Noname manuscript No.
(will be inserted by the editor)

Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation

Thierry Bouwmans, Sajid Javed, Maryam Sultana, Soon Ki Jung

Received: date / Accepted: date

Abstract Conventional neural networks have been demonstrated to be a powerful framework for background subtraction in video acquired by static cameras. Indeed, the well-known Self-Organizing Background Subtraction (SOBS) method and its variants based on neural networks have long been the leading methods on the large-scale CDnet 2012 dataset during a long time. Convolutional neural networks, which are used in deep learning, have been recently and excessively employed for background initialization, foreground detection, and deep learned features. The top background subtraction methods currently used in CDnet 2014 are based on deep neural networks, and have demonstrated a large performance improvement in comparison to conventional unsupervised approaches based on multi-feature or multi-cue strategies. Furthermore, since the seminal work of Braham and Van Droogenbroeck in 2016, a large number of studies on convolutional neural networks applied to background subtraction have been published, and a continual gain of performance has been achieved. In this context, we provide the first review of deep neural network concepts in background subtraction for novices and experts in order to analyze this success and to provide further directions. To do so, we first surveyed the background initialization and background subtraction methods based on deep neural networks concepts, and also deep learned features. We then discuss the adequacy of deep neural networks for

Thierry Bouwmans
Lab. MIA, Univ. La Rochelle, France
E-mail: tbouwman@univ-lr.fr

Sajid Javed
Dept. of Computer Science, University of Warwick, UK
E-mail: s.javed.1@warwick.ac.uk

Maryam Sultana
Dept. of Computer Science and Engineering, Kyungpook National University, Republic of Korea
E-mail: maryam@vr.knu.ac.kr

Soon Ki Jung
Dept. of Computer Science and Engineering, Kyungpook National University, Republic of Korea
E-mail: skjung@knu.ac.kr

the task of background subtraction. Finally, experimental results are presented for the CDnet 2014 dataset.

Keywords Background Subtraction · Restricted Boltzmann Machines · Autoencoders Networks · Convolutional Neural Networks · Generative Adversarial Networks

1 Introduction

During the last two decades, background subtraction for video taken by static cameras has been one of the most active research topics in computer vision owing to a large number of applications including intelligent surveillance of human activities in public spaces, traffic monitoring, and industrial machine vision [25,282]. This low-level operation consists of separating the moving objects called ”foreground” from the static information called ”background” [22,23,28,29,165]. For example, Figure 1 shows original frames of a sequence from the BMC 2012 dataset, the extracted background images and the foreground mask obtained by a well-know method. A big variety of models coming from mathematical theories, machine learning and signal processing have been used for background subtraction, including crisp models [182, 113,267], statistical models [85,248,294,314], fuzzy models [11,12,21], subspace learning models [79,222,241], robust PCA models [153,154,162,164,165], neural networks models [254,256,274] and filter based models [45,58,224,310]. Similar to PCA models, which have generated renewed interest in this area based on the theoretical advances of robust PCA, created in 2009 by Candès et al. [38], after an empty period of development, neural networks have received progressively renewed interest in this field since 2014 [351] owing to the practical advances in deep neural networks, which are now usable owing to the availability of large-scale datasets [112,335] for the training, and the progress in computational hardware ability[1].

Based from mathematical theories, the simplest way to model a background is to compute the temporal average [182], the temporal median [113] or the histogram over time [267]. These methods were widely used in traffic surveillance in 1990s owing to their simplicity but are not robust to the challenges faced in video surveillance such as camera jitter, changes in illumination, and dynamic backgrounds. To consider the imprecision, uncertainty and incompleteness in the observed data (i.e video), statistical models began being introduced in 1999 such as single Gaussian [344], Mixture of Gaussians (MOG) [248,294] and Kernel Density Estimation [85, 387]. These methods based on a Gaussian distribution model proved to be more robust to dynamic backgrounds. More sophisticated statistical models were after developed in literature and can be classified into those based on another distribution that alleviate the strict Gaussian constraint (i.e. general Gaussian distribution [86], Student’s t-distribution [233,120], Dirichlet distribution [131,91], Poisson distribution [95,386]), those based on co-occurrence [187–189] and confidence [265,264], free-distribution models [16,292,293], and regression models [309,177]. These approaches have improved the robustness to various challenges over time. The most ac-

[1] https://www.nvidia.fr/deep-learning-ai/

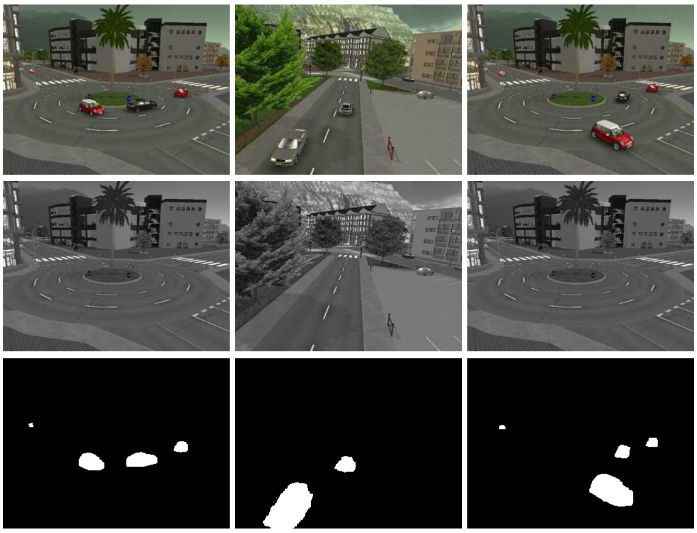


Fig. 1 Background Subtraction: Original image (309), Background extracted, Foreground mask (Sequences from BMC 2012 dataset [313]).

complished methods in this statistical category are ViBe [16], PAWCS [293] and SubSENSE [292]. Another theory that allows the handling of imprecision, uncertainty, and incompleteness is based on the fuzzy concept. In 2006-2008, several authors employed concepts like Type-2 fuzzy sets [12,24,375], Sugeno integral [367,368] and Choquet integral [10,11,53]. These fuzzy models show robustness in the presence of dynamic backgrounds [21]. Dempster-Schafer concepts were also be employed in foreground detection [234].

Based on machine learning, background modeling has been investigated by representation learning (also called subspace learning), support vector machines and neural networks modeling (conventional and deep neural networks). In 1999, reconstructive subspace learning models like Principal Component Analysis (PCA) [241] were introduced to learn the background in an unsupervised manner. Subspace learning models handle illumination changes more robustly than statistical models [79]. In further approaches, discriminative [92,93,221] and mixed [222] subspace learning models have been used to increase the performance for foreground detection. However, each of these regular subspace methods presents a high sensitivity to noise, outliers, and missing data. To address these limitations, since 2009, a robust PCA through decomposition into low-rank plus sparse matrices [38,124–127] has been widely used in the field. TThese methods are not only robust to changes in illumination but also to dynamic backgrounds [157,161,163,289,259,260]. However, they require batch algorithms, making them impractical for real-time applications. To address this limitation, dynamic robust PCA as well as robust subspace tracking [315,316] have been designed to achieve a real-time performance of RPCA-based methods. The most accomplished methods in this subspace learning category are GRASTA [135], incPCP [262], ReProCS [119] and MEROP [236]. However, tensor RPCA based methods

[155,290,201,81] allow to take into account spatial and temporal constraints making them more robust against noise. In 2006, support vector models [197,305–307, 324,325] have been introduced for background modeling in order to be more robust to dynamic backgrounds but their main drawback is their sensitivity to the training data. For all these models, the reader can refer to well-known exhaustive and detailed surveys [22,23,28,29,165]. Below we focus on neural networks models applied to background subtraction.

Schofield et al. [274] were the first to use neural networks for background modeling and foreground detection through the application of a Random Access Memory (RAM) neural network. However, a RAM-NN requires the images to represent the background of the scene correctly, and there is no background maintenance stage because once a RAM-NN is trained with a single pass of background images, it is impossible to modify this information. In a further study, Jimenez et al. [106] classified each zone of a video frame into three classes of background: static, noisy, and impulsive. The classification is conducted using a multilayer perceptron neural network, which requires a training set from specific zones of each training frame. In another study, Tavakkoli [304] proposed a neural network approach under the concept of novelty detector. During the training step, the background is divided in blocks. Each block is associated to a Radial Basis Function Neural Network (RBF-NN). Thus, each RBF-NN is trained with samples of the background corresponding to its associated block. The decision of using RBF-NN is because it works like a detector and not a discriminant, generating a close boundary for the known class. RBF-NN methods is able to address dynamic object detection as a single class problem, and to learn the dynamic background. However, it requires a huge amount of samples to represent general background scenarios. In Wang et al. [339], a hybrid probabilistic and ”Winner Take All” (WTA) neural architectures were combined into a single NN model. The algorithm is named Adaptive Background Probabilistic Neural Network (ABPNN) and it is composed of four layers. In the ABPNN model, each pixel is classified as foreground or background according to a conditional probability of being background. This probability is estimated by a Parzen estimation. The foreground regions are further analyzed in order to classify them as a motion or a shadow region. But, ABPNN needs to define specific initial parameter values (specific thresholds values) for each of the analyzed video. In Culibrk et al. [69], a feed-forward neural network is used for background modeling based on an adaptive Bayesian model called Background Neural Network (BNN). The architecture corresponds to a General Regression Neural Network (GRNN), that works like a Bayesian classifier. Although the architecture is proposed as supervised, it can be extended as an unsupervised architecture in the background model domain. The network is composed of three sub-networks: classification, activation, and replacement. The classifier sub-network maps the features background/foreground of a pixel to a probabilistic density function using the Parzen estimator. The network has two neurons, one of them estimates the probability of being background, and the other neuron computes the probability of being foreground. But, the main disadvantages are that the model is very complex and that it requires of three networks to define if a pixel belongs to the background. In a remarkable work, Maddalena and Petrosino [203,206,204,205] proposed a method called Self Organizing Background Subtraction (SOBS) based on a 2D self-organizing neural network architecture preserving pixel spatial relations. The method is considered as nonparametric, multi-modal, recursive and pixel-based. The background is automatically modeled through the neurons weights of the network. Each pixel is represented by a neural map with $n \times n$ weight vectors. The weights vectors of the neurons are initialized with the corresponding color pixel values using the HSV color space. Once the model is initialized, each new pixel information from a new video frame is compared to its current model to determine if the pixel corresponds to the background or to the foreground. In further works, SOBS was improved in several variants such as Multivalued SOBS [208], SOBS-CF [210], SC-SOBS [211], 3dSOBS+ [213], Simplified SOM [41], Neural-Fuzzy SOM [42] and MILSOBS [105]) which allow this method to be in the leader methods on the CDnet 2012 dataset [112] during a long time. SOBS show also interesting performance for stopped object detection [207, 209,212]. But, one of the main disadvantages of SOBS based methods is the need to manual adjust at least four parameters.

Deep learning methods based on deep neural networks (DNNs) with convolutional neural networks (CNNs), also called ConvNets, have alleviated the disadvantages of the previous approaches based on conventional neural networks [273,198, 117]. Although CNNs have existed for a long time, their success and use in computer vision have long been limited during a long period owing to the size of the available training sets, the size of the considered networks, and the computational capacity. In the area of computer vision the breakthrough was made in the field of image classification in 2012 by Krizhevsky et al. [176] who first used a supervised training of a large network with 8 layers and millions of parameters on the ImageNet dataset [76] with 1 million training images. With the progress made in storage for Big Data and the GPUs used for deep learning, even larger and deeper networks can be trained, and DNNs are now usable and have been widely applied in several computer vision tasks such object detection [377,108,107,258,74,39], semantic segmentation [101,102,123], video object segmentation [147,110,347,89,358,152,272, 184], video anomaly detection [118], person detection and tracking [33], dim small target detection [14], action recognition [356], intelligent transportation system [327, 329,360], remote sensing [328,330] to cite a few. More specifically, conventional object detection methods are built on handcrafted features and shallow trainable architectures but performance easily stagnates by constructing complex ensembles which combine multiple low-level image features with high-level context from object detectors and scene classifiers. In 2014, Girshick et al. [108] used CNNs for object detection obtaining a gap of more than 30% improvement over the previous best results. For intelligent transportation system, Wang et al. [327] designed a siamesed fully CNNs method for road detection from the perspective of moving vehicles in the application of autonomous driving. This method also clearly outperforms conventional approaches on the KITTI road detection benchmark. In 2018, Wang et al. [328] designed an end-to-end Attention Recurrent Convolutional Network (ARCNet) for scene classification of remote sensing. ARCNet gives better performance than handcrafted features and unsupervised learning feature based methods with a gap of 10%-20%. In 2019, Wang et al. [328] developed an end-to-end 2D CNN framework for hyperspectral image change detection in order to provide timely change information about large-scale Earth surface. This method called GETNET outperforms conventional methods based on PCA and SVM.

In the field of background subtraction, DNNs have also been successfully applied to background generation [121,249,349–351], background subtraction [9,19,30,59, 192], foreground detection enhancement [365], ground-truth generation [336], and the learning of deep spatial features [183,238,277,278,369]. More practically, Restricted Boltzman Machines (RBMs) were first employed by Guo and Qi [121] and Xu et al. [349] for background generation to further achieve moving object detection through background subtraction. In a similar manner, Xu et al. [350,351] used deep auto-encoder networks to achieve the same task whereas Qu et al. [249] used context-encoder for background initialization. As another approach, Convolutional Neural Networks (CNNs) has also been employed to background subtraction by Braham and Droogenbroeck [30], Bautista et al. [19] and Cinelli [59]. Other authors have employed improved CNNs such as cascaded CNNs [336], deep CNNs [9], structured CNNs [192] and two stage CNNs [374]. Through another approach, Zhang et al. [369] used a Stacked Denoising Auto-Encoder (SDAE) to learn robust spatial features and modeled the background with density analysis, whereas Shafiee et al. [277] employed Neural Reponse Mixture (NeREM) to learn deep features used in the Mixture of Gaussians (MOG) model [294]. In another study, Chan [43] proposed a deep learning-based scene-awareness approach for change detection in video sequences thus applying the suitable background subtraction algorithm for the correspond type of challenges. The motivations and contributions of this paper can be summarized as follows:

- Numerous studies have been published in the field of background subtraction since the work of Braham and Van Droogenbroeck in 2016, demonstrating the significant interest in deep neural networks in this field. Furthermore, each new method has been a top algorithm applied to the CDnet 2014 dataset, offering a significant improvement in performance compared to conventional approaches. In addition, DNNs have also been employed in background initialization, foreground detection enhancement, ground-truth generation, and deep learned features, showing its potential application in all fields of background subtraction.
- In this context, we provide an exhaustive comparative survey regarding DNN approaches used in the field of background initialization, background subtraction, and foreground detection and their features. To do so, we compare them in terms of the architecture and performance.

The rest of this paper is organized as follows. First, we provide in Section 2 a short summary of different key points in deep neural networks for novices. In Section 3, we review the different methods based on deep neural networks for the background generation of a video sequence. In Section 4, we describe methods based on deep neural networks for background subtraction with a full comparative overview of the architecture and challenges. In Section 5, deep learned features in this field are surveyed. In addition, we also provide a discussion regarding the adequacy of deep neural networks for background subtraction. Finally, experimental results are presented on the CDnet 2014 dataset in Section 8, and some concluding remarks are given in Section 10.

2 Deep Neural Networks: A Short Overview

2.1 Story Aspects: Birth, Empty Periods and Prosperity

Artificial Neural Networks (ANNs) have a long history with two periods of inactivity. Since their first development, an increasing number of sophisticated concepts and related architectures have been created for conventional ANNs, and later for deep neural networks. More precisely, ANNs progress from basic networks (less than three layers) to shallow networks (with three layers), and up to deep networks (more than three layers) [171]. Full surveys can be found studies by Schmidhuber [273] in 2015, Yi et al. [357] in 2016, by Liu et al. [198] in 2017, and by Gu et al. [117] in 2018. In addition, a full description of the different ANNs concepts are available at the Neural Network Zoo website[2]. Here, we briefly summarize the main stages of the ANN development. The use of ANNs began in 1943 with the threshold logic unit (TLU) [70].]. In a further study, in 1957 Rosenblatt [266] designed the first perceptron, whereas in 1962 Widrow [342,343] developed the Adaptive Linear Neuron (ADALINE). This first generation of neural networks were fundamentally limited in what they could learn to do. During the 1970s (the first empty period), research focused more on the XOR problem. The next period concerned the emergence of more advanced neural networks such as multilayer backpropagation neural networks, CNNs, and long short-term memory (LSTMs) for recurrent neural networks (RNNs) [143]. This second generation of neural networks mostly used back-propagation of the error signal to obtain derivatives for learning. During the second empty period, research focused more on a support vector machine (SVM), which is an extremely clever type of perceptron developed by Vapnik et al. [66]. TThus, many researchers abandoned research into neural networks with multiple adaptive hidden layers because an SVM works better with less computational time and training. With the progress of GPUs and the storage of big data, DNNs regained attention, and developments using new deep learning concepts such as deep belief networks [142,141] in 2006 and Generative Adversarial Networks (GANs) [87,271] in 2014. In 2017, Liu et al. [198] classified the deep neural network architectures in the following categories: restricted Boltzmann machines (RBMs), deep belief networks (DBNs), autoencoders (AEs) networks and deep Convolutional Neural Network (CNNs). In addition, deep probabilistic neural networks [104], deep fuzzy neural networks [77,96] and Generative Adversarial Networks (GANs) [87,271] can also be considered as other categories. Thus, the main architectures in deep neural networks can be classified into the following categories [273,198]:

- **Restricted Boltzmann machines:** RBMs have been widely used in deep neural networks owing to their historical importance and relative simplicity [98]. The RBM was designed by Smolensky under the name ”Harmonium” and its use is made popular by Hinton [142] in 2006. RBMs allow to generate stochastic models of ANNs which can learn the probability distribution according to their inputs. RBMs consist of a variant of Boltzmann machines (BMs) that can be considered as NNs with stochastic processing units connected bidirectionally. RBM is a special type of Markov random fields with stochastic visible units in one layer and stochastic observable units in the other layer. More technically, a RBM is a stochastic neural network meaning that the neuron-like units whose activations have a probabilistic element which depends on the neighbors they are connected to, while a classical neural network meaning these activations have binary activations. Figure 2.1 shows an a typical RBMs architecture. The neurons are restricted to form a bipartite graph and here is a full connection between the visible units and the hidden ones, while no connection exists between units from the same layer. To train an RBM, a Gibbs sampler is commonly used.

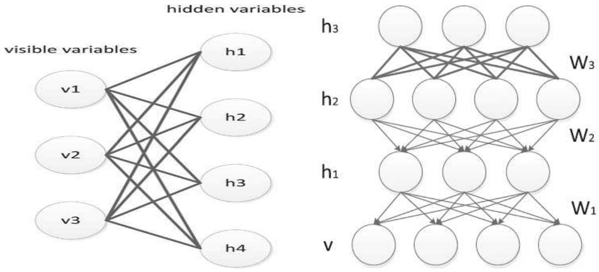


Fig. 2 From left to right: Schematic Illustrations of Restricted Boltzmann Machines (RBMs) and Deep Belief Networks (DBNs) (Images from Liu et al. [198]).

- **Deep Belief Networks:** To study the dependencies between the hidden and visible variables, Hinton [142] constructed the DBNs by stacking a bank of RBMs. Thus, the DBNs are composed of multiple layers of stochastic and latent variables and can be viewed as a special form of the Bayesian probabilistic generative model. DBNs can be viewed as a composition of simple and unsupervised networks that are RBMs with Sigmoid Belief Networks. Indeed, the main building block of a DBN is a bipartite undirected graphical model (i.e. RBM) in order to learn joint probability distribution of hidden and input variables. By generating new data with given joined probability distribution, DBNs are considered more flexible. For the training, the greatest advantage of DBNs is its ability of learning features, which is achieved by a layer-by-layer learning strategies where the higher level features are learned from the previous layers. Thus, DBNs provide a fast and layer-by-layer unsupervised training procedure while CNN required a full training procedure. To make learning easier, the network is designed so that no visible unit is connected to any other visible unit and no hidden unit is connected to any other hidden unit. In addition, DBNs are generative neural networks that stack RBMs which act as generative auto-encoders. DBNs are more effective than ANNs in the presence of problems with unlabeled data. Figure 2.1 shows an a typical DBNs architecture. Every two adjacent layers form an RBM. The visible layer of each RBM is connected to the hidden layer of the previous RBM and the top two layers are non-directional. The directed connection between the above layer and the lower layer is in a top-down manner. For training, different layers of RBMs in a DBN are trained sequentially. First, the lower RBMs are trained then the higher ones. After features are extracted by the top RBM, they are propagated back to the lower layers. In comparison with a single RBM, the stacked model increases the upper bound of the log-likelihood guaranteeing stronger learning abilities.

- **AutoEncoders (AEs) networks:** An autoencoder (also called an auto-associator) is another type of ANN, and is an unsupervised learning algorithm used to efficiently code a dataset for the purpose of a reduction in the dimensionality. AEs are also employed to learn generative data models. Figure 2.1 shows a typical AE architecture. The input data are converted into an abstract representation, which is then converted back into the original format using the encoder function. In practical terms, the AE is trained to encode the input into a representation from which the input can be reconstructed. Thus, the AE attempts to approximate the identity function during this process. The main advantage is that the AE can extract useful features continuously during the propagation and filter out any useless information. Thus, the efficiency of the learning process is improved because the input vector is transformed into a lower dimensional representation during the coding process. Deep autoencoders have demonstrated their effectiveness in discovering non-linear features across many problem domains, but require clean training data. However, in many real applications, data are often corrupted by large outliers or pervasive noise. To address this problem, in 2016, Jiang et al. [167] designed $l_{2,1}$-norm stacked robust autoencoders, whereas in 2017 Zhou and Paffenroth [383]] proposed the use of robust autoencoders based on the principle of an RPCA developed by Candès et al [38]. Thus, the input data A are split into two parts $A = L + S$, where L can be effectively reconstructed by a deep autoencoder and S contains the outliers and noise in the original data A. Because such a split increases the robustness of a conventional deep autoencoder, this model is called a d Robust Deep Autoencoder (RDA) [250]. In a similar manner, based on an RPCA, Chalapathy et al. [250] designed a robust autoencoder that learns a nonlinear subspace capturing the majority of data points, while allowing certain data to have an arbitrary corruption. In 2018, Dai et al. [73] demonstrated that Variational AutoEncoders (VAE) can be viewed as a natural evolution of recent robust PCA models, which are capable of learning nonlinear manifolds of unknown dimension obscured through gross corruptions. In practice, a linear deep autoencoder network (i.e., without the use of nonlinear activation functions at each layer) operates similarly as a dimensionality reduction method such as a PCA. In a similar manner, a robust deep autoencoder can be viewed as an extension of an RPCA in terms of nonlinear dimensions.
- **Deep Convolutional Neural Networks (CNNs):** CNNs are a subtype of the discriminative deep architecture and demonstrate suitable performance in processing 2D data like in images and videos [198]. The architecture of a CNN is inspired by the visual cortex of animals, and the concept is based on a time-delay neural network (TDNN). In a TDNN, the weights are shared in a temporal dimension, whereas the convolution replaces the general matrix multiplication in a CNN. Thus, the number of weights is decreased with a decrease in the complexity of the network. Furthermore, images can be directly imported into a network, avoiding the feature extraction procedure. CNNs were the first truly successful deep learning architecture owing to the successful training of hierarchical lay-

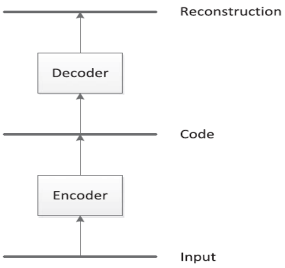


Fig. 3 Schematic Illustrations of AutoEncoders (AEs) networks (Images from Liu et al. [198]).

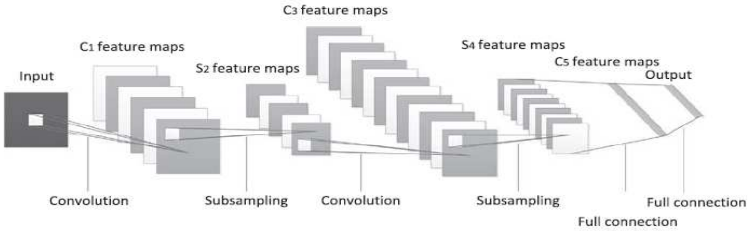


Fig. 4 Schematic Illustrations of Convolutional Neural Networks (CNNs) (Images from Liu et al. [198]).

ers. The CNN topology leverages spatial relationships with a decreasing number of parameters in the network, and the performance is improved using standard back-propagation algorithms. In addition, CNNs require minimal pre-processing, allowing an end-to-end solution. Figure 2.1 shows an a typical CNNs architecture also called ConvNets. However, Cohen and Shashua [62, 60] provided an architecture called SimNets which is a generalization of ConvNets driven by two operators. Experiments demonstrate the capability of achieving state of the art accuracy with networks that are an order of magnitude smaller than comparable ConvNets.

- **Deep probabilistic neural networks:** To consider the uncertainty, thereby providing important information regarding the reliability of predictions and the inner workings of a network, in 2018, Gast and Roth [104] introduced two lightweight deep probabilistic approaches to making supervised learning. Figure 2.1 shows an illustration of these two approaches. First, Gast and Roth [104] proposed the use of probabilistic output layers for classification and regression, which require only minimal changes to existing networks. Second, Gast and Roth [104] used density filtering, demonstrating that activation uncertainties can be propagated through the network. The two probabilistic networks maintain the predictive power of the deterministic counterpart, but yield uncertainties that correlate well with empirical errors induced through their predictions. In addition, the robustness to adversarial examples was significantly improved.

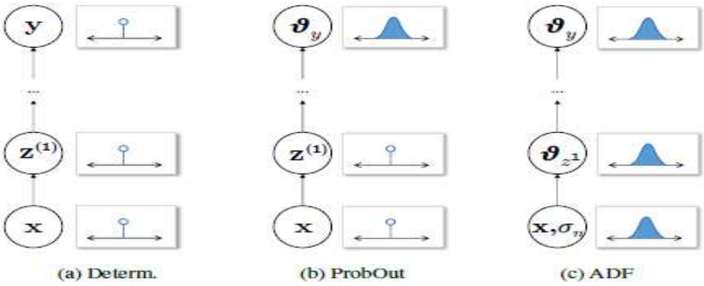


Fig. 5 Probabilistic Convolutional Neural Networks: a) Conventional CNNs with both activations and outputs as deterministic point estimates, b) Probabilistic CNNs with probabilistic output layers, and c) Probabilistic CNNs replacing all intermediate activations by distributions (Images from Gast and Roth [104]).

- **Deep fuzzy neural networks:**: Based on the principle of uncertainty, in 2017, Deng et al. [77] introduced the concept of fuzzy learning, providing a hierarchical deep neural network that derives information from both fuzzy and neural representations. Thus, the knowledge learned from these two respective views are fused, providing the final data representation to be classified. Figure 2.1 shows an illustration of the fuzzy DNNs architecture which consists of four parts. In 2018, Feng and Chen [96] designed a fuzzy RBM by replacing all real-valued parameters with fuzzy numbers. The FRBM then employs the crisp possibilistic mean value of a fuzzy number to defuzzify the fuzzy free energy function.

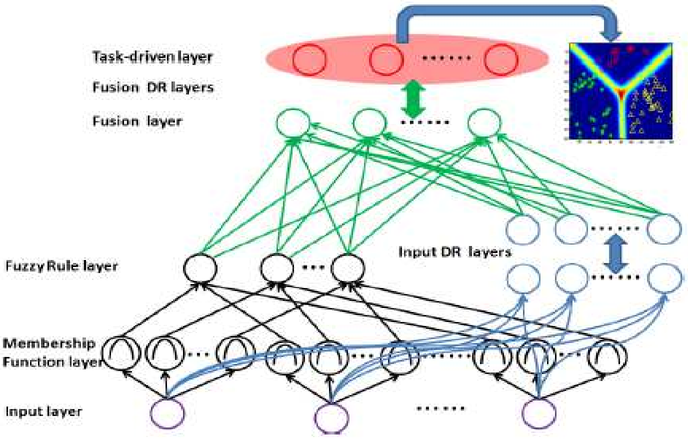


Fig. 6 Fuzzy Deep Neural Networks: Fuzzy logic representation part in black, Neural representation part in blue, Fuzzy-and-deep representation fusion part in green and the task driven learning part in red (Images from Deng et al. [77]).

- **Generative Adversarial Networks (GANs):** Generative Adversarial Networks (GAN) GANs represent a breakthrough in machine learning. Introduced in 2014 by Goodfellow et al. [87,271] in 2014, GANs provide a powerful framework for using unlabeled data in the training of machine learning models, and have become one of the most promising paradigms for unsupervised learning. More precisely, GANs allow estimating generative models using an adversarial process in which two models are trained: a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample was derived from the training data rather than the generative model [87]. To train the generative model, Goodfellow et al. [87] maximize the probability of a discriminative model making a mistake. The main advantages of a GAN is as follows: 1) Markov chains are not required, 2) only a backprop is used to obtain the gradients, 3) no inference is required during learning, and 4) a wide variety of functions can be employed. These advantages offer a low computational time. However, GANs also present a statistical advantage over a generator network that is not updated directly with data examples but with gradients circulating through the discriminator. Thus, the components of the input are not copied directly into the generator parameters [87].

The applications of these deep learning architectures are mainly in the areas of speech separation and recognition [338,302,340,321], computer vision [198] and pattern recognition [198]. In this context, DeepNet architectures for specific applications have emerged, such as the following: AlexNet developed in 2012 by Krizhevsky et al. [176]for image classification, VGG-Net designed in 2015 by Simonyan and Zisserman [287] for large-scale image recognition i, U-Net [263] developed in 2015 by Ronneberger et al. [263] for biomedical image segmentation, GoogLeNet with inception neural network introduced in 2015 by Szegedy et al. [299] for computer vision, and Microsoft Residual Network (ResNet) designed in 2016 by He et al. [136] for image recognition. Thus, all current architectures were designed for a target application such as speech recognition [114], computer vision [239] and pattern recognition [198] the specific features of which provide a very impressive performance in comparison with previous state-of-the-art methods based on a GMM and graph-cut, as with the problem of foreground detection/segmentation/localization. However, in order to obtain performance gains, the deep neural networks have grown larger and deeper, containing millions or even billions of parameters and over a thousand layers. The trade-off is that these large architectures require an enormous amount of memory, storage, and computation, thus limiting their usability [332]. However, many parameters are required with fully-connected layers that employ parameters highly inefficiently. To address this issue, more efficient parameterizations can be designed for fully-connected layers. Such compressed parameter spaces naturally lead to reduced memory and computational costs. Furthermore, high quality parameterizations can extract more meaningful information when relevant data is limited. In this context, several authors proposed deep neural network architecture which replaces matrices by tensors [332,237,64,63,61]. For example, Newman et al. [237] used a tensor neural network (t-NN) whereas Wang et al. [332] used a tensor ring factorization approach

[373] to compress both the fully connected layers and the convolutional layers of deep neural network obtaining Tensor Ring Networks (TR-Nets).

2.2 Features Aspects

Deep neural networks are parametric models that achieve sequential operations on their input data. Each operation, called a layer, consists of a linear transformation followed by a pointwise linear or nonlinear activation function [318]. In deep linear neural networks, the function class of a linear multilayer neural network only contains activation functions that are linear with respect to the input [171]. In contrast, nonlinear activation functions are employed in deep nonlinear neural networks. However, in both cases their loss functions in the weight parameters are non-convex. As shown in the previous section, DNNs are characterized by their architecture, which becomes increasingly sophisticated over time. In practical terms, an architecture consists of different layers, which are classified as input layers, hidden layers, and output layers. Each layer contains many neurons that are either activated or not following an activation function. An activation function can be viewed as the mapping of the input to the output using a non-linear transform function at each node. Different activation functions can be found in the literature, such as the sigmoid function [84], Rectified Linear Unit (ReLU) [246], and Probabilistic ReLU (PReLU) [137].Once the architecture is determined and the activation functions are chosen for each kind of layer, DNNs need to be trained using a large-scale dataset such as the ImageNet dataset [176], CIFAR-10 dataset and ILSVRC 2015 dataset for classification tasks. To do so, the architecture is exposed to the training dataset to learn the weights of each neuron in each layer. The parameters are learned using a cost function and are minimized on the desired and predicted outputs. The most common method for training is back-propagation. The gradient of the error function is typically computed on the correct output, and the predicted output is propagated back to the beginning of the network to update its parameters, which requires a gradient descent algorithm. Batch normalization, which normalizes mini-batches, can also be used to accelerate learning because it employs higher learning rates, and regularizes the learning. For the vocabulary, an epoch is a complete pass through a given dataset, and is thus the number times a neural network has been exposed once to every record of the dataset. An epoch is not an iteration, which corresponds to a single update of the neural net model parameters. Many iterations can occur before an epoch is complete. An epoch and an iteration are only identical if the parameters are updated once for each pass through the entire dataset. The reader can refer to the guide of Dumoulin and Visin [83] for more details.

2.3 Theoretical Aspects

The empirical success of deep learning presents numerous challenges to theoreticians. In 2018, Vidal et al. [319] pointed out three main factors, namely, the archi-

[2] http://www.asimovinstitute.org/neural-network-zoo/

tectures, regularization techniques, and optimization algorithms, which are critical to the training of well-performing DNNs. Understanding the necessity and interplay of these three factors is essential in an analysis of their success. Thus, the theoretical aspects mainly concern an understanding and provability and the stability of the DNNs [240,318,319,361], as well as their properties in the presence of adversarial perturbations [52,228–231,301,381], and their robustness in presence of noisy labels [308]. For this, the principle key features in the design of DNNs need to be mathematically investigated as follows [318,319]:

- **Architecture:** The number, size, and type of layers are the key characteristics of an architecture and the classes of functions that can be approximated using a feed-forward neural network. The key issue is how the chosen architecture, along with its depth and width, impact the expressiveness, which is its ability to approximate arbitrary functions of the input. Several studies [72,146,145,17] have shown that neural networks with a single hidden layer with sigmoidal activations are universal function approximators. However, a wide and shallow network has also been obtained using a deep network with significant improvements in performance [318]. Thus, deep architectures seem to be able to better capture invariant properties of the data as compared to their shallow counterparts. In practice, certain sub-classes of deep neural networks, such as scattering networks [32] are provably stable and locally invariant signal representations, and reveal the fundamental role of the geometry and stability in that both conditions generalize the performance of a modern deep convolution.
- **Optimization:** This concerns the training of the DNNs and contains two aspects, namely, the datasets used for training, and in most cases, the algorithm used to optimize the network. Indeed, the optimization problem is generally non-convex, and the main issues concern the guarantee of the optimality, the success of the stochastic gradient descent (SGD) following the appearance of the error surface, and whether the local minima are global property holds for deep nonlinear networks. To address the issue of non-convexity, a conventional strategy consists of initializing the network weights at random, and updating the weights using a local descent, checking whether the training error decreases sufficiently fast, and if not, choosing another initialization. In practice, this strategy often leads to different solutions for the network weights while giving approximately the same objective values and classification performance. Empirically, when the size of the network is sufficiently large and ReLU non-linearities are used, all local minima may be global [318]. SGD have been rigorously analyzed for convex loss functions; however, a loss is a non-convex function of the deep neural network parameters. Thus, the use of an SGD does not provide a guarantee of finding the global optimum. Moreover, critical points are more likely to be saddle points rather than spurious local minima [57] and the local minima concentrate near the global optimum. However, for certain types of neural networks in which both the loss function and the regularizer are sums of positively homogeneous functions of the same degree, Haeffele and Vidal [130,129] demonstrated that a local optimum, such as when many of the entries are zero, is also a global optimum. In 2016, Kawaguchi [171] demonstrated that, for an expected loss function of a deep nonlinear neural net-

work in which the function is non-convex and non-concave, every local minimum is a global minimum, and every critical point that is not a global minimum is a saddle point. The same statements hold for deep linear neural networks with any depth or width and no unrealistic assumptions.

