CDN-MEDAL: Two-stage Density and Difference Approximation Framework for Motion Analysis

Journal:	*IEEE Transactions on Circuits and Systems for Video Technology*
Manuscript ID	TCSVT-05457-2020.R1
Manuscript Type:	Transactions Papers - Special Issue on Advanced Machine Learning Methodologies for Large-Scale Video Object Segmentation and Detection
Date Submitted by the Author:	01-Mar-2021
Complete List of Authors:	Ha, Synh; International University - National University HCM city, School of Computer Science and Engineering Nguyen, Cuong; International University - National University HCM city, School of Computer Science and Engineering Phan, Hung; International University - National University HCM city, School of Computer Science and Engineering Chung, Nhat; International University - National University HCM city, School of Computer Science and Engineering Ha, Phuong; UiT The Arctic University of Norway, Department of Computer Science
EDICS:	Statistical Methods < 1.4.1 ☐ Processing Methodology < 1.4 ☐ Image/Video Processing < 1 ☐ IMAGE/VIDEO PROCESSING, Tensor-Based Methods < 1.4.1 ☐ Processing Methodology < 1.4 ☐ Image/Video Processing < 1 ☐ IMAGE/VIDEO PROCESSING, 2.4 ☐ Motion Analysis < 2 ☐ IMAGE/VIDEO ANALYSIS AND COMPUTER VISION, 2.3.4 ☐ Foreground/Background Segregation < 2.3 ☐ Segmentation < 2 ☐ IMAGE/VIDEO ANALYSIS AND COMPUTER VISION

CDN-MEDAL: Two-stage Density and Difference Approximation Framework for Motion Analysis

Synh Viet-Uyen Ha, *Member, IEEE,* Cuong Tien Nguyen, Hung Ngoc Phan, Nhat Minh Chung, and Phuong Hoai Ha, *Member, IEEE*

***Abstract*—Background modeling is a promising research area in video analysis with a variety of video surveillance applications. Recent years have witnessed the proliferation of deep neural networks via effective learning-based approaches in motion analysis. However, these techniques only provide a limited description of the observed scenes' insufficient properties where a single-valued mapping is learned to approximate the temporal conditional averages of the target background. On the other hand, statistical learning in imagery domains has become one of the most prevalent approaches with high adaptation to dynamic context transformation, notably Gaussian Mixture Models, combined with a foreground extraction step. In this work, we propose a novel, two-stage method of change detection with two convolutional neural networks. The first architecture is grounded on the unsupervised Gaussian mixtures statistical learning to describe the scenes' salient features. The second one implements a light-weight pipeline of foreground detection. Our two-stage framework contains approximately 3.5K parameters in total but still maintains rapid convergence to intricate motion patterns. Our experiments on publicly available datasets show that our proposed networks are not only capable of generalizing regions of moving objects in unseen cases with promising results but also are competitive in performance efficiency and effectiveness regarding foreground segmentation.**

***Index Terms*—background subtraction, background modeling, change detection, Gaussian Mixture Model, video analysis**

I. Introduction

With the swift progress in computer vision, surveillance systems using static cameras are promising technologies for advanced tasks such as behavior analysis [1], object segmentation [2] and motion analysis [3], [4]. Among their various functionalities, background modeling is a pivotal component when it comes to the proper understanding of scene dynamics, thereby enabling extraction of attributes of interest. An ideal background is a scene containing only stationary objects and components that are not of interest to the system (e.g., streets, houses, trees). Thus, by comparing visual inputs with the background, desired objects, called foregrounds (e.g., cars, pedestrians), can be localized for further analysis. As real-life scenarios involve various degrees of dynamics like illumination changes, scene dynamics or bootstrapping, there are many approaches to the construction of the background [5].

One of the most prominent approaches in tackling the background modeling problem employs pixel-based statistical frameworks such as Gaussian Mixture Models (GMM) [6], [7], [8], [9]. These methods are based on the hypothesis that background intensities appear predominantly throughout a scene, thereby constructing usefully explicit mathematical structures for exploitation of the dominance. In addition, an important property of such approaches is their adaptability to changing conditions of real-world scenarios, even under illumination changes (e.g. moving clouds), view noises (e.g. rain, snow drops) and implicit motions (e.g. moving body of water). However, the generalization of the methods' correctness is hindered when the hypothesis fails under appearances of stopped objects or high degrees of view noises (e.g. camera shaking, abrupt view changes), thereby producing corruptive backgrounds that often lead to poor estimations of foregrounds. Furthermore, the statistical schemes still follow the sequential processing paradigm that under-utilize modern parallel processing units in the presence of big data.

