong camouflage effects, illumination changes, and static foreground object.

Fig. 3.9 shows the final detection result of the proposed algorithm on all the video sequences of the i-LIDS and ISL-ISZM dataset. The results are arranged in numerical order as described in Table 2.1. For instance, the 1st result in Fig. 3.9 refers to the 1st video in Table 2.1. The night sequences are more challenging (2nd and 4th row).

The i-LIDs dataset is a standard benchmark and the scenes were developed in a controlled environment. ISL-ISZM dataset is more challenging as it has illumination changes, dynamic backgrounds, shadows, and camouflaged intruders. It is hard to distinguish between the camouflaged intruders and the background even to the naked eye (4th row). The performance of the proposed algorithm on three different databases with several challenges proves its generalization and effectiveness.

Chapter 4

Supervised Change Detector

This chapter describes the proposed Supervised Foreground Detection Network (SFDNet) in detail with the contributions. The SFDNet takes an input image of size WxHx3, where W, H, 3 are respectively width, height, and depth/channels of an image as shown in Fig. 4.1. SFDNet modifies the Visual Geometry Group (VGG-16) CNN (Simonyan and Zisserman, 2014). There are 13 standard convolutional layers (five blocks) and three fully-connected layers in VGG-16. The number of convolution kernels employed is 64, 128, 256, and 512 in each block. The input is zero padded to maintain spatial dimensions. Rectified Linear Unit (ReLU) is applied after each convolution layer to remove negative values in a feature map.

4.1 Atrous Spatial Feature Extractor (ASFE)

The mainstream feature extractors for foreground detection can be summarized into three different groups as shown in Fig. 4.2. Each block represents stacked convolutional layers. Fig. 4.2 a shows a feature extractor composed of stacked standard convolutional layers. Fig. 4.2 b and 4.2 c show the spatio-temporal feature extractors with convLSTM blocks. These extractors employ either a standard or atrous convolutional layers. The difference between mainstream feature extractors and proposed ASFE is evident from Fig. 4.2 d. The proposed ASFE employs a combination of stacked standard convolutional layers (SSCL) and stacked atrous convolutional layers (SACL) to increase the FOV. Residual-dense (RD) blocks strategy is applied to intermix the mid and high-level features.

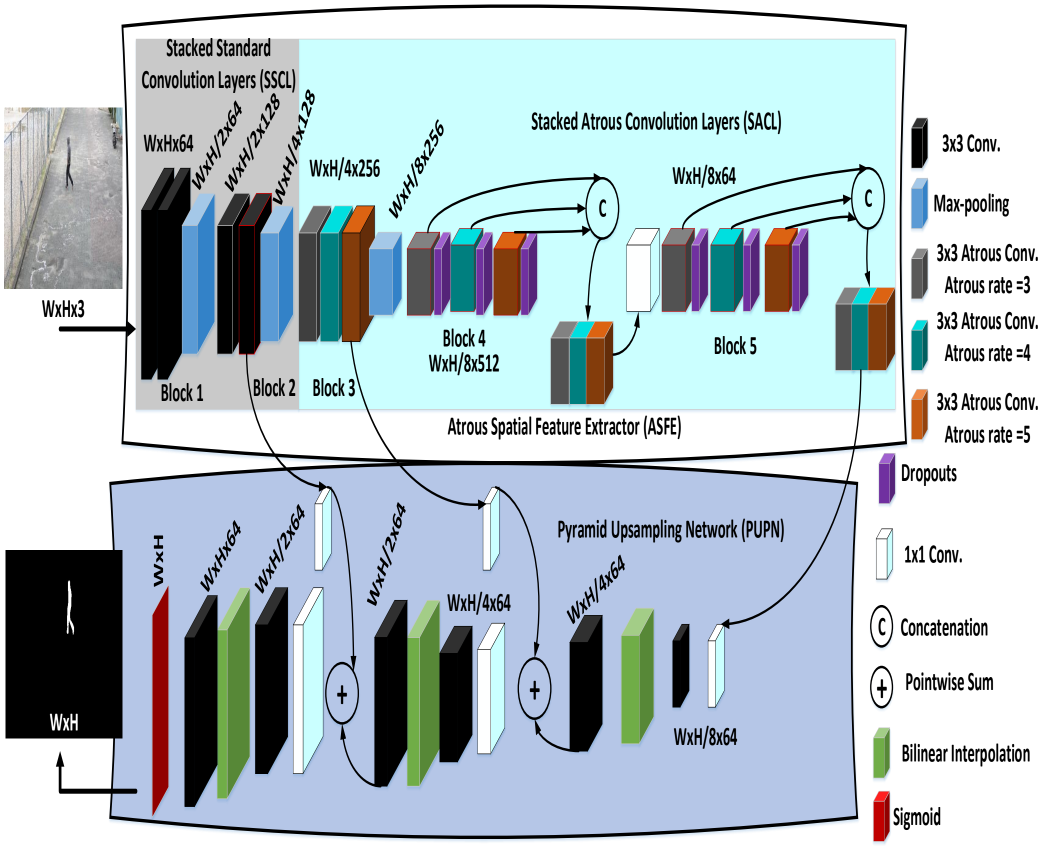
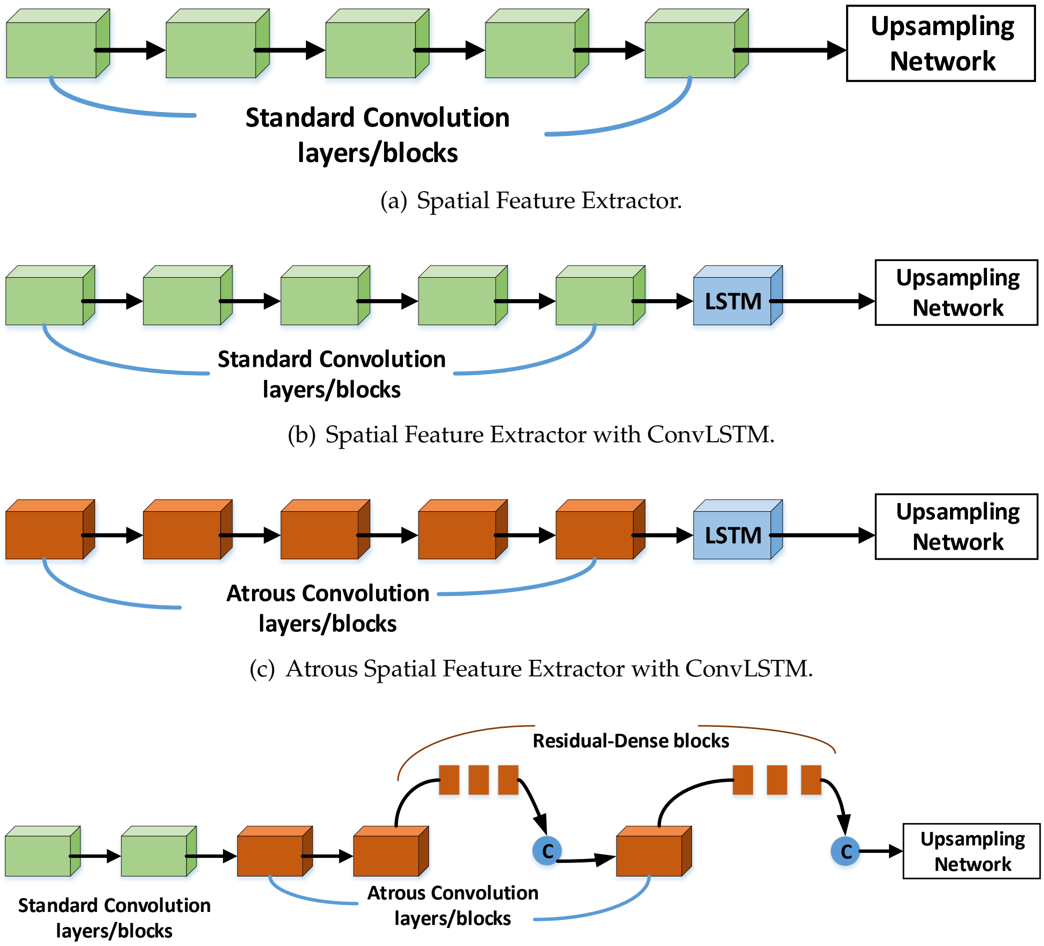


FIGURE 4.1: SFDNet with an atrous spatial feature extractor (ASFE) and pyramid upsampling network (PUPN).