- **Generalization and regularization properties:** The main concerns of this part are how well do DNNs generalize, how should DNNs be regularized, and how should under and over fitting be prevented? Indeed, the main critical issue of a DNN architecture is the ability to generalize from a small number of training examples. Based on statistical learning theory, it has been shown that the number of training examples needed to achieve good generalization increases polynomially with the size of the network. In a DNN, the training set contains much fewer data than the number of parameters, preventing an over-fitting using regularization techniques such as a Dropout [291] which freezes a random subset of the parameters at each iteration. Then, deep architectures produce an embedding of the input data that approximately keeps the distance between data points in the same class (i.e. the inter-class distance), while increasing the separation between classes (i.e. intra-class distance).
- **Stability and robustness properties:** Output instability of deep neural networks are due to small perturbations in the input that can significantly distort the feature embeddings and output of a neural network [276,378]. In 2015, Giryes et al. [109] demonstrate the stability of DNNs with random Gaussian weights that perform a distance-preserving embedding of the data. However, stability can be improved by forward propagation techniques inspired by systems of Ordinary Differential Equations (ODE) [128,44], and an efficient weight normalization technique [220].

Both the architecture and optimization can impact the generalization [240,318,319, 361]. Furthermore, several architectures are easier to optimize than others [318,319]. The first replies regarding the global optimality were provided in 2016 by Kawaguchi by Kawaguchi [171] and in 2018 by Yun et al. [361]. Their main conclusion is that DNNs are more difficult to train than classical neural networks owing to their non-convexity, but not too difficult owing to the nonexistence of poor local minima and the property of the saddle points. In 2018, Wang et al. [322] showed that deep neural networks can be better understood by utilizing the knowledge obtained by the visualization of the output images obtained at each layers. Other authors provided either a theoretical analysis or visualizing analysis in a context of an application. For example, Basu et al. [18] published a theoretical analysis for texture classification whereas Minematsu et al. [225,226] provided a visualizing analysis for background subtraction. Despite these first valuable investigations, an understanding of DNNs remains low. Nevertheless, DNNs have been successfully applied in many computer vision applications, with a large increase in performance. This success is intuitively due to the following reasons: 1) features are learned rather than being manual hand-crafted, 2) more layers capture more invariance characteristics, 3) more data allow a deeper training, 4) more computing CPU, 5) better regularization functions (Dropout [291]) and 6) new non-linearity functions (max-pooling, ReLU [235]).

2.4 Implementation Aspects

For software implementation, many libraries for the development of different programming languages are available for the implementation of DNNs. The most known libraries are Caffe [166], MatConvNet [317] from Matlab, Microsoft Cognitive Toolkit (CNTK), TensorFlow [88], Theano [3] and Torch [4]. All these software support interfaces of C, C++ and/or Python for quick development. For a full list, the reader are referred to go on the deeplearning.net[5] website. There is also a Deep Learning library for Java (DL4J[6]). For hardware implementation and optimization, there are several designed GPUs from NVIDIA with dedicated SDKs[7]. For example, the deep learning GPU Training System (DIGITS[8]) provides fast training of DNNs for computer vision applications like image classification, segmentation and object detection tasks whilst NVIDIA Jetson is designed for embedded systems. For NVIDIA Volta GPUs, TensorRT [9] allows optimizing the deep learning inference and runtime. It also allows the deployment of trained neural networks for inference to hyper-scale data centers, or embedding. A deep neural network accelerator based on FPGA has also been developed [150].

In the following sections, we survey all previous DNN approaches used in background/foreground separation by comparing their advantages and disadvantages, as well as their performance on the CDnet 2014 dataset

3 Background Generation

Background generation [26, 169, 214] (also called background initialization [156, 158] [159, 288], background estimation [65, 132], and background extraction [323]) refers the initialization of the background. In general, a model is often initialized using the first frame or a background model over a set of training frames that either contain or do not contain foreground objects. This background model can be the temporal average or temporal median. However, such a state is impossible in several types of environments owing to the required bootstrapping, and a sophisticated model is then needed to construct the first image. The top algorithms applied to the SBMnet dataset are Motion-assisted Spatio-temporal Clustering of Low-rank (MSCL) [160] and LaBGen [178–180] which are based on robust PCA [28, 29] and the robust estimation of the median, respectively. Figure 7 shows samples of background generation of three videos from the SBMI dataset [215]. In practical terms, the main challenge is to obtain the first background model when more than half of the training contains

[3] http://deeplearning.net/software/theano/

[4] http://torch.ch/

[5] http://deeplearning.net/software-links/

[6] https://deeplearning4j.org/

[7] https://developer.nvidia.com/deep-learning-software

[8] https://developer.nvidia.com/digits

[9] https://developer.nvidia.com/tensorrt

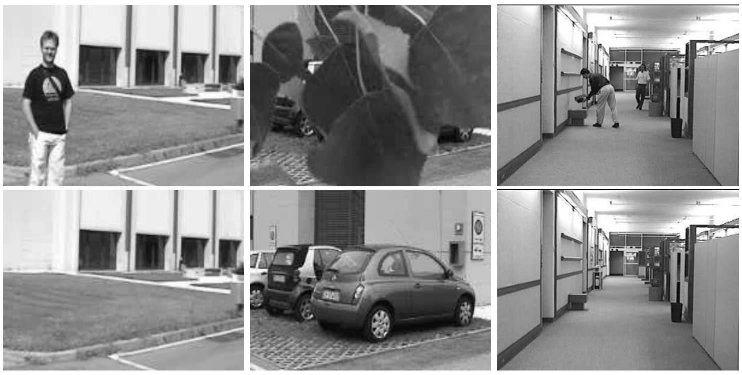


Fig. 7 Background Generation: The first row shows an original image of three videos from the SBMI dataset [215] and the second row shows the corresponding ground truth in the following order from left to right: CaVignal, Foliage and "Hall and Monitor".

Categories	Methods	Authors - Dates
Restricted Boltzmann Machines	Partially-Sparse RBM (PS-RBM)	Guo and Qi [121] (2013)
	Temp. Adaptive RBM (TARBM)	Xu et al. [349] (2015)
	Gaussian-Bernoulli RBM	Sheri et al. [283] (2018)
	RBM (PTZ Cameras)	Rafique et al. [253] (2014)
Deep Auto-encoders Networks	Deep Auto-encoder Networks (DAN)	Xu et al. [351] (2014)
	DAN with Adaptive Tolerance Measure	Xu et al. [350] (2014)
	Encoder-Decoder CNN (ED-CNN)	Qu et al. [249] (2016)
Convolutional Neural Networks	FC-Flownet	Halfaoui et al. [132] (2016)
	BM-Unet	Tao et al. [303] (2017)
Generative Adversarial Networks	Deep Context Prediction (DCP)	Sultana et al. [296] (2018)
	ForeGAN-RGBD	Sultana et al. [297] (2018)
	Illumination Invariant ForeGAN	Sultana and Jung [295] (2019)

Table 1 Deep Neural Networks in Background Generation: An Overview

foreground objects. This learning process can be achieved off-line and thus a batch-type algorithm can be applied. Deep neural networks are suitable for this type of task and several DNN methods have recently been used in this field. We classified such networks into the following categories described below. Table 1 shows an overview of these methods. In addition, a list of publications dealing with these networks is available at the Background Subtraction Website[10] and is updated regularly .

3.1 Restricted Boltzmann Machines (RBMs)

In 2013, Guo and Qi [121] were the first authors who applied Restricted Boltzmann Machine (RBM) to background generation by using a Partially-Sparse RBM (PS-RBM) framework in order to detect moving objects by background subtraction. This framework models the image as the integration of RBM weights as shown in Figure

[10] https://sites.google.com/site/backgroundsubtraction/background-initialization/neural-networks

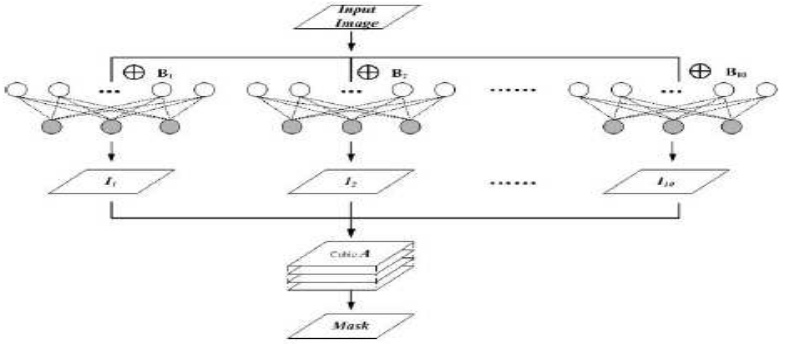


Fig. 8 PS-RBM Architecture (Image from Guo and Qi [121]).

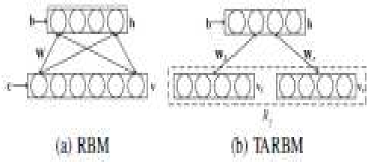


Fig. 9 Comparison between conventional RBM and TARBM (Image from Xu et al. [349]).

3.1. By introducing a sparsity target, the learning process alleviate the tendency of growth in weights. Once the sparse constraints are added to the objective function, the hidden units only keep active in a rather small portion on the specific training data. In this context, Guo and Qi [121] proposed a controlled redundancy technique, that allow the hidden units to learn the distinctive features as sparse as possible, meanwhile, the redundant part rapidly learns the similar information to reduce the total error. The PS-RBM provides accurate background modeling even in dynamic and noisy environments. Practically, PS-RBM provided similar results than DPGMM [131], KDE [85], KNN [387], and SOBS [203] methods on the CDnet 2012 dataset.

In 2015, Xu et al. [349] proposed a Temporally Adaptive RBM (TARBM) background subtraction to take into account the spatial coherence by exploiting possible hidden correlations among pixels while exploiting the temporal coherence too. Figure 3.1 illustrates the difference between a conventional RBM and a Temporally Adaptive RBM. As a result, the augmented temporally adaptive model can generate a more stable background given noisy inputs and adapt quickly to changes in the background while maintaining all advantages of PS-RBM including an exact inference and effective learning procedure. Figure 3.1 shows the pipeline of TARBM background subtraction. TARBM outperforms the standard RBM, and is robust in the presence of dynamic changes to the background and illumination.

In 2018, Sheri et al. [283] employed a Gaussian-Bernoulli restricted Boltzmann machine (GRBM), which differs from an ordinary restricted Boltzmann machine

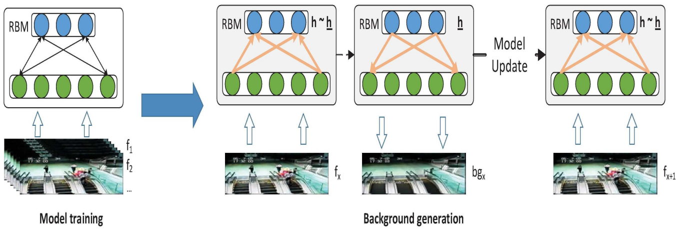


Fig. 10 TARBM Pipeline (Image from Xu et al. [349]).

(RBM), using real numbers as inputs. This network results in a constrained mixture of Gaussians, which is one of the most widely used techniques for solving the background subtraction problem. GRBM then easily learns the variance of the pixel values and takes advantage of the generative model paradigm of an RBM. In the case of PTZ cameras, Rafique et al. [253] modeled a background scene using an RBM. The generative modeling paradigm of an RBM provides an extensive and nonparametric background learning framework. An RBM was then trained using one-step contrastive divergence.

3.2 Deep Auto Encoder Networks (DAE)

In 2014, Xu et al. [351] designed a background generation method based on two auto-encoder neural networks. First, the approximate background images are computed using an auto-encoder network called a reconstruction network from the current video frames. Second, the background model is learned based on these background images using another auto-encoder network called a background network (BN). In addition, the background model is updated on-line to incorporate more training samples over time. Figure 3.2 shows the background generation pipeline. Experimental results on the I2R dataset [185] shows that DAN outperforms MOG [294], Dynamic Group Sparsity (DGS) [149], Robust Dictionary Learning (RDL) [372] and Online RDL (ORDL) [202]. In a further work, Xu et al. [350] improved this method by using an Adaptive Tolerance Measure Thus, DAN-ATM can handle large variations of dynamic background more efficiently than DAN. Experimental results on the I2R dataset [185] confirm this increase in performance.

Qu et al. [249] employed a context-encoder network for a motion-based background generation method by removing the moving foreground objects and learning the features. After removing the foreground, a context-encoder is also applied to predict the missing pixels of an empty region and to generate a background model of each frame. The architecture is based on AlexNet, which produces a latent feature representation of the input image samples with empty regions. The decoder has five upper convolutional layers and uses the feature representation to fill in the missing regions of the input samples. The encoder and the decoder are connected through a channel-wise fully connected layer. This allows information to be propagated within

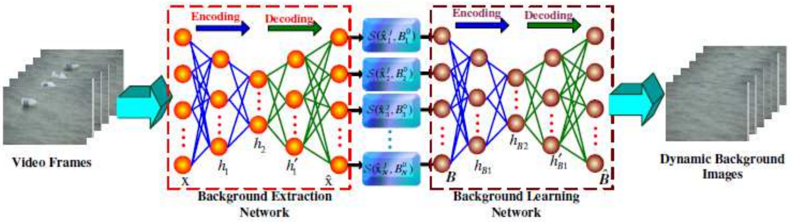


Fig. 11 Deep Auto Encoder Networks Pipeline (Image from Xu et al. [351]).

the activations of each feature map. The experiments conducted by Qu et al. [249] are limited but convincing.

3.3 FC-FlowNet

Halfaoui et al. [132] employed a CNN architecture for background estimation, which can provide a background image with only a small set of frames containing foreground objects. The CNN is trained using estimated background patches, followed by a post-processing step to obtain the final background image. More precisely, this architecture is based on FlownNetSimple [99],], which is a two-stage architecture developed for the prediction of the optical flow motion vectors. The first stage is contraction, whereas the second stage is refinement. The contraction stage is a succession of convolutional layers. This rather generic stage extracts high-level abstractions of the stacked input images, and forwards the gained feature maps to the upper convolutional refinement stage to enhance the coarse-to-fine transformations. Halfaoui et al. [132] adapted this architecture by providing a Fully-concatenated version called FCFlowNet. Experimental results on the SBMC 2016 dataset[11]demonstrate the robustness against very short or long sequences, a dynamic background, changes in illumination, and intermittent object motion.

3.3.1 U-Net

In 2017, Tao et al. [303]proposed an unsupervised deep learning model for background modeling called BM-Unet. This method is based on the generative U-Net architecture [263] which for a given frame (input) provides the corresponding background image (output) with a probabilistic heat map of the color values. However, to tackle camera jitter and quick changes in illumination, this method learns parameters automatically and uses the differences in intensity and optical flow features in addition to the color features. Moreover, BM-Unet can be applied to a new video sequence without the need for re-training. More precisely, Tao et al. [303] proposed

[11] http://pione.dinf.usherbrooke.ca/sbmc2016/

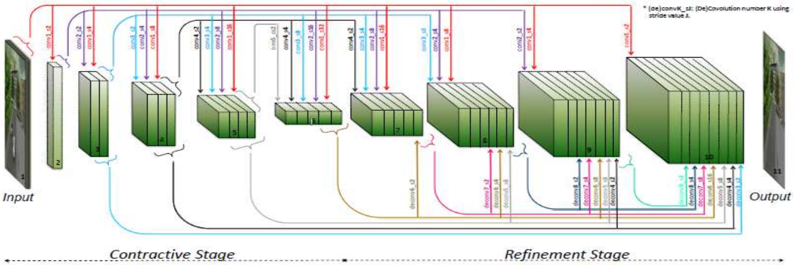


Fig. 12 FC-FlowNet Architecture (Image from Halfaoui et al. [132]).

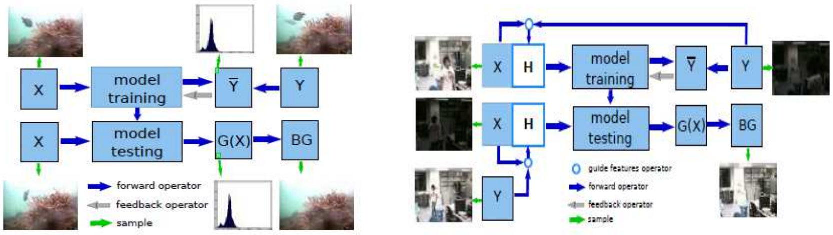


Fig. 13 From left to right: Baseline BM-Unet and Augmented BM-Unet (Image from Tao et al. [303]).

two algorithms named Baseline BM-Unet and Augmented BM-Unet that can handle static background and background with illumination changes and camera jitter, respectively. Figure 13 shows an illustration of the baseline BM-Unet and augmented BM-Unet architectures. The Augmented BM-Unet is based on the so called guide features which are used to guide the network to generate the background corresponding to the target frame. Experimental results [303] on the SBMnet dataset[12] [169] demonstrate promising results over neural networks methods (BEWiS [115], BE-AAPSA [255], and FC-FlowNet [132]), and state-of-the-art methods (Photomontage [2], LabGen-P [178]).

3.4 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GAN) have been a breakthrough in machine learning. Introduced in 2014, a GAN [87, 271] provides a powerful framework for applying unlabeled data to the training of machine learning models, and is one of the most promising paradigms for unsupervised learning. Based on a hybrid GAN, Sultana et al. [296] designed an unsupervised Deep Context Prediction (DCP) for background initialization in the context of background/foreground separation. Figure 3.4 shows

[12] http://scenebackgroundmodeling.net/

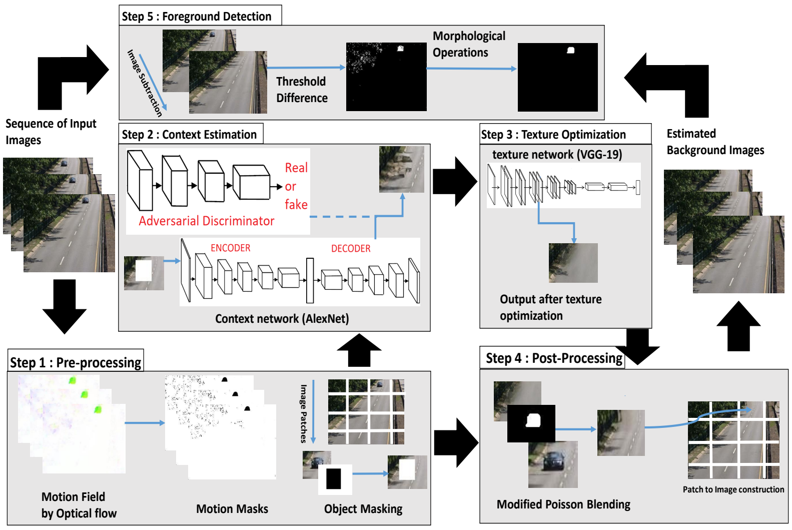

Step 5 : Foreground Detection
Image Subtraction
Threshold Difference
Morphological Operations
Sequence of Input Images
Estimated Background Images
Step 2 : Context Estimation
Real or fake
Adversarial Discriminator
ENCODER
DECODER
Context network (AlexNet)
Step 3 : Texture Optimization
texture network (VGG-19)
Output after texture optimization
Step 1 : Pre-processing
Image Patches
Motion Field by Optical flow
Motion Masks
Object Masking
Step 4 : Post-Processing
Patch to Image construction
Modified Poisson Blending

Fig. 14 Unsupervised GAN Deep Context Prediction (DCP) Pipeline (Image from Sultana et al. [296]).

the pipeline of DCP. More precisely, DCP is an unsupervised visual feature learning hybrid GAN based on context prediction. It is followed by a semantic inpainting network for texture optimization. Sultana et al. [296] additionally trained a context prediction model using scene-specific data patches with a resolution of 128×128 for three epochs. The texture optimization is done with VGG-16 network pre-trained on ImageNet [76] for classification. The frame selection for inpainting the background is then achieved through a summation of the pixel values using a forward frame difference technique. If the sum of the difference pixels is small, the current frame is then selected. Experimental results on the SBMnet dataset [169] show that DCP achieves an average gray level error of 8.724 which is the lowest among all compared low-rank methods, namely, RFSA [122], GRASTA [134], GOSUS [348], SSGoDec [384], and DECOLOR [385].

In a further study, Sultana et al. [297] used a GAN model for RGB-D video sequences by separately training two GANs: one for RGB video and one for depth video to generate background images. Each generated background sample is then subtracted from the given test sample to detect the foreground objects either in terms of the RGB or depth. Finally, the final foreground mask is obtained by combining the two foreground masks using a logical AND. Experiments on the SBM-RGBD[13] dataset [37] show that ForeGAN-RGBD model outperforms cwisardH+ [116], RGB-SOBS [211], and SRPCA [157] with an average F-Measure score of 0.8966.

In 2019, Sultana and Jung [295] provided an illumination invariant method using ForeGAN. Thus, this method proposed is inspired from ForeGAN-RGBD model designed by Sultana et al. [297], which has been adapted for background generation

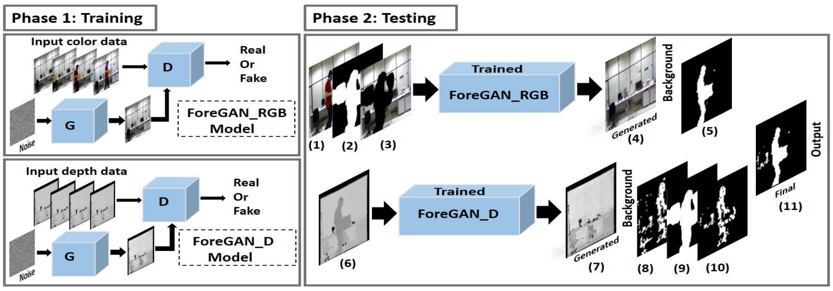


Fig. 15 Deep Context Prediction (DCP) for RGB-D videos (Image from Sultana et al. [297]).

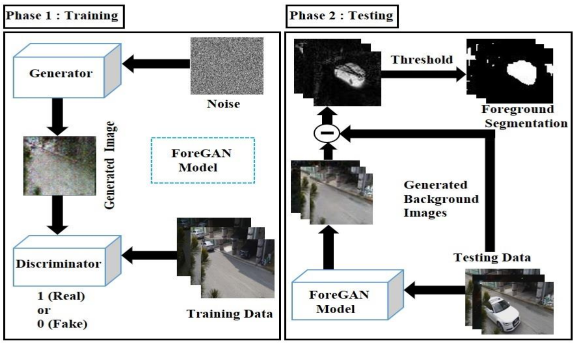


Fig. 16 Illumination Invariant ForeGAN Pipeline (Image from Sultana an Jung [295]).

by introducing scene-specific illumination information into DCGAN model [251]. First, the ForeGAN model is trained on background image samples with various illumination conditions including dynamic changes. For testing, the GAN model generates the same background sample as test sample with similar illumination conditions via back-propagation technique. The generated background sample is then subtracted from the given test sample to segment foreground objects. Experimental results on the Illumination Conditions from Dawn until Dusk (ICDD[14]) dataset show that Illumination-Invariant ForeGAN outperforms robust subspace learning methods, namely, GRASTA [134], DECOLOR [385], 3TD [242] RMAMC [353] TVRPCA [40].

[13] http://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html

[14] https://sites.google.com/view/icdddataset/

Categories	Methods	Authors - Dates
Encoder-Decoder Networks	Multi-scale Recurrent ED (MSRNN) [Semi-supervised]	Choo et al. [56] (2018)
	Modified MSRNN [Unsupervised]	Choo et al. [55] (2018)
	Variational autoencoders (DeepPBM)) [Unsupervised]	Farnoosh et al. [94] (2019)
Convolutional Neural Networks [Supervised]	CNN (ConvNets)	Braham and Van Droogenbroeck [30] (2016)
	CNN (ConvNets)	Bautista et al. [19] (2016)
	CNN (ConvNets) (Analysis) (2)	Minematsu et al. [225] (2017)
	CNN (Pedestrian Detection)	Yan et al. [352] (2018)
	CNN (GoogLeNet)	Weinstein [341] (2018)
	CNN (RPoTP feature)	Zhao et al. [371] (2018)
	CNN (Depth feature)	Wang et al. [333] (2018)
	CNN (Moving camera)	Afonso et al. [1] (2018)
Multi-scale and cascaded CNN [Supervised]	cascaded CNN (Ground-Truth)	Wang et al. [186] (2016)
	FgSegNet-M	Lim and Keles [193] (2018)
	FgSegNet-S	Lim and Keles [194] (2018)
	FgSegNet-V2	Lim et al. [195] (2018)
	MCSS	Liao et al. [191] (2018)
	Guided Multi-scale CNN	Liang et al. [190] (2018)
	MsEDNet	Patil et al. [245] (2018)
Fully CNNs [Supervised]	Basic Fully CNN	Cinelli [59] (2017)
	Basic Fully CNN	Yang et al. [355] (2018)
	Multiview recep. field FCN (MV-FCN)	Akilan et al.[3] (2018)
	Multiscale Fully CNN (MFCN)	Zeng and Zhu [366] (2018)
	MFCN with Contrast Layers (MFCN-CL)	Zeng and Zhu [364] (2018)
	CNN-SFC (Foreground Masks)	Zeng et al. [365] (2018)
	Fully Conv. Semantic Net. (FCSN)	Lin et al. [196] (2018)
Deep CNNs [Supervised]	Deep CNNs	Babaee et al. [9] (2017)
	TCNN/Joint TCNN	Zhao et al [374] (2017)
	Adaptive deep CNN (ADCNN)	Li et al. [186] (2018)
	SFEN	Chen et al. [50] (2018)
	MSFgNet/MSFgNet-I	Patil and Murala [244] (2018)
Structured CNNs [Supervised]	Struct CNNs	Lim et al. [192] (2017)
	Encoder-Decoder Structured CNNs	Le and Pham [181] (2018)
Double Encoding CNNs [Supervised]	Double Encoding/Slow Decoding CNNs (DESD)	Akilan and Wu [5] (2018)
	sEnDec	Akilan [4] (2018)
3D CNNs [Supervised]	3D-CNNs	Sakkos et al. [270] (2017)
	3D-CNNs	Gao et al. [100] (2018)
	STA-3D ConvNets (ReMoteNet)	Yu et al. [359] (2017)
	3D Atrous CNN (ConvLSTM)	Hu et al. [148] (2018)
	FC3D	Wang et al. [337] (2018)
	Multi-scale FC3D (MFC3D)	Wang et al. [337] (2018)
	3D-CNN with LSTM	Akilan [4] (2018)
Retrospective Convolutions [Supervised]	Atrous retrospective convolution (ARConv)	Chen et al. [46] (2018)
	Atrous Retrospective Pyramid Pooling (ARPP)	Chen et al. [46] (2018)
Generative Adversarial Networks [Unsupervised]	BScGAN	Bakkay et al. [15] (2018)
	Bayesian GAN (BGAN)	Zheng et al. [379] (2018)
	Bayesian Parallel Vision GAN (BPVGAN)	Zheng et al. [380] (2018)
	Neural Unsupervised Moving Object Detection (NUMOD)	Bahri et al. [13] (2018)
	Multi-Task GAN (MT-GAN)	Sakkos et al. [269] (2018)

Table 2 Deep Neural Networks in Background Subtraction: An Overview

Methods	Input	Output	Architecture Encoder/Decoder	Additional Architecture	Activation Function	Conv. Layers	Fully Conv.	Implementation Framework
Basic CNNs								
ConvNets [30]	Backg. (Median) Current Image	Foreground	LeNet-5 [71]	-	ReLU/Sigm.	2	1	-
Basic CNNs [186]	Current Image	Foreground	CNN-1	-	ReLU/Sigm.	4	2	Caffe [166]/MatConvNet [317]
Basic CNNs [352]	Backg. Visible (Median) Backg. Thermal (Median) Current Image (Visible) Current Image (Thermal)	GT	CNN	-	ReLU/Sigm.	4	-	-
Basic CNNs [341]	Backg. (Median) Current Image	Foreground (Bound. Box)	GoogLeNet [299]	-	ReLU/Sigm.	-	-	Tensorflow [88]
Basic CNNs [371]	Current Image (RPoTP)	Foreground	CNN	-	ReLU	-	1	-
Basic CNNs [333]	Background Image (Average) (Depth)	Foreground	CNN	(MLP)	ReLU/Sigmoid	3	3	-
	Current Image (Depth)			-	-	-	-	-
Multi-scale and cascaded CNNs								
Multi-scale CNNs [186]	Current Image	GT	CNN-1	-	ReLU/Sigm.	-	-	Caffe [166]/MatConvNet [317]
cascaded CNNs [186]	Current Image	GT	CNN-1	CNN-2	ReLU/Sigm.	-	-	Caffe [166]/MatConvNet [317]
FgSegNet-M [193]	Current Image	Foreground	VGG-16 [287]	TCNN	ReLU/Sigm.	4	-	Keras [54]/TensorFlow [88]
FgSegNet-S [194]	Current Image	Foreground	VGG-16 [287]	TCNN/FPM	ReLU/Sigm.	4	-	Keras [54]/TensorFlow [88]
FgSegNet-V2 [195]	Current Image	Foreground	VGG-16 [287]	TCNN/FPM Feat. Fusions	ReLU/Sigm.	4	-	Keras [54]/TensorFlow [88]
MCSS [191]	Backg. Current Image	Foreground	ConvNets [30]	-	ReLU/Sigm.	2	2	-
Guided Multi-scale CNN [190]	Current Image	Foreground	ConvNets [30]	Guided Learning	ReLU/Sigm.	4	-	-
MsEDNet [245]	Back. (Temp. Histogram)	Foreground	Compact CNN	Saliency Map	-	2	-	-
Fully CNN								
Fully CNNs [59]	Backg. (Median) Current Image	Foreground	LeNet-5 [71]	-	ReLU/Sigm.	4	-	Torch7
Fully CNNs [59]	Backg. (Median) Current Image	Foreground	ResNet [139]	-	ReLU/Sigm.	-	-	Torch7
Deep FCNNs [355]	Current Image	Foreground	Multi. Branches (4)	CRF	PReLU [137]	5 (Atrous)	1	-
MV-FCN [3]	Current Image	Foreground	U-Net [263]	2CFFs/PFF	ReLU/Sigm.	(2D Conv.)	1	Keras/Python
MFCN [366]	Current Image	Foreground	VGG-16 [287]		ReLU/Sigm.	5	-	TensorFlow [88]
CNN-SFC [365]	3 For. Masks	Foreground	VGG-16 [287]		ReLU/Sigm.	13	None	TensorFlow [88]
FCSN [196]	Backg. (SuBSENSE) Current Image	Foreground	FCN/VGG-16 [199]		ReLU/Sigm.	20	3	TensorFlow [88]
Deep CNNs								
Deep CNN [9]	Backg. (SuBSENSE /FTSG) Current Image	Foreground	CNN	Multi-Layer Perceptron (MLP)	ReLU/Sigm.	3	-	-
TCNN/Joint TCNN [374]	Backg. Current Image	Foreground	MCFC (VGG-16)	DCGAN [252]/ Context Enc. [243]	ReLU/Sigm.	-	-	Caffe [166]/DeepLab [47]
ADCNN [186]	Current Image	Foreground (Bound. Box)	T-CNN S-CNN, C-CNN	-	ReLU/Sigm.	7	None	Caffe [166]
SFEN [50]	Current Image	Foreground	VGG-16 GoogLeNet [299] ResNet	Attention ConvLSTM/ STN/CRF	ReLU/Sigm.	-	-	-
MSFgNet [244]	Background (BENet [244]) Current Image	Foreground		SMNet [244]	BiReLU [34,346]	2	1	-
Structured CNN								
Struct CNN [192]	Back. (Median) Current Image t Image t-1	Foreground	VGG-16	-	PReLU [137]	13	-	Caffe [166]
3D CNNs								
3D ConvNet [270]	10 Frames	Foreground	C3D Branch [311]	-	-	6 (3D Conv.)	-	Caffe [166]
3D CNNs [100]	5 Frames	Foreground		-	tanh	4 (3D Conv.)	2	-
STA-3D ConvNets (ReMoteNet) [359]	Current Image	Foreground (Bound. Box)	Modified C3D Branch [359]	ST Attention ConvLSTM	ReLU	(3D Conv.)	-	TensorFlow [88]
3D Atrous CNN [3]	Current Image	Foreground	3D Atrous ConvLSTM	-	ReLU	5 (3D Conv.)	-	TensorFlow [88]
FC3D [337]	16 frames	Foreground	3D-CNN	-	ReLU	3 (3D Conv.)	-	TensorFlow [88]
MFC3D [337]	16 frames	Foreground	3D-CNN	-	ReLU	3 (3D Conv.)	-	TensorFlow [88]
Generative Adversarial Networks								
BScGAN [15]	Back. (Median)	Foreground	cGAN [151]	-	Leaky ReLU/Tanh	8	-	Pytorch
	Current Image		Discrim. net		Leaky ReLU/Sigm	4	-	Pytorch
BGAN [379]	Back. (Median) Current Image	Foreground	Bayesian GAN	-	-	-	-	-
BPVGAN [379]	Back. (Median) Current Image	Foreground	Paralell Bayesian GAN	-	-	-	-	-
NUMOD [13]	Current Image Illum. Image Foreground	Back.	GFCN Bayesian GAN Bayesian GAN	-	ReLU/Sigm.	-	-	-

Table 3 Deep Neural Networks Architecture in Background Subtraction: A Comparative Overview. ”-” stands for ”not indicated” by the authors.