On the other hand, riding on the increasing advancement wave of specialized processing units for large-scale data, Deep Neural Networks (DNNs) have emerged as a prominent pattern matching and visual prediction mechanism. Deep learning approaches for the motion detection problem are rapidly demonstrating their effectiveness not only in utilizing tremendous sets of processing cores of modern parallel computing technologies, but also in producing highly accurate predictions from data-learning. However, the typical DNNs' architectures are very computationally expensive if they actually can produce highly accurate results, especially regarding those providing solutions to the problem of background modeling and foreground detection. Furthermore, the DNNs in the literature have experienced two primary shortcomings:

A requirement of a huge-scale dataset of labeled images: DNNs-based models for motion detection exploit weak statistical regularities between input sequences of images and annotated background scenes. Thus, to generalize all practical scenarios in real life, a prohibitively large dataset consisting of all practical scenarios and effects is needed. With few training labels in video sequences for building generalized background models [10], there are currently no universal data-driven experiments to assure that the scenes' true properties are appropriately presented.

A prevailing fail on contextual variation: Recently, foreground segmentation has been considered from the perspective of binary classification schemes. It has been proposed to minimize a sum-of-squares or a cross-entropy error function in DNNs-based approaches to reflect the motion analysis problem's true objective as closely as possible. In this ap-

S. V.-U. Ha, C. T. Nguyen, H. N. Phan and N. M. Chung are with the School of Computer Science and Engineering, International University, Vietnam National University, Ho Chi Minh City, Vietnam.

P. H. Ha is with the Department of Computer Science, UiT The Arctic University of Norway.

Corresponding email: hvusynh@hcmiu.edu.vn

proach, models are usually trained to represent the semantics properties on the training sets when the actual aim is to generalize well to experimental datasets' specific target video sequence. This conditional average will be inadequate for various unseen contextual semantics and dynamics that might occur in real-world [11], [12]. In other words, DNNs-based methods usually perform well on experimental datasets of background modeling and change detection but can still fail on unseen situations in real-world scenarios.

Nevertheless, the DNNs-based approach is particularly promising as the literature has rapidly demonstrated their ability to approximate any functions up to arbitrary accuracy within highly parallelizable architectures. In other words, we can exploit their parallelizable capability to approximate the mechanism behind the optimization of GMM, in a way that boosts the construction of statistical model estimations of our data using modern parallel computing technologies. Hence, it becomes possible to efficiently exploit GMM-based background models' characteristics, which are clear and consistent with their mathematical framework, for functional extension, i.e. tackling stopped objects and high-degreed view shifts via DNNs' common data-driven effectiveness. In this article, to address the issues of DNNs while also utilizing its benefits, we incorporate the mathematics of modeling statistical GMM into our processes, and introduce a novel, light-weighted, dual framework of two convolutional neural networks (CNN): (1) the **C**onvolutional **D**ensity **N**etwork of **G**aussian **M**ixtures (**CDN-GM**) for the task of generalistically modeling backgrounds; and (2) the **M**otion **E**stimation with **D**ifferencing **A**pproximation via **L**earning on a convolutional network (**MEDAL-net**), for context-driven foreground extraction. Specifically, our contributions are actually three-fold, and they are summarized as follows:

Firstly, by leveraging existing technologies and being inspired by Bishop [13], we propose our CDN-GM, a feed-forward, highly parallelizable CNN representing a conditional probability density function that models the temporal history for each pixel location in the first pipeline of the proposed framework. In this architecture, conditioned on pixel-wise vectors of intensity values across a time period, the network approximates a Gaussian-Mixture statistical mapping function to efficiently produce models of their underlying multimodular distributions. Accordingly, at each pixel, the mixture is characterized by the weighted combination of its Gaussian components, where each capture and highlight a context-relevant range of pixel-wise values in the manner of a mean and variance. Thus, from statistical models of data in Gaussian Mixtures at pixel level, backgrounds are extracted from the most informative components, resulting in our compressed, light-weighted and efficient architecture.