4.1.1 Stacked Standard Convolution Layers (SSCL)

Like VGG-16, the first two blocks of SFDNet are SSCL with two convolutional layers and max-pooling layers each (Fig. 4.1). The convolutional layers are stacked to define a block to increase field-of-view (FOV) without increasing trainable parameters. The FOV of a 3x3 convolution kernel is the same as its size. However, the FOV of two stacked 3x3 convolution kernels would be effectively equal to that of a 5x5 convolution kernel. This decreases the number of trainable parameters appreciably. For example, a 5x5 convolution filter has 25 trainable parameters. But, a stack of two 3x3 convolution filters has 18 parameters. Hence, a feature from the same FOV can be extracted with fewer trainable parameters.


Upsampling Network
Standard Convolution layers/blocks
LSTM
Upsampling Network
Standard Convolution layers/blocks
LSTM
Upsampling Network
Atrous Convolution layers/blocks
Residual-Dense blocks
C
C
Upsampling Network
Standard Convolution layers/blocks
Atrous Convolution layers/blocks

(a) Spatial Feature Extractor.

(b) Spatial Feature Extractor with ConvLSTM.

(c) Atrous Spatial Feature Extractor with ConvLSTM.

(d) Proposed Atrous Spatial Feature Extractor (ASFE).

FIGURE 4.2: The difference between mainstream feature extractors and the proposed ASFE.

4.1.2 Stacked Atrous Convolution Layers (SACL)

The function of the 3rd-5th blocks is the same as before, i.e., increasing FOV without increasing trainable parameters. The FOV of three 3x3 convolution kernels is effectively equal to that of a 7x7 convolution kernel (Simonyan and Zisserman, 2014). A 7x7 convolution filter has 49 trainable parameters, while, a stack of three 3x3 convolution filters have 27 parameters (45% fewer parameters).

Unlike VGG-16, the 3rd-5th blocks of proposed ASFE are three SACL with varied atrous rates a. The number of convolution kernels is 256, 512, and 64. Motivated from (Simonyan and Zisserman, 2014), an atrous convolution kernel (ACK) is employed to extract the features. ACK is expanded by inserting zeros in the appropriate positions of the kernel mask. The ACK increases the FOV of a kernel mask without increasing the number of parameters (Chen et al., 2017).

The feature maps produced by the ACK are the same size as the input. But, each neuron in the feature map has rich global context information due to a larger FOV. An increase in the FOV of an ACK can be formulated as $[(a\text{-}1)\times(K\text{-}1) + K]$, where K is kernel size and a is atrous rate. Thus, the FOV of a 3x3 ACK with a= 3 is 7x7. Varied atrous rates (a= 3, 4, and 5) are applied on each ACK to increase the FOV to 7x7, 9x9, and 11x11. Atrous rates a were chosen after extensive experimentation. As objects often have various scales in an image, the varied atrous rates help to obtain feature maps with multi-scales forming a feature pyramid.

Similarly, the FOV of three stacked convolution kernels can be written as (K_1+K_2+K_3-2), where K_1, K_2, K_3 are respective atrous kernel sizes. Thus, stacking three atrous convolution layers with FOV 7x7, 9x9, and 11x11 will result in the FOV of 25x25 as compared to the FOV of 7x7 in VGG-16, without an increase in the trainable parameters.

4.1.3 Residual-Dense Blocks (RD)

Inspired by (Chen et al., 2017) and (Huang, Liu, and Weinberger, 2016), the atrous convolutional layers in the 4th block are further concatenated channel-wise to form a residual-dense block and fed into the 5th block. For example, feature maps of block4conv1 are 1-512. Then, feature maps of block4conv2 would be from 513-1024, and so on. This residual-dense blocks (RD) strategy helps to aggregate a global context and retain foreground information lost due to several convolutional operations on low-resolution high-level feature maps (Chen et al., 2018). Furthermore, concatenating features extracted from different FOVs helps to build a better global context via a feature pyramid. The feature maps are squeezed back from 1536 to 64 using 1x1 pointwise convolutions and fed in the 5th block.

Similarly, atrous convolutional layers of 5th block are also concatenated channel-wise and fed into the pyramid upsampling network (PUPN). The low resolution of high-level features often results in decreased pixel-level prediction. Unlike VGG-16, SFDNet removes the max-pooling layers in the 4th and 5th blocks and reduces the input size by 8 times. The spatial dropouts are added after each convolution layer of the 4th and 5th block. It helps to generalize the network (avoid over-fitting) and further decrease trainable parameters by zeroing out the whole feature map.

4.2 Pyramid Upsampling Network (PUPN)

The final feature maps from the ASFE are expanded to the original size to get the pixel-wise prediction. The design of an upsampling network is crucial for better foreground object detail extraction. Bilinear interpolation (BI) has been widely employed to expand the feature maps for pixel-wise prediction (Sakkos et al., 2018, Braham and Droogenbroeck, 2016, and Cioppa, Droogenbroeck, and Braham, 2020). BI can be regarded as a simple approach and cannot guarantee the recovery of the foreground object's detail. Another drawback is that it utilizes fixed weights, which cannot be learned during training. This is problematic during back-propagation as the gradient does not flow through an upsampling network.

Later works have utilized transposed convolution to expand feature maps (Wang, Luo, and Jodoin, 2017, Hu et al., 2018, and Yang et al., 2019). The feature maps are zero-padded and convolved with a transposed convolution kernel. The weights of the kernel are tuned during training, but this approach results in a checkerboard effect on the expanded feature maps (Chen et al., 2018).

SFDNet proposes a new pyramid upsampling network (PUPN). The PUPN is designed as BI sandwiched between 3x3 convolutional layers (conv.→BI→conv.). There are three such sandwiched structures in PUPN. BI is applied by a factor of 2 to expand the feature maps. The expanded feature maps are then refined using a 3x3 convolutional kernel. Such a strategy allows a gradient flow during the backpropagation process and mitigates the checkerboard effect. After the first BI, the spatial size increases from $\frac{1}{8}$ to $\frac{1}{4}$. The expanded feature maps are added with the corresponding mid-level features (block3conv3). This helps in two folds. First, the expanded features which have more global features representation from ASFE are mixed with the locally extracted mid-level features. Secondly, it helps to retain the lost foreground information after the application of several convolution layers. The depth of feature maps is again squeezed to 64 via 1×1 pointwise convolution. Then, the spatial dimension is expanded to $\frac{1}{2}$ using BI.

The expanded feature maps are again added with corresponding low-level feature maps (block2conv2) and pass through BI to get the original input size of an image. The depth of feature maps is always squeezed to 64 using a 1×1 pointwise convolution. This helps to control the model size (number of parameters).