Methods	Muti-scale (Size)	Training (Over-fitting)	Training (GT)	Spatial (Pixel)	Computation	End-to-End	Long-Term (Temporal)	Features	Type
Basic CNNs									
ConvNets [30]	No (27×27)	Scene-specific	GT/IUTIS	No	Yes	No (Pre-proc.)	No	Grey	Generator
Basic CNNs [352]	No (64×64)	Scene-specific	GT	No	No	No (Pre-proc.)	No	RGB/IR	Generator
Basic CNNs [186]	No (31×31)	Scene-specific	GT	No	Yes	Yes	No	RGB	Generator
Basic CNNs [371]	Frame	-	one GT	No	-	No (RPoTP)	Yes	RPoTP feature [371]	Generator
Basic CNNs [333]	Patch	-	GT (SBM-RGBD)	No	No	No (Pre-process.)	No	Depth feature	Generator
Multi-scale and cascaded CNNs									
Multi-scale CNNs [186]	3 scales	Scene-specific	GT	cascaded (2)	Yes	Yes	No	RGB	Generator
cascaded CNNs [186]	3 scales	Scene-specific	GT	cascaded (2 levels)	Yes	Yes	No	RGB	Generator
FgSegNet-M [193]	3 scales	Imbalanced data	GT	TNN	18 fr/s	Yes	No	RGB	Generator
FgSegNet-S [194]	FPM	Imbalanced data	GT	TNN	-	Yes	No	RGB	Generator
FgSegNet-V2 [195]	M-FPM	Imbalanced data	GT	TNN	-	Yes	No	RGB	Generator
MCSS [191]	3 scales (27×27)	Scene-specific	GT (Small Number)	cascaded (2 levels)	-	Yes	No	Grey	Generator
Guided Multi-scale [190]	3 scales (31×31)	Scene-specific	GT	-	-	No (Post-proc.)	No	RGB	Generator
MsEDNet [245]	2 scales (512×512)	Scene-specific	GT	Saliency Map	103 msec/fr	No (Pre-proc.)	No	Grey	Generator
Fully CNNs									
Fully CNNs [59]	No	Scene-specific	GT	No	Yes	Yes	No	Grey	Generator
Deep FCNNs [355]	No	-	GT	Atrous	Yes	Yes	No	-(RGB?)	Generator
MV-FCN [3]	Inception Mod.	-	GT	-	-	Yes	Encoder	-(RGB?)	Generator
MFCN [366]	Yes ($224 \times 244 \times 3$)	-	Mean	-	27 fr/s	Yes	No	Infrared	Generator
	Yes ($224 \times 244 \times 3$)	-	Mean	-	-	Yes	No	RGB	Generator
CNN-SFC [365]	Semantic	No	GT	No	-	No	No	Black/White	Generator
FCSN [196]	Semantic	No	GT/SuBSENSE	Semantic	48 fr/s	Yes	No	-(RGB?)	Generator
Deep CNNs									
Deep CNN [9]	No (37×37)	Scene-specific	GT	No	Yes	No (Post-proc.)	No	RGB)	Generator
TCNN/Joint TCNN [374]	Yes (961×961)	Background	GT	No	5 fr/s	Yes	No	RGB	Generator
	Atrous Sampling Rate	Generation	(PASCAL VOC 2012)						
ADCNN [186]	Yes	Discriminative	GT	No	-	Yes	No	RGB	Generator
		Features	(CUHK, MIT, PETS)						
SFEN [50]	Semantic	No	GT	STN	15 fr/s	Yes	No	RGB	Generator
SFEN+CRF [50]	Semantic	No	GT	STN/CRF	6 fr/s	Yes	No	RGB	Generator
SFEN+PSL+CRF [50]	Semantic ($224/times224$)	No	GT	STN/CRF/PSL	5 fr/s	Yes	ConvLSTM	RGB	Generator
MSFgNet/MSFgNet-I [244]	256×256	Scene-specific	GT (CDnet 2014, LASIESTA, I2R)	Yes	-	Yes	Yes	RGB	Generator
Structured CNNs									
Struct CNN [192]	Contours (336×336)	No	GT	Superpixel	-	No (Post-proc)	No	Grey	Generator
3D CNNs									
3D ConvNet [270]	Multi-kernel upsampling	Yes	GT	No	-	Yes	3D	-(RGB?)	Generator
3D CNNs [100]	$17 \times 17 \times 3$	Yes	GT (CDnet 2012)	Yes	-	Yes	3D	-(RGB?)	Generator/Predictor
STA-3D ConvNets (ReMoteNet) [359]	1280×720)	No	GT	STA ConvLSTM	Fast	Yes	STA ConvLSTM	RGB	Generator
3D Atrous CNN [3]	320×240	No	GT	Atrous	-	Yes	3D/ConvLSTM	-(RGB?)	Generator
FC3D [337]	No	Yes	GT (Sports1-M)	3D	-	Yes	Yes	RGB	Generator
MFC3D [337]	Yes	Yes	GT (Sports1-M)	3D	12fps	yes	Yes	RGB	Generator
Generative Adversarial Networks									
BScGAN [15]	256×256)	No	GT	No	10 fr/s	Yes	No	-(RGB?)	Generator/Discriminator
BGAN [379]	-	-	GT	-	-	Yes	-	-	Generator/Discriminator
BPVGAN [379]	-	-	GT	-	Parallel Implem.	Yes	-	-	Generator/Discriminator
NUMOD [13]	Frame	No	$I = B + C + F$	No	-	Yes	No	RGB	Generator/

Table 4 Deep Neural Networks in Background Subtraction: A Comparative Overview for Challenges. ”-” stands for ”not indicated” by the authors.

4 Background Subtraction

Background subtraction consists of comparing the background image with the current image to label pixels as background or foreground pixels.The top-three algorithms on the large-scale dataset CDnet 2014 for supervised approaches are DNN-based methods, namely, FgSegNet [192], BSGAN [379], cascaded CNN [336] followed by three non-supervised multi-feature/multi-cue approaches, namely, SuBSENSE [292], PAWCS [293], IUTIS [20]. This is a classification task, which can be successfully achieved using a DNN. Different methods for this have been developed, and we review them in the following sub-sections. Table 2 shows an overview of these methods. In addition, the list of publications is available at the Background Subtraction Website[14] and is regularly updated.

4.1 Convolutional Neural Networks

In 2016, Braham and Van Droogenbroeck [30] were the first authors to use Convolutional Neural Networks (CNNs) for background subtraction. This model named ConvNet has a similar structure than LeNet-5 [71] (See Figure 4.1). TThus, the background subtraction model involves four stages: background image extraction using a temporal gray-scale median, specific-scene dataset generation, network training, and background subtraction. More precisely, the background model is built for a specific scene. For each frame in a video sequence, image patches that are centered on each pixel are extracted and are then combined with the corresponding patches from the background model. Braham and Van Droogenbroeck [30] used a patch size of 27×27. After, these combined patches are fed to the network to predict probability of foreground pixels. For the architecture, Braham and Van Droogenbroeck [30] employed 5×5 local receptive fields, and 3×3 non-overlapping receptive fields for all pooling layers. The first and second convolutional layers have 6 and 16 feature maps, respectively. The first fully connected layer has 120 hidden units and the output layer consists of a single sigmoid unit. The algorithm needs for training the foreground results of a previous segmentation algorithm named IUTIS [20] or the ground truth information provided in CDnet 2014 [335]. Half of the training examples are used for training ConvNet and the remaining frames are used for testing. By using the results of the IUTIS method [20], the segmentation produced by the ConvNet is very similar to other state-of-the-art methods whilst the algorithm outperforms all other methods significantly when using the ground-truth information especially in videos of hard shadows and night videos. Evaluated on the CDnet 2014 dataset (excluding the IOM and PTZ categories), this method with IUTIS and GT achieved an average F-Measure of 0.7897 and 0.9046, respectively. In 2016, Baustita et al. [19] also used a simple CNN but for the specific task of vehicle detection. For pedestrian detection, Yan et al. [352] employed the similar scheme with both visible and thermal images. Then, the inputs of the network have a size of $64 \times 64 \times 8$ which includes the visible frame (RGB), thermal frame (IR), visible background (RGB) and thermal background (IR). The outputs of the network have a size of $64 \times 64 \times 2$. Experiments on OCTBVS

[14] https://sites.google.com/site/backgroundsubtraction/recent-background-modeling/deep-learning

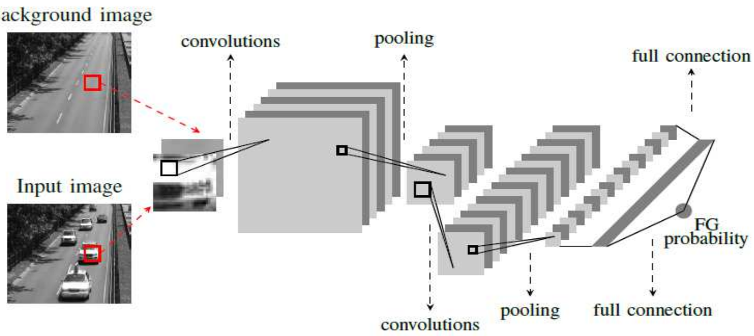


Fig. 17 ConvNet's Architecture: The network is trained with two small patches extracted from the input and background images in gray-scale. The network is inspired by LeNet-5 network (Image from Braham and Van Droogenbroeck [30]).

dataset[16] show that this method outperforms T2-FMOG [12], SuBSENSE [292], and DECOLOR [385]. For biodiversity detection in terrestrial and marine environments, Weinstein [341] employed the GoogLeNet architecture integrated in a software called DeepMeerkat[17]. Experiments on humming bird videos show robust performance in challenging outdoor scenes where moving foliages occur.

Remarks: ConvNet is the simplest way to learn the differences between the background and foreground when using a CNN. The study by Braham and Van Droogenbroeck [30] has a significant merit of being the first application of deep learning for background subtraction, and can thus be used as a reference for comparison in terms of the improvement in performance. But, it presents several limitations: 1) It has difficulty learning high-level information through patches [196]; 2) Because of an over-fitting caused by highly redundant data for training, the network is scene-specific. In practice, it can only process a certain type of scenery, and needs to be retrained for other video scenes [9]. This fact is usually not a problem because the camera is fixed when filming similar scenes. However, this may not be the case for certain applications, as pointed out by Hu et al. [148]; 3) Each pixel is processed independently, and the foreground mask may then contain isolated false positives and false negatives ; 4)) It is computationally expensive owing to a large number of patches extracted from each frame, as stated by Lim and Keles [193]; 5) It requires a preprocessing or post-processing of the data, and hence is not based on an end-to-end learning framework [148]; 6) ConvNet uses few frames as input, and thus cannot consider the long-term dependencies of the input video sequences [148]; and finally 7) ConvNet is a deep encoder-decoder network, namely, a generator network. However, a classical generator network produces blurry foreground regions, and such networks cannot preserve the object edges because they minimize the classical loss functions

[16] http://vcipl-okstate.org/pbvs/bench/

[17] http://benweinstein.weebly.com/deepmeerkat.html

(e.g., Euclidean distance) between the predicted output and the ground-truth [196]. Since the introduction of this valuable work, posterior methods developed in the literature have attempted to alleviate these limitations, which are the main challenges to the use of a DNN in background subtraction. Table 3 shows a comparative overview with all the posterior methods whereas Table 4 show an overview in terms of the challenges. These tables are discussed in detail in Section 6.

4.2 Multi-scale and cascaded CNNs

In 2016, Wang et al. [336] proposed a deep learning method for an iterative ground-truth generation process in the context of background modeling algorithms validation. In order to yield the ground truths, this method segments the foreground objects by learning the appearance of foreground samples. Figure 4.2 illustrates the pipeline. First, Wang et al. [336] designed basic CNN and the multi-scale CNN which processed each pixel independently based on the information contained in their local patch of size 31×31 in each channel RGB. The basic CNN model consists of 4 convolutional layers and 2 fully connected layers. The first 2 convolutional layers come with 2×2 max pooling layer. Each convolutional layer uses a filter size of 7×7 and Rectified Linear Unit (ReLU) as the activation function. By considering the CNN output as a likelihood probability, a cross entropy loss function is employed for training. Figure 4.2 shows the corresponding basic CNN architecture. Because, this basic model processes patches of size 31×31, its performance is limited to distinguish foreground and background objects with the same size or less. This limitation is alleviated using a multi-scale CNN model, which gives three outputs of three different sizes that are further combined in the original size. Figure 4.2 shows the multi-scale CNN architecture.T o model the dependencies among adjacent pixels and thus enforce the spatial coherence, Wang et al. [336] employed a multi-scale CNN model with a cascaded architecture, called a cascaded CNN. A CNN has the advantage of learning or extracting its own features, which may be better than hand-designed features. To learn the foreground features, a CNN is fed with manually generated foreground objects from some frames of a video sequence. After this step, the CNN employs generalization to segment the remaining frames of the video. Wang et al. [336] trained scene specific networks using 200 frames by manual selection. cascaded CNN provides an overall F-Measure of 0.9209 in CDnet2014 dataset [335]. For the cascaded CNN's implementation[18] available online, Wang et al. [336] used the Caffe library[19] [166] and MatConvNet[20]. The limitations of cascaded CNN are as follows: 1) it is more dedicated to ground-truth generation than an automated background/foreground separation method, and 2) it is also computationally expensive.

In 2018, Lim and Keles [193] proposed a method called FgSegNet-M[21] based on a triplet CNN and a transposed convolutional neural network (TCNN) attached to the end of the network in an encoder-decoder structure. Figure 4.2 illustrates the FgSegNet architecture. Practically, the four blocks of the pre-trained VGG-16 [287] are employed at the beginning of the proposed CNN under a triplet framework as a

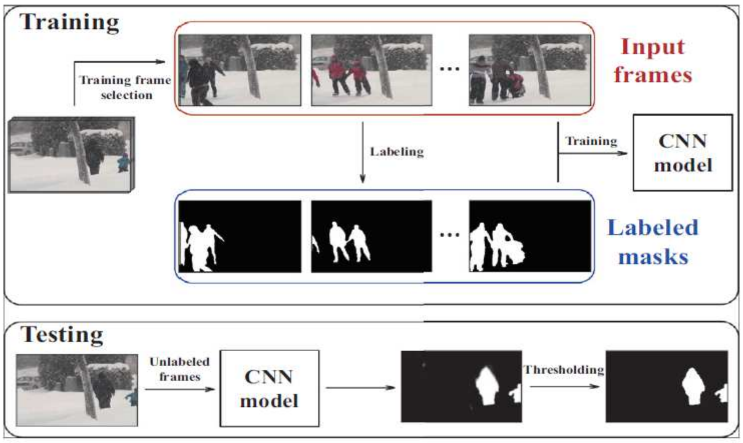


Fig. 18 Pipeline for Ground-truth Generation Process via Multi-scale and Cascade CNNs (Image from Wang et al. [336]).

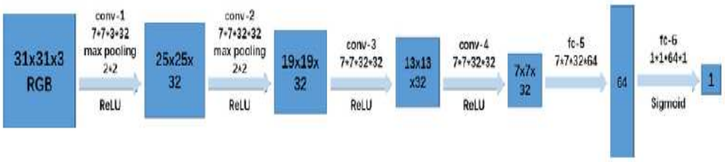


Fig. 19 Basic CNN Architecture: 4 convolutional layers, 2 fully connected layer. The first 2 convolutional layers come with a 2 × 2 max pooling layer (Image from Wang et al. [336]).

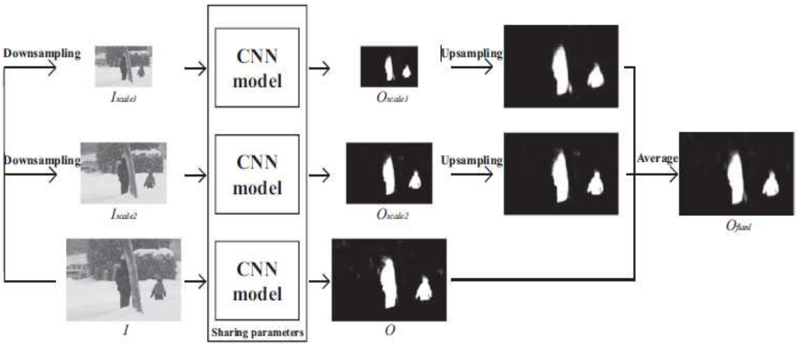


Fig. 20 Multi-scale CNN Architecture (Image from Wang et al. [336]).

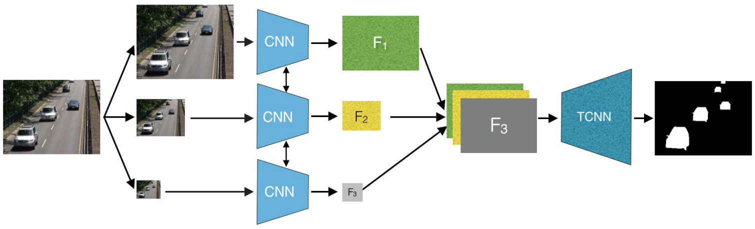


Fig. 21 FgSegNet Architecture (Image from Lim and Keles [193]).

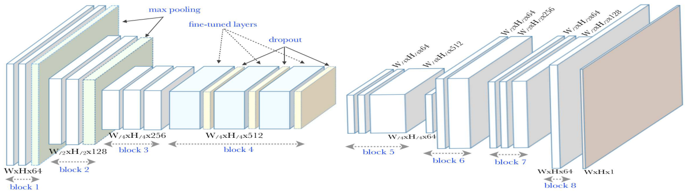


Fig. 22 From left to right: The first image shows the architecture of each CNN in the triplet network. The second image shows the TCNN architecture (Images from Lim et al. [193]

multiscale feature encoder. Furthermore, a decoder network is integrated at the end to map the features to a pixel-level foreground probability map. A threshold is then applied to this map to obtain binary segmentation labels. Figure 4.2 shows the architecture of each CNN in the triplet network. The first four blocks are modified copies of the pre-trained VGG-16 [287]. In addition, the third and fourth max pooling layers were removed and dropouts between each layer of fourth convolutional block were inserted. Figure 4.2 illustrates the TCNN architecture. The output of the encoding network is a concatenated form of the feature maps in three different scales. This map is fed to the TCNN to learn the weights for decoding the feature maps. Finally, the output will be a dense probability mask. Practically, Lim and Keles [193] generated scene specific models using only a few frames (to 50 up to 200) similar to Wang et al. [336]. Experimental results [193] show that TCNN outperforms both ConvNet [30] and cascaded CNN [336], and practically outperformed all the reported methods by an overall F-Measure of 0.9770. In a further study, Lim and Keles [194] designed a variant of FgSegNet-M called FgSegNet-S by adding a Feature Pooling Module (FPM) which operates on top of the final encoder (CNN) layer. In an additional study, Lim et al. [195] proposed an improved architecture called FgSegNet-V2. Figure 4.2 illustrates the FgSegNet-V2 architecture. Lim et al. [195] also provided a modified FPM module with feature fusion. Figure 4.2 shows both the FPM module of the FgSegNet-S and the modified FPM module of FgSegNet-V2. FgSegNet-V2[22] ranks number one on the CDnet 2014 dataset.

These previous methods usually require a large amount of densely labeled video training data. To solve this problem, Liao et al. [191] designed a multi-scale cascaded scene-specific (MCSS) CNN-based background subtraction method with a

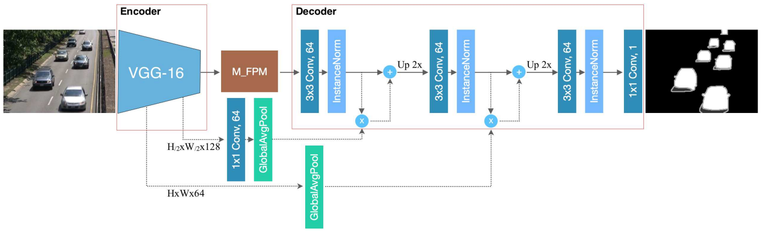


Fig. 23 FgSegNet-V2 Architecture (Image from Lim et al.[195]).

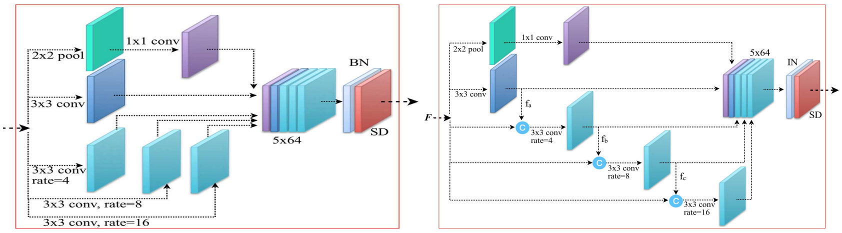


Fig. 24 From left to right: The first image shows the Feature Pooling Module (FPM) with BN (BatchNormalization) and SD (Spatial- Dropout) for FgSegNet-M (Image from Lim and Keles [194]). The second image shows the Modified FPM module (M-FPM) with IN (InstanceNormalization) and SD (Spatial-Dropout). All convolution layers have 64 features (Image from Lim and Keles [195]).

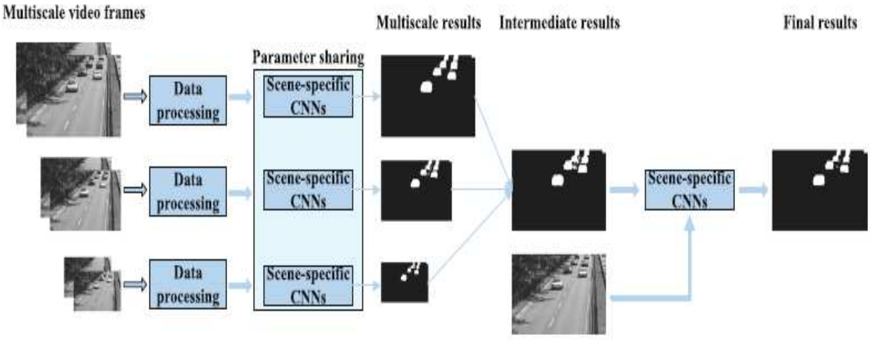


Fig. 25 Pipeline of the Multi-Scale Cascaded Scene-Specific (MCSS) (Image from Liao et al. [191]).

novel training strategy. The architecture combines ConvNets [30] and the multiscale-cascaded architecture [336] using a training that takes advantage of the balance of positive and negative training samples. Figure 4.2 shows the pipeline of Multi-MCSS. Experimental results show that MCSS outperforms Deep CNN [9], TCNN [374] and SFEN [50] with a score of 0.904 on the CDnet 2014 dataset when excluding the PTZ category.

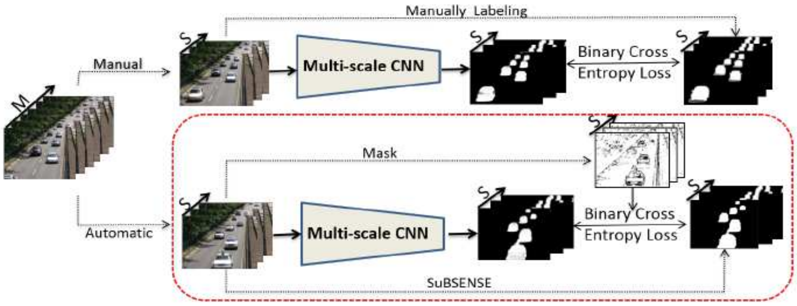


Fig. 26 Top: Pipeline learning for manual labeling in Wang et al. [336]. Bottom: Pipeline for guided automatic learning method in Liang et al. [190] (Image from Liang et al. [190]).

In 2018, Liang et al. [190] developed a multi-scale CNN based background subtraction method by learning a specific CNN model for each video to ensure accuracy, while avoiding manual labeling by using a guided learning scheme. First, Liang et al. [190] applied the SubSENSE algorithm [292] to obtain an initial foreground mask. An adaptive strategy is then applied to select reliable pixels to guide the CNN training because the outputs of SubSENSE cannot be directly used as ground truth owing to a lack of accuracy in the results. A simple strategy was also proposed to automatically select informative frames for the guided learning. Figure 4.2 shows the pipeline for the manual labeling and the pipeline for the guided automatic scheme. Experiments on the CDnet 2014 dataset show that Guided Multi-scale CNN achieves a better F-Measure score of 0.7591 than DeepBS [9] and SuBSENSE [292].

In 2018, Patil et al. [245] proposed a compact multi-scale CNN for deep saliency map in order to detect moving objects. First, the background image is estimated using a temporal histogram based on several input frames in order to generate the saliency map. Second, a compact multi-scale encoder-decoder network is used to learn multi-scale semantic feature of estimated saliency to obtain the foreground masks. Practically, the encoder allows to extract multi-scale features from multi-scale saliency map and the decoder allows to learn the mapping of low resolution multi-scale features into high resolution output frame. Experimental results show that MsEDNet outperforms SuBSENSE [292], DeepBS [9], SFEN [50] with VGG16, SFEN+PSL [50] with VGG16 and SFEN+PSL+CRF [50] with VGG16 on the CDnet 2014 dataset when excluding the four challenging ”LFR”, ”NVD”, ”PTZ”, and ”TBL” categories.

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

[19] http://caffe.berkeleyvision.org/tutorial/solver.html

[20] http://www.vlfeat.org/matconvnet/

[21] https://github.com/lim-anggun/FgSegNet

[22] https://github.com/lim-anggun/FgSegNet-v2

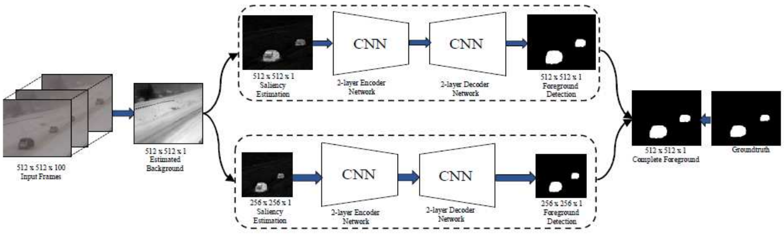


Fig. 27 Pipeline of MsEDnet Network (Image from Patil et al. [245]).

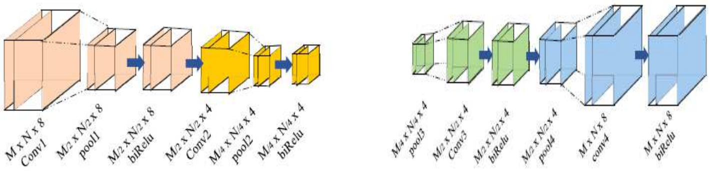


Fig. 28 From left to right: Encoder architecture, decoder architecture for MsEDnet Network (Image from Patil et al. [245]).

4.3 Fully CNNs

Cinelli [59] proposed a similar method to that of Braham and Droogenbroeck [30] by exploring the advantages of Fully Convolutional Neural Networks (FCNNs) [199] to diminish the computational requirements. A FCNN uses a convolutional layer to replace the fully connected layer in traditional convolution networks, which can avoid the disadvantages caused by a fully connection layer. Cinelli tested both the LeNet5 [71] and ResNet [136] architectures. ABecause the ResNet presents a greater degree of hyperparameter setting (namely, the size of the model and even the layer organization) compared to LeNet5, Cinelli also used different features of the ResNet architectures for optimization of the background/foreground separation. To do so, Cinelli used networks designed for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC [23]), which deal with 224 × 224 pixel images, and those for the CIFAR-10 and CIFAR-100 datasets[24], which have 32 × 32 pixel-images as input. The FAIR[25] implementation is employed. From this study, the best models on the CDnet 2014 dataset [335] are the 32-layer CIFAR-derived dilated network and the pre-trained 34-layer ILSVRC-based dilated model adapted through direct substitution. However, Cinelli [59] only provided visual results without an F-measure score.

[23] http://www.image-net.org/challenges/LSVRC/

[24] https://www.cs.toronto.edu/ kriz/cifar.html

[25] https://github.com/facebook/fb.resnet.torch

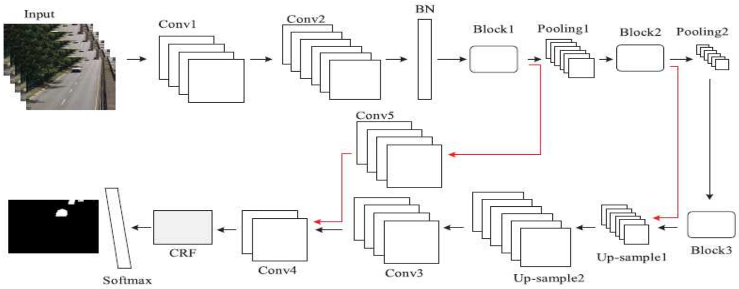


Fig. 29 Fully convolutional network (Image from Yang et al. [355]).

In another study, Yang et al. [355] also used a FCNN but with a structure of shortcut connected block with multiple branches. Each block provides four different branches. Figure 4.3 shows the structure of the FCNN for background modeling. The front of three branches is used to calculate different features by applying a different atrous convolution, and the last branch is the shortcut connection. Figure 4.3 shows the shortcut connected block with multiple branches. For the spatial information, atrous convolution [138] is employed instead of a common convolution to avoid considerable details by expanding the receptive fields. For the activation layers, PReLU Parametric Rectified Linear Unit (PReLU) [137]] was introduced as a learned parameter to transform values of less than zero. Yang et al. [355] also employed a refinement method using Conditional Random Fields (CRF). Experimental results show that this method outperforms traditional background subtraction methods (MOG [294] and Codebook [345]) as well as recent state-of-art methods (ViBe [16], PBAS [144] and P2M [354]) on the CDnet 2012 dataset [112]. But, Yang et al. [355] evaluated their method on a subset of 6 sequences of CDnet 2012 [112] instead of all the categories of CDnet 2014 [335] making a comparison with other DNN methods more difficult to apply.

In 2018, Akilan [3,6,4] designed a Multi-View receptive field Fully CNN (MV-FCN) based on fully convolutional structure, inception modules [300], and residual networking. MV-FCN is based on inception module [299] designed by Google that performs convolution of multiple filters with different scales on the same input to simulate human cognitive processes in perceiving multi-scale information, and ResNet [136] developed by Microsoft that acts as lost feature recovery mechanism. In addition, Akilan [3] exploits intra-domain transfer learning that boosts the correct foreground region prediction. Figure 4.3 shows the MV-FCN architecture. MV-FCN consists of two Complementary Feature Flows (CFF) and a Pivotal Feature Flow (PFF). The PFF is essentially an encoder-decoder CNN whereas CFF1 and CFF2 complement its learning ability. The PFF only employs convolution kernels size of 3×3, whereas CFF1 and CFF2 uses filters size of 5×5 and 9×9 respectively in their first conv layers. Practically, MV-FCN employs inception modules at early and late stages with three different sizes of receptive fields to capture invariance at var-

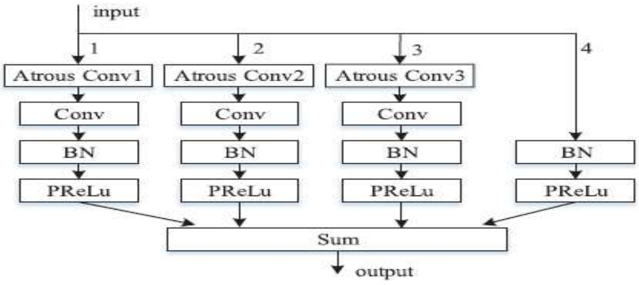


Fig. 30 Structure of shortcut connected block with multiple branches. This block contains four different branches with the same data flow into each branch but different features flows out from each branch because each branch has different layers. From left to right: the front of three branches computes different features by using different atrous convolution whilst the last branch is the shortcut connection. (Image from Yang et al. [355]).

ious scales. The features learned in the encoding phase are fused with appropriate feature maps in the decoding phase through residual connections for achieving enhanced spatial representation. These multi-view receptive fields and residual feature connections provide generalized features for a more accurate pixel-wise foreground region identification. The training is made using the CDnet 2014 [335]. Akilan et al. [3] evaluated MV-FCN against classical neural networks (Stacked Multi-Layer [376], Multi-Layered SOM [105]), and two deep learning approaches (SDAE [369], Deep CNN [9]) on the CDnet 2014 [335] but only on selected sequences making the comparison less complete.

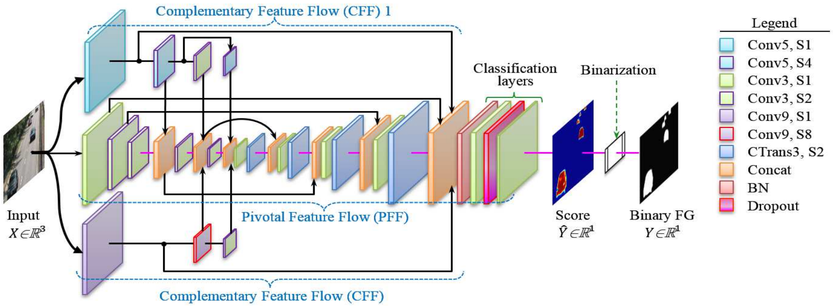


Fig. 31 MV-FCN Architecture: Convk, Si, CTransk, Concat, and BN stand for convolution using kernel size of k and stride of i, transpose convolution with filter size of k, activation maps concatenation, and batch normalization operations, respectively (Image from Akilan [3]).

In 2018, Zeng and Zhu [366] developed a Multiscale Fully Convolutional Network (MFCN) for moving object detection in infrared videos. MFCN does not need to extract the background images. The input is frames from different sequences, and

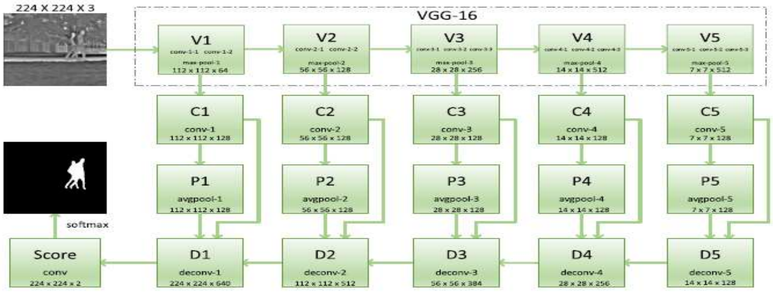


Fig. 32 MFCN Architecture for IR videos: A FCN architecture covering multi-scale convolution and deconvolution operations. As CNN features are learned from multiple scales, the feature representation contains both category-level semantics and fine-grain details. (Image from Zeng and Zhu [364]).

the output is a probability map. Practically, Zeng and Zhu [366] used the VGG-16 as architecture and the inputs have a size of 224×224. The VGG-16 network is split into five blocks with each block containing some convolution and max pooling operations. The The lower blocks have a higher spatial resolution and contain more low-level local features, whereas the deeper blocks contain more high-level global features at a lower resolution. A contrast layer is added behind the output feature layer based on the average pooling operation with a kernel size of 3×3. To exploit multi-scale features from multiple layers, Zeng and Zhu [366] employed a set of deconvolution operations to up-sample the features, creating an output probability map the same size as the input. For the loss function, the cross-entropy is used. The layers from VGG-16 are initialized with pre-trained weights, whereas the other weights are randomly initialized with a truncated normal distribution. The adam optimizer method is used for updating the model parameters. Experimental results on the THM category of CDnet 2014 [335] dataset show that MFCN obtains a score of 0.9870 in this category whereas cascaded CNN [336] obtains 0.8958 and MFCN achieves a score of 0.96 over all the categories. In a further study, Zeng and Zhu [364] provided an improved version of MFCN with contrast layers, which obtains an average measure of 0.9830 on CDnet 2014 [335] dataset. In another study, Zeng and Zhu [365] fused the results produced by different background subtraction algorithms (SuBSENSE [292], FTSG [331], and CwisarDH+ [116]) in order to output a more precise result. This method called CNN-SFC outperforms its direct competitor IUTIS [20] on the CDnet 2014 dataset.