Secondly, with the goal of modeling the underlying generator of the data, we propose a loss function in the manner of unsupervised learning. This loss function serves to direct the proposed CDN-GM's architectural parameters into approximating the mathematical structure behind GMM-driven modeling of the data with expectation maximization. Thus, because of this, the resulting inferences will consist of mixtures of Gaussian components describing the data, and the most likely background description of actually observed data can be made, with the trained network being subsequently presented with new values of input. In conjunction with CDN-GM, the proposed background modeling architecture not only achieves higher degrees of interpretability compared to the idea of estimating an implicit hidden function in previous neural network methods, but it also gains better capability of adaptation under contextual dynamics with statistical learning, as it is able to utilize a virtually inexhaustible amount of data for incorporation of expectation maximization into the neural-network parameters.

Thirdly, in the latter pipeline of the proposed framework, we design a compact convolutional auto-encoder for context-driven foreground extraction called MEDAL-net, which simulates a context-driven difference mapping between input frames and their corresponding background scenes. This is greatly encouraged because even though real-life scenarios involve various degrees of contextual variations that yet any existing mathematical framework can completely capture, we can construct consistently GMM-driven background models of those variations with CDN-GM to provide semantic understanding of the scene. Thus, we are able to make good use of information from features in images from the first module of background modeling, and even from features seemingly corruptive to motion extractions (e.g. stopped objects), for formulating foreground extraction from raw inputs, thereby resulting in a very light-weighted and efficient structure with high accuracy. The network is trained in a supervised manner in such a way that it maintains good generalization to various views, and to even unseen situations of similar scenery dynamics.

The organization of this paper is as follows: Section II encapsulates the synthesis of recent approaches in background initialization and foreground segmentation. The proposed method is described in Section III. Experimental evaluations are discussed in Section IV. Finally, our conclusion and motivations towards future works are reached in Section V.

II. Related Works

The new era of video analysis has witnessed a proliferation of methods that concentrate on background modeling and foreground detection. Prior studies in recent decades were encapsulated in various perspectives of feature concepts [11], [14], [15]. Among published methods that meet the requirements of robustness, adaptation to scene dynamics, memory efficiency, and real-time processing, two promising approaches of background subtraction are statistical methods and neural-network-based models. Statistical studies aim to characterize the history of pixels' intensities with a model of probabilistic analysis. On the other hand, neural-network-driven approaches implicitly estimate a mapping between an input sequence of observed scenes and hand-labeled background/foreground images on non-linear regularities.

In statistical approaches, the pixels' visual features are modeled with an explainable probabilistic foundation regarding either pixel-level or region-level in temporal and spatial resolution perspectives. In the last decades, there have been a

variety of statistical models that were proposed to resolve the problem of background initialization. Stauffer and Grimson [6] proposed a pioneering work that handled gradual changes in outdoor scenes using pixel-level GMM with a sequential K-means distribution matching algorithm. To enhance the foreground/background discrimination ability regarding scene dynamics, Pulgarin-Giraldo *et al.* [16] improved GMM with a contextual sensitivity that used a Least Mean Squares formulation to update the parameter estimation framework. Validating the robustness of background modeling in a high amount of dynamic scene changes, Ha *et al.* [17] proposed a GMM with high variation removal module using entropy estimation. To enhance the performance, Lu *et al.* [18] applied a median filter on an input frame to reduce its spatial dimension before initializing its background. To address the sequential bottleneck among statistical methods in pixel-wise learning, an unsupervised, tensor-driven framework of GMM was proposed by Ha *et al.* [9] with balanced trade-off between satisfactory foreground mask and exceptional processing speed. However, the approach's number of parameters requires a lot of manual tuning. In addition to GMM, Cauchy Mixture Models (CMM) was exploited to detect foreground objects via eliminating noise and capturing periodical perturbations in varying lighting conditions and dynamic scenarios [19]. Overall, statistical models were developed with explicit probabilistic hypotheses to sequentially present the correlation of history observation at each image point or a pixel block, added with a global thresholding approach to extract foreground. This global thresholding technique for foreground detection usually leads to a compromise between the segregation of slow-moving objects and rapid adaptation to sudden scene changes within short-term measurement. This trade-off usually damages the image-background subtraction in multi-contextual scenarios, which is considered as a sensitive concern in motion estimation. Hence, regarding foreground segmentation from background modeling, it is critical to improve frame differencing from constructed background scene with a better approximation mechanism, and utilize parallel technologies.