The final layer of the PUPN outputs the probability of a pixel being a foreground using the sigmoid function. The threshold (Th=0.9) is applied to the final layer to get the foreground and background pixels. The threshold is chosen after extensive experimentation. Binary cross entropy loss L_i is employed during training (Equation 4.1). It compares probabilities of a pixel being a foreground or background with the ground-truth defined as:

$$L_i = \frac{-1}{M} \sum_{j=1}^{M} [y_j^i \log{(p_j^i)} + (1 - y_j^i) \log{(1 - p_j^i)}], \tag{4.1}$$

where y_j^i is the ground-truth label and p_j^i is the predicted value of the pixel i at location j. M is the total number of pixels in an image.

4.3 Hybrid Training Strategy

The SFDNet is trained using a hybrid training strategy. The weights of the first two blocks of the ASFE were borrowed from the pretrained VGG model. Such a scheme reduces the number of weights to be learned. It gives the SFDNet a head start and the network trains faster. The intuition is that the initial convolutional layers extract low-level features such as edges, color, corners, etc. (Chen et al., 2017). The weights of the 3rd-5th blocks are calibrated via fine-tuning with the specific video or dataset, e.g., Change Detection dataset (CDNet). Such a strategy helps the network to learn deep atrous spatial features according to the specific video or dataset.

4.4 Experimental Analysis and Results

SFDNet is compared with high-ranked foreground detection algorithms. The unsupervised algorithms used include SuBSENSE (St-Charles, Bilodeau, and Bergevin, 2015), PAWCS (St-Charles, Bilodeau, and Bergevin, 2016), WeSamBE (Jiang and Lu, 2018), IUTIS (Bianco, Ciocca, and Schettini, 2017), and ML-RPCA (Zuluaga et al., 2017), whereas DeepBS (Babaee, Dinh, and Rigoll, 2018), CascadeCNN (Wang, Luo, and Jodoin, 2017), RT-BGS (Cioppa, Droogenbroeck, and Braham, 2020), M2DC4 (Hu et al., 2018), MS-ST (Yang et al., 2019), MvRF-CNN (Akilan, Wu, and Zhang, 2019) and 3D CNN-LSTM (Akilan et al., 2020) are supervised algorithms.

4.4.1 Training Details

SFDNet is trained on Intel Core i5 Hardware with 8 GB RAM with a low-end NVIDIA GTX570 GPU. It is implemented in Keras (Chollet et al., 2015). SFDNet was trained for 50 epochs on 50 frames with 20% validation frames, i.e., 10 from 50 training frames. The SFDNet was trained for each video sequence of the CDNet. The training frames and respective ground-truths for the CDNet were provided by CascadeCNN authors[1]. CascadeCNN provided subsets of 50 and 200 training frames. The training and testing details for i-LIDS and our dataset is shown in Table 4.1. The ground-truths (50 training frames) for both datasets were made via the Interactive Segmentation tool[2]. It is assumed that all the possible foreground objects appear in the limited training frames.

4.4.2 Ablation Study

Extensive experiments were performed to demonstrate the effectiveness of the network design of the SFDNet using CDNet as shown in Table 4.2, where $SSCL$, $SACL$,

[1] https://github.com/zhimingluo/MovingObjectSegmentation

[2] http://www.cs.cmu.edu/ mohitg/segmentation.html

TABLE 4.1: Train/Test Details.

Video	# Training Frames	# Testing Frames	Scenario
i-LIDS dataset			
1	50	950	Day, Normal walk
2	50	950	Day, Running
3	-	1000	Day, Crawling
4	-	1000	Day, Slow walk
5	-	1000	Day, Walking fast
6	50	950	Night, Walking away
7	-	1000	Night, Walking slowly
8	-	1000	Night, Far from camera
9	-	1000	Night, Camouflage intruder
10	-	1000	Night, Camouflage intruder
ISL-ISZM dataset			
11	50	2250	Day, Normal walk
12	-	2300	Day, Walking fast
13	-	2300	Day, Running
14	-	2300	Day, Slow walking
15	-	2300	Day, Multiple intruders
16	50	2250	Day, Normal walk
17	-	2300	Day, Running
18	-	2300	Day, Slow walking
19	-	2300	Day, Dynamic background
20	-	2300	Day, Dynamic background
21	50	950	Night, Walking fast
22	-	1000	Night, Camouflage intruder
23	-	1100	Night, Camouflage intruder
24	-	1100	Night, Camouflage intruder
25	50	1050	Night, Multiple intruder

TABLE 4.2: Ablation study using Change detection dataset (CDNet).

Atrous Spatial Feature Extractor (*ASFE*)			
SSCL	*SACL*	*RD*	*F*
1,2,3,4,5	-	X	0.8941
-	1,2,3,4,5	✓	0.9224
1,2,3,4	5	✓	0.9264
1,2,3	4,5	✓	0.9388
1,2	3,4,5	✓	**0.9541**
1	2,3,4,5	✓	0.9407
Pyramid Upsampling Network (*PUPN*)			
	Upsampling	Pyramid	*F*
	BIConv	X	0.9541
	BI	✓	0.9328
	TConv	✓	0.9534
	BIConv	✓	**0.9747**

TABLE 4.3: Quantitative analysis on all the categories of the CDNet.

Algorithm	# Frames	*R*	*Sp*	*FPR*	*FNR*	*PWC*	*P*	*F*
SFDNet	50	0.944	0.9997	0.0002	0.0551	0.12	0.973	0.958
SFDNet	200	0.969	**0.9999**	**0.0001**	0.0301	**0.06**	**0.980**	**0.974**
MS-ST	630-5600	0.965	0.9995	0.0005	0.0350	0.11	0.965	0.967
M2DC4	630-5600	**0.970**	0.9991	0.00012	**0.0224**	0.2	0.966	0.961
3D CNN-LSTM	630-5600	-	-	-	-	-	-	0.94
MvRF-CNN	630-5600	-	-	-	-	-	-	0.948
CascadeCNN	200	0.950	0.9968	0.0032	0.0494	0.40	0.899	0.920
RT-BGS	-	0.789	0.996	0.0039	0.2110	1.07	0.830	0.789
DeepBS	200	0.754	0.9905	0.0095	0.2455	1.99	0.833	0.745
SuBSENSE	200	0.812	0.9904	0.0096	0.1876	1.6	0.750	0.740
IUTIS-5	200	0.784	0.9948	0.0052	0.2151	1.1	0.771	0.808

and *RD* are stacked standard convolutional layer, stacked atrous convolutional layer, and residual-dense blocks strategy. 1, 2, 3, 4, and 5 refer to the block number.

The experiments are split into two levels, i.e., atrous spatial feature extractor (ASFE) and pyramid upsampling network (PUPN). In the ASFE, the network design of stacked standard convolution layers (SSCL), stacked atrous convolution layers (SACL), and residual-dense blocks strategy (RD) were evaluated. In the PUPN design, however, the effectiveness of the upsampling technique and pyramid structure was demonstrated.

Initially, pretrained VGG-16 (1st row) with 5 SSCL blocks were employed. Then, all the VGG-16 blocks were replaced by SACL blocks (2nd row). Later, experiments were performed by supplementing SSCL and SACL blocks together with hybrid training. SFDNet benefits from SSCL, SACL, CT, and hybrid training (5th row). Its F-measure was 5% more than the only SSCL model (1st row).

The pyramid upsampling network (PUPN) was switched with other upsampling techniques (UP), i.e., bilinear interpolation BI (2nd row) and transposed convolution TConv (3rd row). The proposed PUPN (4th row) performed 2-5% better than BI and TConv.

TABLE 4.4: F-measure of each category of the CDNet.