In 2018, Lin et al. [196] designed a deep Fully Convolutional Semantic Network (FCSN) for background subtraction. First, an FCN can learn the global differences between the foreground and the background. Second, SuBSENSE [292] algorithm is employed to generate robust background image with better performance, which is concatenated into the input of the network together with the video frame. Furthermore, Lin et al. [196] initialized the weights of FCSN by partially using pre-trained weights of FCN-VGG16, because these weights are applied to semantic segmentation. Then, FCSN can understand semantic information of images and converge

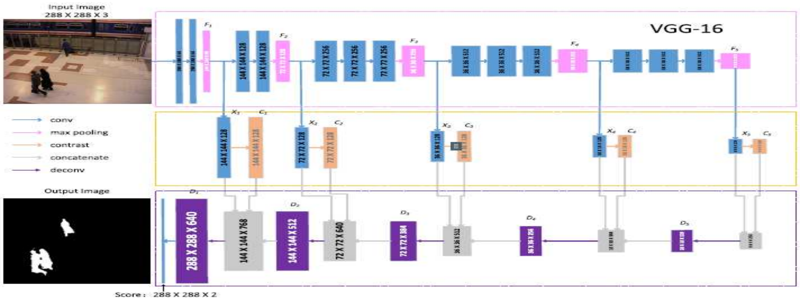


Fig. 33 MFCN Architecture for color videos: Based on VGG, MFCN is divided into five stages by max pooling operations. To effectively use multiscale features, a set of convolution and deconvolution operations with the stepwise upsampling strategy aggregate multiscale features, making a feature representation that contains more category-level information and fine-grain details (Image from Zeng and Zhu [364]).

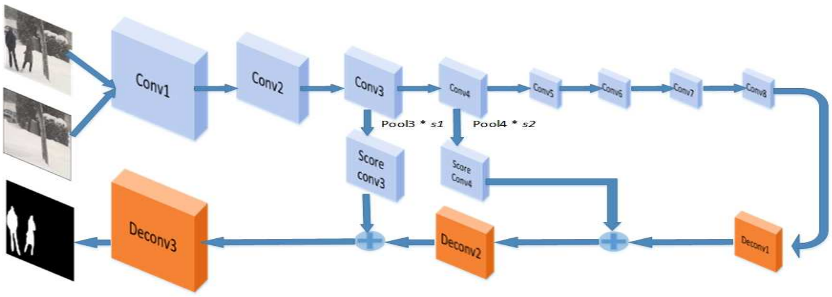


Fig. 34 FCSN Architecture: The Pool3 and Pool4 are the result of max pooling layer and the parameter s1 and the parameter s2 are the scale parameters (Image from Lin et al. [196]).

faster. In addition, FCSN uses less training data and get better result with the help of pre-trained weights. Figure 4.3 show the FCSN architecture. For two input images with a current frame and a background image, corresponding output image with foreground obtained by proposed fully convolutional networks model. FCSN contains 20 convolutional layers and 3 deconvolutional layers. Experimental results show that FCSN outperforms MOG, ViBe, PAWCS and SuBSENSE on several challenging videos of CDnet 2014 dataset.

4.4 Deep CNNs

In 2017, Babaee et al. [9] proposed a deep CNNs based moving objects detection method that contains the following components: an algorithm for background initialization via an average model in RGB, a CNN model for background subtraction, and a post-processing module of the networks output using a spatial median filter. First, Babaee et al. [9] proposed distinguishing the foreground and background pixels using the SuBSENSE algorithm [292], and then only use the background pixel values to obtain the background averaging model. To achieve an adaptive memory length based on the motion of the camera and objects in the video frames, Babaee et al. [9] used Flux Tensor with Split Gaussian Models (FTSG [331]) algorithm. For the network architecture and training, Babaee et al. [9] trained the CNNs with background images obtained by the SuBSENSE algorithm [292]. With images of size 240×320 pixels, the network is trained with pairs of RGB image patches (triplets of size 37×37) from video, background frames and the respective ground truth segmentation patches (CDnet 2014 [335] with around 5% of the data). Thus, instead of training a network for a specific scene, Babaee et al. [9] trained their model all at once by combining training frames from various video sequences including 5% of frames from each video sequence. On the other hand, the same training procedure than ConvNet [30] is employed. Each image-patches are combined with background-patches then fed to the network. The network contains 3 convolutional layers and a 2-layer Multi-Layer Perceptron (MLP). Rectified Linear Unit (ReLU) [235] is used as activation function after each convolutional layer and the sigmoid function after the last fully connected layer. In addition, batch normalization layers are used before each activation layer to decrease over-fitting and to also provide higher learning rates for training. Finally, a spatial-median filtering is applied in the post-processing step. This method provided foreground mask more precise than ConvNet [30] and not very prone to outliers in presence of dynamic backgrounds. Finally, deep CNN based background subtraction outperforms the existing algorithms when the challenge does not lie in the background modeling maintenance. Deep CNN obtained an F-Measure score of 0.7548 in CDnet2014 dataset [335]. The limitations of Deep CNN are as follows: 1) It cannot handle the camouflage regions well within foreground objects, 2) It provides a poor performance on PTZ video sequences, and 3) owing to the corruption of the background images, it performs poorly in presence of large changes in the background.

In a further study, Zhao et al. [374] proposed an end-to-end two-stage deep CNN (TS-CNN) framework. Figure 4.4 shows the pipeline of TS-CNN. The current frame is the input of the network to reconstruct the background. The reconstructed background image is then concentrated to the current frame and fed into the following fully convolutional network to obtain the foreground mask. More precisely, a convolutional encoder-decoder sub-network is used to reconstruct the background images and encode rich prior knowledge of the background scenes, whereas the reconstructed background and current frame are the inputs into a multi-channel fully convolutional sub-network for accurate foreground detection in the second stage. In the two-stage CNN, the reconstruction and segmentation losses are jointly optimized. The encoder contains a set of convolutions, and represents the input image as a latent feature vector. The decoder restores the background image from the feature vector. The l_2 loss was employed as the reconstruction loss. After training, the encoder-decoder network separates the background from the input image and restores a clean background image. The second network can learn semantic knowledge of the foreground and background. Therefore, it could handle various challenges such as the nighttime lighting, shadows and camouflaged foreground objects. Experimental results [374] show that the TS-CNN outperforms SuBSENSE [292], PAWCS [293], FTSG [331] and Shared-Model [49] in the case of night videos, camera jitter, shadows, thermal imagery and

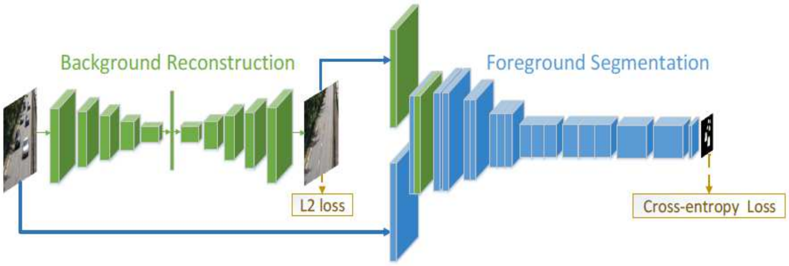


Fig. 35 Pipeline of TS-CNN (Image from Zhao et al. [374]).

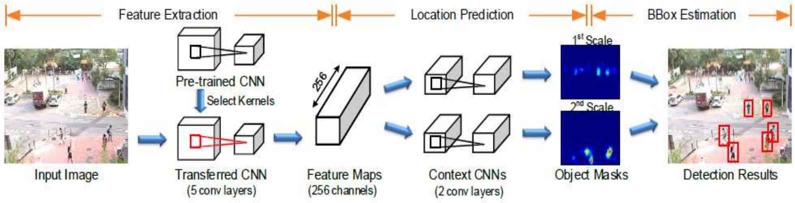


Fig. 36 Pipeline of ADCNN (Image from Li et al. [186]).

bad weather. In CDnet2014 dataset [335], TS-CNN and Joint TS-CNN obtained an F-Measure score of 0.7870 and 0.8124, respectively.

In 2017, Li et al. [186] designed an adaptive deep CNN (ADCNN) to predict object locations in a surveillance scene. Figure 4.4 illustrates the pipeline of ADCNN. First, the current image is the input into the transferred CNN, which outputs 256 feature maps. The 256 feature maps are then forward propagated using several context CNNs. Thus, an equal number of object masks at their corresponding scales are generated. Finally, the detection results are obtained by merging the bounding boxes, which are estimated on object masks. More precisely, a generic CNN-based classifier is transferred to the surveillance scene by selecting useful kernels. The context information of the surveillance scene is then learned using the regression model for an accurate location prediction. Although they focus on object detection and thus do not use the principle of background subtraction, ADCNNs have achieved very interesting performance on several surveillance datasets for pedestrian detection and vehicle detection. Furthermore, Li et al. [186] provided results with the CUHK square dataset [326], the MIT traffic dataset [334] and the PETS 2007[26] instead of the CDnet2014 dataset [335].

In 2017, Chen et al. [50] proposed the detection of moving objects using an end-to-end deep sequence learning architecture with the pixel-level Semantic Features

[26] http://www.cvg.reading.ac.uk/pets2007/data.html

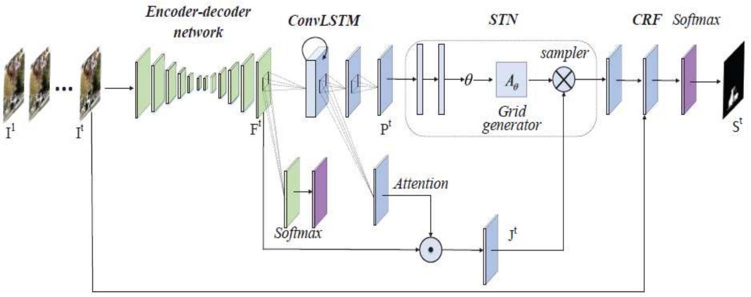


Fig. 37 Pipeline of SFEN (Image from Chen et al. [50]).

(SFEN). Figure 4.4 shows the pipeline of SFEN. Video sequences are the input into a deep convolutional encoder-decoder network to extract pixel-level Semantic Features (SFEN). Practically, Chen et al. [50] used the VGG-16 [287] as encoder-decoder network, although other architectures, such as GoogLeNet [300], ResNet50 [136] can also be used in this framework. An attention long short-term memory model called Attention ConvLSTM is used to integrate pixel-wise changes over time. A Spatial Transformer Network (STN) model and a Conditional Random Fields (CRF) layer are then employed to reduce the sensitivity to camera motion and to smooth the foreground boundaries, respectively. Experimental results [50] on the two large-scale dataset CDnet 2014 dataset [335] and LASIESTA [68] indidate that the proposed method obtained similar results as Convnet [30] with a better performance for the category ”Night videos”, ”Camera jitter”, ”Shadow” and ”Turbulence”. Attention ConvLSTM obtained an F-Measure score of 0.8292 with VGG-16, 0.7360 with GoogLeNet and 0.8772 with ResNet50 as can be seen in Table 12.

In 2018, Patil and Murala [244] designed a compact end-to-end convolutional neural network architecture called motion saliency foreground network (MSFgNet) in order to estimate the background and to extract the foreground from video frames. Figure 4.4 shows the pipeline of MSFgNet. First, a long video is divided into a number of small video streams (SVS) that are the input of MSFgNet which estimates the background frame for each SVS. Second, the saliency map is obtained using the estimated background and the current frame. In addition, a compact encoderdecoder network extracts the foreground from the estimated saliency maps. In practice, MSFgNet consists of two main networks: 1) a Motion-saliency network (MSNet) composed of a Background Estimation Network (BENet) and Saliency Estimation Network (SMNet), and 2) a Foreground extraction network (FgNet). Figure 4.4 shows the MSFgNet architecture. However, MSFgNet handles approximately 168 and 87 times less parameters compared to cascaded CNN [336] and SFEN [50], respectively. MSFgNet also obtains better performance compared to cascaded CNN [336] and SFEN [50] in terms of the average F-measure score on the CDnet 2014 dataset.

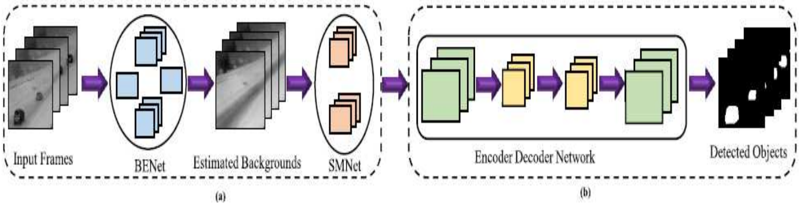


Fig. 38 Pipeline of MSFgNet (Image from Patil and Murala [244]).

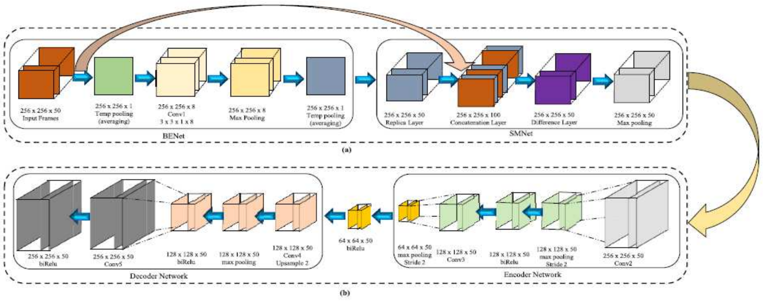


Fig. 39 MSFgNet Architecture (Image from Patil and Murala [244]).

4.5 Structured CNNs

In 2017, Lim et al. [192] developed an encoder-encoder structured CNN (Struct-CNN) for background subtraction. Thus, the background subtraction model involves the following components: a background image extraction via a temporal median in RGB, network training, background subtraction and foreground extraction based on super-pixel information. Figure 4.5 illustrates the structure of Struct-CNN. The structure is thus similar to the VGG16 network [287] after excluding the fully connected layers. The encoder converts the 3 (RGB) channel input (images of size 336×336 pixels) into 512-channel feature vector through convolutional and max-pooling layers yielding a $21 \times 21 \times 512$ feature vector. Then, the decoder converts the feature vector into a 1-channel image of size 336×336 pixels providing the foreground mask through deconvolutional and unpooling layers. Lim et al.[192] trained this encoder-decoder structured network in the end-to-end manner using CDnet 2014 [335]. For the architecture, the decoder consists of six deconvolutional layers and 4 unpooling layers. In all deconvolutional layers, except for the last one, features are batch-normalized and the Parametric Rectified Linear Unit (PReLU) [138] is employed as an activation function. The last deconvolutional layer which is the prediction layer used the sigmoid activation function to normalize outputs and then to provide the foreground mask. 5×5 kernels are used in all convolutional while a 3×3 kernel is employed in the prediction layer. In order to suppress the incorrect boundaries and holes in the foreground mask, Lim et al. [192] used the superpixel information

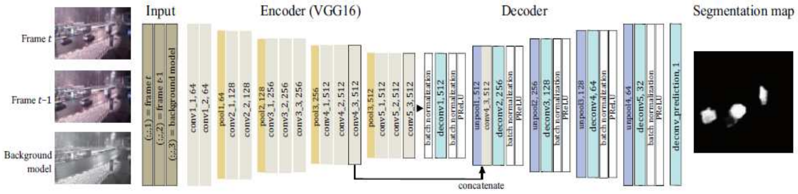


Fig. 40 Struct-CNN Architecture: Three grayscale images are used as inputs. The encoder is based on the VGG16. The decoder extracts a foreground mask using the features from the encoder (Image from Lim et al. [192]).

obtained by an edge detector. Experimental results [192] show that Struct-CNN outperforms SuBSENSE [292], PAWCS [293], FTSG [331] and SharedModel [49] in the case of bad weather, camera jitter, low frame rate, intermittent object motion and thermal imagery. Struct-CNN obtained an F-Measure score of 0.8645 on the CDnet 2014 dataset [335] excluding the ”PTZ” category. Lim et al. [192] excluded this category, arguing that they focused only on static cameras.

Le and Pham [181] also proposed an encoder-decoder structured CNN for background subtraction. In the encoder, features of both the target frame and background frame are extracted and then subtracted to obtain the foreground mask. Le and Pham [181] also combined features of target frame passed from the low-lever block CNN through skip connection to enhance the representation of changing description. Next, the decoder part estimates the change map with finest resolution. Experimental results provided only on several challenging videos of the CDnet 2014 dataset, show that EDS-CNN outperforms bothe SubSENSE [292] and DeepBS [9].

4.6 Double Encoding-Slow Decoding CNNs

In 2018, Akilan and Wu [5] proposed a strategy called Double Encoding-Slow Decoding (DESD) to improve a basic encoder-decoder CNN. This method has also been called sEnDec by Akilan [4], and by Akilan and Wu [7]. The DESD EnDec CNN consists of two sub-networks, as shown in Figure 4.6, namely, encoding and decoding networks. Both networks exploit structured residual feature fusions. Instead of ConvNets [30], DeepBS [9], FCNN [355] and Struct-CNN [192], this architecture does not use any pooling or hidden FC layers, but subsumes conv, transpose convolution (convT), and cat layers, which are interconnected to capture spatio-temporal contextual cues of moving objects. An input feature map applied at the sub-sampling stage is encoded twice before reaching to the next level of reduced spatial dimension. This process works as a micro auto-encoder. In the up-sampling sub-network, each spatial dimension of decoded feature maps is improved using two sets of residual feature cat operations interspersed with a BN, thereby fusing two individual encoded feature maps from the sub-sampling stages.

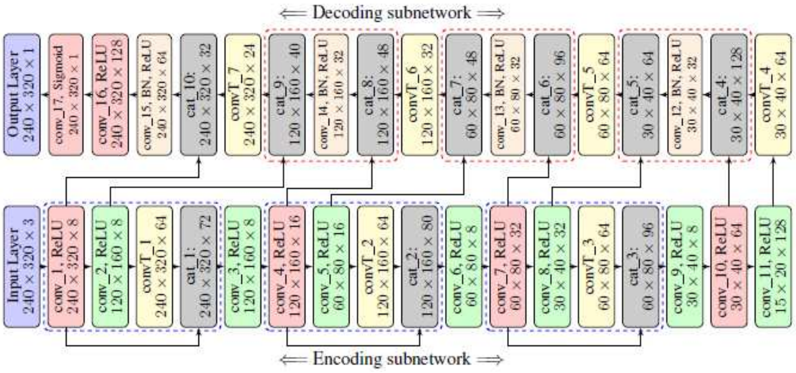

⟸ Decoding subnetwork ⟹
Output Layer 240 × 320 × 1
conv_17, Sigmoid 240 × 320 × 1
conv_16, ReLU 240 × 320 × 128
conv_15, BN, ReLU 240 × 320 × 64
cat_10: 240 × 320 × 32
convT_7 240 × 320 × 24
cat_9: 120 × 160 × 40
conv_14, BN, ReLU 120 × 160 × 32
cat_8: 120 × 160 × 48
convT_6 120 × 160 × 32
cat_7: 60 × 80 × 48
conv_13, BN, ReLU 60 × 80 × 32
cat_6: 60 × 80 × 96
convT_5 60 × 80 × 64
cat_5: 30 × 40 × 64
conv_12, BN, ReLU 30 × 40 × 32
cat_4: 30 × 40 × 128
convT_4 30 × 40 × 64
Input Layer 240 × 320 × 3
conv_1, ReLU 240 × 320 × 8
conv_2, ReLU 120 × 160 × 8
convT_1 240 × 320 × 64
cat_1: 240 × 320 × 72
conv_3, ReLU 120 × 160 × 8
conv_4, ReLU 120 × 160 × 16
conv_5, ReLU 60 × 80 × 16
convT_2 120 × 160 × 64
cat_2: 120 × 160 × 80
conv_6, ReLU 60 × 80 × 8
conv_7, ReLU 60 × 80 × 32
conv_8, ReLU 30 × 40 × 32
convT_3 60 × 80 × 64
cat_3: 60 × 80 × 96
conv_9, ReLU 30 × 40 × 8
conv_10, ReLU 30 × 40 × 64
conv_11, ReLU 15 × 20 × 128
⟸ Encoding subnetwork ⟹

Fig. 41 DESD's Architecture: Layer diagram of Double Encoding-Slow Decoding EnDec CNN (Image from Akilan and Wu [5]).

4.7 3D-CNNs

In 2017, Sakkos et al. [270] designed an end-to-end 3D-CNNs to track temporal changes in video sequences avoiding the use of a background model for the training. Here, 3D-CNNs can handle multiple scenes without further fine-tuning on each scene individually. Figure 4.7 illustrates the 3D-CNNs architecture. More precisely, Sakkos et al. [270] used C3D branch [311]. The input employs a video of ten frames connected to the first group of layers (CRP-1) in groups of four frames with stride 2. CRP-1 is then connected to CRP-2 in the same manner and CRP-3 has access to the features of all frames. CRP-4 is performing 2D operations only, while CR has no pooling layer. The upsampling layers (US-1, US-2, US-3 and US-4) are connected to CRP-2, CRP-3, CRP-4 and CR, respectively. Then, they are concatenated before applying the final convolution. Experimental results [270] reveal that 3D-CNN provides a better performance than ConvNet [30] and deep CNN [9]. Furthermore, experiments on the ESI dataset [320], which presents extreme and sudden changes in illumination, show that 3D-CNN outperforms two designed illumination invariant background subtraction methods that are Universal Multimode Background Subtraction (UMBS) [268] and ESI [320]. 3D-CNNs obtained an average F-Measure score of 0.9507 in CDnet 2014 dataset. In 2018, Gao et al. [100] also employed 3D-CNNs for background subtraction. Figure 4.7 shows the comparison between a 2D convolution operation and a 3D convolution operation demonstrating the advantage of a 3D convolution for the background subtraction task. Figure 4.7 illustrates the 3D CNNs architecture. Practically, Gao et al. [100] only provided experimental results on several sequences of the CDnet 2012 dataset, making it more difficult to compare their algorithm than had the results been provided on the CDnet 2014 dataset.

In 2018, Yu et al. [359] employed a spatial-temporal attention-based 3D ConvNets to jointly model the appearance and motion of objects-of-interest in a video for a Relevant Motion Event detection Network (ReMotENet). Figure 4.7 shows the

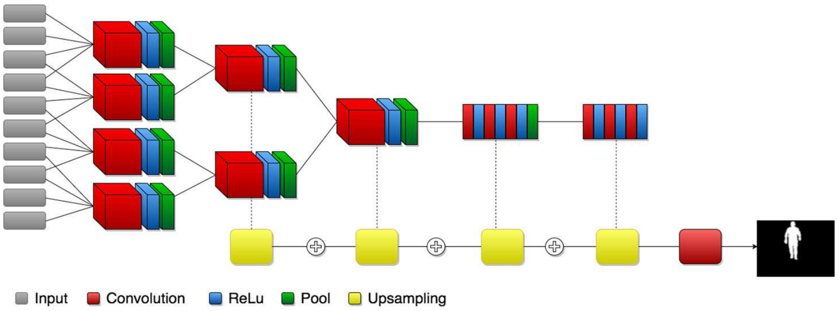


Fig. 42 3D-CNNs Architecture. Cubes indicate 3D operations across the temporal dimension. Rectangles indicate 2D (spatial only) operations. The plus sign indicates concatenation (Image from Sakkos et al. [270].

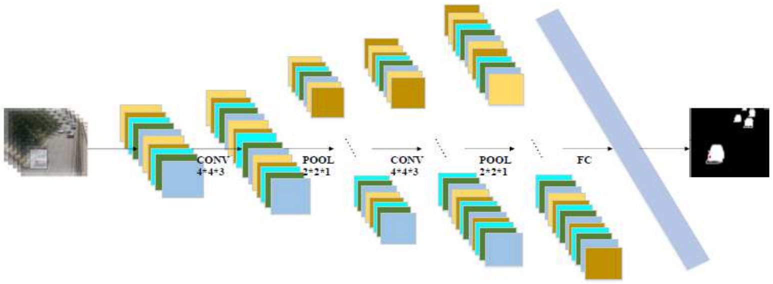


Fig. 43 3D CNNs Architecture: Two convolution layers, two pooling layers, one full connection layer and one output layer (Image from Gao et al. [100].

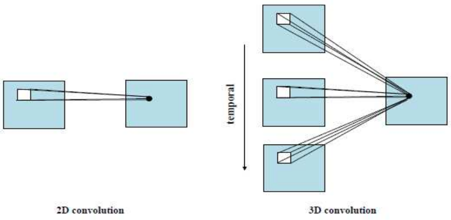


Fig. 44 Comparison between a 2D convolution operation and a 3D convolution operation (Image from Gao et al. [100].

ReMotENet architecture. The input is a 4D representation of a video and the outputs are binary predictions of relevant motion involving different moving objects. The architecture is based on the C3D branch [311]. However, instead of using max pooling both spatially and temporally, Yu et al. [359] separated the spatial and temporal max pooling to capture fine-grained temporal information, and deepen the network to learn better representations. Experimental results demonstrate that ReMotENet achieves a comparable or even better performance, and is three- to four-orders of magnitude

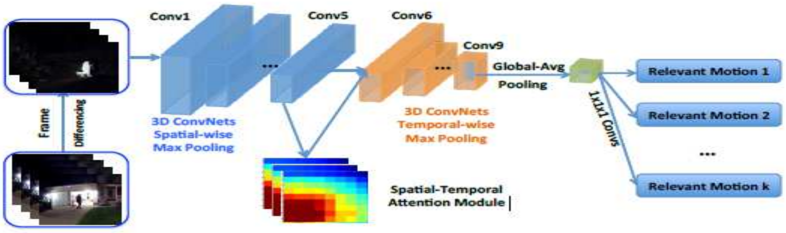


Fig. 45 3D CNNs Architecture: The low-level 3D ConvNets only keeps spatial features with spatial-wise max pooling. The high-level 3D ConvNets keeps temporal features using temporal-wise max pooling. Spatial-temporal mask is multiplied with the extracted features from Conv5 before it is fed as the input to Conv6 (Image from Yu et al. [359].

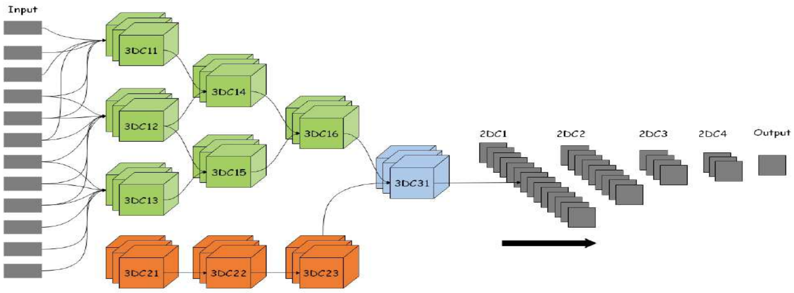


Fig. 46 3D Atrous CNN Architecture (10 layers): Layer 1 is the input layer. Two parallel structures in layers 2, 3, 4 to gain different temporal information. Their outputs are concatenated in 3DC31 in layer 5. 2D atrous convolution is used to the remaining layers 6, 7, 8, 9 to suppress the time dimension and perform foreground detection. Layer 10 is the output layer (Image from Hu et al. [148].

faster than the object detection based method. It can detect relevant motion in a 15s video in $4-8$ milliseconds on a GPU and a fraction of second on a CPU with model size of less than 1 MB.

In another study, Hu et al. [148] developed a 3D atrous CNN model to learn deep spatial-temporal features without losing resolution information. Figure 4.7 illustrates of 3D atrous convolution demonstrating its interest for the background subtraction task. Figure 4.7 shows the architecture of the 3D atrous CNN model, whereas Figure 4.7 shows how the 3D atrous ConvLSTM network at time steps $t-1$, t and $t+1$. More precisely, this model is combined with two convolutional long short-term memory (ConvLSTM) networks in order to capture both short- and long-term spatiotemporal information of the input video data. Furthermore, 3D Atrous ConvLSTM is a completely end-to-end framework that does not require any pre- or post-processing of the data. Experiments on CDnet 204 dataset show that 3D atrous CNN outperforms SuBSENSE [293], cascaded CNN [336] and DeepBS [9].

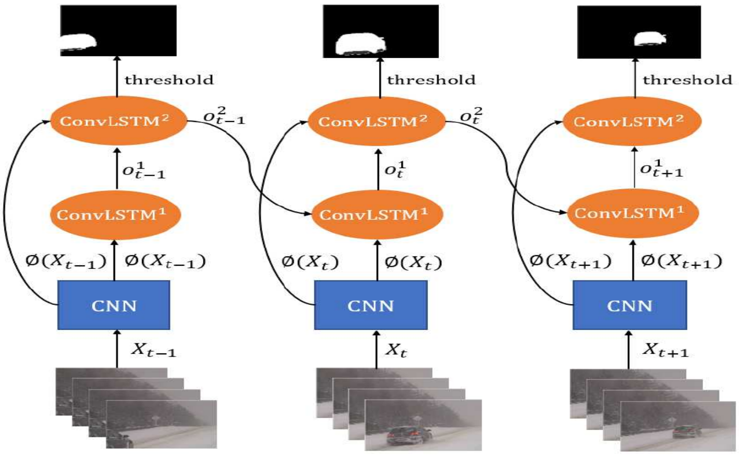


Fig. 47 2-level 3D atrous ConvLSTM network at time steps $t-1$, t and $t+1$. The input of ConvLSTM1 at time step t consists of the output of the feature extractor CNN and the output of ConvLSTM2 for time step $t-1$. The input of ConvLSTM2 at time step t consists of the output of our feature extractor CNN and the output of ConvLSTM1. The input consists of 12 frames (Image from Hu et al. [148].

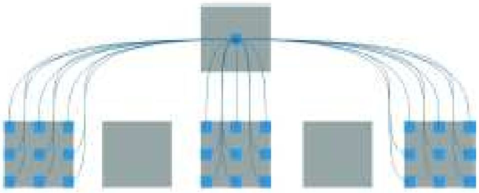


Fig. 48 3D atrous convolution with kernel size (3,3,3) and rate (2,2).(Image from Hu et al. [148].

In 2018, Wang et al. [337] proposed a multi-scale 3D Fully CNN (MFC3D) architecture in order to learn multi-scale features in both spatial and temporal domains. The MFC3D uses an encoder-decoder structure. Figure 4.7 shows the architecture of MFC3D. The input of the network is a video with 16 consecutive frames, including the current frame and 15 previous frames. The encoder extracts multiscale spatial-temporal features, namely, two spatial scale and two temporal scale features from the input sequences, whereas the decoder merges the features to reconstruct the pixel-wise detection result, which is the probability of each pixel belong to the foreground. The probability is then thresholded to obtain the foreground mask. Therefore, the network establishes a mapping from a video sequence to the pixel-wise classification results. Experiments on CDnet 204 dataset show that MFC3D obtains better a F-Measure score than cascaded CNN [336] and DeepBS [9] over all categories. MFC3D reaches an average F-measure score 0.9619 whereas FC3D (MFC3D without multi-scale process) obtains a score of 0.9524.

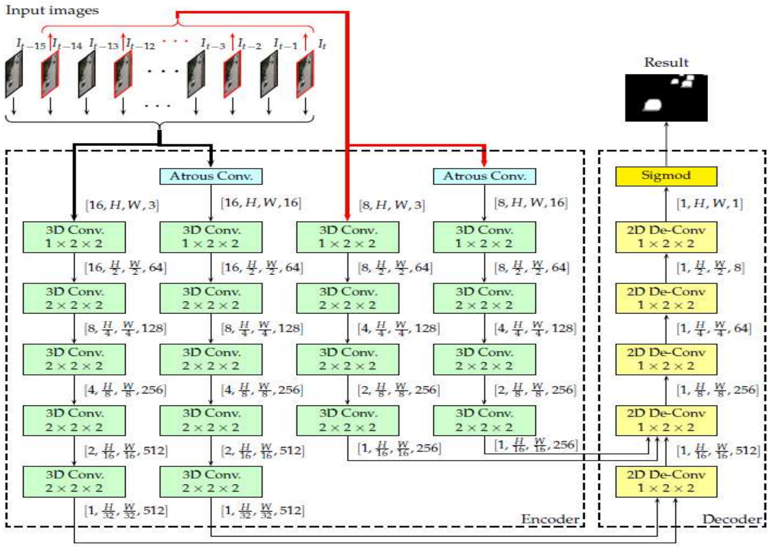
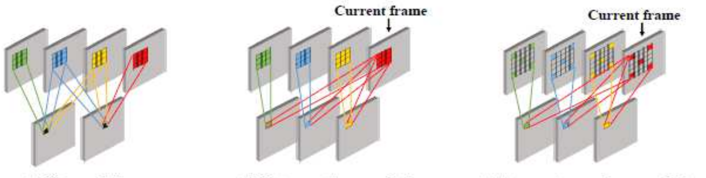

Input images
I_{t-15} I_{t-14} I_{t-13} I_{t-12} I_{t-3} I_{t-2} I_{t-1} I_t
Result
Atrous Conv.
Atrous Conv.
Sigmod
[16, H, W, 3]
[16, H, W, 16]
[8, H, W, 3]
[8, H, W, 16]
[1, H, W, 1]
3D Conv. 1 × 2 × 2
2D De-Conv 1 × 2 × 2
[16, $\frac{H}{2}$, $\frac{W}{2}$, 64]
[8, $\frac{H}{2}$, $\frac{W}{2}$, 64]
[1, $\frac{H}{2}$, $\frac{W}{2}$, 8]
3D Conv. 2 × 2 × 2
[8, $\frac{H}{4}$, $\frac{W}{4}$, 128]
[4, $\frac{H}{4}$, $\frac{W}{4}$, 128]
[1, $\frac{H}{4}$, $\frac{W}{4}$, 64]
[4, $\frac{H}{8}$, $\frac{W}{8}$, 256]
[2, $\frac{H}{8}$, $\frac{W}{8}$, 256]
[1, $\frac{H}{8}$, $\frac{W}{8}$, 256]
[2, $\frac{H}{16}$, $\frac{W}{16}$, 512]
[1, $\frac{H}{16}$, $\frac{W}{16}$, 256]
[1, $\frac{H}{16}$, $\frac{W}{16}$, 512]
[1, $\frac{H}{32}$, $\frac{W}{32}$, 512]
Encoder
Decoder

Fig. 49 MFC3D Architecture: The downsampling rate or the upsampling rate are indicated for each layer. The dimensions of the tensors are shown beside corresponding arrows (Image from Wang et al. [337].

4.8 Retrospective Convolutions

Chen et al. [46] proposed the use of retrospective convolutions to avoid the temporal limitation of 3D CNNs. Retrospective convolution directly links the current frame to any previous frame and detects instantaneous changes. Figure 4.8 illustrates the comparison between 3D convolution, retrospective convolution and atrous retrospective convolution. The 3D convolution kernel of works on three consecutive frames, and a frame can not be linked directly to another one with more than 2-frame interval. A retrospective convolution kernel of spatial size relate the current frame to each of all preceding frames. An atrous retrospective convolution kernel with dilation expands the FoV from 3 × 3 to 5 × 5. An Atrous Retrospective Pyramid Pooling (ARPP) module is further employed to enhance retrospective convolution with multi-scale field-of-views. Figure 4.8 shows the architecture based on ResNet-18, ARPP and multi-level encoder-decoder modules. To address the problem of foreground-specific overfitting in learning-based methods, Chen et al. [46] employed a data augmentation method called static sample synthesis which guides the network to focus on learning change-cued information rather than specific spatial features of foreground. Finally, an end-to-end framework allows to fuse change features of different scales and realizes pixel-wise prediction. Experimental results provided on several challenging videos of the CDnet 2014 dataset show that ResNet-18 + ARPP outperforms MOG [294], ViBe [16] and SuBSENSE [293].