Recently, there have also been many attempts to apply DNNs into background subtraction and background modeling problems with supervised learning. Inspired from LeNet-5 [20] used for handwritten digit recognition, one of the earliest efforts to subtract the background from the input image frame was done by Braham *et al.* [21]. This work explores the potential of visual features learned by hidden layers for foreground-background pixel classification. Similarly, Wang *et al.* [22] proposed a deep CNN trained on only a small subset of frames as there is a large redundancy in a video taken by surveillance systems. The model requires a hand-labeled segmentation of moving regions as an indicator in observed scenes. Lim *et al.* [23] constructed an encoder-decoder architecture with the encoder inherited from VGG-16 [24]. The proposed encoder-decoder network takes a video frame, along its corresponding grayscale background and its previous frame as the network's inputs to compute their latent representations, and to deconvolve these latent features into a foreground binary map. Another method is DeepBS [25] which was proposed by Babaee *et al.* to compute the background model using both SuBSENSE [26] and Flux Tensor method [27]. The authors extract the foreground mask from a small patch from the current video frame and its corresponding background to feed into the CNN, and the mask is later post-processed to give the result. Nguyen *et al.* proposed a motion feature network [28] to exploit motion patterns via encoding motion features from small samples of images. The method's experimental results showed that the network obtained a promising results and well-performed on unseen data sequences. Another method that used a triplet convolutional autoencoder to learn multi-scale hidden representations for motion mask extraction of the observed scenes was proposed as FgSegNet [29]. Recently, there is also a work from Chen *et al.* [30] which aims to exploit high-level spatial-temporal features with a deep pixel-wise attention mechanism and convolutional long short-term memory (ConvLSTM).

All things considered, most neural-network-based methods are benefitted from a significant number of weak statistical regularities in associative mapping, where the aim is to learn a transformation from an input batch of consecutive frames to the target hand-labeled foreground or background. There is little evidence that this supervised learning approach ensures that DNNs possess true properties of observed scenes from the sampling peculiarities of training datasets, and be able to generalize to varying degree of contextual dynamics in the real-world. Furthermore, recent CNN methods do not ensure real-time performance, which is a crucial requirement for any practical systems. However, CNN's ability to utilize the parallelism mechanism of modern hardware very efficiently, and the effective use of data for high-accuracy prediction is appealing to investigate. Therefore, in this work, we propose a scheme of two compact CNN with a couple of strategies. First, grounded on a probabilistic model, the former network models a conditional density function via exploiting temporal information to construct background scenes. Second, the latter CNN-based encoder-decoder aims to approximate frame-background differencing to extract moving regions.

III. The Proposed Method

As shown with an overview in Fig. 1, the primary goal of our proposed framework is to address the previously listed problems of DNNs and statistical methods, via adaptively acquiring the underlying properties of a sequence of images to construct corresponding background scenes with CDN-GM (the left subfigure), and extract foregrounds of interest through data-driven learning with MEDAL-net (the right lower subfigure). Following pixel-wise temporal data reformation (the right upper subfigure), a batch of video frames is decompressed into a sequence of pixel histories to estimate each pixel's true background intensity with CDN-GM. After reconstructing the background image from the output intensity sequence of CDN-GM, the input frame is concatenated along the channel dimension with the background to estimate the final segmentation map. The concatenation before the foreground extraction step provides information to engender context-driven difference mapping within MEDAL-net, rather than memorizing the single-valued mapping between input