Algorithm	*BW*	*LF*	*NV*	*PTZ*	*T*	Average
SFDNet (ours)	**0.9879**	**0.9294**	**0.9659**	**0.9817**	0.9273	**0.9747**
MS-ST	0.9846	0.9013	0.9390	0.9314	**0.9568**	0.9657
M2DC4	0.9609	0.8994	0.9489	0.9582	0.9488	0.9615
3D CNN-LSTM	0.9583	0.9660	-	-	0.9624	0.9402
MvRF-CNN	0.9480	-	0.8963	-	-	0.9489
CascadeCNN	0.9431	0.8370	0.8965	0.9168	0.9108	0.9209
RT-BGS	0.8260	0.7888	0.5014	0.5673	0.6921	0.7892
DeepBS	0.8301	0.6002	0.5835	0.3133	0.8455	0.7458
SuBSENSE	0.8619	0.6445	0.559	0.3476	0.7792	0.7408
IUTIS-5	0.8248	0.7743	0.5290	0.4282	0.7836	0.7717
Algorithm	*B*	*CJ*	*DB*	*IOM*	*Sh*	*Th*
SFDNet (ours)	**0.9938**	**0.9857**	**0.9888**	**0.9897**	**0.9938**	0.9808
MS-ST	0.9895	0.9802	0.9791	0.9893	0.9874	**0.9840**
M2DC4	0.9897	0.9645	0.9789	0.9637	0.9813	0.9833
3D CNN-LSTM	0.9470	0.9525	0.9502	-	0.9446	0.8870
MvRF-CNN	0.9632	0.9507	0.9590	0.9660	0.9579	-
CascadeCNN	0.9786	0.9758	0.9658	0.8505	0.9593	0.8958
RT-BGS	0.9604	0.8388	0.9489	0.7878	0.9478	0.8219
DeepBS	0.9580	0.8990	0.8761	0.6098	0.9304	0.7583
SuBSENSE	0.9503	0.8152	0.8177	0.6569	0.8986	0.8171
IUTIS-5	0.9567	0.8332	0.8902	0.7296	0.8766	0.8303

TABLE 4.5: Quantitative Analysis using recall R, precision P and F-measure F.

	i-LIDS dataset			ISL-ISZM dataset		
Algorithm	R	P	F	R	P	F
SFDNet (ours)	1.0	1.0	1.0	1.0	1.0	1.0
SuBSENSE	0.80	0.80	0.80	0.73	0.73	0.73
ML-RPCA	1.00	0.50	0.66	1.00	0.50	0.66
DeepBS	0.70	0.70	0.70	0.66	0.66	0.66
RT-BGS	0.80	0.80	0.80	0.86	0.86	0.86

4.5 Quantitative Analysis

4.5.1 Change Detection Dataset (CDNet)

Table 4.3 shows the quantitative analysis of SFDNet and baselines on the CDNet using seven performance metrics. The performance metrics are recall R, specificity Sp, false positive rate FPR, false negative rate FNR, percentage of wrong classifications PWC, precision P, and F-measure F. CDNet defines strict pixel-wise performance evaluation as compared to frame-level. Thus, CDNet is better in bench-marking the foreground detection algorithms.

Most supervised algorithms were trained for 200 frames except MS-ST and M2DC4, which were trained for significantly more frames (630-5600). This is due to the use of temporal data for ConvLSTM. Each video in CDNet comprises of 900-8000 frames. Hence, the number of test frames varies from 700-7800. For example, the highway sequence in the baseline category consists of 900 frames. Thus, 200 frames were used for training and 700 frames for testing.

SFDNet gets an average of F of 0.9747 and PWC of 0.0653, which is within the error margin of human annotation accuracy. SFDNet outperformed the baselines in 5 out of 7 performance metrics. However, the difference is quite low, e.g., recall R of M2DC4 (0.9701) is 0.003 more than SFDNet (0.9698). False Negative Rate FNR of M4DC4 (0.0224) is 0.0077 less than SFDNet (0.0301). Despite this, SFDNet performed significantly better than baselines in other categories such as F-measure and precision. 3D CNN-LSTM and MvRF-CNN only reported F-measure.

Table 4.4 shows the category-wise F-measure F of SFDNet and the baselines. The categories are baseline B, bad weather BW, dynamic background DB, camera jitter CJ, low frame rate LFR, night videos NV, pan tilt zoom PTZ, thermal Th, shadow Sh, intermittent object motion IOM, and turbulence T. SFDNet is significantly better than the baselines on 9 categories of CDNet. SFDNet achieved more than 98% F-measure on the 8 categories. It achieved 96% F-measure on the night category, which is considered the most challenging for video surveillance systems. The turbulence T category is also challenging due to the very small foreground object. 3D CNN-LSTM and MvRF-CNN did not test all the categories of CDNet.

4.5.2 i-LIDS Dataset

Table 4.5 shows the quantitative analysis of SFDNet and baselines on the i-LIDS dataset. Unlike CDNet, the i-LIDS dataset employs soft frame-level evaluation as compared to pixel-wise prediction. An intruder must be detected for atleast 75% of a video to be marked as successful (Branch, 2006). For instance, each video weighs 10% of the final F-measure. SuBSENSE detected an intruder in 8 videos. Thus, its F=80%. ML-RPCA detected an intruder in all videos. However, it gave false positives, decreasing its overall performance.

Supervised algorithms suffered due to a strong camouflage effect in night-time videos. RT-BGS and DeepBS detected an intruder in, respectively, 8 and 7 videos. MS-ST and M2DC4 were not evaluated due to their high requirement of labeled training data. SFDNet detected an intruder in all the videos without any false positives.

4.5.3 ISL-ISZM Dataset

Table 4.5 shows quantitative analysis on our dataset using the i-LIDS dataset criterion. It is more challenging than the i-LIDS dataset. Each video weighs 6.66% of the final F-measure. SuBSENSE detected an intruder in all day-time videos with illumination noise and shadows. But, it detected an intruder in one night video only due to a strong camouflage effect. Thus, its F=73%.

Although ML-RPCA detected the intruder in all day and night videos, it suffered due to illumination changes and the shadow of the intruder, severely decreasing its overall performance. RT-BGS and DeepBS detected the intruder in 13 and 10 sequences, respectively. The SFDNet detected an intruder in all the videos.

4.6 Qualitative Analysis

4.6.1 Change Detection Dataset (CDNet)

It is evident from Fig. 4.3 that the SFDNet (3rd column) foreground masks are comparable with the ground-truths. It was able to detect the foreground object precisely. The detection of foreground objects in challenging situations shows promise. The baselines were unable to detect the foreground objects precisely. It is worth mentioning that SFDNet was only trained for 50 training frames, while the baselines might be trained for at least 200 training frames.

CascadeCNN (4th column) detected foreground objects partially in thermal sequences. It was unable to segment the object geometry precisely, e.g., in bad weather (2nd row) and thermal sequence (6th row). DeepBS (5th column) did well on some challenges of the CDNet like baseline and dynamic background categories. It could not extract foreground details. It partially detected the foreground object in shadows (5th row) and thermal (6th row) sequences. It even gave false positives in the thermal sequence (6th row).