(a) 3D convolution (b) Retrospective convolution (c) Atrous retrospective convolution

Fig. 50 Comparison between 3D convolution, retrospective convolution and atrous retrospective convolution. (Image from Chen et al. [46].

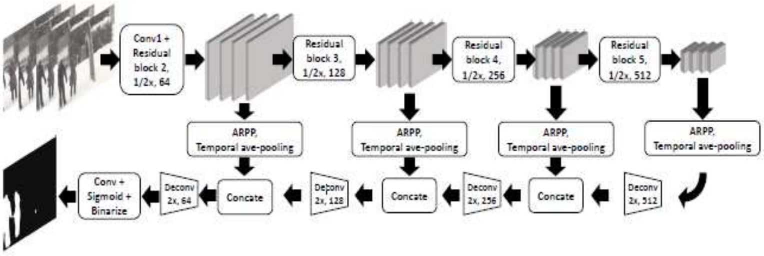

Conv1 + Residual block 2, 1/2x, 64
Residual block 3, 1/2x, 128
Residual block 4, 1/2x, 256
Residual block 5, 1/2x, 512
ARPP, Temporal ave-pooling
ARPP, Temporal ave-pooling
ARPP, Temporal ave-pooling
ARPP, Temporal ave-pooling
Conv + Sigmoid + Binarize
Deconv 2x, 64
Concate
Deconv 2x, 128
Concate
Deconv 2x, 256
Concate
Deconv 2x, 512

Fig. 51 Atrous Retrospective Architecture based on ResNet-18, ARPP and multi-level encoder-decoder modules (Image from Chen et al. [46].

4.9 CNNs with Different Input Features

4.9.1 Random Permutation of Temporal Pixels (RPoTP) feature

Zhao et al. [371] designed a Deep Pixel Distribution Learning (DPDL) model for background subtraction. For the input of the CNNs, Zhao et al. [371] employed Random Permutation of Temporal Pixels (RPoTP) features instead of using the intensity values, as in the previous methods. Figure 4.9.1 illustrates the RPoTP features used to represent the distribution of past observations for a particular pixel, in which the temporal correlation between observations is deliberately no ordered over time. The RPoTP features from all pixels are fed into the convolutional neural network to learn a classifier to achieve background subtraction. A convolutional neural network (CNN) is then used to learn the distribution and thereby determine whether the current observation is foreground or background. The random permutation allows the framework to focus primarily on the distribution of observations, rather than be disturbed by spurious temporal correlations. For a large number of RPoTP features, the pixel representation is captured even with a small number of ground-truth frames. Figure 4.9.1 shows the architecture of DPDL. Experiments on the CDnet 2014 dataset show that DPDL is effective even with only a single ground-truth frame giving similar performance than the MOG model in this case. With 20 GTs, DPDL obtains similar scores

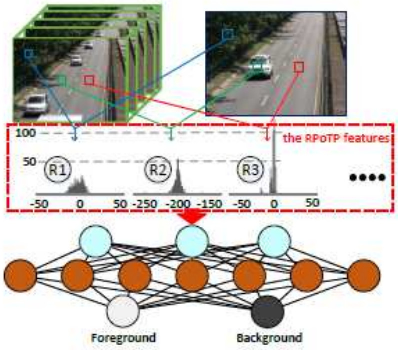


Fig. 52 RPoTP features encode the distributions of pixel observations that belong to dynamical background R1, moving objects R2 and static background R3 respectively.(Image from Zhao et al. [371].

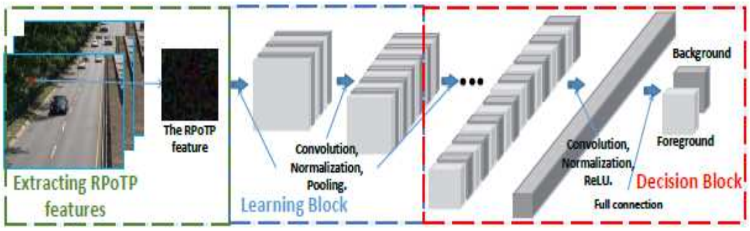

The RPoTP feature
Extracting RPoTP features
Convolution, Normalization, Pooling.
Learning Block
Background
Foreground
Convolution, Normalization, ReLU.
Decision Block
Full connection

Fig. 53 Deep Pixel Distribution Learning (DPDL) Architecture (Image from Zhao et al. [371].

as SubSENSE [293]. Finally, DPDL[27] with 40 GTs achieves an average F-Measure score of 0.8106, outperforming DeepBS [9].

4.9.2 Depth feature

Wang et al. [333] proposed the use of a BackGround Subtraction neural Networks for Depth videos (BGSNet-D) to detect moving objects in scenes in which the color information cannot be obtained. Thus, BGSNet-D is suitable for dark scenes, where the color information is difficult to obtain. CNNs can extract features in color images, but cannot be applied to depth images directly because edge noises occur and there is an absence of pixels in the captured data. To address this problem, Wang et al. [333] designed an extended min-max normalization method to pre-process the depth images. After pre-processing, the two inputs of the CNNs are the average background image in depth and the current image. The architecture is therefore similar to that of ConvNets with three convolutional layers. In each convolutional layer, a filter with 3×3 local receptive fields and a 1×1 stride is used. ReLU follows as the activation function in hidden layers. The batch normalization layer and pooling layer are both applied after each ReLU layer. Finally, all feature maps are employed as inputs of an

[27] hhttps://github.com/zhaochenqiu/DPDL

MLP, which contains three fully connected layers. A sigmoid is used as an activation function, and the output only consists of a single unit. Experiments on the SBM-RGBD[28] dataset [37]show that BGSNet-D outperforms existing methods that use only the depth data, and even reaches a level of performance similar to those methods that use RGB-D data.

4.10 Generative Adversarial Networks

In 2018, Bakkay et al. [15] proposed a background subtraction method based on conditional Generative Adversarial Network (cGAN). Figure 4.10 shows the pipeline of this model, called BScGAN, which consists of two successive networks: generator and discriminator networks. The generator learns the mapping from the background and the current image for the foreground mask. The discriminator then learns a loss function to train this mapping by comparing the ground truth and predicted output by observing the input image and background. For the architecture, the generator network follows the encoder-decoder architecture of Unet network with skip connections [151]. The encoder part includes down-sampling layers that decrease the size of the feature maps followed by convolutional filters. It consists of eight convolutional layers. The first layer uses a 7×7 convolution to provide 64 feature maps. The 8th layer generates 512 feature maps with a 1×1 size. Their weights are randomly initialized. In addition, the six middle convolutional layers are ResNet blocks. In all encoder layers, leaky-ReLU non-linearities are used. The decoder part uses up-sampling layers followed by deconvolutional filters to construct an output image with the same resolution as the input image. Its architecture is similar to that of the encoder, including eight deconvolutional layers, but with reverse layer ordering and down-sampling layers being replaced by up-sampling layers. For the discriminator network, the architecture is composed of four convolutional and down-sampling layers. The first layer generates 64 feature maps. Moreover, the fourth layer generates 512 feature maps with a 30×30 size. The convolutions are 3×3 spatial filters and their corresponding weights are randomly initialized. Leaky ReLU functions are employed as activation functions. Experimental results on CDnet 2014 datasets shows that BScGAN outperforms ConvNets [30], cascaded CNN [336], and Deep CNN [9] with an average F-Measure score of 0.9763 when excluding the ”PTZ” category.

In 2018, Zheng et al. [379] employed a Bayesian GAN (BGAN) approach. First, a median filter algorithm is used to extract the background, and a network based on a BGAN is then trained to classify each pixel, thereby dealing with the challenges of sudden and slow illumination changes, a non-stationary background, and ghosting. Deep CNNs are adopted to construct the generator and discriminator of a BGAN. In a further study, Zheng et al. [380] proposed a parallel version of the BGAN algorithm called (BPVGAN).

In 2018, Bahri et al. [13] designed an end-to-end framework called Neural Unsupervised Moving Object Detection (NUMOD), which is based on a batch method

[28] http://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html

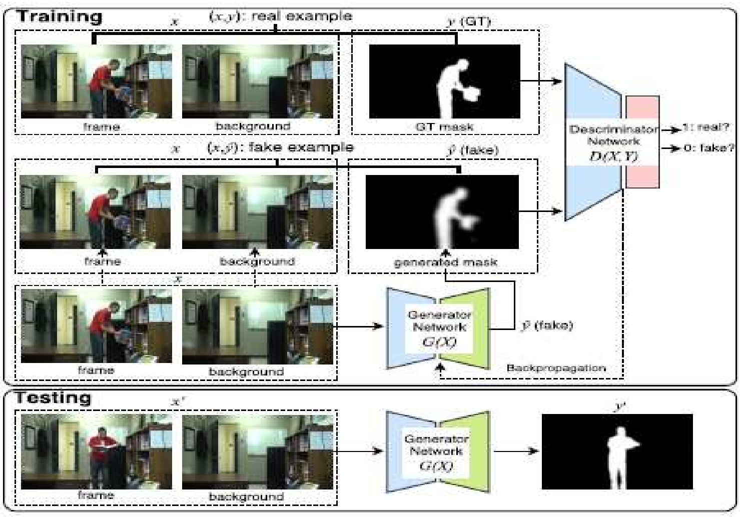


Fig. 54 Pipeline of BScGAN (Image from Bakkay et al. [15]).

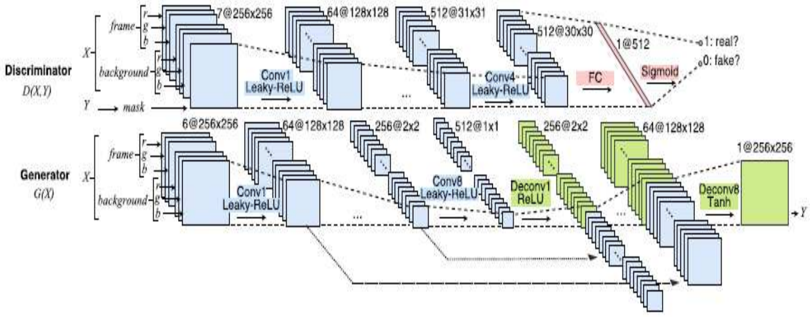


Fig. 55 cGAN Architecture (Image from Bakkay et al. [15].

named ILISD [281]. NUMOD can work in either online or batch mode thanks to the parametrization through a generative neural network. NUMOD decomposes each frame into three parts: changes in the background, foreground, and illumination. It uses a fully connected generative neural network to generate a background model by finding a low-dimensional manifold for the background of the image sequence. For the architecture, NUMOD uses two generative fully connected networks (GFCNs). Net1 estimates the background image from the input image, whereas Net2 generates a background image from an illumination-invariant image. These two networks have the exact same architecture. Thus, the input to the GFCN is an optimizable low-dimensional latent vector. Then, two fully connected hidden layers are followed by

ReLU non-linearity. The second hidden layer is fully connected to the output layer, which is followed by the sigmoid function. A loss term is employed to impose the output of the GFCN to be similar to the current input frame. A GFCN is similar to the decoder part of an auto-encoder. In an auto-encoder, the low-dimensional latent code is learned by the encoder, whereas in a GFCN, it is a free parameter that can be optimized and input into the network. During training, this latent vector learns a low-dimensional manifold of the input distribution.

4.11 Encoder-Decoder Networks

CNNs can difficulty deal with temporal events in video sequences that have long-term dependencies. In particular, a dense pixel-wise prediction is a hard problem for CNNs owing to the huge memory and large numbers of parameters needed to learn the temporal correlation. To address this problem, Choo et al. [56] designed in 2018 a Multi-Scale Recurrent encoder-decoder Neural Network (MSRNN), which compresses the spatio-temporal features at the encoder and restores them to the original sized results at the decoder. Figure 4.11 shows the architecture which has recurrent layers both in the encoder and decoder at each scale level. The recurrent layers are convolutional LSTM, which maintain the shapes of features. These multi-scale LSTM layers stacked with the convolutional layers enable the network to learn the temporal information from the consecutive frames and produce the dense predictions. More precisely, Choo et al. [56] employed a convolutional long short-term memory (LSTM) into the encoder-decoder architecture. MSRNN successfully learns the spatio-temporal relation with a small number of parameters compared to CNNs. MSRN is trained with limited duration of video frames, and shows robustness against different challenges under different time duration. MSRNN outperforms IUTIS-5 [20] and STSOM [82] on CDnet 2014 dataset. In addition, Choo et al. [56] studied the influence of recurrent layers through ablation showing that the performance of the architecture is then reduced as can be seen in Table 12. In a further study, Choo et al. [55] proposed an unsupervised version of MSRNN. Figure 4.11 shows the corresponding structure which is divided into two branches. The recurrent branch learns the spatiotemporal information by stacking the convolutional LSTM in the form of multi-scale encoder-decoder. The semantic branch extracts visual information from each frames. The tensors of the two branches are piled with the original resolution of the image. Then, pixels are classified as background or foreground according to the softmax value. Binary labels are then created through the augmentation. Because it is not possible to synthesize semantic and optical flow labels with unlabeled training phase video, the semantic branch is also trained for background subtraction. In 2019, Farnoosh et al. [94] designed a Deep Probabilistic Background Model (DeepPBM) based on Variational autoencoders (VAEs) [78, 173]. DeepPBM is a generative modeling of the background allowing to compute backgrounds of a specific scene in presence of illumination changes and variations in the background. However, DeepPBM is based on two main hypotheses. First, the background lies on a low-dimensional subspace represented by a series of latent variables. Second, the latent subspace of the background embedded by a non-linear mapping of the video frames fit a Gaussian

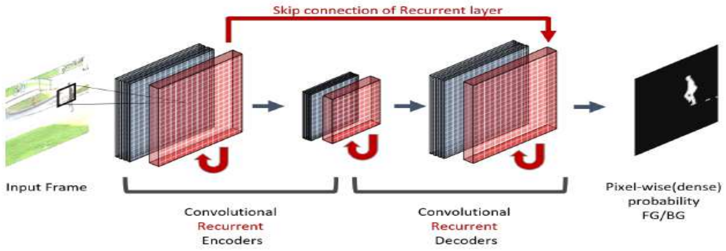


Fig. 56 Multi-Scale Recurrent encoder-decoder Neural Network Architecture (Image from Choo et al.[56]).

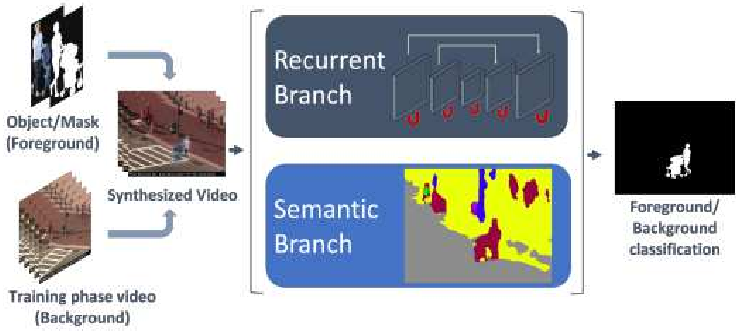


Fig. 57 Pipeline of the unsupervised version of MSRNN (Image from Choo et al. [55]).

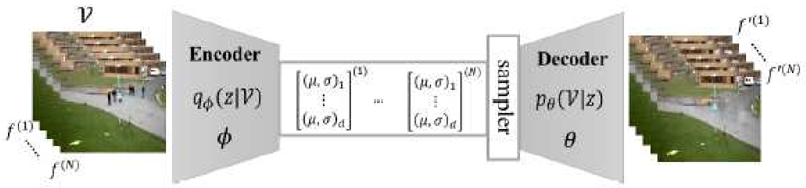

$\mathcal{V}$
$f^{(1)}$
$f^{(N)}$
Encoder
$q_\phi(z|\mathcal{V})$
ϕ
$\begin{bmatrix}(\mu,\sigma)_1\\ \vdots \\ (\mu,\sigma)_d\end{bmatrix}^{(1)}$
...
$\begin{bmatrix}(\mu,\sigma)_1\\ \vdots \\ (\mu,\sigma)_d\end{bmatrix}^{(N)}$
sampler
Decoder
$p_\theta(\mathcal{V}|z)$
θ
$f'^{(1)}$
$f'^{(N)}$

Fig. 58 DeepPBM Pipeline based on Variational autoencoders (VAEs) (Image from Farnoosh et al. [94]).

distribution model. Figure 4.11 illustrated that the encoder learns an efficient representation of the input video and projects that into a stochastic lower dimensional space determined by latent variables. The decoder attempts to recover the original data, given the probabilistic latent variables from the encoder. The entire network is trained by comparing the original input data with its reconstructed output. For long-term videos, experimental results show that DeepPBM outperforms RPCA [38] on the BMC 2012 dataset [313].

Categories	Methods	Authors - Dates
Convolutional Neural Networks	CNN features	Dou et al. [80] (2018)
Deep Auto-encoders Networks	Stacked Denoising AutoEncoders (SDAE) Stacked Denoising AutoEncoders (SDAE)	Zhang et al. [369] (2015) Garcia-Gonzalez et al. [103] (2018)
Neural Reponse Mixture	NeREM Real-Time NeREM	Shafiee et al. [277] (2016) Shafiee et al. [278] (2017)
Motion Feature Networks	MF-Net Factored 3-Way RBM	Nguyen et al. [238] (2018) Lee and Kim [183] (2018)

Table 5 Deep Neural Networks for Deep Learned Features: An Overview

5 Deep Learned Features

The features used play an important role in the robustness against the challenge met in a video sequence[27]. Historically, low-level and hand-craft features such as color [200,280], edge [67,172], texture [140,284], motion [111,227], and depth [217,35, 36,97,154,232] features have often been employed to deal with illumination changes, dynamic background, and camouflage. However, an operator needs to be chosen[10, 11,53] to fuse the results derived from the different features or a feature selection scheme [285,286]. Nevertheless, none of these approaches can finally compete with approaches based on deep learned features.

5.1 Convolutional Neural Networks

Dou et al. [80] proposed employing CNN features to deal with challenges met in video surveillance. First, given a cleaned background image without moving objects, Dou et al. [80] constructed adjustable neighborhood of each pixel in the background image to form windows. The CNN features are then extracted with a pre-trained CNN model for each window to obtain a features based background model. Second, Dou et al. [80] extracted features for the current frame with the same operation as the background model. After, a distance map between the background image and the current frame is constructed by using the Euclidean distance. Third, the distance map is fed into graph cut algorithm to obtain the foreground mask. The background model is also updated with a learning rate. Figure 5.1 illustrates the architecture with 8 layers conv-net model. A 224 by 224 crop of an image in RGB is the input which is convolved with 96 different 1st layer filters (red), each of size 7 × 7 employing a stride of 2 in both x and y. The resulting feature maps are then passed through a ReLu (, pooled, and contrast normalized across feature maps to give 96 different 55 × 55 element feature maps. Similar operations are repeated in layers 2-5. The last two layers are fully connected. The final layer is a c-way soft-max function with c being the number of classes. Experimental results on the Wallflower dataset [310] show that the proposed method outperforms MOG [294] and LBP [140].

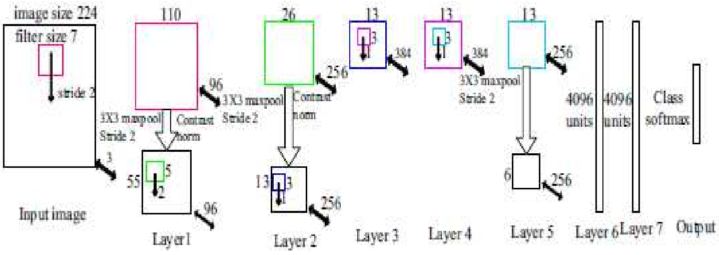


Fig. 59 Deep CNN's features (Image from Dou et al. [80]).

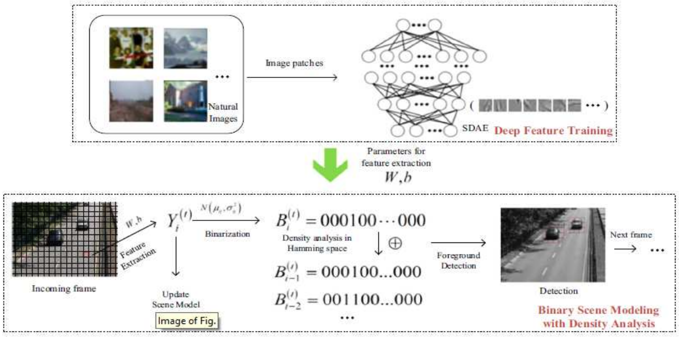


Fig. 60 Deep Feature Learning and Binary Background Modeling (Image from Zhang et al. [369]).

5.2 Stacked Denoising AutoEncoders

Zhang et al. [369] designed a deep learned features based block-wise method with a binary spatio-temporal background model. Figure 5.2 shows the corresponding pipeline that consists of two parts: Stacked Denoising AutoEncoder (SDAE) learning binary background modeling. Based on SDAE, the deep learning module learns a deep image representation encoding the intrinsic scene information. This leads to the robustness of feature description. Figure 5.2 illustrates the SDAE network. The binary background model captures the spatio-temporal scene distribution information in the Hamming space to perform foreground detection. Experimental results [369] on the CDnet 2012 dataset [112] demonstrate that SDAE provides a better performance than traditional methods, namely, MOG [294], KDE [85], and LBP [140], and therecent state-of-art model PBAS [144]. To address the robustness against stationary noise, Garcia-Gonzalez et al. [103] also used a stacked denoising autoencoders to generate a set of robust features for each patch of the image. This set is then considered as the input of a probabilistic model to determine whether that region is part of the background or foreground.

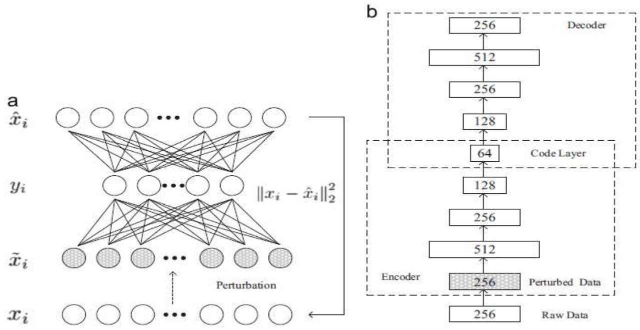


Fig. 61 SDAE Architecture: (a) Denoising Autoencoder. (b) Four Stacked Denoising Autoencoder with the input patch of size 16×16 (Image from Zhang et al. [369]).

5.3 Neural Reponse Mixture

Shafiee et al. [277,278] proposed a Neural Reponse Mixture (NeRM) framework to extract rich deep learned features with which to build a reliable MOG background model. Figure 5.3 shows the motion detection based on the NeRM framework. The first synaptic layer of StochasticNet [279] is trained on the ImageNet dataset [76] as a primitive, low-level, feature representation. Thus, the neural responses of the first synaptic layer at all pixels in the frame is then used as a feature to distinguish motion caused by objects moving in the scene. It is worth noting that the formation of StochasticNets used in the NeRM framework is a one-time and off-line procedure which is not implemented on an embedded system. The final formed StochasticNet is transferred to the embedded system. Then, MOG model is employed using the deep learned features. Experimental results [277] on the CDnet 2012 dataset [112] show that MOG-NeRM globally outperforms both the MOG model with RGB features and Color based Histogram model called CHist [48], but does not achieve the best scores for the ”intermittentObjectMotion”', ”Low frame rate”, ”Night video”, and ”Thermal” categories.

5.4 Motion Feature Networks

Nguyen et al. [238] combined a sample-based background model with a feature extractor obtained by training a triplet network. This network is constructed by three identical CNNs, each of which is called a Motion Feature Network (MF-Net). Thus, each motion patterns is learned from small image patches and each input images of any size is transformed into feature embeddings for high-level representations. A sample based background model is then used with the color feature and the extracted deep motion features. To classify whether a pixel is background or foreground, Nguyen et al. [238] employed the l_1 distance. Furthermore, an adaptive feedback scheme is also employed. The training is made with the CDNet 2014 dataset [335]

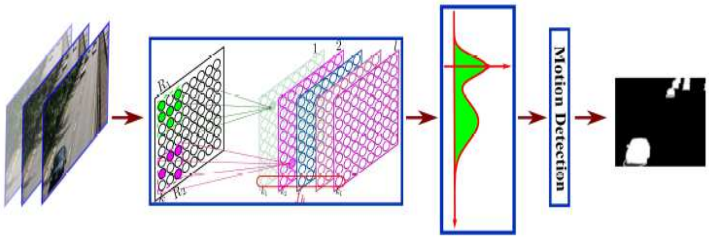


Fig. 62 (NeRM Architecture: The neural responses from a highly efficient StochasticNet are used as rich deep features that are used in the MOG model (Image from Shafiee et al. [277]).

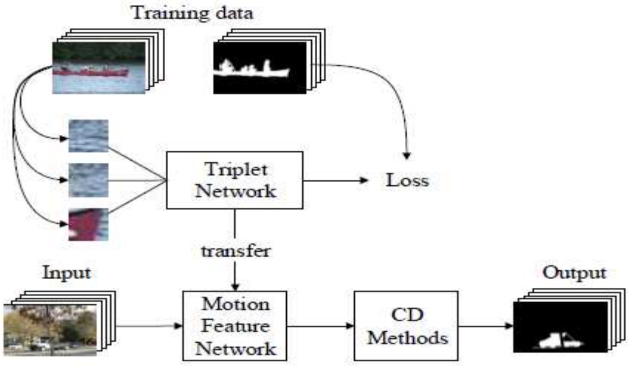


Fig. 63 (Block diagram of MF-Net. The triplet network are trained with a dataset. The trained CNN is then split and modified to work as a feature extractor (Image from Nguyen et al. [238]).

and the offline trained network is then used on the fly without re-training on any video sequence before each execution. Experimental results [238] on BMC 2012 dataset and CDNet 2014 dataset [335] show that MF-Net outperforms SOBS, LOBSTER and SuBSENSE in the case of dynamic backgrounds. Lee and Kim [183] proposed a method for learning the pattern of the motions using the Factored 3-Way Restricted Boltzmann Machines (RBM) [257] and obtaining the global motion from the sequential images. Once this global motion is identified between frames, background subtraction is achieved by selecting the regions that do not respect the global motion. These regions are thus considered as the foreground region

6 Adequacy for the background subtraction task

All the previous works demonstrated the performance of DNN for background subtraction but not discuss the reason why DNN works well. A first way to analyze these performance is to compare these different methods. For this, we have grouped in Table 3 a comparative overview of the architectures while we show an overview in terms of the challenges in Table 4. From Table 3, we can see that it is possible to have three type of input: current image only, background and current images. In the first case, the authors works either with the current images without computing a background image or with a end-to-end solution that first generates a background image. In the second case, the authors have to compute the background image by using the temporal median or another model like SuBSENSE. The output is always the foreground mask except for NUMOD which provide the background and the foreground mask but also an illumination change mask. For the architecture, most of the authors employed a well-know architecture (LeNet-5, VGG-16 and U-Net) that they slighly adapted to the task of background subtraction. Only few authors proposed a full designed architecture for background subtraction. Table 4 groups the solutions of the different methods for the limitations of ConvNets [30]. To learn the process at different level, the most common solutions are multi-scale and cascaded strategies alleviating the drawback to work with patches. For the training, over-fitting is often the case producing scene-specific methods. For the dataset used for the training, most of the authors employed the CDnet 2014 dataset with a part devoted to the training phase and another part for the testing phase. End-to-end solutions are well proposed as well as spatial and temporal strategies. Most of the time, the architecture is a generative one even if a combination of generative and discriminative would be better suitable for background subtraction. Indeed, the background modeling is more a reconstructive task while the foreground detection is more a discriminative task.

To analyze how and why the DNN works well for this application, Minematsu et al. [225, 226] provided a valuable analysis by testing a quasi-similar method than ConvNet [30] and found that the first layer performs the role of background subtraction using several filters whilst the last layer categorizes some background changes into a group without supervised signals. Thus, DNN automatically discovers background features through feature extraction by background subtraction and the integration of the features [225] showing its potential for background/foreground separation. This first analysis is very valuable but the adequacy of a DNN method for the application of background/foreground separation should also be investigated in other key issues, that are the challenges and requirements met in background subtraction, and the adequacy of the architecture for background subtraction. More experimentally, Karadag and Erdas [170] observed that deep learning approaches detect changes in presence of static backgrounds successfully but they are more sensitive in the case of dynamic backgrounds and camera jitter although they provide better performance than conventional approaches. In 2018, Akilan et al. [8] studied the gap of performance between traditional models (i.e. statistical models and conventional ANNs) and two deep neural networks models that achieve about 9% and 7% improvements in terms of F-Measure.

To be effective, a background/foreground separation method should address the following challenges and requirements met in this application: (1) its robustness to noise, (2) its spatial and temporal coherence, (3) the existence of an incremental version, (4) the existence of a real-time implementation, and (5) the ability to deal with the challenges met in video sequences. Issue (1) is ensured for deep learning methods because a DNN learns the deep features of the background and foreground during the training phase. For issue (2), spatial and temporal processing need to be added to pixel-wise DNN methods because, as explained in Akilan [3], one of the main challenges in DNN methods is dealing with objects of very different scales and the dithering effect at bordering pixels of foreground objects. In literature, several authors have added spatial and temporal constraints using several spatial and/or temporal strategies. These strategies can be either incorporated in an end-to-end solution or can be done via a post-processing applied to the foreground mask. For example, cascaded CNN [336] and MV-FCN [3] employed a multi-scale strategy while DeepBS [9] used a spatial median filter. Struct-CNN [192] is based on a superpixel strategy whilst Attention ConvLSTM+CRF [192] with Conditional Random Field (CRF). In another manner, Sakkos et al. [270] used directly 3D-CNN for temporal coherence while Chen et al. [50] used a spatial and temporal processing in Attention ConvLSTM. For issue (3), there is no need to update the background model in the DNN method if the training is sufficiently large to learn all appearances of the model in terms of changes in illumination and dynamics (waving trees, water rippling, waves, etc.), but is required otherwise. In this last case, several authors employed an end-to-end solution in which a DNN method is used for background generation to determine the background image over time. The output of this DNN-based background generation is then the input of the DNN-based background subtraction with the current image to determine the foreground mask. For issue (4), DNNs are time consuming when not applying a specific GPU and optimizer. Thus, the key point in achieving a suitable DNN method for background subtraction is to have a large training dataset and additional spatial/temporal strategies, and to apply them using a specific graphics card if possible. For issue (5), which regards the challenges met in video sequences, such as changes in illumination and dynamic backgrounds, a DNN alone may be sufficient if the architecture allows learning these changes, as applied in several studies, or if additional networks can be added.

For the adequacy of the architecture, it is necessary to check the features of the DNNs, namely, (1) type of architecture, and (2) parameters such as number of neurons, number of layers, etc. In the literature, we can only find two works comparing different architectures for background/foreground separation: Cinelli [30] tested both LeNet5 [71] and ResNet [136] architectures while Chen et al. [50] compared the VGG-16 [287], the GoogLeNet [300], and the ResNet50 [136]. In these two works, ResNet [136] provided the best results. However, these architectures were first designed for different classification tasks using the ImageNet dataset [176], CIFAR-10 dataset or ILSVRC 2015 dataset, , but not for background/foreground separation using a corresponding dataset such as the CDnet 2014 dataset.

7 Experimental Results for Background Generation

For comparison, we analyzed the results obtained by different algorithms on the well-known publicly available SBMnet dataset [169] in a quantitative manner. Practically, only FCFlowNet [132] was fully evaluated on this dataset. Looking at SBMnet dataset, the top algorithm is MSCL [160] based on RPCA decomposition followed by a super-pixel approach [362] and the LabGen's group algorithms [178–180]. The rank of FCFlowNet is only 19. However, FCFlowNet is also outperformed by conventional neural networks approaches like BEWiS [115], SC-SOBS-C4 [216], and BE-AAPSA [255]. This counter performance can be explained by the fact that deep learning is difficult in presence of several challenges like very short sequences, and thus can not outperform methods with specific designed strategies using optical flow for example.

8 Experimental Results for Background Subtraction

For comparison, we present the results obtained on the well-known publicly available CDnet 2014 dataset [335] both in a qualitative and quantitative manner.

8.1 CDnet 2014 dataset and Challenges

CDnet 2014 dataset [335] was developed as part of Change Detection Workshop challenge (CDW 2014). This dataset includes all the videos from the CDnet 2012 dataset [112] plus 22 additional camera-captured videos providing 5 different categories that incorporate challenges that were not addressed in the 2012 dataset. The categories are as follows: baseline, dynamic backgrounds, camera jitter, shadows, intermittent object motion, thermal, challenging Weather, low frame-rate, night videos, PTZ and turbulence. In addition, whereas ground truths for all frames were made publicly available for the CDnet 2012 dataset for testing and evaluation, in the CDnet 2014, ground truths of only the first half of every video in the 5 new categories is made publicly available for testing. The evaluation will, however, be across all frames for all the videos (both new and old) as in CDnet 2012. All challenges of these different categories have different spatial and temporal properties. It is important to determine both the solved and unsolved challenges. Both the CDnet 2012 and CDnet 2014 datasets allow highlighting under which situations it is difficult to provide robust foreground detection for existing background subtraction methods. The following remarks can be made regarding the development described in [168]:

- Conventional background subtraction methods can efficiently deal with challenges met in ”baseline” and ”bad weather” sequences.
- The ”Dynamic backgrounds”, ”thermal video” and ”camera jitter” categories are a reachable challenge for top-performing background subtraction.
- The ”Night videos”, ”low frame-rate”, and ”PTZ” video sequences represent significant challenges.