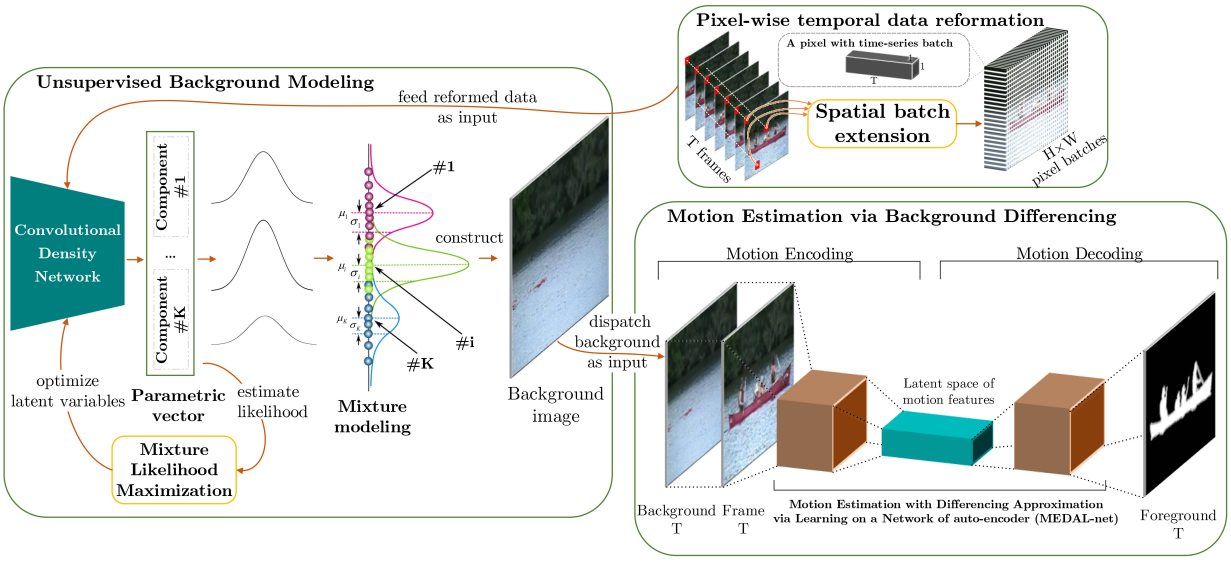


Fig. 1. The overview of the proposed method for background modeling and foreground detection

frames and labeled foregrounds. This difference mapping idea effectively limits MEDAL-net's parameter search space, while enabling our proposed foreground extraction network to be more robust against various real-world motion dynamics.

A. Convolution Density Network of Gaussian Mixture

According to Zivkovic's study [7], let $\boldsymbol{\chi}_c^T = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T | \mathbf{x}_i \in [0, 255]^c\}$ be the time series of the T most recently observed color signals of a pixel where the dimension of the vector $\mathbf{x}_i$ in the color space is c, the distribution of pixel intensity $\mathbf{x}_i$ can be modeled by a linear combination of K probabilistic components $\boldsymbol{\theta}_k$ and their corresponding conditional probability density functions $P(\mathbf{x}_i|\boldsymbol{\theta}_k)$. The marginal probability $P(\mathbf{x}_i)$ of the mixture is defined in:

$$P(\mathbf{x}) = \sum_{k=1}^{K} P(\boldsymbol{\theta}_k) P(\mathbf{x}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}|\boldsymbol{\theta}_k) \tag{1}$$

where π_k is the non-negative mixing coefficient that sums to unity, representing the likelihood of occurrence of the probabilistic component $\boldsymbol{\theta}_k$.

Because of the multimodality of observed scenes, the intensity of target pixels is assumed to be distributed normally in a finite mixture. Regarding RGB space of analyzed videos, each examined color channel in $\mathbf{x}_i$ was assumed to be distributed independently and can be described with a common variance σ_k to avoid performing costly matrix inversion as indicated in [6]. Hence, the multivariate Gaussian distribution can be re-formulated as:

$$\begin{aligned} P(\mathbf{x}|\boldsymbol{\theta}_k) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^c \sigma_k^c}} \exp\left(-\frac{\| \mathbf{x} - \boldsymbol{\mu}_k \|^2}{2\sigma_k}\right) \end{aligned} \tag{2}$$

where $\boldsymbol{\mu}_k$ is the estimated mean and σ_k is the estimated universal covariance of examined color channels in the k^{th} Gaussian component.