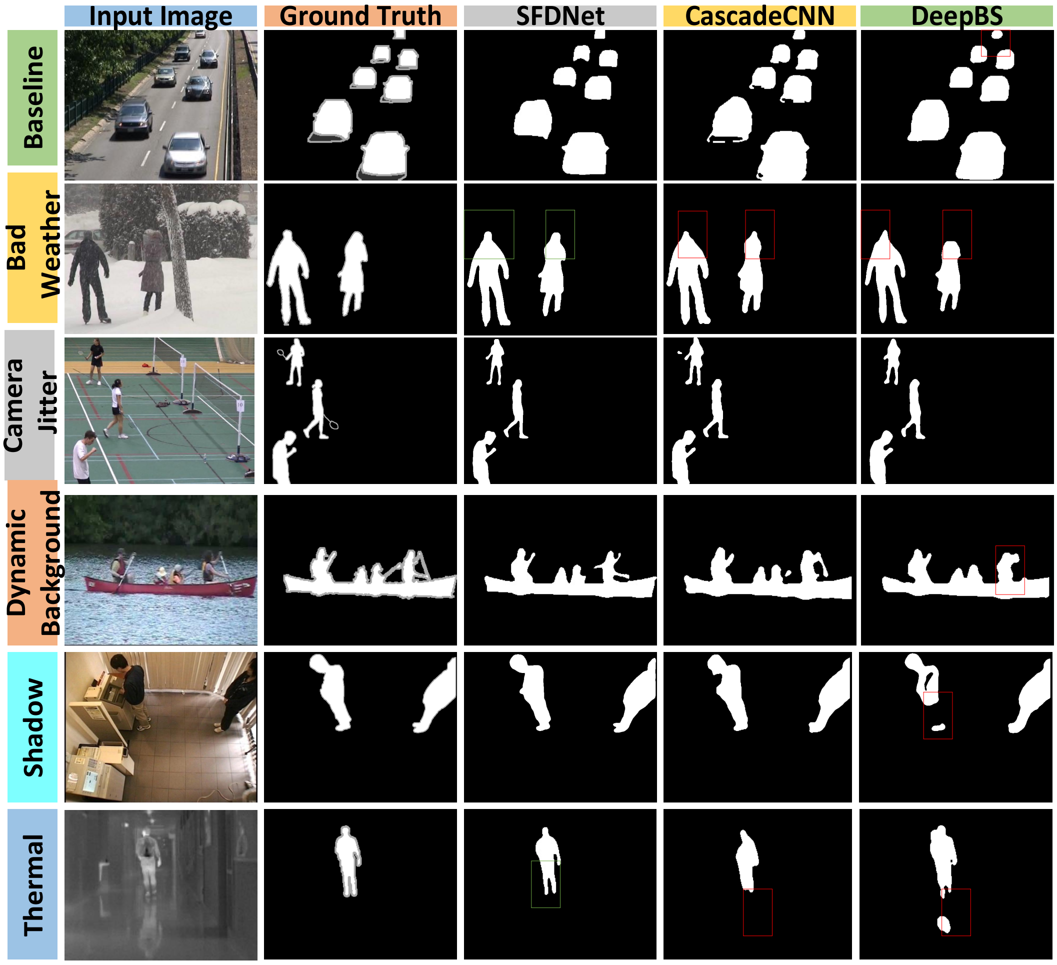


FIGURE 4.3: The qualitative comparison of SFDNet and the supervised baselines on the CDNet.

4.6.2 i-LIDS Dataset

Fig. 4.4 a shows the foreground masks from SFDNet and the baselines. Each column depicts different video sequences from day and night. The background setting offers the challenge of illumination changes and dynamic background, while the foreground object comes with the challenge of speed-variance, scale-variance, shadows, camouflage, and static intruder. The IR camera videos are also challenging for baselines. This is because currently available datasets like CDNet do not test algorithms against the IR camera.

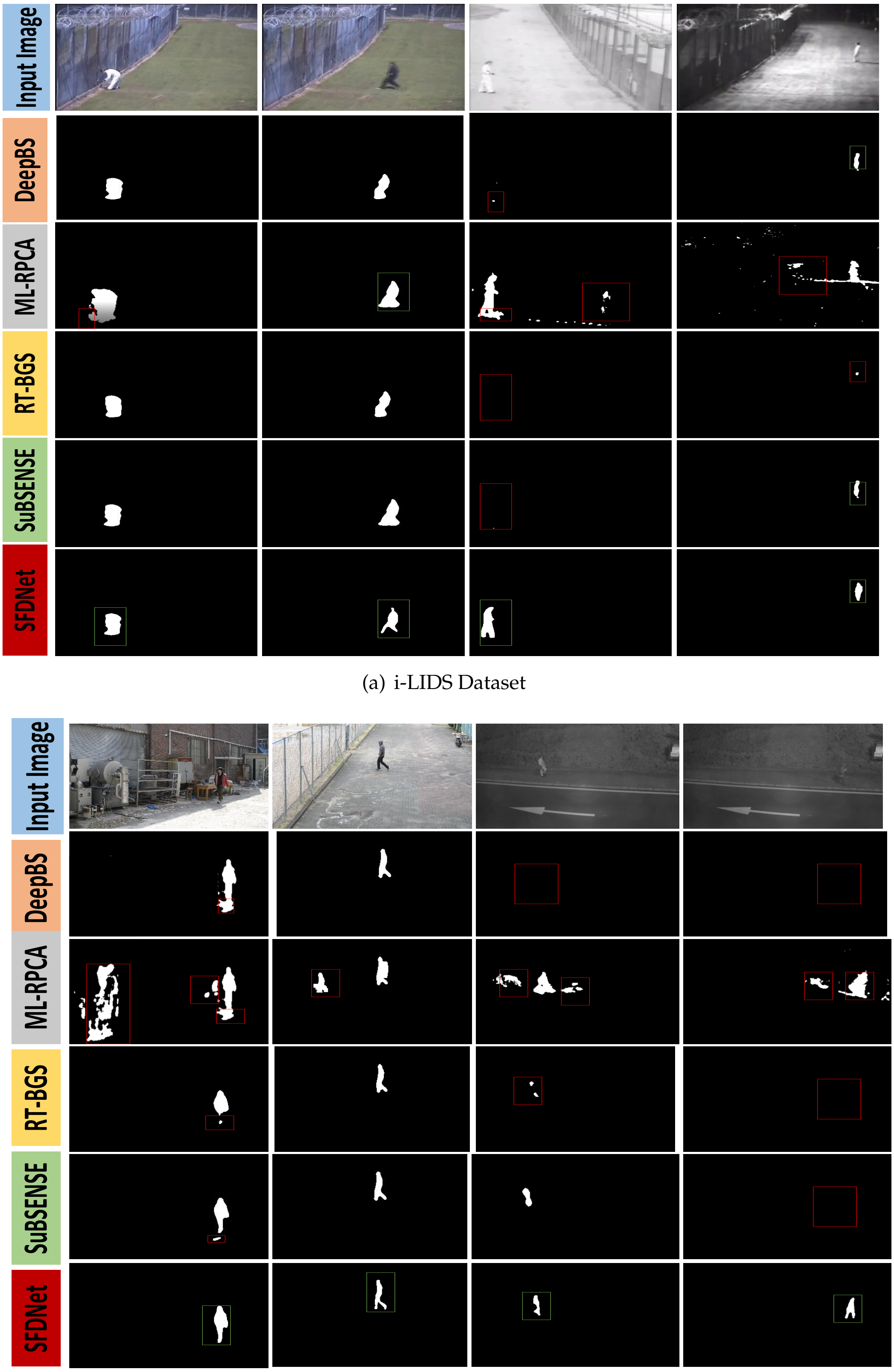
The baselines showed similar trends towards the challenges of illumination changes, shadows, and camouflaged intruders. DeepBS (2nd row) performed well in the day sequences. It suffered from the camouflage effect in IR videos. It detected a distinct part of an intruder. ML-RPCA (3rd row) performed well among baselines by detecting the intruders in both datasets. However, it gave false positives owing to illumination changes shadows. RT-BGS (4th row) performed well on the day sequences. It missed the intruders in the night sequences. SFDNet detected intruder with precise object details.