8.2 Performance Evaluation

8.2.1 Qualitative Evaluation

A) Comparison setup

We compared the visual results obtained on the CDnet 2014 dataset by the different deep learning algorithms with visual results of other representative background subtraction algorithms that are:

- Two statistical models, namely, the well-known MOG [294] and RMOG [314]. The Mixture of K Gaussians (MOG) was introduced in 1999 by Stauffer and Grimson [294] to model dynamic backgrounds. Each pixel is thus characterized by a mixture of K Gaussians. Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized. The parameters of the MOG's model are the number of Gaussians K, the weight $\omega_{i,t}$ associated to the i^{th} Gaussian at time t, the mean $\mu_{i,t}$ and the covariance matrix $\Sigma_{i,t}$. K determines the multi-modality of the background and by the available memory and computational power and it is commonly set from 3 to 7 [294]. This model can handle better dynamic backgrounds than the mean, median, or single Gaussian model owing to its multi-modality. In 2013, Varadarajan et al. [314] improved the MOG by taking into account the spatial relationship between pixels. Thus, regions are modeled as mixture distributions rather than as individual pixels.
- One multi-cues model called Self-Balanced SENsitivity SEgmenter (SubSENSE) [292] proposed in 2014 by St-Charles et al. [292]. SubSENSE is a sample-based method that allows building a background model rather than building a model based on a specific distribution. SubSENSE is also non-parametric. Its primary goal is to address the issue of dynamic background modeling while increasing the foreground detection sensitivity through awareness of spatio-temporal variations, and decreasing the sensitivity to illumination variations. SuBSENSE offers a very effective feedback scheme that is able to identify static and dynamic background regions, adjust the model parameters to promote sample matching, and increase the overall foreground detection accuracy. It works at the pixel level, leading to better segmentation results in complex heterogeneous scenes. Because it is based on a sample consensus modeling approach, it still holds a significant memory footprint, while offering a fast processing speed. However, it does not handle intermittently moving foreground objects particularly well owing to the memoryless nature of its model, and to the random nature of its updating rules.
- Two conventional neural networks, namely, SC-SOBS [211] and AAPSA [255]. SC-SOBS [211] is an extension of SOBS that uses the spatial coherence and takes into account uncertainty in the background model. The SC-SOBS algorithm outperforms the crisp SOBS for moving object detection [210] and parked vehicles detection [209]. In the auto-adaptive parallel SOM architecture (AAPSA), a suspicious foreground analysis is conducted by continuously monitoring the segmentation results and thereby obtaining a reduction of the false positive rates.

Deep learning models include the following: five CNNs based methods (cascaded CNN [336], DeepBS [9], FgSegNet [193], FgSegNet-SFPM [194], FgSegNet-V2 [195]) and two GANs based methods (BSPVGAN [380], DCP [296]). All visual results come from the CDnet 2014 website except for DCP, for which the authors kindly provided the results. We also let in the four figures the number ID as well as the name as it is provided in the CDnet 2014 website.

B) Qualitative Analysis

Figure 6 shows the visual results obtained using MOG, RMOG, and SuBSENSE. We can see that SuBSENSE clearly improves the foreground mask by reducing false positives and negative detections. From Figure 7, we can remark that cascaded CNN outperforms the classical neural networks SC-SOBS and AAPSA except in the ”Low-frame Rate” and ”Night Videos” categories. In Figure 8, FgSegNet and FgSegNet-SFPM (that are top methods in CDnet 2014 dataset) visually outperforms DeepBS in the ”Baseline” and ”Thermal”' Categories. In Figure 9, we can remark that Semantic BGS [31] obtains similar visual results than semi-supervised MSRNN [56] and worse than unsupervised MSRNN [56]. In Figure 10, FgSegNet-V2 which is the top method in CDnet 2014 dataset is compared with GAN based methods that give similar visual results. Finally, we can state that the foreground mask was progressively improved over time when using statistical models, multi-cue models, conventional neural networks, and deep learning models in order of quality.

8.2.2 Quantitative Evaluation

A) Comparison setup

We compared the F-measures obtained using the different algorithms with the F-measures of other representative background subtraction algorithms over a complete evaluation dataset, namely, **(A)** two conventional statistical models (MOG [294], RMOG [314], **(B)** three advanced non-parametric models (SubSENSE [292], PAWCS [293], and Spectral-360 [275]), and **(C)** two conventional neural networks models (SOBS-CF [210], SC-SOBS [211]). Deep learning models for background separation are classified based on their applied architecture:

- **Convolutional Neural Networks:** We grouped the scores of 22 algorithms based on a CNN, namely, two basic CNN algorithms (two variants of ConvNet [30]), seven multi-scale or/and cascaded CNN algorithms (cascaded CNN [336], FgSegNet-M [193], FgSegNet-S [194], FgSegNet-V2 [195], MCSS [191], Guided Multi-scale CNN [190], and MsEDNet [245]), 1 fully CNN algorithms (MFCN [366]), seven deep CNN algorithms (DeepBS [9], TS-CNN [374], Joint TS-CNN [374], five variants of Attention ConvLSTM [50]), one structured CNN algorithm (Struct-CNN [192]), and four 3D CNN algorithms (3D CNN [270], 3D Atrous CNN [148], FC3D [337], MFC3D [337]).
- **Generative Adversarial Networks:** We grouped scores of four GAN algorithms, namely, DCP [296], BScGAN [15], BGAN [379], and BPVGAN [380].

Furthermore, these algorithms can be labeled as pixel-wise, spatial-wise, temporal wise, and spatio-temporal-wise algorithms. For pixel-wise algorithms, they were directly applied by the authors to background/foreground separation without specific processing by considering the spatial and temporal constraints. With these algorithms, each pixel is processed independently based or not on the information contained in their local patch, such as in ConvNet [30]. Thus, they may produce isolated false positives or false negatives. For spatial-wise algorithms, these algorithms model the dependencies among adjacent spatial pixels and thus enforce spatial coherence, as in cascaded CNN [336] and MFCN [366] with a multi-scale strategy, Deep CNN (DeepBS) [9] with spatial median filtering, Struct-CNN [192] with

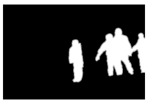
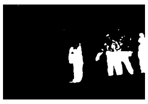
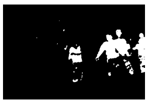
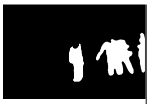
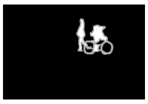
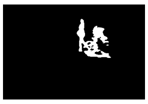
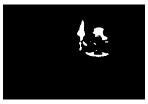
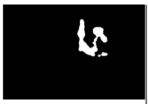
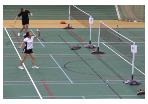
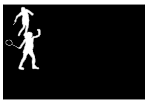
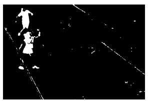
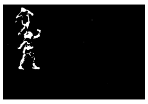
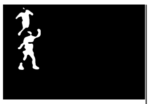
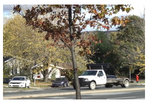
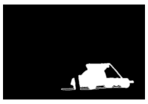
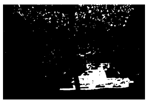
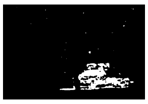
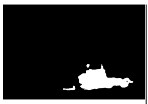
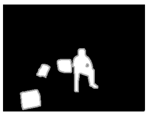
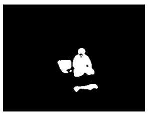
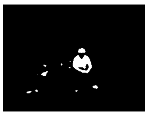
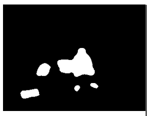
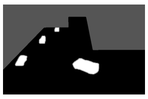
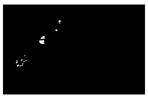
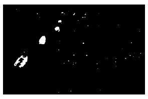
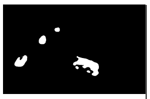
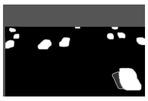
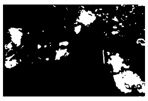
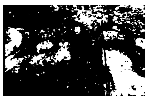
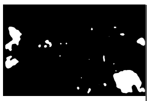
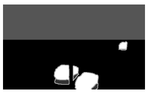
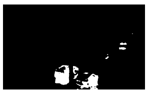
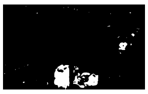
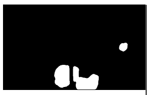
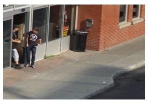
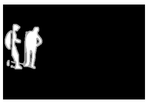
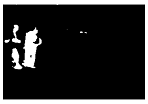
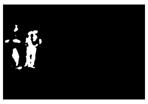
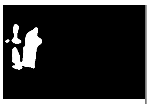
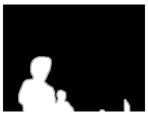
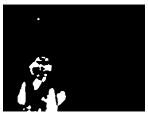
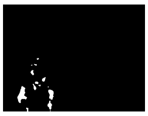
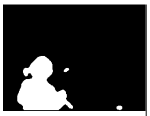
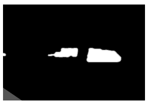
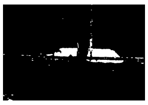
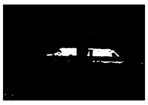
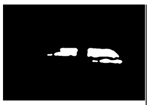
Categories	Original	Ground Truth	4-MOGStauffer	16-MOGMiller	14-SuBSENSE
B-Weather Skating (in002349)					
Baseline Pedestrians (in000490)					
C-Jitter Badminton (in001123)					
Dynamic-B Fall (in002416)					
I-O-Motion Sofa (in001314)					
Low-F TunnelExit (in002781)					
NightVideos F-Highway (in000450)					
PTZ TwoPosition (in001216)					
Shadow BusStation (in000394)					
Thermal D-Room (in002656)					
Turbulence T-3 (in000999)					

Table 6 Visual results on CDnet 2014 dataset (Part 1): From left to right: Original images, Ground-Truth images, MOG (4-MOG-Stauffer) [294], RMOG (16-MOGMiller) [314], SubSENSE [292].

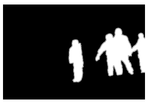
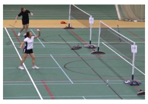
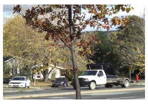
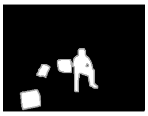
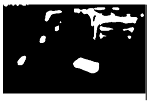
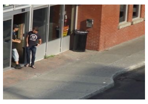
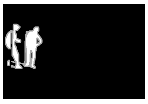
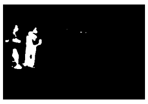
Categories	Original	Ground Truth	10-SC-SOBS	18-AAPSA	29-cascaded CNN
B-Weather Skating (in002349)					
Baseline Pedestrians (in000490)					
C-Jitter Badminton (in001123)					
Dynamic-B Fall (in002416)					
I-O-Motion Sofa (in001314)					
Low-F TunnelExit (in002781)					
NightVideos F-Highway (in000450)					
PTZ TwoPosition (in001216)					
Shadow BusStation (in000394)					
Thermal D-Room (in002656)					
Turbulence T-3 (in000999)					

Table 7 Visual results on CDnet 2014 dataset (Part 2): From left to right: Original images, Ground-Truth images, SC-SOBS [211], AAPSA [255], cascaded CNN [336].

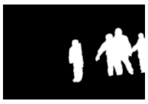
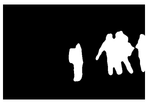
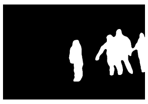
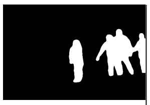
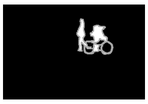
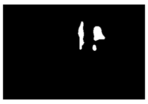
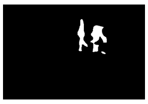
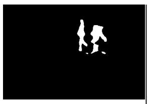
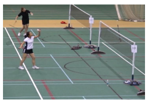
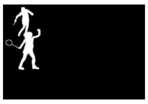
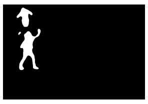
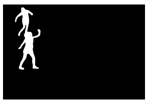
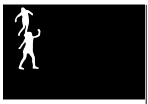
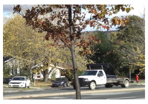
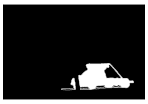
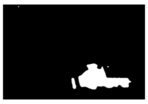
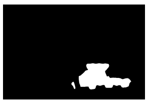
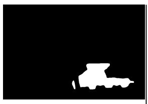
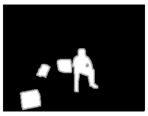
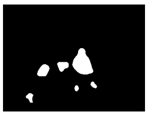
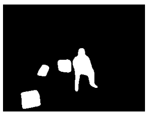
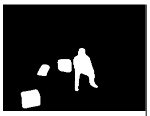
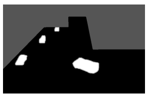
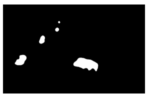
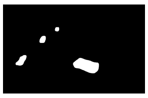
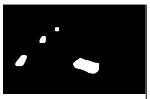
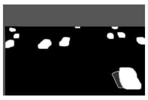
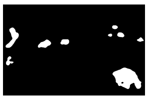
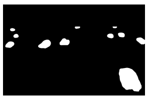
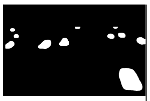
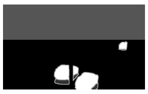
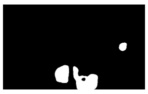
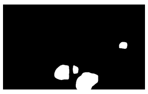
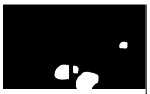
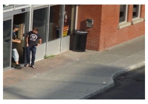
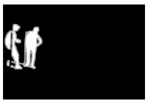
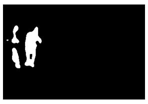
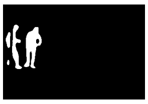
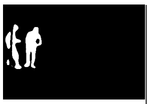
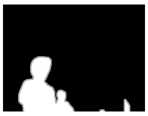
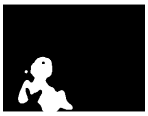
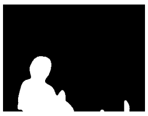
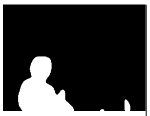
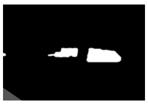
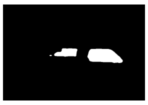
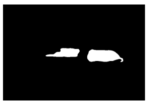
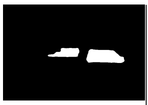
Categories	Original	Ground Truth	34-DeepBS	39-FgSegNet	44-FgSegNet-SFPM
B-Weather Skating (in002349)					
Baseline Pedestrians (in000490)					
C-Jitter Badminton (in001123)					
Dynamic-B Fall (in002416)					
I-O-Motion Sofa (in001314)					
Low-F TunnelExit (in002781)					
NightVideos F-Highway (in000450)					
PTZ TwoPosition (in001216)					
Shadow BusStation (in000394)					
Thermal D-Room (in002656)					
Turbulence T-3 (in000999)					

Table 8 Visual results on CDnet 2014 dataset (Part 3): From left to right: Original images, Ground-Truth images, DeepBS [9], FgSegNet [193], FgSegNetSFPM [194].

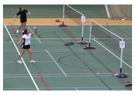
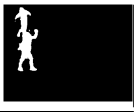
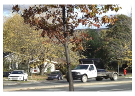
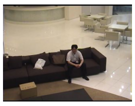
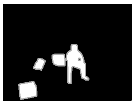
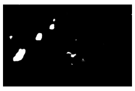
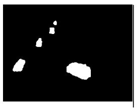
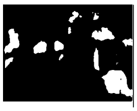
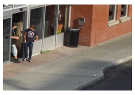
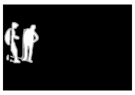
Categories	Original	Ground Truth	Semantic BGS	semi-supervised MSRNN	unsupervised MSRNN
B-Weather Skating (in002349)					
Baseline Pedestrians (in000490)					
C-Jitter Badminton (in001123)					
Dynamic-B Fall (in002416)					
I-O-Motion Sofa (in001314)					
Low-F TunnelExit (in002781)					
NightVideos F-Highway (in000450)					
PTZ TwoPosition (in001216)					
Shadow BusStation (in000394)					
Thermal D-Room (in002656)					
Turbulence T-3 (in000999)					

Table 9 Visual results on CDnet 2014 dataset (Part 4): From left to right: Original images, Ground-Truth images, Semantic BGS [31], semi-supervised MSRNN [56], unsupervised MSRNN [55]. For semanticBGS, the authors did not tested their algorithm on five categories

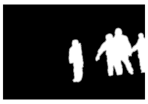
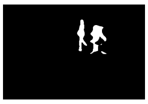
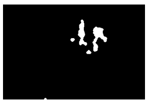
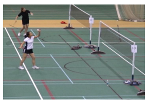
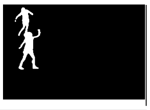
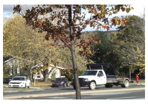
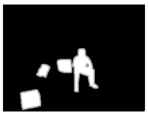
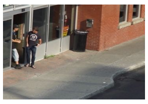
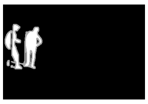
Categories	Original	Ground Truth	45-FgSegNet-V2	DCP	41-BSPVGAN
B-Weather Skating (in002349)					
Baseline Pedestrians (in000490)					
C-Jitter Badminton (in001123)					
Dynamic-B Fall (in002416)					
I-O-Motion Sofa (in001314)					
Low-F TunnelExit (in002781)					
NightVideos F-Highway (in000450)					
PTZ TwoPosition (in001216)					
Shadow BusStation (in000394)					
Thermal D-Room (in002656)					
Turbulence T-3 (in000999)					

Table 10 Visual results on CDnet 2014 dataset (Part 5): From left to right: Original images, Ground-Truth images, FgSegNet-V2 [195], DCP [296], BPVGAN [380]. For DCP, the authors did not tested their algorithm on four categories.

super-pixel filtering, and Attention ConvLSTM+CRF [192] with Conditional Random Field. The temporal-wise algorithms model the dependencies among adjacent temporal pixels, and thus enforce temporal coherence, such as Joint TS-CNN [374] with background reconstruction feedback and 3D-CNN [270]. The spatio-temporal-wise algorithms model both the dependencies among adjacent spatial and temporal pixels, and thus enforce both spatial and temporal coherence, such as Attention ConvLSTM+PSL+CRF [50] with different architectures. Table 12 groups the different F-measures which come either from the corresponding papers or directly from ChangeDetection.net website.

Barnich and Van Droogenbroeck [30] did not test ConvNet on the ”Intermittent Motion Object” and ”PTZ” categories because they claimed that their method is not designed for it. Similarly, Lim et al. [192] did not evaluate Struct-CNN on the ”PTZ” category, nor were MCSS and BScGAN. Zeng and Zhu [366] only evaluated MFCN on the ”THM” category because this method was designed for infrared video. For these methods, the average F-measure is achieved by indicating the missing category or number of missing categories.

For FgSegNet-M [193], FgSegNet-S [194], FgSegNet-V2 [195], we noticed that the F-measure reported by the authors are different than those available on the CDnet website. We report one of the official CDnet, and the overall score provided by the authors are given in parentheses.

B) Quantitative Analysis

Table 12 groups the different F-measures that come either from the corresponding papers or directly from changedetection.net website. We highlighted in bold the best algorithm score in each category. The top-ten methods are indicated along with their rank. Figure 66 and Figure 67 show graphics of the F-measures for the key methods, from MOG to the current leading method, FgSegNet-V2 [193].]. In these figures, the more the curve of the method reaches closer to a circle with a radius of 1, the more the method is robust over the eleven categories of CDnet 2014 dataset.

By analyzing Table 12 and looking at Figure 8.2.2 and Figure 66.a, we can first see that the representative conventional neural networks, namely, Coherence-based and Fuzzy SOBS (SOBS-CF) [210] and SOBS with Spatial Coherence (SC-SOBS) [211] slightly outperform the basic statistical models such as MOG [294] designed in 1999 even with improvements (i.e. RMOG [314] developed in 2013). . However, SOBS and its variants were the leading methods for the CDnet 2012 dataset [112] for a long time (approximately two years), demonstrating the interest in neural networks for background subtraction. However, the F-measure did not exceed $0./9$ on average, which is relatively low. The F-measure exceeded only 0.9 for the baseline category making these methods only usable and reliable in applications where the environments were not overly complex.

Second, we can also see in Table 12, Figure 8.2.2 and Figure 66.b that advanced non-parametric models such as SuBSENSE [292] and PAWCS [293] developed in 2014 and 2015, respectively, achieve a chronologically better performance than SOBS-CF and SC-SOBS because of multi-features and multi-cues strategies. The gain in F-measure score was approximately 25%. The average F-measure was approximately 0.75, which becomes more acceptable in terms of reliable use under real conditions. In particular, the F-measure was approximately 0.9 for several challenges (baseline, dynamic backgrounds, camera jitter, and shadow). Thus, these methods are more applicable in more complex environments.

Third, we can observe that CNN-based methods can achieve a maximum increase in average F-measure of approximately 30% compared to SuBSENSE [292] and PAWCS [293], demonstrating their superiority on this task. Figure 8.2.2 compares the performance of PAWCS [293], SuBSENSE [292], Cascaded CNN [336] and FgSegNet-V2 [193] and Figure 66.c also compares SuBSENSE [292] with several CNNs based methods. The first CNN-based method provides a better performance than SuBSENSE in all categories. In addition, we can see in Figure 67.a that the top DNNs based methods clearly outperforms SuBSENSE. In Figure 66.(d), we can also see an increase in performance between the first cascaded CNNs method published in 2016 and one of the top method FgSegNet-M [195] which was designed in 2018, thereby showing the progress made during a two year period. Such an increase in performance required approximately 5 years before the use of deep neural networks. However, CNNs significantly increase the F-measure under dynamic backgrounds, camera jitter, intermittent object motion, and turbulence categories. For the "PTZ" category, the performance is mitigated as can be seen in works of several authors who did not provide results on this category, arguing that they did not design their method for this type of challenge, although their scores obtained using GANs are extremely interesting. These methods appear to be usable and reliable in an extremely large spectrum of environments, but are mostly scene-specific with supervised mode. We can also see that the training has a significant influence on the performance. Indeed, the results obtained by ConvNet using manual foreground masks (GT) obtained a F-Measure around 0.9 whereas this value falls to approximately 0.79 using the foreground masks from IUTIS, demonstrating a slight increase in performance in comparison with SuBSENSE [292] and PAWCS [293]. This fact also highlights that the increase in performance obtained by DNN-based methods is essentially due to their supervised aspects. In addition, their current computation times, as shown in Table 4, are too slow to be currently employed in real applications.

The top-ten DNN-based methods can be decomposed into three main groups.. The first group consists of FgSegNet methods developed by Lim and Keles [193–195]. Indeed, FgSegNet-V2 [193], FgSegNet-S [194] and FgSegNet-M [195] take the top-three places. Their success seems to be due to the architecture of FgSegNet, which is particularly designed for background subtraction, and by their spatial-wise aspects. The second group consists of 3D-CNNs based methods (MCF3D [337], 3D Atrous CNN [148], FC3D [337], and 3D-CNN[270]). This good performance of 3D-CNN based methods is due to their ability to take into account both spatial and temporal constraints, which are extremely important in this field. Figure 67.(d) compare the different 3D-CNNs based methods. We can state that MCF3D [337] offers the closest curve to a circle with a radius of 1 but present a weakness for the IOM category, as compared to the other 3D-CNN based methods. Finally, the third group consists of unsupervised GAN-based methods (BPVGAN [380], BVGAN [379] and BScGAN [15]). However, their performance can be improved because these methods are pixel-wise without taking into account either the spatial or temporal constraints. Figure 67.b compare three top DNNs that belongs each to one of the three top groups. We can note that FgSegNet-V2 [193] outperforms both MFC3D [337] and BPVGAN

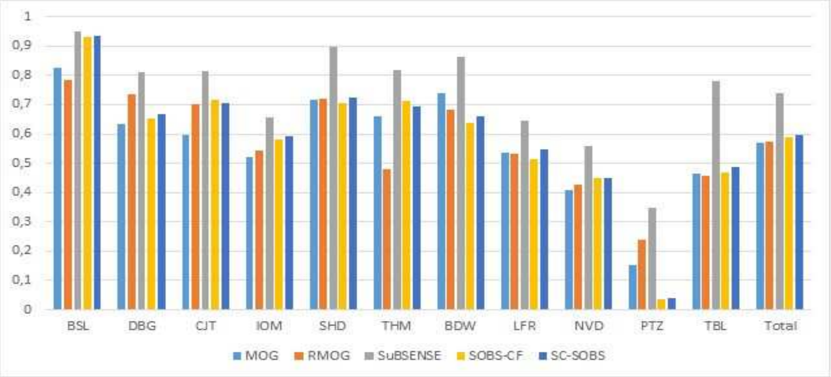


Fig. 64 Comparison of F-Measure between MOG [294], RMOG [314], SOBS-CF, SC-SOBS and SuBSENSE [292]. It can be noted that SOBS-CF and SC-SOBS outperform MOG except on the ”BDW” and ”PTZ” categories. SuBSENSE provides the best performance.

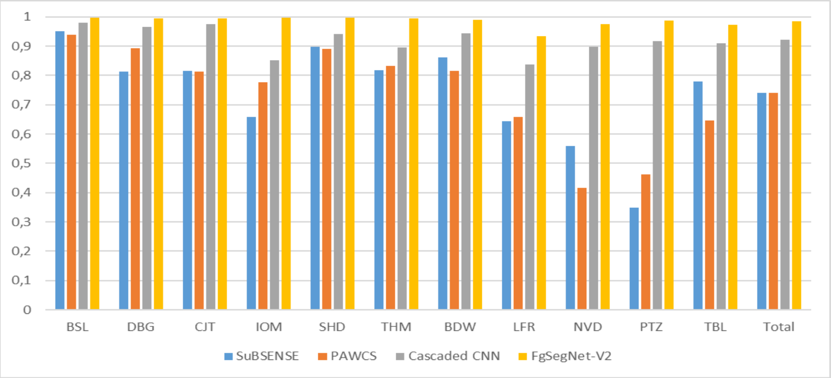


Fig. 65 Comparison of F-Measure between PAWCS [293], SuBSENSE [292], Cascaded CNN [336] and FgSegNet-V2 [193]. It can be noted that Cascaded CNN and FgSegNet-V2 outperform PAWCS and SuBSENSE on all the categories. FgSegNet-V2 provides the best performance.

[380]. Moreover, FgSegNet-V2 [193] presents no main weaknesses in a single category. Figure 67.c highlights the increase in performance over 20 years of research between MOG developed in 1999 to FgSegNet-V2 [193] designed in 2018. We can state that the curve of the compared methods progressively increases from the first method, MOG, to FgSegNet-V2 [193], highlighting our quantitative analysis. Furthermore, the curve of FgSegNet-V2 [193] is close to a circle with a radius of 1, indicating that deep learning methods are able to reach a quasi-ideal performance.

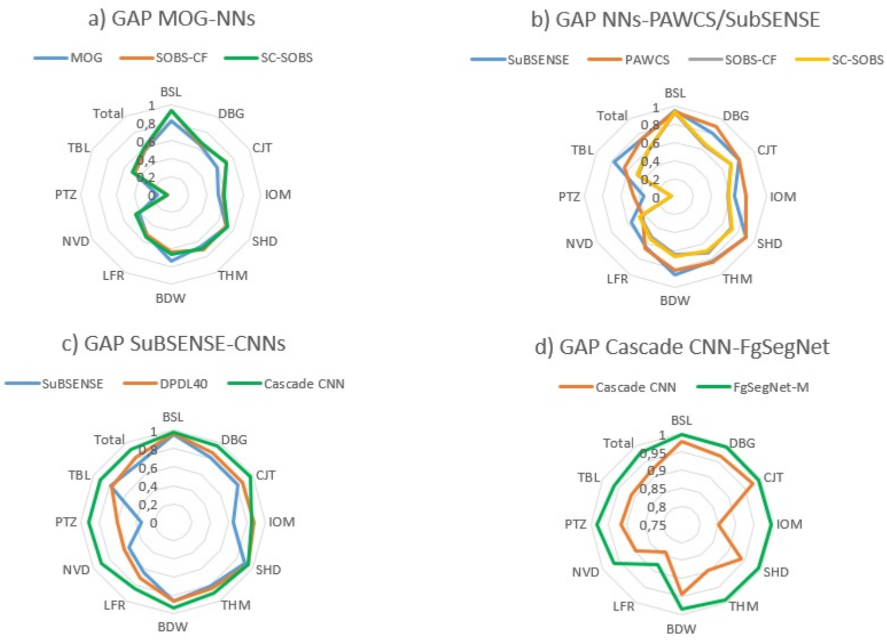


Fig. 66 First row: a) Gap between MOG [294] and conventional neural networks (SOBS-CF, SC-SOBS). b) Gap between conventional NNs, and PAWCS [293]/SuBSENSE [292]. Second row: c) Gap between SuBSENSE and CNNs, d) GAP between the first cascaded CNNs and one of the best DNN method (FgSegNet-M [195]).

9 How far are DNNs from the ideal method?

To evaluate the progress of background subtraction methods since MOG was developed in 1999 until the advent of DNN-based methods in 2018, we computed different key increases in the F-measure in terms of percentage. To do so, we considered a) the gap between MOG and the best conventional neural network (SC-SOBS), b) the gap between SC-SOBS and the best non-parametric multi-cues methods (SubSENSE), c) the gap between SuBSENSE and Cascaded CNNs, d) the gap between SuBSENSE and the best DNNs based method (FgSegNet-V2), and e) the gap between FgSegNet-V2 and the ideal method (F-Measure$=1$ in each category). From Table 11, we can see than the big gap was obtained by DNNs methods againts SuBSENSE with 24.31 and 32.92 for Cascaded CNN and FgSegNet-V2, respectively. We can also note that the gap of 1.55% that remains between FgSegNet-V2 and the ideal method is less than the gap of 6.93% between Cascaded CNN and FgSegNet-V2. This gap can be partially filled by three main directions: robust deep auto-encoders [73, 167, 250, 382, 383] probabilistic [104] and fuzzy [77, 96] DNNs, and GANs architecture specifically designed for background subtraction. Nevertheless, it is important to note that the large gap between cascaded CNN and FgSegNet-V2 is mainly due to their supervised aspect, and a required drawback of training using labeling data. However, when

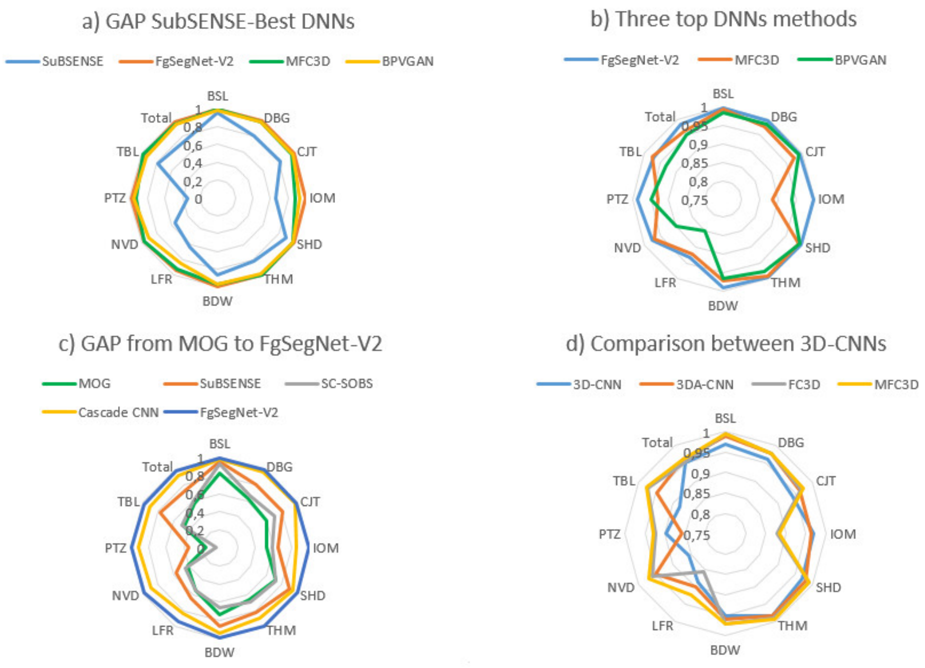


Fig. 67 First row: a) Gap between SuBSENSE [292] and three top DNNs based methods. b) Comparison of three top DNNs. c) Second row: Gap between from MOG (1999) to FgSegNet-V2 [193] (2018) that represent 20 years of research. d) gap between the different 3D-CNNs based methods.

labeling data are unavailable, efforts should be concentrated on unsupervised GANs as well as unsupervised methods based on semantic background subtraction [31,363] and robust subspace tracking [236,247,261,262,315,316] that are still of interest in the field of background subtraction.

Challenges/Gap	MOG/SC-SOBS	SC-SOBS/SuBSENSE	SuBSENSE/Cascaded CNN	SuBSENSE/FgSegNetV2	Cascaded CNN/FgSegNetV2	FgSegNetV2/Ideal
BSL	**13.20%**	1.82%	2.98%	5.00%	1.96%	0.22%
DBG	5.62%	21.40%	18.98%	**22.59%**	3.03%	0.49%
CJT	18.13%	15.61%	19.70%	**21.91%**	1.84%	0.62%
IOM	13.65%	11.00%	29.47%	**51.64%**	17.12%	0.39%
SHD	1.03%	**24.29%**	4.76%	10.78%	5.75%	0.45
THM	4.56%	18.03%	9.63%	**21.63%**	10.94%	0.62
BDW	-10.30%	**30.20%**	9.42%	14.91%	5.02%	0.97%
LFR	1.68%	17.98%	29.87%	**44.86%**	11.54%	7.11%
NVD	9.91%	24.34%	60.12%	**73.94%**	8.63%	2.68%
PTZ	-73.13%	**749.88%**	163.75%	183.72%	7.57%	1.40%
TBL	4.65%	**59.67%**	16.89%	24.83%	6.80%	2.81%
Average	4.45%	24.27%	24.31%	**32.92%**	6.93%	1.55%

Table 11 Gain in terms of F-measure score in percentage over the eleven categories of the CDnet2014, namely, Baseline (BSL), Dynamic background (DBG), Camera jitter (CJT), Intermittent Motion Object (IOM), Shadows (SHD), Thermal (THM), Bad Weather (BDW), Low Frame Rate (LFR), Night Videos (NVD), PTZ, Turbulence (TBL). In bold, maximum gain.

Algorithms (Authors)	BSL	DBG	CJT	IOM	SHD	THM	BDW	LFR	NVD	PTZ	TBL	Average F-Measure
A) Basic statistical models												
MOG (Stauffer and Grimson [294] 1999)	0.8245	0.6330	0.5969	0.5207	0.7156	0.6621	0.7380	0.5373	0.4097	0.1522	0.4663	0.5707
RMOG (Varadarajan et al. [314] 2013)	0.7848	0.7352	0.7010	0.5431	0.7212	0.4788	0.6826	0.5312	0.4265	0.2400	0.4578	**0.5735**
B) Advanced non parametric models												
SuBSENSE (St-Charles et al. [292] 2014)	0.9503	0.8117	0.8152	0.6569	0.8986	0.8171	0.8619	0.6445	0.5599	0.3476	0.7792	**0.7408**
PAWCS (St-Charles et al. [293] 2015)	0.9397	0.8938	0.8137	0.7764	0.8913	0.8324	0.8152	0.6588	0.4152	0.4615	0.6450	0.7403
Spectral-360 (Sedky et al. [275] 2014)	0.9330	0.7872	0.7156	0.5656	0.8843	0.7764	0.7569	0.6437	0.4832	0.3653	0.5429	0.7054
C) Conventional Neural Networks												
SOBS-CF (Maddalena and Petrosino [210] 2010)	0.9299	0.6519	0.7150	0.5810	0.7045	0.7140	0.6370	0.5148	0.4482	0.0368	0.4702	0.5883
SC-SOBS (Maddalena and Petrosino [211] 2012)	0.9333	0.6686	0.7051	0.5918	0.7230	0.6923	0.6620	0.5463	0.4503	0.0409	0.4880	**0.5961**
D) Deep Neural Networks (*Structure*)												
1) Encoder-Decoder Networks												
MSRNN* [Semi-supervised] (*Spatial/Temporal-wise*) (Choo et al. [56] 2018)	0.9737	0.8632	0.9255	0.8012	0.8796	0.8371	0.7775	0.7331	0.7921	0.5724	0.8796	0.8214 (0.8440)
MSRNN without recurrent layers* (*Pixel-wise*) (Choo et all. [56] 2018)	0.9147	0.5529	0.8624	0.6985	0.7638	0.7694	0.6739	0.6420	0.6189	0.3990	0.4578	0.6685 (0.6930)
1 LSTM* (*Pixel-wise*) (Choo et all. [56] 2018)	0.9074	0.5117	0.8314	0.7314	0.7577	0.7721	0.6556	0.6290	0.7080	0.4718	0.4602	0.6760 (0.6946)
MSRNN* [Unsupervised] (*Spatial/Temporal-wise*) (Choo et al. [55] 2018)	0.9595	0.9135	0.9246	0.8717	0.9557	0.8483	0.8874	0.8435	0.5576	-	0.7998	**0.8562** (PTZ)
2) Convolutional Neural Networks [Supervised]												
2.1) Basic CNNs												
CNN* (ConvNet-GT) (*LeNet-5*) (*Pixel-wise*) (Barnich and Droogenbroeck [30] 2016)	0.9813	0.8845	0.9020	-	0.9454	0.8543	0.9254	0.9612	0.7565	-	0.9297	0.9044 (IOM, PTZ)
CNN* (ConvNet-IUTIS) (*LeNet-5*) (*Pixel-wise*) (Barnich and Droogenbroeck [30] 2016)	0.9647	0.7923	0.8013	-	0.8590	0.7559	0.8849	0.8273	0.4715	-	0.7506	**0.7897** (IOM, PTZ)
$DPDL_1{}^*$ (One GT) (*CNN*) (*Temporal-wise*) (Zhao et al. [371] 2018)	0.7886	0.6566	0.5456	0.5115	0.6957	0.6697	0.6036	0.5966	0.3953	0.2942	0.6301	0.5807 (0.5602)
$DPDL_{20}{}^*$ (20 GTs) (*CNN*) (*Temporal-wise*) (Zhao et al. [371] 2018)	0.9620	0.8369	0.8627	0.8174	0.8763	0.8311	0.8107	0.6646	0.5866	0.4654	0.7173	0.7665 (0.7195)
$DPDL_{40}{}^*$ (40 GTs) (*CNN*) (*Temporal-wise*) (Zhao et al. [371] 2018)	0.9692	0.8692	0.8661	0.8759	0.9361	0.8379	0.8688	0.7078	0.6110	0.6087	0.7636	0.8106 (0.7491)
2.2) Multi-scale or/and cascaded CNNs												
cascaded CNN (*CNN-1/CNN-2*) (*Spatial-wise*) (Wang et al. [336] 2016)	0.9786	0.9658	0.9758	0.8505	0.9414	0.8958	0.9431	0.8370	0.8965	0.9168	0.9108	0.9209
FgSegNet-M (-) (*Spatial-wise*) (Lim and Keles [193] 2018)	0.9973	0.9958	0.9954	0.9951	0.9937	0.9921	0.9845	0.8786	0.9655	0.9843	0.9648	0.9770 (0.9865*) **Rank 3**
FgSegNet-S (-) (*Spatial-wise*) (Lim and Keles [194] 2018)	0.9977	0.9958	0.9957	0.9940	0.9927	0.9937	0.9897	0.8972	0.9713	0.9879	0.9681	0.9804 (0.9878*) **Rank 2**
FgSegNet-V2 (-) (*Spatial-wise*) (Lim et al. [195] 2018)	0.9978	0.9951	0.9938	0.9961	0.9955	0.9938	0.9904	0.9336	0.9739	0.9862	0.9727	**0.9847** (0.9890*) **Rank 1**
MCSS* (-) (*Spatial-wise*) (Liao et al. [191] 2018)	0.9940	0.881	0.794	0.770	0.915	0.883	0.861	0.725	0.788	-	0.884	0.844
Guided Multi-scale CNN* (-) (*Spatial-wise*) (Liang et al. [190] 2018)	0.9791	0.8266	0.8818	0.6229	0.8910	0.7490	0.8711	0.6396	0.5048	0.6057	0.8114	0.7591
MsEDNet* (*Compact VGG-16*) (*Spatial-wise*) (Patil et al. [245] 2018)	0.9368	0.9068	0.9173	0.8187	0.9196	0.8825	0.9055	-	-	-	-	0.8988 (LFR, NVD, PTZ, TBL)
2.3) Fully CNNs												
MFCN (-) (*Spatial-wise*) (Zeng and Zhu [366] 2018)	-	-	-	-	-	0.9870	-	-	-	-	-	0.9870 (only THM)
MFCN-CL (-) (*Spatial-wise*) (Zeng and Zhu [364] 2018)	0.9931	0.9956	0.9931	0.9822	0.9911	0.9873	0.9881	0.9353	0.9764	0.9818	0.9709	**0.9830**
2.4) Deep CNNs												
Deep CNN (DeepBS) (-) (*Pixel-wise*) (Babaee et al. [9] 2017)	0.9580	0.8761	0.8990	0.6098	0.9304	0.7583	0.8301	0.6002	0.5835	0.3133	0.8455	0.7548
Two-Stage CNN* (TS-CNN) (-) (*Pixel-wise*) (Zhao et al. [374] 2018)	0.9630	0.7405	0.8689	0.8734	0.9216	0.8536	0.8004	0.8075	0.6851	0.4493	0.6929	0.7870
Joint TS-CNN* (-) (*Temporal-wise*) (Zhao et al. [374] 2017)	0.9680	0.7716	0.8988	0.9066	0.9286	0.8586	0.8550	0.7491	0.7695	0.5168	0.7143	0.8124
Attention ConvLSTM* (*VGG-16*) (*Temporal-wise*) (Chen et al. [50] 2018)	0.9243	0.6030	0.9053	0.572	0.8916	0.7181	0.8493	0.5920	0.5060	0.7436	0.7347	0.7314
Attention ConvLSTM+CRF* (*VGG-16*) (*Spatial/Temporal-wise*)(Chen et al. [50] 2018)	0.9383	0.6207	0.9251	0.6058	0.8962	0.7271	0.8846	0.6113	0.5188	0.7697	0.7404	0.7489
Attention ConvLSTM+PSL+CRF* (*VGG-16*) (*Spatial/Temporal-wise*) (Chen et al. [50] 2018)	0.9594	0.7356	0.9422	0.7538	0.9084	0.8546	0.8949	0.6175	0.7526	0.7816	0.9207	0.8292
Attention ConvLSTM+PSL+CRF* (*GoogleLeNet*) (*Spatial/Temporal-wise*) (Chen et al. [50] 2018)	0.8557	0.6588	0.8864	0.6488	0.8049	0.7725	0.7961	0.5947	0.6003	0.7136	0.7637	0.7360
Attention ConvLSTM+PSL+CRF* (*ResNet*) (*Spatial/Temporal-wise*) (Chen et al. [50] 2018)	0.9294	0.8220	0.9518	0.8453	0.9647	0.9444	0.9461	0.8080	0.8585	0.7776	0.8011	0.8772
MSFgNet (*Spatial/Temporal-wise*) (Patil and Murala [244] 2018)	0.9211	0.8514	0.8313	0.7797	0.9345	0.8016	0.8504	0.8422	0.8099	0.7870	0.8624	0.8430
MSFgNet-I (*Spatial/Temporal-wise*) (Patil and Murala [244] 2018)	0.9217	0.9687	0.8974	0.8705	0.9972	0.9458	0.9741	0.8281	0.8472	0.9477	0.8583	**0.9140**
2.5) Structured CNNs												
Struct-CNN* (*VGG-16*) (*Spatial-wise*) (Lim et al. [192] 2017)	0.9586	0.9112	0.8990	0.8780	0.8565	0.8048	0.8757	0.9321	0.7715	-	0.7573	**0.8645**
2.6) 3D CNNs												
3D CNN* (*C3D branch*) (*Temporal-wise*) (Sakkos et al. [270] 2017)	0.9691	0.9614	0.9396	0.9698	0.9706	0.9830	0.9509	0.8862	0.8565	0.8987	0.8823	0.9507 **Rank 7**
3D Atrous CNN* (ConvLTSM) (-) (*Spatial/Temporal-wise*) (Hu et al. [148] 2018)	0.9897	0.9789	0.9645	0.9637	0.9813	0.9833	0.9609	0.8994	0.9489	0.8582	0.9488	0.9615 **Rank 5**
FC3D* (*C3D branch*) (*Spatial/Temporal-wise*) (Wang et al. [337] 2018)	0.9941	0.9775	0.9651	0.8779	0.9881	0.9902	0.9699	0.8575	0.9595	0.9240	0.9729	0.9524 **Rank 6**
MFC3D* (*C3D branch*) (*Spatial/Temporal-wise*) (Wang et al. [337] 2018)	0.9950	0.9780	0.9744	0.8835	0.9893	0.9924	0.9703	0.9233	0.9696	0.9287	0.9773	**0.9619 Rank 4**
3) Generative Adversarial Networks [Unsupervised]												
DCP* (*VGG-19*) (Sultana et al. [296] 2018)	0.8178	0.7757	0.8376	0.5979	0.7665	0.8212	0.8212	-	-	-	-	0.7620 (LFR,NVD, PTZ, TBL))
BScGAN* (*UNet/ResNet*) (*Pixel-wise*) (Bakkay et al. [15] 2018)	0.9930	0.9784	0.9770	0.9623	0.9828	0.9612	0.9796	0.9918	0.9661	-	0.9712	**0.9763** (PTZ) **Rank 10**
BGAN (-) (*Pixel-wise*) (Zheng et al. [379] 2018)	0.9814	0.9763	0.9828	0.9366	0.9849	0.9064	0.9465	0.8472	0.8965	0.9194	0.9118	0.9339 (0.9368) **Rank 9**
BPVGAN (-) (*Pixel-wise*) (Zheng et al. [380] 2018)	0.9837	0.9849	0.9893	0.9366	0.9927	0.9764	0.9644	0.8508	0.9001	0.9486	0.9310	0.9501 (0.9509) **Rank 8**

Table 12 F-measure metric over the 6 categories of the CDnet2014, namely Baseline (BSL), Dynamic background (DBG), Camera jitter (CJT, Intermittent Motion Object (IOM), Shadows (SHD), Thermal (THM), Bad Weather (BDW), Low Frame Rate (LFR), Night Videos (NVD), PTZ, Turbulence (TBL). $*$ indicated that the measures come from the corresponding papers otherwise the measures comes from the ChangeDetection.net website. In bold, the best score in each algorithm's category. The top 10 methods are indicated with their rank. There are three groups of leading methods: FgSegNet's group, 3D-CNNs group and GANs group.

10 Conclusion

In this paper, we first presented a full review of recent advances in deep neural networks as applied to background generation, background subtraction, and deep learned features for the detection of moving objects in video taken by a static camera. Experiment results on the large-scale CDnet 2014 dataset show the increase in performance obtained using supervised deep neural network methods in this field. Although deep neural networks have recently received significant attention for their use in background subtraction during the last two years since the seminal study by Braham and Van Droogenbroeck [30], there remain many important and unresolved issues:

- The main question remains what is the most suitable type of deep neural network and its corresponding architecture for background initialization, background subtraction, and deep learned features in the presence of complex backgrounds?
- Looking at the various experiments conducted, it can be observed that deep learning approaches detect the changes in images with static backgrounds successfully but are more sensitive in the case of dynamic backgrounds and camera jitter, although they do provide a better performance than conventional approaches [170]. In addition, several authors avoid experiments on the "IOM" and the "PTZ" categories. In addition, when the F-Measure is provided for these categories, the score is not very high. Thus, it seems that the current deep neural networks tested face problems in theses cases perhaps because they have difficulties in how to learn the duration of sleeping moving objects and how to handle changes from moving cameras.
- For the inputs, all of the authors employed either gray or color images in RGB, with the exception of Zhao et al. [371] who used a distribution learning feature to improve the performance of a basic CNNs. However, it would be interesting to employ RGB-D images because depth information is extremely helpful in several challenges such as in camouflage images, as developed by Maddalena and Petrosino [218]. In addition, the conventional neural networks SOBS [219] is the top algorithm on the SBM-RGBD dataset [37]. Thus, we can expect that CNNs with RGB-D features as inputs will also achieve a significant performance as a ForeGAN-RGBD [297] model. . However, multi-spectral data would also be interesting to test. Furthermore, a study on the influence of the input feature type would be an area of interest.
- Rather than working in the pixel domain, DNNs may also be applied to the measurement domain for use in conjunction with compressive sensing data such as in RPCA models [75,247].

Currently, mainly CNNs and basic GANs have been employed for background subtraction. Thus, a future direction may be to investigate the adequacy and use of pyramidal deep CNNs [312], deep belief neural networks, deep restricted kernel neural networks [298], probabilistic neural networks [104], deep fuzzy neural networks [77, 96] and fully memristive neural networks [51,90,133,174,175,370] for both static and moving cameras [223].

References

1. B. Afonso, L. Cinelli, L. Thomaz, A. da Silva, E. da Silva, and S. Netto. Moving-camera video surveillance in cluttered environments using deep features. *IEEE International Conference on Image Processing, ICIP 2018*, pages 2296–2300, 2018.
2. A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(1):294–302, 2004.
3. T. Akilan. A foreground inference network for video surveillance using multi-view receptive field. *Preprint*, January 2018.
4. T. Akilan. Video foreground localization from traditional methods to deep learning. *PhD Thesis, University of Windsor, Canada*, 2018.
5. T. Akilan and J. Wu. Double Encoding - Slow Decoding Image to Image CNN for Foreground Identification with Application Towards Intelligent Transportation. *IEEE Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing*, pages 395–403, 2018.
6. T. Akilan and J. Wu. An Improved Video foreground Extraction Strategy using Multi-view Receptive Field and EnDec CNN. *IEEE Transactions on Industrial Informatics*, 2019.
7. T. Akilan and J. Wu. sEnDec: An improved image to image CNN for foreground localization. *IEEE Intelligent Transportation Systems Transactions*, 2019.
8. T. Akilan, J. Wu, W. Jiang, A. Safaei, and J. Huo. New trend in video foreground detection using deep learning. *IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2018*, pages 889–892, 2018.
9. M. Babaee, D. Dinh, and G. Rigoll. A deep convolutional neural network for background subtraction. *Pattern Recognition*, September 2017.
10. F. El Baf, T. Bouwmans, and B. Vachon. Foreground detection using the Choquet integral. *International Workshop on Image Analysis for Multimedia Interactive Integral, WIAMIS 2008*, pages 187–190, May 2008.
11. F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008*, pages 1729–1736, June 2008.
12. F. El Baf, T. Bouwmans, and B. Vachon. Type-2 fuzzy mixture of Gaussians model: Application to background modeling. *International Symposium on Visual Computing, ISVC 2008*, pages 772–781, December 2008.
13. F. Bahri, M. Shakeri, and N. Ray. Online illumination invariant moving object detection by generative neural network. *Preprint*, 2018.
14. J. Bai, H. Zhang, and Z. Li. The Generalized Detection Method for the Dim Small Targets by Faster R-CNN Integrated with GAN. *IEEE International Conference on Communication and Information Systems, ICCIS 2018*, pages 1–5, 2018.
15. M. Bakkay, H. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek. BSCGAN: deep background subtraction with conditional generative adversarial networks. *IEEE International Conference on Image Processing, ICIP 2018*, October 2018.
16. O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.
17. A. Barron. Approximation and estimation bounds for artificial neural networks. *Neural networks*, 14(1):115–133, 1994.
18. S. Basu, S. Mukhopadhyay, ManoharKarki, R. Biano, S. Ganguly, R. Nemani, and S. Gayaka. Deep neural networks for texture classification: A theoretical analysis. *Neural Networks*, 97:173–182, January 2018.
19. C. Bautista, C. Dy, M. Manalac, R. Orbe, and M. Cordel. Convolutional neural network for vehicle detection in low resolution traffic videos. *TENCON 2016*, 2016.
20. S. Bianco, G. Ciocca, and R. Schettini. How far can you get by combining change detection algorithms? *CoRR*, abs/1505.02921, 2015.
21. T. Bouwmans. Background Subtraction For Visual Surveillance: A Fuzzy Approach. *Chapter 5, Handbook on Soft Computing for Video Surveillance, Taylor and Francis Group, S.K. Pal, A. Petrosino, L. Maddalena*, pages 103–139, March 2012.
22. T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11(31-66), May 2014.
23. T. Bouwmans. Traditional Approaches in Background Modeling for Video Surveillance. *Handbook Background Modeling and Foreground Detection for Video Surveillance, Taylor and Francis Group, T. Bouwmans, B. Hoferlin, F. Porikli, A. Vacavant*, July 2014.

24. T. Bouwmans and F. El Baf. Modeling of Dynamic Backgrounds by Type-2 Fuzzy Gaussians Mixture Models. *MASAUM Journal of Basic and Applied Sciences*, 1(2):265–277, November 2009.
25. T. Bouwmans and B. Garcia-Garcia. Background subtraction in real applications: Challenges, current models and future directions. *Preprint*, 2019.
26. T. Bouwmans, L. Maddalena, and A. Petrosino. Scene Background Initialization: a Taxonomy. *Pattern Recognition Letters*, January 2017.
27. T. Bouwmans, C. Silva, C. Marghes, M. Zitouni, H. Bhaskar, and C. Frelicot. On the role and the importance of features for background modeling and foreground detection. *Computer Science Review*, 28:26–91, May 2018.
28. T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23:1–71, February 2017.
29. T. Bouwmans and E. Zahzah. Robust PCA via Principal Component Pursuit: A Review for a Comparative Evaluation in Video Surveillance. *Special Issue on Background Models Challenge, Computer Vision and Image Understanding, CVIU 2014*, 122:22–34, May 2014.
30. M. Braham and M. Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. *International Conference on Systems, Signals and Image Processing, IWSSIP 2016*, pages 1–4, May 2016.
31. M. Braham, S. Pierard, and M. Van Droogenbroeck. Semantic Background Subtraction. *IEEE International Conference on Image Processing, ICIP 2017*, September 2017.
32. J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 35(8):8721886, 2013.
33. A. Brunettia, D. Buongiorno, G. Trotta, and V. Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:1733, July 2018.
34. B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. DehazeNet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25:1–13, 2016.
35. M. Camplani, C. Blanco, L. Salgado, F. Jaureguizar, and N. Garca. Advanced background modeling with RGB-D sensors through classifiers combination and inter-frame foreground prediction. *Machine Vision and Applications*, 2014.
36. M. Camplani, L. Maddalena, G. Moya Alcover, A. Petrosino, and L. Salgado. A Benchmarking Framework for Background Subtraction in RGBD Videos. *ICIAP 2017*, pages 219–229, September 2017.
37. M. Camplani, L. Maddalena, G. Moya Alcover, A. Petrosino, and L. Salgado. RGB-D dataset: Background learning for detection and tracking from RGBD videos. *IEEE ICIAP-Workshops 2017*, 2017.
38. E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *International Journal of ACM*, 58(3), May 2011.
39. T. Cane and J. Ferryman. Evaluating deep semantic segmentation networks for object detection in maritime surveillance. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2018*, pages 1–6, 2018.
40. X. Cao, L. Yang, and X. Guo. Total variation regularized RPCA for irregularly moving object detection under dynamic background. *IEEE Transactions on Cybernetics*, 46(4):1014–1027, April 2016.
41. M. Chacon-Muguia, S. Gonzalez-Duarte, and P. Vega. Simplified SOM-neural model for video segmentation of moving objects. *International Joint Conference on Neural Networks, IJCNN 2009*, pages 474–480, 2009.
42. M. Chacon-Murguia, G. Ramirez-Alonso, and S. Gonzalez-Duarte. Improvement of a neural-fuzzy motion detection vision model for complex scenario conditions. *International Joint Conference on Neural Networks, IJCNN 2013*, August 2013.
43. Y. Chan. Deep learning-based scene-awareness approach for intelligent change detection in videos. *Journal of Electronic Imaging*, 28(1):013038, February 2019.
44. B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham. Reversible Architectures for Arbitrarily Deep Residual Neural Networks. *AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 2811–2818, 2018.
45. T. Chang, T. Ghandi, and M. Trivedi. Vision modules for a multi sensory bridge monitoring approach. *International Conference on Intelligent Transportation Systems, ITSC 2004*, pages 971–976, October 2004.
46. C. Chen, S. Zhang, and C. Du. Learning to detect instantaneous changes with retrospective convolution and static sample synthesis. *Preprint*, 2018.

47. L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution and fully connected CRFs. *arXiv preprint arXiv:1606.00915*, 2016.
48. Y. Chen, C. Chen, C. Huang, and Y. Hung. Efficient hierarchical method for background subtraction. *Pattern Recognition*, 10:40, 2007.
49. Y. Chen, J. Wang, and H. Lu. Learning sharable models for robust background subtraction. *IEEE International Conference on Multimedia and Expo, ICME 2015*, pages 1–6, 2015.
50. Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu. Pixel-wise deep sequence learning for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
51. M. Cheng, L. Xia, Z. Zhu, Y. Cai, Y. Xie, Y. Wang, and H. Yang. Time: A training-in-memory architecture for memristor-based deep neural networks. *ACM/EDAC/IEEE Design Automation Conference, DAC 2017*, pages 1–6, June 2017.
52. Y. Cheng, I. Diakonikolas, D. Kane, and A. Stewart. Robust Learning of Fixed-Structure Bayesian Networks. *NIPS 2018*, 2018.
53. P. Chiranjeevi and S. Sengupta. Interval-valued model level fuzzy aggregation-based background subtraction. *IEEE Transactions on Cybernetics*, 2016.
54. F. Chollet. Keras. *https://github.com/fchollet/keras*, 2015.
55. S. Choo, W. Seo, D. Jeong, and N. Cho. Learning background subtraction by video synthesis and multi-scale recurrent networks. *Asian Conference on Computer Vision, ACCV 2018*, December 2018.
56. S. Choo, W. Seo, D. Jeong, and N. Cho. Multi-scale recurrent encoder-decoder network for dense temporal classification. *IAPR International Conference on Pattern Recognition, ICPR 2018*, pages 103–108, 2018.
57. A. Choromanska, M. Henaff, M. Mathieu, G. Arous, and Y. LeCun. The loss surfaces of multilayer networks. *International Conference on Artificial Intelligence and Statistics*, pages 192–204, 2015.
58. G. Cinar and J. Principe. Adaptive background estimation using an information theoretic cost for hidden state estimation. *International Joint Conference on Neural Networks, IJCNN 2011*, August 2011.
59. L. Pinheiro Cinelli. Anomaly detection in surveillance videos using deep residual networks. *Master Thesis, Universidade de Rio de Janeiro*, February 2017.
60. N. Cohen, O. Sharir, and A. Shashua. Deep SimNets. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, June 2016.
61. N. Cohen, O. Sharir, and A. Shashua. On the Expressive Power of Deep Learning: A Tensor Analysis. *Conference on Learning Theory, COLT 2016*, 2016.
62. N. Cohen and A. Shashua. SimNets: A Generalization of Convolutional Networks. *NIPS workshop on Deep Learning*, December 2014.
63. N. Cohen and A. Shashua. Convolutional Rectifier Networks as Generalized Tensor Decompositions. *International Conference on Machine Learning, ICML 2016*, 2016.
64. N. Cohen, R. Tamari, and A. Shashua. Boosting Dilated Convolutional Networks with Mixed Tensor Decompositions. *International Conference on Learning Representations, ICLR 2018*, April 2018.
65. S. Cohen. Background Estimation as a Labeling Problem. *International Conference on Computer Vision, ICCV 2005*, 2:1034–1041, October 2005.
66. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
67. C. Cuevas and N. Garcia. Tracking-based non-parametric background-foreground classification in a chromaticity-gradient space. *IEEE International Conference on Image Processing, ICIP 2010*, September 2010.
68. C. Cuevas, E. Yaoez, and N. Garcia. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. *Computer Vision and Image Understanding*, 2016.
69. D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht. A neural network approach to Bayesian background modeling for video object segmentation. *International Conference on Computer Vision Theory and Applications, VISAPP 2006*, February 2006.
70. W. Mc Culloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
71. Y. Le Cun, L. Bottou, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86:2278–2324, November 1998.
72. G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals and Systems*, 2(4):303314, 1989.
73. B. Dai, Y. Wang, J. Aston, G. Hua, and D. Wipf. Connections with Robust PCA and the Role of Emergent Sparsity in Variational Autoencoder Models. *Journal of Machine Learning Research*, 19:1–42, 2018.

74. J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *NIPS 2016*, 2016.
75. R. Davies, L. Mihaylova, N. Pavlidis, and I. Eckley. The effect of recovery algorithms on compressive sensing background subtraction. *Workshop Sensor Data Fusion: Trends, Solutions, and Applications*, 2013.
76. J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 2009.
77. Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai. A hierarchical fused fuzzy deep neural network for data classification. *IEEE Transactions on Fuzzy Systems*, 25(4):1006–1012, 2017.
78. C. Doersch. Tutorial on variational autoencoders. *Preprint*, 2016.
79. Y. Dong and G. DeSouza. Adaptive learning of multi-subspace for foreground detection under illumination changes. *Computer Vision and Image Understanding*, 2010.
80. J. Dou, Q. Qin, and Z. Tu. Background subtraction based on deep convolutional neural networks features. *Multimedia Tools and Applications*, pages 1–23, 2018.
81. D. Driggs, S. Becker, and J. Boyd-Graberz. Tensor robust principal component analysis: Better recovery with atomic norm regularization. *Preprint*, January 2019.
82. Y. Du, C. Yuan, W. Hu, and S. Maybank. Spatio-temporal self-organizing map deep network for dynamic object detection from videos. *IEEE International Conference on Computer Vison and Pattern Recognition, CVPR 2017*, June 2017.
83. V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *Preprint*, January 2018.
84. S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, November 2018.
85. A. Elgammal and L. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision, ECCV 2000*, pages 751–767, June 2000.
86. T. Elguebaly and N. Bouguila. Finite asymmetric generalized gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 2013.
87. I. Goodfellow et al. Generative adversarial networks. *NIPS 2014*, 2014.
88. M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *ACM International Conference on Multimedia*, March 2016.
89. S. Caelles et al. One-shot video object segmentation. *CVPR 2017*, 2017.
90. Z. Wang et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nature Electronics*, 1:137–145, 2018.
91. W. Fan and N. Bouguila. Online variational learning of finite Dirichlet mixture models. *Evolving Systems*, January 2012.
92. D. Farcas and T. Bouwmans. Background modeling via a supervised subspace learning. *International Conference on Image, Video Processing and Computer Vision, IVPCV 2010*, 1-7, July 2010.
93. D. Farcas, C. Marghes, and T. Bouwmans. Background subtraction via incremental maximum margin criterion: A discriminative approach. *Machine Vision and Applications*, 23(6):1083–1101, October 2012.
94. A. Farnoosh, B. Rezaei, and S. Ostadabbas. DeepPBM: deep probabilistic background model estimation from video sequences. *Preprint*, February 2019.
95. A. Faro, D. Giordano, and C. Spampinato. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1398–1412, December 2011.
96. S. Feng and C. Chen. A Fuzzy Restricted Boltzmann Machine: Novel Learning Algorithms Based on the Crisp Possibilistic Mean Value of Fuzzy Numbers. *IEEE Transactions on Fuzzy Systems*, 26(1):117–130, 2018.
97. E. Fernandez-Sanchez, L. Rubio, J. Diaz, and E. Ros. Background subtraction model based on color and depth cues. *Machine Vision and Applications*, 2014.
98. A. Fischer and C. Igel. An Introduction to Restricted Boltzmann Machines. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 7441:14–36, September 2012.
99. P. Fischer, A. Dosovitskiy, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.
100. Y. Gao, H. Cai, X. Zhang, L. Lan, and Z. Luo. Background subtraction via 3d convolutional neural networks. *IAPR International Conference on Pattern Recognition, ICPR 2018*, pages 1271–1276, 2018.

101. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *Preprint*, 2017.

102. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

103. J. Garcia-Gonzalez, J. Ortiz de Lazcano-Lobato, R. Luque-Baena, and M. Molina-Cabello. Background modeling for video sequences by stacked denoising autoencoders. *Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2018*, pages 341–350, September 2018.

104. J. Gast and S. Roth. Lightweight probabilistic deep networks. *Preprint*, 2018.

105. G. Gemignani and A. Rozza. A novel background subtraction approach based on multi-layered self organizing maps. *IEEE International Conference on Image Processing*, 2015.

106. P. Gil-Jimenez, S. Maldonado-Bascon, R. Gil-Pita, and H. Gomez-Moreno. Background pixel classification for motion detection in video image sequences. *International Work Conference on Artificial and Natural Neural Network, IWANN 2003*, 2686:718–725, 2003.

107. R. Girshick. Fast R-CNN. *IEEE International Conference on Computer Vision, ICCV 2015*, page 14401448, 2015.

108. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, page 580587, 2014.

109. R. Giryes, G. Sapiro, and A. Bronstein. On the stability of deep networks. *International Conference on Machine Learning, ICLR 2015*, 2015.

110. V. Goel, J. Weng, and P. Poupart. Unsupervised Video Object Segmentation for Deep Reinforcement Learning. *Preprint*, 2018.

111. M. Gong and L. Cheng. Incorporating estimated motion in real-time background subtraction. *IEEE International Conference on Image Processing, ICIP 2011*, pages 3265–3268, September 2011.

112. N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. *IEEE Workshop on Change Detection, CDW 2012 in conjunction with CVPR 2012*, June 2012.

113. P. Graszka. Median mixture model for background-foreground segmentation in video sequences. *Conference on Computer Graphics, Visualization and Computer Vision, WSCG 2014*, 2014.

114. A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

115. M. Gregorio and M. Giordano. Background modeling by weightless neural networks. *SBMI 2015 Workshop in conjunction with ICIAP 2015*, September 2015.

116. M. Gregorio and M. Giordano. CwisarDH+: Background detection in RGBD videos by learning of weightless neural networks. *ICIAP 2017*, pages 242–253, 2017.

117. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, and X. Wang. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

118. K. Gunale and P. Mukherji. Deep Learning with a Spatiotemporal Descriptor of Appearance and Motion Estimation for Video Anomaly Detection. *MDPI Journal of Imaging*, 4(6):79, 2018.

119. H. Guo, C. Qiu, and N. Vaswani. Practical ReProCS for separating sparse and low-dimensional signal sequences from their sum. *Preprint*, October 2013.

120. L. Guo and M. Du. Student's t-distribution mixture background model for efficient object detection. *IEEE International Conference on Signal Processing, Communication and Computing, IC-SPCC 2012*, pages 410–414, August 2012.

121. R. Guo and H. Qi. Partially-sparse restricted Boltzmann machine for background modeling and subtraction. *International Conference on Machine Learning and Applications, ICMLA 2013*, pages 209–214, December 2013.

122. X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma. Robust foreground detection using smoothness and arbitrariness constraints. *European Conference on Computer Vision, ECCV 2014*, September 2014.

123. Y. Guo, Y. Liu, T. Georgiou, and M. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 2017.

124. C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection based on low-rank and block-sparse matrix decomposition. *IEEE International Conference on Image Processing, ICIP 2012*, September 2012.

125. C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection by robust PCA solved via a linearized alternating direction method. *International Conference on Image Analysis and Recognition, ICIAR 2012*, June 2012.
126. C. Guyon, T. Bouwmans, and E. Zahzah. Moving object detection by robust PCA solved via a linearized symmetric alternating direction method. *International Symposium on Visual Computing, ISVC 2012*, July 2012.
127. C. Guyon, T. Bouwmans, and E. Zahzah. Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis. *INTECH, Principal Component Analysis, Book 1, Chapter 12*, pages 223–238, March 2012.
128. E. Haber and L. Ruthotto. Stable Architectures for Deep Neural Networks. *Inverse Problems*, 34(1):014004, 2017.
129. B. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond. *Preprint*, 2015.
130. B. Haeffele and R. Vidal. Global optimality in neural network training. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
131. T. Haines and T. Xiang. Background subtraction with Dirichlet processes. *European Conference on Computer Vision, ECCV 2012*, October 2012.
132. I. Halfaoui, F. Bouzaraa, and O. Urfalioglu. CNN-Based Initial Background Estimation. *Scene Background Modeling Contest in conjunction with ICPR 2016*, 2016.
133. R. Hasan, T. Taha, and C. Yakopcic. On-chip training of memristor based deep neural networks. *International Joint Conference on Neural Networks, IJCNN 2017*, pages 3527–3534, May 2017.
134. J. He, L. Balzano, and J. Luiz. Online robust subspace tracking from partial information. *IT 2011*, September 2011.
135. J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. *International on Conference on Computer Vision and Pattern Recognition, CVPR 2012*, June 2012.
136. K. He, X. Zhang, and S. Ren. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, June 2016.
137. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision, ICCV 2015*, page 10261034, 2015.
138. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision, ICCV 2015*, page 10261034, 2015.
139. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778, 2016.
140. M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 2006*, 28(4):657–62, 2006.
141. G. Hinton. Deep belief nets. *NIPS Tutorial*, 2007.
142. G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.
143. S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
144. M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. *IEEE Workshop on Change Detection, CVPR 2012*, June 2012.
145. K. Hornik. Approximation capabilities of multilayer feedforwardnetworks. *Neural networks*, 4(2):251–257, 1991.
146. K. Hornik, M. Stinchcombe, and H. Whit. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
147. Y. Hu, J. Huang, and A. Schwing. MaskRNN: Instance Level Video Object Segmentation. *NIPS 2017*, 2017.
148. Z. Hu, T. Turki, N. Phan, and J. Wang. 3D Atrous Convolutional Long Short-Term Memory Network for Background Subtraction. *IEEE Access*, 2018.
149. J. Huang, X. Huang, and D. Metaxas. Learning with dynamic group sparsity. *International Conference on Computer Vision, ICCV 2009*, October 2009.
150. T. Huynh. Deep neural network accelerator based on FPGA. *NAFOSTED 2017*, pages 254–257, 2017.