4.6.3 ISL-ISZM Dataset

ISL-ISZM dataset offers strong challenges of illumination changes, shadows, and camouflaged effect. Like the i-LIDS dataset, baselines showed similar trends on our dataset as shown in Fig. 4.4 b. DeepBS (2nd row) could not cope with shadows and camouflaged effect. It detected the shadow of an intruder in day sequences. It was unable to detect an intruder in night videos due to the camouflaged effect. Like the i-LIDS dataset, ML-RPCA (3rd row) performed well with false positives due to illumination and shadows.

RT-BGS (4th row) only detected distinct parts of the intruder from a background as shown in the red box. Similarly, in the day sequence of our dataset, it missed detecting the intruder's legs shown as a red box. DeepBS and RT-BGS suffering from the camouflage effect at night sequences may be due to the use of a background model for training the CNN model.

SuBSENSE (5th row) followed similar trends to other baselines and detected the shadow of the intruder. It was challenged by the night videos as it only segmented distinct parts of the camouflaged intruder from the background. The SFDNet was able to detect an intruder precisely (6th row) and differentiate between its shadow as well. Similarly, it was able to detect the camouflaged intruder in the night videos. Additionally, it did not give any false positives.



(a) i-LIDS Dataset

(b) ISL-ISZM Dataset

FIGURE 4.4: The qualitative results of the SFDNet with baselines such as DeepBS, ML-RPCA, RT-BGS, and SuBSENSE.

Chapter 5

System Implementation and Computational Analysis

This chapter describes the system implementation and computational analysis of proposed algorithms. The proposed unsupervised algorithm was implemented in the OpenCV based C++ environment. It utilized hardware with Intel Core i5-3.80 GHz and 8 GB RAM. The video sequences were resized to 640×480. The comparative algorithms were also implemented on the same machine.

The proposed supervised algorithm (SFDNet) was trained on a low-end NVIDIA GTX570 GPU and GTX1080Ti. The later was used to better compare the other supervised change detectors. It is implemented in Keras with TensorFlow backend. SFDNet was trained for 50 epochs on 50 frames with 20% validation frames, i.e., 10 from 50 training frames. It has 11.2 Million parameters in total with 7.6 Million as learnable. The total number of Giga FLOating Points Operations (GFLOPs) is 0.66. Regularization method RMSProp and 0.001 learning rate was used. A spatial dropout of 0.4 was applied.

The SFDNet was trained for each video sequence of the CDNet. The training frames and respective ground-truths for the CDNet were provided by CascadeCNN authors[1]. CascadeCNN provided subsets of 50 and 200 training frames. The training and testing details for the i-LIDS and ISL-ISZM dataset is shown in Table 4.1. The ground-truths (50 training frames) for both datasets were made via the Interactive Segmentation tool[2]. It is assumed that all the possible foreground objects appear in the limited training frames.

[1] https://github.com/zhimingluo/MovingObjectSegmentation

[2] http://www.cs.cmu.edu/ mohitg/segmentation.html

TABLE 5.1: Computational Analysis

Algorithm	Processing Speed (fps)
GMM	25-35
Proposed+GMM	28-30
SuBSENSE	4
Proposed+SuBSENSE	3.9
PAWCS	2
Proposed+PAWCS	2
WeSamBE	2
Proposed+WeSamBE	2
ML-RPCA	0.5
Proposed+ML-RPCA	0.5

TABLE 5.2: Operation-wise Processing Time

Operation	Processing Time (ms)
Camera Switching Detection	1.2
Optimal Color Space Selection	2.3
Contrast Enhancement	±3.1
Adaptive Background Update	5.6
Foreground Mask Purging	0.3
Background Modeling	16.6
Foreground Detection	6.6
Total	32.6±3.1

5.1 Computational Analysis of Unsupervised Detector

Table 5.1 shows the computational analysis in terms of the average frames per second (fps). GMM and its improvements have good processing speed but failed to detect an intruder overall. SuBSENSE, PAWCS, and WeSamBE have low processing speed as they built the background models using the texture information. Similarly, ML-RPCA builds models using sub-space and requires batch processing which is computation inefficient. Thus, such methods are unsuitable for a real-time system with low-cost hardware. The proposed algorithm integrated with GMM outperformed the comparative algorithms with real-time performance.

Table 5.2 shows the operation-wise processing time in milliseconds (ms). The background modeling and foreground detection operations from GMM constitute most of the processing time. The enhancements contribute to 28-34% (9.1±3.1 ms) of the total processing time (32.6±3.1 ms). The camera switch detection scheme helps to apply contrast enhancement only on IR frames. This saves significant processing time (3.1 ms).

TABLE 5.3: Minimum Hardware Evaluation For real-time performance

CPU	RAM	Image Size	Processing Speed (fps)
Core i5-3.80 GHz	8 GB	640×480	30
Core i3-2.66 GHz	8 GB	640×480	21
Quad core-2.90 GHz	4 GB	640×480	13

Table 5.3 shows the minimum hardware requirement to achieve real-time performance in terms of the average frames per second (fps). The proposed algorithm was further tested on two low-end hardware. It runs at 21 fps and 13 fps on Intel Core i3 and Intel Quad-core CPUs respectively, which is fit for the real-time requirement of the i-LIDS benchmark for video surveillance systems (10 fps) (Branch, 2006).

5.2 Computational Analysis of Supervised Change Detector

Table 5.4 shows the computational complexity of SFDNet and baselines in terms of training time, number of parameters, and testing speed in frames per second (fps). The analysis is performed on the NVIDIA GTX1080Ti GPU. It is evident that SFDNet is better in training time (12 minutes) and testing speed (32 fps) as compared to baselines. However, 3D CNN-LSTM has fewer parameters owing to the pretrained network and 3D convolution operation. Still, SFDNet is significantly better than 3D CNN-LSTM in training and testing time. SFDNet is suitable for real-time systems. Depending on image size (720×640-320×240), it performs at 8-15 fps on low-end NVIDIA GTX570 GPU which is within the real-time performance requirement of 10 fps (Branch, 2006).

TABLE 5.4: Computational complexity on 320×240 image size.

Method	Training time	# Parameters	Testing speed
SFDNet (ours)	12 minutes	7.3 Million	32 fps
MS-ST	90 minutes	>14 Million	11 fps
M2DC4	120 minutes	>11 Million	17 fps
3D CNN-LSTM	80 minutes	2.9 Million	24 fps
MvRF-CNN	>90 minutes	8.6 Million	25 fps
DeepBS	90 minutes	-	22 fps
RT-BGS	-	-	24 fps
SuBSENSE	-	-	2 fps
ML-RPCA	-	-	0.3 fps

Chapter 6

Conclusion

This work presented the unsupervised and supervised change detector powerful enough to perform real-time on low-end hardware. First, this work proposed an enhanced unsupervised change detector for IVAs particularly industrial sterile zone monitoring. Its ability to be integrated with other change detectors show promising prospects.

It was tested on three databases, 45 videos, and more than 100,000 video sequences. It outperformed top-ranked change detection algorithms with real-time performance. It improves other change detector's performance to the IR camera. Also, it improves their overall performance on the change detection dataset from 2-5%. The proposed enhancements are light-weight and only contribute to 28-34% of total processing time.

Secondly, considering the promise of deep learning algorithms, an efficient supervised change detector named SFDNet is proposed to tackle the challenges of camera-based surveillance systems. SFDNet benefits from atrous-convolved feature maps that extract rich encoded semantic features from the input, without increasing computational complexity.

Qualitative, quantitative, and computational results with top-ranked algorithms on three standard datasets are presented to demonstrate the effectiveness of SFDNet. The trained models are scene specific, i.e., a model can only perform better in the specific trained background setting. The algorithm requires a minimum low-end GPU to perform real-time.