151. P. Isola, J. Zhu, T. Zhou, and A. Efros. Image to- image translation with conditional adversarial networks. *Preprint*, 2017.
152. W. Jang and C. Kim. Online video object segmentation via convolutional trident network. *CVPR 2017*, 2017.
153. S. Javed, T. Bouwmans, and S. Jung. Combining ARF and OR-PCA background subtraction of noisy videos. *International Conference in Image Analysis and Applications, ICIAP 2015*, September 2015.
154. S. Javed, T. Bouwmans, and S. Jung. Depth Extended Online RPCA with Spatiotemporal Constraints for Robust Background Subtraction. *Korea-Japan Workshop on Frontiers of Computer Vision, FCV 2015*, January 2015.
155. S. Javed, T. Bouwmans, and S. Jung. Stochastic decomposition into low rank and sparse tensor for robust background subtraction. *ICDP 2015*, July 2015.
156. S. Javed, T. Bouwmans, and S. Jung. SBMI-LTD: Stationary Background Model Initialization based on Low-rank Tensor Decomposition. *ACM Symposium on Applied Computing, SAC 2017*, 2017.
157. S. Javed, T. Bouwmans, M. Sultana, and S. Jung. Moving Object Detection on RGB-D Videos using Graph Regularized Spatiotemporal RPCA. *ICIAP 2017*, September 2017.
158. S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Motion-Aware Graph Regularized RPCA for Background Modeling of Complex Scenes. *International Conference on Pattern Recognition, ICPR 2016*, 2016.
159. S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Spatiotemporal Low-rank Modeling for Complex Scene Background Initialization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.
160. S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Background-Foreground Modeling Based on Spatio-temporal Sparse Subspace Clustering. *IEEE Transactions on Image Processing*, 26(12):5840–5854, December 2017.
161. S. Javed, A. Mahmood, T. Bouwmans, and S. Jung. Superpixels based Manifold Structured Sparse RPCA for Moving Object Detectio. *International Workshop on Activity Monitoring by Multiple Distributed Sensing, BMVC 2017*, September 2017.
162. S. Javed, S. Oh, T. Bouwmans, and S. Jung. Robust background subtraction to global illumination changes via multiple features based OR-PCA with MRF. *Journal of Electronic Imaging*, 2015.
163. S. Javed, S. Oh, A. Sobral, T. Bouwmans, and S. Jung. Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints. *Workshop on Robust Subspace Learning and Computer Vision, ICCV 2015*, December 2015.
164. S. Javed, A. Sobral, T. Bouwmans, and S. Jung. OR-PCA with dynamic feature selection for robust background subtraction. *ACM Symposium On Applied Computing, SAC 2015,*, 2015.
165. S. Javed, A. Sobral, S. Oh, T. Bouwmans, and S. Jung. OR-PCA with MRF for Robust Foreground Detection in Highly Dynamic Backgrounds. *Asian Conference on Computer Vision, ACCV 2014*, November 2014.
166. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *ACM International Conference on Multimedia*, pages 675–678, 2014.
167. W. Jiang, H. Gao, F. Chung, and H. Huang. The $l_{2,1}$-Norm Stacked Robust Autoencoders for Domain Adaptation. *AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.
168. P. Jodoin. Motion detection: Unsolved issues and [potential] solutions. *Invited Talk, SBMI 2015 in conjunction with ICIAP 2015*, September 2015.
169. P. Jodoin, L. Maddalena, A. Petrosino, and Y. Wang. Extensive Benchmark and Survey of Modeling Methods for Scene Background Initialization. *IEEE Transactions on Image Processing*, 26(11):5244–5256, November 2017.
170. O. Karadag and O. Erdas. Evaluation of the robustness of deep features on the change detection problem. *IEEE Signal Processing and Communications Applications Conference, SIU 2018*, pages 1–4, 2018.
171. K. Kawaguchi. Deep learning without poor local minima. *NIPS 2016*, 2016.
172. J. Kim, A. Rivera, B. Kim, K. Roy, and O. Chae. Background modeling using adaptive properties of hybrid features. *International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2017*, 2017.
173. D. Kingma and M. Welling. Auto-Encoding Variational Bayes. *Preprint*, 2013.
174. O. Krestinskaya, K. Salama, and A. James. Analog back propagation learning circuits for memristive crossbar neural networks. *IEEE International Symposium on Circuits and Systems, ISCAS 2018*, 2018.

175. O. Krestinskaya, K. Salama, and A. James. Learning in memristive neural network architectures using analog backpropagation circuits. *Preprint*, 2018.
176. A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet: Classification with Deep Convolutional Neural Networks. *International Conference on Neural Information Processing Systems, NIPS 2012*, pages 1097–1105, 2012.
177. A. Lanza, F. Tombari, and L. Di Stefano. Accurate and efficient background subtraction by monotonic second-degree polynomial fitting. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, 2010.
178. B. Laugraud, S. Pierard, and M. Van Droogenbroeck. LaBGen-P: A pixel-level stationary background generation method based on LaBGen. *Scene Background Modeling Contest in conjunction with ICPR 2016*, 2016.
179. B. Laugraud, S. Pierard, and M. Van Droogenbroeck. A method based on motion detection for generating the background of a scene. *Pattern Recognition Letters*, 2017.
180. B. Laugraud, S. Pierard, and M. Van Droogenbroeck. LaBGen-P-Semantic: A First Step for Leveraging Semantic Segmentation in Background Generation. *MDPI Journal of Imaging*, 4(7), 2018.
181. D. Le and T. Pham. Encoder-decoder convolutional neural network for change detection. *CITA 2018*, 2018.
182. B. Lee and M. Hedley. Background estimation for video surveillance. *Image and Vision Computing New Zealand, IVCNZ 2002*, pages 315–320, 2002.
183. S. Lee and D. Kim. Background subtraction using the factored 3-way restricted boltzmann machines. *Preprint*, 2018.
184. D. Li, M. Jiang, Y. Fang, Y. Huang, and C. Zhao. Deep video foreground target extraction with complex scenes. *IEEE International Conference on Sensor Networks and Signal Processing, SNSP 2018*, pages 440–445, 2018.
185. L. Li and W. Huang. Statistical modeling of complex background for foreground object detection. *IEEE Transaction on Image Processing*, 13(11):1459–1472, November 2004.
186. X. Li, M. Ye, Y. Liu, and C. Zhu. Adaptive deep convolutional neural networks for scene-specific object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, September 2017.
187. D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, and X. Zhao. Co-occurrence Probability based Pixel Pairs Background Model for Robust Object Detection in Dynamic Scenes. *Pattern Recognition*, 48(4):1374–1390, 2015.
188. D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, X. Zhao, and Y. Satoh. Co-occurrence-based adaptive background model for robust object detection. *International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2013*, September 2013.
189. D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, X. Zhao, and Y. Satoh. Robust object detection in severe imaging conditions using co-occurrence background model. *International Journal of Optomechatronics*, pages 14–29, April 2014.
190. X. Liang, S. Liao, X. Wang, W. Liu, Y. Chen, and S. Li. Deep background subtraction with guided learning. *IEEE International Conference on Multimedia and Expo, ICME 2018*, July 2018.
191. J. Liao, G. Guo, Y. Yan, and H. Wang. Multiscale cascaded scene-specific convolutional neural networks for background subtraction. *Pacific Rim Conference on Multimedia, PCM 2018*, pages 524–533, 2018.
192. K. Lim, W. Jang, and C. Kim. Background subtraction using encoder-decoder structured convolutional neural network. *IEEE International Conference on Advanced Video and Signal based Surveillance, AVSS 2017*, 2017.
193. L. Lim and H. Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *Preprint*, January 2018.
194. L. Lim and H. Keles. Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, 112:256–262, 2018.
195. L. Lim, l. Ang, and H. Keles. Learning multi-scale features for foreground segmentation. *Preprint*, September 2018.
196. C. Lin, B. Yan, and W. Tan. Foreground detection in surveillance video with fully convolutional semantic network. *IEEE International Conference on Image Processing, ICIP 2018*, pages 4118–4122, October 2018.
197. H. Lin, T. Liu, and J. Chuang. A probabilistic SVM approach for background scene initialization. *International Conference on Image Processing, ICIP 2002*, 3:893–896, September 2002.
198. W. Liu, Z. Wang, X.Liu, N. Zeng, Y. Liuc, and F. Alsaadid. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, April 2017.

199. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR 2015*, pages 3431–3440, 2015.
200. F. Lopez-Rubio, E. Lopez-Rubio, R. Luque-Baena, E. Dominguez, and E. Palomo. Color space selection for self-organizing map based foreground detection in video sequences. *International Joint Conference on Neural Networks, IJCNN 2014*, pages 3347–3354, July 2014.
201. C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
202. C. Lu, J. Shi, and J. Jia. Online robust dictionary learning. *EURASIP Journal on Image and Video Processing, IVP 2011*, January 2011.
203. L. Maddalena and A. Petrosino. A self-organizing approach to detection of moving patterns for real-time applications. *Advances in Brain, Vision, and Artificial Intelligence*, 4729:181–190, 2007.
204. L. Maddalena and A. Petrosino. Neural model-based segmentation of image motion. *KES 2008*, pages 57–64, 2008.
205. L. Maddalena and A. Petrosino. A self organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, July 2008.
206. L. Maddalena and A. Petrosino. A self-organizing neural system for background and foreground modeling. *International Conference on Artificial Neural Networks, ICANN 2008*, pages 652–661, 2008.
207. L. Maddalena and A. Petrosino. 3D neural model-based stopped object detection. *International Conference on Image Analysis and Processing, ICIAP 2009*, pages 585–593, 2009.
208. L. Maddalena and A. Petrosino. Multivalued background/foreground separation for moving object detection. *International Workshop on Fuzzy Logic and Applications, WILF 2009*, pages 263–270, June 2009.
209. L. Maddalena and A. Petrosino. Self organizing and fuzzy modelling for parked vehicles detection. *Advanced Concepts for Intelligent Vision Systems, ACVIS 2009*, pages 422–433, 2009.
210. L. Maddalena and A. Petrosino. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Computing and Applications, NCA 2010*, pages 1–8, March 2010.
211. L. Maddalena and A. Petrosino. The SOBS algorithm: What are the limits? *IEEE Workshop on Change Detection, CVPR 2012*, June 2012.
212. L. Maddalena and A. Petrosino. Stopped object detection by learning foreground model in videos. *IEEE Transactions on Neural Networks and Learning Systems*, 24(5):723–735, May 2013.
213. L. Maddalena and A. Petrosino. The 3dSOBS+ algorithm for moving object detection. *Computer Vision and Image Understanding, CVIU 2014*, 122:65–73, May 2014.
214. L. Maddalena and A. Petrosino. Background Model Initialization for Static Cameras. *Handbook on Background Modeling and Foreground Detection for Video Surveillance, CRC Press*, July 2014.
215. L. Maddalena and A. Petrosino. Towards benchmarking scene background initialization. *New Trends in Image Analysis and Processing, SBMI 2015 with ICIAP 2015*, pages 469–476, September 2015.
216. L. Maddalena and A. Petrosino. Extracting a Background Image by a Multi-modal Scene Background Model. *Scene Background Modeling workshop, ICPR 2016*, 2016.
217. L. Maddalena and A. Petrosino. Exploiting Color and Depth for Background Subtraction. *ICIAP 2017*, pages 254–265, September 2017.
218. L. Maddalena and A. Petrosino. Background subtraction for moving object detection in rgb-d data: A survey. *MDPI Journal of Imaging*, 2018.
219. L. Maddalena and A. Petrosino. Self-organizing background subtraction using color and depth data. *Multimedia Tools and Applications*, October 2018.
220. S. Malladi and I. Sharapov. FastNorm: Improving Numerical Stability of Deep Network Training with Efficient Normalization. *International Conference on Machine Learning, ICLR 2018*, 2018.
221. C. Marghes and T. Bouwmans. Background modeling via incremental maximum margin criterion. *International Workshop on Subspace Methods, ACCV 2010 Workshop Subspace 2010*, November 2010.
222. C. Marghes, T. Bouwmans, and R. Vasiu. Background modeling and foreground detection via a reconstructive and discriminative subspace learning approach. *International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2012*, July 2012.
223. Y. Mehran and T. Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:1257–117, May 2018.
224. S. Messelodi, C. Modena, N. Segata, and M. Zanin. A Kalman filter based background updating algorithm robust to sharp illumination changes. *International Conference on Image Analysis and Processing, ICIAP 2005*, 3617:163–170, September 2005.

225. T. Minematsu, A. Shimada, and R. Taniguchi. Analytics of deep neural network in change detection. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*, September 2017.
226. T. Minematsu, A. Shimada, H. Uchiyama, and R. Taniguchi. Analytics of deep neural network-based background subtraction. *MDPI Journal of Imaging*, MDPI 2018.
227. A. Mittal. Motion-based background subtraction using adaptive kernel density estimation. *International Conference on Computer Vision and Pattern Recognition, CVPR 2004*, July 2004.
228. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, July 2017.
229. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto. Analysis of universal adversarial perturbations. *Preprint*, 2017.
230. K. Mopuri, U. Garg, and R. Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *British Machine Vision Conference, BMVC 2017*, 2017.
231. K. Mopuri, U. Ojha, U. Garg, and R. Babu. NAG: Network for Adversary Generation. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 742–751, 2018.
232. G. Moya-Alcover, A. Elgammal, A. Jaume i Capo, and J. Varona. Modelling depth for nonparametric foreground segmentation using RGBD devices. *Pattern Recognition Letters*, 2016.
233. D. Mukherjee and J. Wu. Real-time video segmentation using Student's t mixture model. *International Conference on Ambient Systems, Networks and Technologies, ANT 2012*, pages 153–160, 2012.
234. O. Munteanu, T. Bouwmans, E. Zahzah, and R. Vasiu. The detection of moving objects in video by background subtraction using Dempster-Shafer theory. *Transactions on Electronics and Communications*, 60(1), March 2015.
235. V. Nair and G. Hinton. Rectified linear units improve restricted Boltzmann machines. *International Conference on Machine Learning, ICML 2010*, 2010.
236. P. Narayanamurthy and N. Vaswani. A Fast and Memory-efficient Algorithm for Robust PCA (MEROP). *IEEE International Conference on Acoustics, Speech, and Signal, ICASSP 2018*, April 2018.
237. E. Newman, L. Horesh, H. Avron, and M. Kilmer. Stable Tensor Neural Networks for Rapid Deep Learning. *Preprint*, 2018.
238. T. Nguyen, C. Pham, S. Ha, and J. Jeon. Change detection by training a triplet network for motion feature extraction. *IEEE Transactions on Circuits and Systems for Video Technology*, January 2018.
239. E. Nishani and B. Cico. Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation. *Mediterranean Conference on Embedded Computing, MECO 2017*, pages 1–4, 2017.
240. M. Nouiehed and M. Razaviyay. Learning deep models: Critical points and local openness. *Preprint*, 2018.
241. N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. *ICVS 1999*, January 1999.
242. O. Oreifej, X. Li, and M. Shah. Simultaneous video stabilization and moving object detection in turbulence. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 2012*, 2012.
243. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016.
244. P. Patil and S. Murala. MSFgNet: A Novel Compact End-to-End Deep Network for Moving Object Detection. *IEEE Transactions on Intelligent Transportation Systems*, December 2018.
245. P. Patil, S. Murala, A. Dhall, and S. Chaudhary. MsEDNet: Multi-Scale Deep Saliency Learning for Moving Object Detection. *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018*, pages 1670–1675, 2018.
246. P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, December 2018.
247. S. Prativadibhayankaram, H. Luong, T. Le, and A. Kaup. Compressive online video background-foreground separation using multiple prior information and optical flow. *MDPI Journal of Imaging*, 2018.
248. J. Pulgarin-Giraldo, A. Alvarez-Meza, D. Insuasti-Ceballos, T. Bouwmans, and G. Castellanos-Dominguez. GMM Background Modeling using Divergence-based Weight Updating. *Conference Ibero-american Congress on Pattern Recognition, CIARP 2016*, 2016.

249. Z. Qu, S. Yu, and M. Fu. Motion background modeling based on context-encoder. *IEEE International Conference on Artificial Intelligence and Pattern Recognition, ICAIPR 2016*, September 2016.
250. S. Chawla R. Chalapathy, A. Menon. Robust, Deep and Inductive Anomaly Detection. *Preprint*, 2017.
251. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Preprint*, 2015.
252. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*, 2015.
253. A. Rafique, A.Sheri, and M. Jeon. Background scene modeling for PTZ cameras using RBM. *International Conference on Control, Automation and Information Sciences, ICCAIS 2014*, pages 165–169, 2014.
254. G. Ramirez-Alonso and M. Chacon-Murguia. Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recognition*, April 2015.
255. G. Ramirez-Alonso, J. Ramirez-Quintana, and M. Chacon-Murguia. Temporal weighted learning model for background estimation with an automatic re-initialization stage and adaptive parameters update. *Pattern Recognition Letters*, 2017.
256. J. Ramirez-Quintana and M. Chacon-Murguia. Self-organizing retinotopic maps applied to background modeling for dynamic object segmentation in video sequences. *International Joint Conference on Neural Networks, IJCNN 2013*, August 2013.
257. M. Ranzato, A. Krizhevsky, and G. Hinton. Factored 3-way restricted Boltzmann machines for modeling natural images. *AISTATS 2010*, 2010.
258. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2017.
259. B. Rezaei and S. Ostadabbas. Background Subtraction via Fast Robust Matrix Completion. *International Workshop on RSL-CV in conjunction with ICCV 2017*, October 2017.
260. B. Rezaei and S. Ostadabbas. Moving Object Detection through Robust Matrix Completion Augmented with Objectness. *IEEE Journal of Selected Topics in Signal Processing*, December 2018.
261. P. Rodriguez and B. Wohlberg. Translational and rotational jitter invariant incremental principal-component pursuit for video background modeling. *IEEE International Conference on Image Processing, ICIP 2015*, 2015.
262. P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 55(1):1–18, 2016.
263. O. Ronneberger and and T. Brox. P. Fischer. U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.
264. J. Rosell-Ortega, G. Andreu, V. Atienza, and F. Lopez-Garcia. Background modeling with motion criterion and multi-modal support. *International Conference on Computer Vision Theory and Applications, VISAPP 2010*, May 2010.
265. J. Rosell-Ortega, G. Andreu-Garcia, A. Rodas-Jorda, and V. Atienza-Vanacloig. Background Modelling in Demanding Situations with Confidence Measure. *IAPR International Conference on Pattern Recognition, ICPR 2008*, December 2008.
266. F. Rosenblatt. The perceptron–a perceiving and recognizing automaton. *Report 85-460-1, Cornell Aeronautical Laboratory*, 1957.
267. S. Roy and A. Ghosh. Real-time adaptive histogram min-max bucket (hmmb) model for background subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
268. H. Sajid and S. Cheung. Universal multimode background subtraction. *IEEE Transactions on Image Processing*, 26(7):3249–3260, May 2017.
269. D. Sakkos, E. Ho, and H. Shum. Illumination-aware multi-task gans for foreground segmentation. *IEEE Access*, 2018.
270. D. Sakkos, H. Liu, J. Han, and L. Shao. End-to-end video background subtraction with 3D convolutional neural networks. *Multimedia Tools and Applications*, pages 1–19, December 2017.
271. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. *NIPS 2016*, 2016.
272. J. Sasikumar. Investigating the Application of Deep Convolutional Neural Networks in Semi-supervised Video Object Segmentation. *Master Science Thesis, Technological University Dublin*, 2018.

273. J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, pages 85–117, January 2015.
274. A. Schofield, P. Mehta, and T. Stonham. A system for counting people in video images using neural networks to identify the background scene. *Pattern Recognition*, 29:1421–1428, 1996.
275. M. Sedky, M. Moniri, and C. Chibelushi. Spectral-360: A Physics-Based Technique for Change Detection. *IEEE Change Detection Workshop, CDW 2014*, June 2014.
276. B. Sengupta and K. Friston. How Robust are Deep Neural Networks? *Preprint*, 2018.
277. M. Shafiee, P. Siva, P. Fieguth, and A. Wong. Embedded motion detection via neural response mixture background modeling. *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2016*, June 2016.
278. M. Shafiee, P. Siva, P. Fieguth, and A. Wong. Real-time embedded motion detection via neural response mixture modeling. *Journal of Signal Processing Systems*, June 2017.
279. M. Shafiee, P. Siva, and A. Wong. Stochasticnet: Forming deep neural networks via stochastic connectivity. *IEEE Access*, 2016.
280. A. Shahbaz, D. Hernandez, and K.Jo. Optimal color space based probabilistic foreground detector for video surveillance systems. *IEEE International Symposium on Industrial Electronics, ISIE 2017*, pages 1637–1641, 2017.
281. M. Shakeri and H. Zhang. Moving object detection in time-lapse or motion trigger image sequences using low-rank and invariant sparse decomposition. *IEEE International Conference on Computer Vision, ICCV 2017*, pages 5133–5141, 2017.
282. L. Sharma and N. Lohan. Performance analysis of moving object detection using bgs techniques in visual surveillance. *International Journal of Spatio-Temporal Data Science, Inderscience*, 1(1):22–53, 2019.
283. A. Sheri, M. Rafique, M. Jeon, and W. Pedrycz. Background subtraction using Gaussian−Bernoulli restricted Boltzmann machine. *IET Image Processing*, 2018.
284. C. Silva, T. Bouwmans, and C. Frelicot. An eXtended center-symmetric local binary pattern for background modeling and subtraction in videos. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015*, March 2015.
285. C. Silva, T. Bouwmans, and C. Frelicot. Online weighted one-class ensemble for feature selection in background/foreground separation. *International Conference on Pattern Recognition, ICPR 2016*, December 2016.
286. C. Silva, T. Bouwmans, and C. Frelicot. Superpixel-based online wagging one-class ensemble for feature selection in background/foreground separation. *Pattern Recognition Letters*, 2017.
287. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representation, ICLR 2015*, 2015.
288. A. Sobral, T. Bouwmans, and E. Zahzah. Comparison of Matrix Completion Algorithms for Background Initialization in Videos. *ICIAP 2015*, 2015.
289. A. Sobral, T. Bouwmans, and E. Zahzah. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. *ISBC 2015 Workshop conjunction with AVSS 2015*, 2015.
290. A. Sobral, S. Javed, S. Jung, T. Bouwmans, and E. Zahzah. Online stochastic tensor decomposition for background subtraction in multispectral video sequences. *Workshop on Robust Subspace Learning and Computer Vision, ICCV 2015*, 2015.
291. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, June 2014.
292. P. St-Charles, G. Bilodeau, and R. Bergevin. Flexible background subtraction with self-balanced local sensitivity. *IEEE Change Detection Workshop, CDW 2014*, June 2014.
293. P. St-Charles, G. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. *IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015.
294. C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1999*, pages 246–252, 1999.
295. M. Sultana and S. Jung. Illumination Invariant Foreground Object Segmentation using ForeGAN. *Preprint*, February 2019.
296. M. Sultana, A. Mahmood, S. Javed, and S. Jung. Unsupervised deep context prediction for background estimation and foreground segmentation. *Machine Vision and Applications*, October 2018.
297. M. Sultana, A. Mahmood, S. Javed, and S. Jung. Unsupervised RGBD Video Object Segmentation using GANs. *ACCV-Workshops 2018*, December 2018.

298. J. Suykens. Deep restricted kernel machines using conjugate feature duality. *Neural Computation*, 29:2123–2163, 2017.
299. C. Szegedy, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, June 2016.
300. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 1–9, 2015.
301. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014.
302. K. Tan, J. Chen, and D. Wang. Gated residual networks with dilated convolutions for monaural speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:189–198, 2019.
303. Y. Tao, P. Palasek, Z. Ling, and I. Patras. Background modelling based on generative Unet. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*, September 2017.
304. A. Tavakkoli. Foreground-background segmentation in video sequences using neural networks. *Intelligent Systems: Neural Networks and Applications*, May 2005.
305. A. Tavakkoli, A. Ambardekar, M. Nicolescu, and S. Louis. A genetic approach to training support vector data descriptors for background modeling in video data. *International Symposium on Visual Computing, ISVC 2007*, November 2007.
306. A. Tavakkoli, M. Nicolescu, and G. Bebis. Novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. *International Symposium on Visual Computing, ISVC 2006*, pages 40–49, November 2006.
307. A. Tavakkoli, M. Nicolescu, M. Nicolescu, and G. Bebis. Incremental svdd training: Improving efficiency of background modeling in videos. *International Conference on Signal and Image Processing, ICSIP 2008*, August 2008.
308. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh. Robustness of conditional GANs to noisy labels. *NIPS 2018*, 2018.
309. F. Tombari, A. Lanza, L. Di Stefano, and S. Mattoccia. Non-linear Parametric Bayesian Regression for Robust Background Subtraction. *IEEE Workshop on Motion and Video Computing, MOTION 2009*, December 2009.
310. K. Toyama, J. Krumm, B. Brumiit, and B. Meyers. Wallflower: Principles and practice of background maintenance. *International Conference on Computer Vision, ICCV 1999*, pages 255–261, September 1999.
311. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *IEEE International Conference on Computer Vision, ICCV 2015*, 2015.
312. I. Ullah and A. Petrosino. About pyramid structure in convolutional neural networks. *Preprint*, 2018.
313. A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for foreground/background extraction. *International Workshop on Background Models Challenge, ACCV 2012*, November 2012.
314. S. Varadarajan, P. Miller, and H. Zhou. Spatial mixture of Gaussians for dynamic background modelling. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2013*, pages 63–68, 2013.
315. N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurth. Robust PCA and Robust Subspace Tracking: A Comparative Evaluation. *Statistical Signal Processing Workshop, SSP 2018*, June 2018.
316. N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurth. Robust Subspace Learning: Robust PCA, Robust Subspace Tracking and Robust Subspace Recovery. *IEEE Signal Processing Magazine*, 35(4):32–55, July 2018.
317. A. Vedaldi and K. Lenc. MatConvNet: Convolutional Neural Networks for MATLAB. *http://www.vlfeat.org/matconvnet/matconvnet-manual.pdf*, 2016.
318. R. Vidal. Mathematics of deep learning. *Seminar, Univ. La Rochelle*, 2017.
319. R. Vidal, J. Bruna, R. Giryes, and S. Soatto. Mathematics of deep learning. *Preprint*, 2018.
320. L. Vosters, C. Shan, and T. Gritti. Real-time robust background subtraction under rapidly changing illumination conditions. *Image Vision and Computing*, 30(12):10041015, 2012.
321. D. Wang and J. Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26:1702–1726, 2018.
322. F. Wang, H. Liu, and J. Cheng. Visualizing deep neural network by alternately image blurring and deblurring. *Neural Networks*, 97:162–172, January 2018.

323. H. Wang, Y. Lai, W. Cheng, C. Cheng, and K. Hua. Background Extraction Based on Joint Gaussian Conditional Random Fields. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
324. J. Wang, G. Bebis, and R. Miller. Robust video-based surveillance by integrating target detection with tracking. *IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum in conjunction with CVPR 2006*, June 2006.
325. J. Wang, G. Bebis, M. Nicolescu, M. Nicolescu, and R. Miller. Improving target detection by coupling it with tracking. *Machine Vision and Application*, pages 1–19, 2008.
326. M. Wang, W. Li, and X. Wang. Transferring a generic pedestrian detector towards specific scenes. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pages 3274–3281, 2012.
327. Q. Wang, J. Gao, and Y. Yuan. Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):230–241, January 2018.
328. Q. Wang, S. Liu, J. Chanussot, and X. Li. Scene classification with recurrent attention of vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.
329. Q. Wang, J. Wan, and X. Li. Robust Hierarchical Deep Learning for Vehicular Management. *IEEE Transactions on Vehicular Technology*, 2018.
330. Q. Wang, Z. Yuan, Q. Du, and X. Li. GETNET: A General End-to-End 2-D CNN Framework for Hyperspectral Image Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):3–13, January 2019.
331. R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappa. Static and moving object detection using flux tensor with split Gaussian models. *IEEE International Conference on Computer Vision, CVPR 2014*, 2014.
332. W. Wang, Y. Sun, B. Eriksson, W. Duke, and V. Aggarwal. Wide Compression: Tensor Ring Net. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 9329–9338, 2018.
333. X. Wang, L. Liu, G. Li, X. Dong, P. Zhao, and X. Feng. Background subtraction on depth videos with convolutional neural networks. *IEEE International Joint Conference on Neural Networks, IJCNN 2018*, pages 1–7, 2018.
334. X. Wang, X. Ma, and W Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539555, March 2009.
335. Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: an expanded change detection benchmark dataset. *IEEE Workshop on Change Detection, CDW 2014 in conjunction with CVPR 2014*, June 2014.
336. Y. Wang, Z. Luo, and P. Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 2016.
337. Y. Wang, Z. Yu, and L. Zhu. Foreground detection with deeply learned multi-scale spatial-temporal features. *MDPI Sensors*, 2018.
338. Z. Wang and D. Wang. Combining spectral and spatial features for deep learning based blind speaker separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:457–468, 2019.
339. Z. Wang, L. Zhang, and H. Bao. PNN based motion detection with adaptive learning rate. *International Conference on Computational Intelligence and Security, CIS 2009*, pages 301–306, December 2009.
340. Z. Wang, X. Zhang, and D. Wang. Robust speaker localization guided by deep learning-based time-frequency masking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:178–188, 2019.
341. B. Weinstein. Scene-specific convolutional neural networks for video-based biodiversity detection. *Methods in Ecology and Evolution*, 2018.
342. B. Widrow. Generalization and information storage in networks of ADALINE. *Self Organizing Systems*, 1962.
343. B. Widrow and M. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.
344. C. Wren and A. Azarbayejani. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780 –785, July 1997.
345. M. Wu and X. Peng. Spatio-temporal context for codebook-based dynamic background subtraction. *AEU-Int. J. Electron. Commun.*, 64(8):739–747, 2010.

346. Z. Wu, D. Lin, , and X. Tang. Adjustable bounded rectifiers: Towards deep binary representations. *Preprint*, 2015.
347. H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. MoNet: Deep Motion Exploitation for Video Object Segmentation. *CVPR 2018*, pages 1140–1148, 2018.
348. J. Xu, V. Ithapu, L. Mukherjee, J. Rehg, and V. Singh. GOSUS: Grassmannian Online Subspace Updates with Structured-sparsity. *International Conference on Computer Vision, ICCV 2013*, September 2013.
349. L. Xu, Y. Li, Y. Wang, and E. Chen. Temporally adaptive restricted Boltzmann machine for background modeling. *American Association for Artificial Intelligence, AAAI 2015*, January 2015.
350. P. Xu, M. Ye, X. Li, Q. Liu, Y. Yang, and J. Ding. Dynamic background learning through deep auto-encoder networks. *ACM International Conference on Multimedia*, November 2014.
351. P. Xu, M. Ye, Q. Liu, X. Li, L. Pei, and J. Ding. Motion detection via a couple of auto-encoder networks. *International Conference on Multimedia and Expo, ICME 2014*, 2014.
352. Y. Yan, H. Zhao, F. Kao, V. Vargas, S. Zhao, and J. Ren. Deep background subtraction of thermal and visible imagery for pedestrian detection in videos. *International Conference on Brain Inspired Cognitive Systems, BICS 2018*, 2018.
353. J. Yang, J. Yang, X. Yang, and H. Yue. Background recovery from video sequences via online motion-assisted rpca. *Visual Communications and Image Processing, VCIP 2016*, pages 1–4, 2016.
354. L. Yang, H. Cheng, J. Su, and X. Li. Pixel-to-model distance for robust background reconstruction. *IEEE Transactions on Circuits Systems and Video Technology*, 26(5):903–916, May 2016.
355. L. Yang, J. Li, Y. Luo, Y. Zhao, H. Cheng, and J. Li. Deep background modeling using fully convolutional network. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):254262, 2018.
356. G. Yao, T. Lei, and J. Zhong. A review of Convolutional-Neural-Network-based action recognition. *Pattern Recognition Letters*, 118:14–22, 2019.
357. H. Yi, S. Shiyu, D. Xiusheng, and C. Zhigang. A study on deep neural networks framework. *IMCEC 2016*, pages 1519–1522, 2016.
358. J. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmen-tation using convolutional neural networks. *ICCV 2017*, 2017.
359. R. Yu, H. Wang, and L. Davis. ReMotENet: efficient relevant motion event detection for large-scale home surveillance videos. *Preprint*, January 2018.
360. Y. Yuan and and Q. Wang Z. Xiong. ACM: Adaptive Cross-Modal Graph Convolutional Neural Networks for RGB-D Scene Recognition. *AAAI Conference on Artificial Intelligence, AAAI 2019*, 2019.
361. C. Yun, S. Sra, and A.Jadbabaie. A critical view of global optimality in deep learning. *International Conference on Machine Learning Representations, ICLR 2018*, 2018.
362. R. Cheung Z. Xu, B. Min. A Robust Background Initialization Algorithm with Superpixel Motion Detection. *Signal Processing: Image Communication*, 71:1–12, February 2019.
363. D. Zeng, X. Chen, M. Zhu, M. Goesele, and A. Kuijper. Background Subtraction with Real-time Semantic Segmentation. *Preprint*, December 2018.
364. D. Zeng and M. Zhu. Background subtraction using multiscale fully convolutional network. *IEEE Access*, pages 16010–16021, March 2018.
365. D. Zeng and M. Zhu. Combining background subtraction algorithms with convolutional neural network. *Preprint*, 2018.
366. D. Zeng and M. Zhu. Multiscale fully convolutional network for foreground object detection in infrared videos. *IEEE Geoscience and Remote Sensing Letters*, 2018.
367. H. Zhang and D. Xu. Fusing color and gradient features for background model. *International Conference on Signal Processing, ICSP 2006*, 2(7), 2006.
368. H. Zhang and D. Xu. Fusing color and texture features for background model. *International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2006*, 4223(7):887–893, September 2006.
369. Y. Zhang, X. Li, Z. Zhang, F. Wu, and L. Zhao. Deep learning driven blockwise moving object detection with binary scene modeling. *Neurocomputing*, June 2015.
370. Y. Zhang, X. Wang, and E. Friedman. Memristor-based circuit design for multilayer neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(2):677–686, February 2018.
371. C. Zhao, T. Cham, X. Ren, J. Cai, and H. Zhu. Background subtraction based on deep pixel distribution learning. *IEEE International Conference on Multimedia and Expo, ICME 2018*, pages 1–6, 2018.
372. C. Zhao, X. Wang, and W. Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing, IVP 2011*, January 2011.

373. Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki. Tensor ring decomposition. *Preprint*, 2016.
374. X. Zhao, Y. Chen, M. Tang, and J. Wang. Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network. *Preprint*, 2017.
375. Z. Zhao, T. Bouwmans, X. Zhang, and Y. Fang. A Fuzzy Background Modeling Approach for Motion Detection in Dynamic Backgrounds. *International Conference on Multimedia and Signal Processing*, December 2012.
376. Z. Zhao, X. Zhang, and Y. Fang. Stacked multi-layer self-organizing map for background modeling. *IEEE Transactions on Image Processing*, 2015.
377. Z. Zhao, P. Zheng, S. Xu, and X. Wu. Object Detection with Deep Learning: A Review. *Preprint*, July 2018.
378. S. Zheng, Y. Song, T.Leung, and I. Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 4480–4488, 2018.
379. W. Zheng, K. Wang, and F. Wang. Background Subtraction Algorithm based on Bayesian Generative Adversarial Networks. *Acta Automatica Sinica*, 2018.
380. W. Zheng, K. Wang, and F. Wang. A novel background subtraction algorithm based on parallel vision and Bayesian GANs. *Neurocomputing*, 2018.
381. Z. Zheng and P. Hong. Robust Detection of Adversarial Attacks by Modeling the Intrinsic Properties of Deep Neural Networks. *NIPS 2018*, 2018.
382. C. Zhou. Robust Auto-encoders. *PhD Thesis, Worcester Institute, USA*, 2016.
383. C. Zhou and R. Paffenroth. Anomaly Detection with Robust Deep Autoencoders. *KDD 2017*, 2017.
384. T. Zhou and D. Tao. GoDec: randomized low-rank and sparse matrix decomposition in noisy case. *International Conference on Machine Learning, ICML 2011*, 2011.
385. X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:597–610, 2013.
386. T. Zin, P. Tin, T. Toriu, and H. Hama. A new background subtraction method using bivariate Poisson process. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 419–422, August 2014.
387. Z. Zivkovic. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, January 2006.