6.1 Future Works

Undoubtedly, deep learning has taken over the field of computer vision. Deep learning has proven its worth by outperforming unsupervised or traditional computer vision algorithms. The task of dense estimation like change detection and semantic segmentation which requires assigning a class label to each pixel in an image was thought to be an tedious task at hand. Deep learning based computer vision algorithms changed that factor of impossibility.

The requirement of high-end hardware to train the algorithms with pixel-wise annotated data is a huge hindrance. Thus, extending the current work to further decrease the computational complexity using a lightweight network would be a possible research direction. The authors wish to integrate SFDNet into other high-level tasks of video surveillance systems such as abandoned object detection and illegally parked vehicle detection in the future.

Appendix A

Publications

A.1 Journal

1. Ajmal Shahbaz and Kang-Hyun Jo, Enhanced Unsupervised Change Detector for Industrial Surveillance Systems, IEEE Transactions on Industrial Electronics, 2020.

2. Ajmal Shahbaz and Kang-Hyun Jo, Deep Atrous Spatial Features based Supervised Foreground Detection Algorithm for Industrial Surveillance Systems, IEEE Transactions on Industrial Informatics, 2020.

3. Ajmal Shahbaz and Kang-Hyun Jo, Improved Change Detector using Dual-Camera Sensors for Intelligent Surveillance Systems, IEEE Sensors, 2020.

A.2 Conference

1. Ajmal Shahbaz and Kang-Hyun Jo, Deep Atrous Spatial Features-based Foreground Segmentation for Moving Camera, ISIE 2020, Delft, Netherland, Jun 17, 2020.

2. Ajmal Shahbaz, and Kang-Hyun Jo, Dilated CNN based Human Verifier for Intrusion Detection, IW-FCV 2020, Ibusuki, Japan, Feb 20, 2020.

3. Van-Thanh Hoang, Ajmal Shahbaz and Kang-Hyun Jo, Deep Residual Networks with Pyramid Depthwise Separable Convolution, IECON 2019, Lisbon, Portugal, Oct 14, 2019.

4. Ajmal Shahbaz, Van-Thanh Hoang, and Kang-Hyun Jo, Convolutional Neural Network based Foreground Segmentation for Video Surveillance Systems, IECON 2019, Lisbon, Portugal, Oct 14, 2019.

5. Ajmal Shahbaz, and Kang-Hyun Jo, Deep Foreground Segmentation using Convolutional Neural Network, ISIE 2019 , Vancouver, Canada, Jun 12, 2019.

6. Ajmal Shahbaz, and Kang-Hyun Jo, Foreground Segmentation using Convolutional Neural Network, ICCAS 2018, PyeongChang, Korea, Oct 17, 2018.

7. Ajmal Shahbaz, and Kang-Hyun Jo, Probabilistic Change Detector with Human Verifier for Intelligent Sterile Zone Monitoring, ISIE 2018, Cairns, Australia, Jun 12, 2018.

8. Ajmal Shahbaz and Kang-Hyun Jo, Optimal Background Modeling for Cluttered Scenes, IECON 2017, Beijing, China, Oct 29, 2017.

9. Ajmal Shahbaz, and Kang-Hyun Jo, Exploiting Color Spaces for the Task of Foreground Detection, ICCAS 2017, Jeju, Korea, Oct 18, 2017.

10. Ajmal Shahbaz, Wahyono, and Kang-Hyun Jo, Sterile Zone Monitoring with Human Verification, HSI 2017, Ulsan, Korea, Jul 17, 2017.

11. Ajmal Shahbaz, Danilo Caceres Hernandez, and Kang-Hyun Jo, Optimal Color Space based Probabilistic Foreground Detector for Video Surveillance System, ISIE 2017, Edinburgh, Scotland, Jun 19, 2017.

12. Ajmal Shahbaz, Laksono Kurnianggoro, Wahyono, and Kang-Hyun Jo, Recent Advances in the Field of Foreground Detection: An Overview, ACIIDS 2017, Kanazawa, Japan, Apr 3, 2017.

13. Laksono Kurnianggoro, Dongwook Seo, Joko Hariyono, Ajmal Shahbaz, and Kang-Hyun Jo, Coarse-to-fine Approach for Fast Correlation-based Visual Tracking, IECON 2016, Florence, Italy, Oct 23, 2016.

14. Ajmal Shahbaz, Laksono Kurnianggoro, Kang-Hyun Jo, Parameter Analysis of Probabilistic Foreground Detector, IECON 2016, Florence, Italy, Oct 23, 2016.

15. Joko Hariyono, Ajmal Shahbaz, Laksono Kurnianggoro and Kang-Hyun Jo, Estimation of Collision Risk for Improving Driver"s Safety, IECON 2016, Florence, Italy, Oct 23, 2016.

16. Laksono Kurnianggoro, Ajmal Shahbaz, and Kang-Hyun Jo, Dense Optical Flow in Stabilized Scenes for Moving Object Detection from aMoving Camera, ICCAS 2016, Gyeongju, Korea, Oct 16, 2016.

17. Ajmal Shahbaz, Laksono Kurnianggoro, Kang-Hyun Jo, A Comparative Study for Foreground Detection using Gaussian Mixture Models-Novice to Novel, ICCAS 2016, Gyeongju, Korea, Oct 16, 2016.

18. Wahyono, Alexander Filonenko, Ajmal Shahbaz, and Kang-Hyun Jo, Vision-based Intelligent Surveillance System: Multi-tasks Implementation, URAI 2016, Xian , China, Aug 19, 2016 pp.296.

19. Ajmal Shahbaz, Alexander Filonenko, Joko Hariyono, Wahyono, Kang-Hyun Jo, Probabilistic Foreground Detector for Sterile Zone Monitoring at Night Time, URAI 2016, Xian , China, Aug 19, 2016.

20. Alexander Filonenko, Danilo Caceres Hernandez, Ajmal Shahbaz, and Kang-Hyun Jo, Unified Smoke and Flame Detection for Intelligent Surveillance System, ISIE 2016, Santa Clara, USA, Jun 8, 2016.

21. Ajmal Shahbaz, Danilo Caceres Hernandez, Alexander Filonenko, Joko Hariyono, Kang-Hyun Jo, Probabilistic Foreground Detector with Camouflage detection for Sterile Zone Monitoring, ISIE 2016, Santa Clara, USA, Jun 8, 2016.

22. Danilo Caceres HernandezAlexander FilonenkoJoko HariyonoAjmal Shahbaz, Laser Based Collision Warning System for High Conflict Vehicle-Pedestrian Zones , ISIE 2016, Santa Clara, USA, Jun 8, 2016.

23. Alexander Filonenko, Wahyono, Joko Hariyono, Ajmal Shahbaz, Youlkyeong Lee, and Kang-Hyun Jo, Adaptive Frame Rate Streaming Strategy for Intelligent Surveillance System, ICROS 2016, Seoul, Korea, Mar 10, 2016.

24. Wahyono, Alexander Filonenko, Ajmal Shahbaz, Joko Hariyono, Kang, Hyun-Deok, and Kang-Hyun Jo, Integrating Multiple Tasks of Vision-based Surveillance System: Design and Implementation, FCV 2016, Takayama, Japan, Feb 17, 2016 pp.91-94.

25. Joko Hariyono, Ajmal Shahbaz and Kang-Hyun Jo, Estimation of Walking Direction for Pedestrian Path Prediction from Moving Vehicle, SII 2015, Nagoya, Japan, Dec 11, 2015.

26. Ajmal Shahbaz, Kang-Hyun Jo, Probabilistic Foreground Detector for Sterile Zone Monitoring, URAI 2015, Goyang City, Korea, Oct 28, 2015.

27. Ajmal Shahbaz, Joko Hariyono, and Kang-Hyun Jo, Evaluation of Background Subtraction Algorithms for Video Surveillance, FCV 2015, Mokpo, Korea, Jan 28, 2015.

Bibliography

Akilan, T., Q. M. J. Wu, and W. Zhang (2019). “Video Foreground Extraction Using Multi-View Receptive Field and Encoder–Decoder DCNN for Traffic and Surveillance Applications”. In: *IEEE Transactions on Vehicular Technology* 68.10, pp. 9478–9493.

Akilan, T. et al. (2020). “A 3D CNN-LSTM-Based Image-to-Image Foreground Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.3, pp. 959–971.

Babaee, Mohammadreza, Duc Tung Dinh, and Gerhard Rigoll (Apr. 2018). “A Deep Convolutional Neural Network for Video Sequence Background Subtraction”. In: *Pattern Recogn.* 76.C, pp. 635–649. ISSN: 0031-3203.

Bianco, S., G. Ciocca, and R. Schettini (2017). “Combination of Video Change Detection Algorithms by Genetic Programming”. In: *IEEE Transactions on Evolutionary Computation* 21.6, pp. 914–928.

Bouwmans, Thierry et al. (2019). “Deep neural network concepts for background subtraction:A systematic review and comparative evaluation”. In: *Neural Networks* 117, pp. 8 –66. ISSN: 0893-6080.

Braham, M. and M. Van Droogenbroeck (2016). “Deep background subtraction with scene-specific convolutional neural networks”. In: *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 1–4.

Branch, Home Office Scientific Development (2006). “Imagery Library for Intelligent Detection Systems (i-LIDS)”. In: *Crime and Security. The Institution of Engineering and Technology Conference on*, pp. 445–448.

Candes, Emmanuel J. et al. (June 2011). “Robust Principal Component Analysis”. In: *J. ACM* 58.3, 11:1–11:37. ISSN: 0004-5411. DOI: 10.1145/1970392.1970395.

Chen, Liang-Chieh et al. (2017). “Rethinking Atrous Convolution for Semantic Image Segmentation”. In: *CoRR* abs/1706.05587. arXiv: 1706.05587.

Chen, Liang-Chieh et al. (2018). "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *CoRR* abs/1802.02611. arXiv: 1802.02611. URL: http://arxiv.org/abs/1802.02611.

Chen, Yingying, Jinqiao Wang, and Hanqing Lu (2015). "Learning sharable models for robust background subtraction". In: *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6.

Chollet, François et al. (2015). *Keras*. https://github.com/fchollet/keras.

Cioppa, Anthony, Marc Van Droogenbroeck, and Marc Braham (2020). "Real-Time Semantic Background Subtraction". In: *CoRR* abs/1409.1556. arXiv: 2002.04993.

Filonenko, A., D. C. Hernandez, and K. Jo (2018). "Fast Smoke Detection for Video Surveillance Using CUDA". In: *IEEE Transactions on Industrial Informatics* 14.2, pp. 725–733. ISSN: 1551-3203. DOI: 10.1109/TII.2017.2757457.

Hu, Z. et al. (2018). "A 3D Atrous Convolutional Long Short-Term Memory Network for Background Subtraction". In: *IEEE Access* 6, pp. 43450–43459.

Huang, Gao, Zhuang Liu, and Kilian Q. Weinberger (2016). "Densely Connected Convolutional Networks". In: *CoRR* abs/1608.06993. arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

Jiang, S. and X. Lu (2018). "WeSamBE: A Weight-Sample-Based Method for Background Subtraction". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.9, pp. 2105–2115. ISSN: 1051-8215.

KaewTraKulPong, P. and R. Bowden (2002). "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection". In: *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*. Springer US, pp. 135–144.

Lin, H., J. Chuang, and T. Liu (2011). "Regularized Background Adaptation: A Novel Learning Rate Control Scheme for Gaussian Mixture Modeling". In: *IEEE Transactions on Image Processing* 20.3, pp. 822–836.

Liu, G. and S. Yan (2012). "Active Subspace: Toward Scalable Low-Rank Learning". In: *Neural Computation* 24.12, pp. 3371–3394. ISSN: 0899-7667.

Sajid, H. and S. C. S. Cheung (2015). "Background subtraction for static amp; moving camera". In: *IEEE International Conference on Image Processing (ICIP)*, pp. 4530–4534.

Sakkos, Dimitrios et al. (2018). "End-to-end video background subtraction with 3d convolutional neural networks". In: *Multimedia Tools and Applications* 77.17, pp. 23023–23041.

Shahbaz, A., J. Hariyono, and K. H. Jo (2015). "Evaluation of background subtraction algorithms for video surveillance". In: *21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pp. 1–4.

Shahbaz, A., V. Hoang, and K. Jo (2019). "Convolutional Neural Network based Foreground Segmentation for Video Surveillance Systems". In: *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1, pp. 86–89. DOI: 10.1109/IECON.2019.8927776.

Shahbaz, Ajmal et al. (2017). "Recent Advances in the Field of Foreground Detection: An Overview". In: *Advanced Topics in Intelligent Information and Database Systems*. Springer International Publishing, pp. 261–269.

Simonyan, Karen and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556. arXiv: 1409.1556.

St-Charles, P. L., G. Bilodeau, and R. Bergevin (2016). "Universal Background Subtraction Using Word Consensus Models". In: *IEEE Transactions on Image Processing* 25.10, pp. 4768–4781.

St-Charles, P. L., G. A. Bilodeau, and R. Bergevin (2015). "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity". In: *IEEE Transactions on Image Processing* 24.1, pp. 359–373.

Stauffer, Chris and W.E.L. Grimson (1999). "Adaptive background mixture models for real-time tracking". In: *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on.* Vol. 2, 252 Vol. 2.

Ullah, A. et al. (2019). "Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM". In: *IEEE Transactions on Industrial Electronics* 66.12, pp. 9692–9702.

Wahyono, A. Filonenko, and K. Jo (2016). "Unattended Object Identification for Intelligent Surveillance Systems Using Sequence of Dual Background Difference". In: *IEEE Transactions on Industrial Informatics* 12.6, pp. 2247–2255.

Wahyono and K. Jo (2017). "Cumulative Dual Foreground Differences for Illegally Parked Vehicles Detection". In: *IEEE Transactions on Industrial Informatics* 13.5, pp. 2464–2473.

Wang, R. et al. (2014a). “Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models”. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 420–424.

Wang, Y. et al. (2014b). “CDnet 2014: An Expanded Change Detection Benchmark Dataset”. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 393–400.

Wang, Yi, Zhiming Luo, and Pierre-Marc Jodoin (Sept. 2017). “Interactive Deep Learning Method for Segmenting Moving Objects”. In: *Pattern Recogn. Lett.* 96.C, pp. 66–75. ISSN: 0167-8655.

Yang, Y. et al. (2019). “End-to-End Background Subtraction via a Multi-Scale Spatio-Temporal Model”. In: *IEEE Access* 7, pp. 97949–97958.

Zhang, T. et al. (2013). “Mining Semantic Context Information for Intelligent Video Surveillance of Traffic Scenes”. In: *IEEE Transactions on Industrial Informatics* 9.1, pp. 149–160.

Zivkovic, Z. (2004). “Improved adaptive Gaussian mixture model for background subtraction”. In: *Pattern Recognition. ICPR. Proceedings of the 17th International Conference on*. Vol. 2, 28–31 Vol.2.

Zuluaga, Jhony Heriberto Giraldo et al. (2017). “Camera-Trap Images Segmentation using Multi-Layer Robust Principal Component Analysis”. In: *CoRR* abs/1701.08180.