From this hypothesis, in this work, we propose an architecture of convolutional neural network, called Convolutional Density Network of Gaussian Mixtures (CDN-GM), which employs a set of non-linearity transformations $f_\theta(\cdot)$ to formulate a conditional formalism of GMM density function of $\mathbf{x}$ given a set of randomly selected, vectorized data points $\boldsymbol{\chi}_T$:

$$\mathbf{y}_T = f_\theta(\boldsymbol{\chi}_c^T) \sim P(\mathbf{x}|\boldsymbol{\chi}_\mathbf{c}^\mathbf{T}) \tag{3}$$

The ability of multilayer neural networks that was trained with an optimization algorithm to learn complex, high-dimensional, nonlinear mappings from large collections of examples increases their capability in pattern recognition via gathering relevant information from the input and eliminating irrelevant variabilities. With respect to problems of prediction, the conditional average represents only a very limited statistic. For applicable contexts, it is considerably beneficial to obtain a complete description of the probability distribution of the target data. In this work, we incorporate the mixture density model with the convolutional neural network instead of a multi-layer perceptron as done by Bishop *et al.* in the vanilla research [13]. In the proposed scheme, the network itself learns to act as a feature extractor to formulate statistical inferences on temporal series of intensity values. First, regarding recently proposed CNN methods, the local connectivity characteristics in convolution layers motivate CNN to learn common visual patterns in a local region of images. Literally, a background image contains most frequently presented intensities in the sequence of observed scenes. Hence, in CDN-GM, we take advantage of this mechanism to exploit the most likely intensity value that will raise in the background image via consideration of temporal arrangement. Second, the memory requirement to store so many weights may rule out certain hardware implementations. In convolutional layers, shift invariance is automatically obtained by forcing the replication of weight configurations across space. Hence, the scheme of weight sharing in the proposed CNN reduces the number of parameters, making CDN-GM lighter and exploiting the

parallel processing of a set of multiple pixel-wise analysis within a batch of video frames.

The architecture of CDN-GM contains seven learned layers, not counting the input – two depthwise convolutional, two convolutional and three dense layers. Our network is summarized in Fig. 2. The input of our rudimentary architecture of the proposed network is a time series of color intensity at each pixel, which was analyzed with noncomplete connection schemes in four convolution layers regarding temporal perspective. Finally, the feature map of the last convolution layer was connected with three different configurations of dense layers to form a three-fold output of the network which present the kernel parameter of the Gaussian Mixture Model.

The main goal of CDN-GM is to construct an architecture of CNN which presents multivariate mapping in forms of Gaussian Mixture Model with the mechanism of offline learning. With the simulated probabilistic function, we aim to model the description of the most likely background scenes from actual observed data. In other words, the regularities in the proposed CNN should cover a generalized presentation of the intensity series of a set of consecutive frames at pixel level. To achieve this proposition, instead of using separate GMM for each pixel-wise statistical learning, we consider to use a single GMM to formulate the temporal history of all pixels in the whole image. Accordingly, CDN-GM architecture is extended through a spatial extension of temporal data at image points with an extensive scheme defined in Table I.

The network output $\mathbf{y_T}$, whose dimension is $(c+2) \times K$, is partitioned into three portions $\mathbf{y}_\mu\left(\boldsymbol{\chi}_c^T\right)$, $\mathbf{y}_\sigma\left(\boldsymbol{\chi}_c^T\right)$, and $\mathbf{y}_\pi\left(\boldsymbol{\chi}_c^T\right)$ corresponding to the latent variables of GMM model:

TABLE I
ARCHITECTURE OF CONVOLUTIONAL DENSITY NETWORK

Type / Stride	Filter Shape	Output Size
Input	-	$(H*W) \times 1 \times T \times 3$
Conv dw / s7	$1 \times 7 \times 1$ dw	$(H*W) \times 1 \times 35 \times 3$
Conv / s1	$1 \times 1 \times 3 \times 7$	$(H*W) \times 1 \times 35 \times 7$
Conv dw / s7	$1 \times 7 \times 7$ dw	$(H*W) \times 1 \times 5 \times 7$
Conv / s1	$1 \times 1 \times 7 \times 7$	$(H*W) \times 1 \times 5 \times 7$
Dense / s1	$K \times C$	$(H*W) \times K \times d$
Dense / s1 / Softmax	K	$(H*W) \times K$
Dense / s1	K	$(H*W) \times K$

$$\begin{aligned}\mathbf{y_T} &= \left[\mathbf{y}_\mu\left(\boldsymbol{\chi}_c^T\right), \mathbf{y}_\sigma\left(\boldsymbol{\chi}_c^T\right), \mathbf{y}_\pi\left(\boldsymbol{\chi}_c^T\right)\right] \\ &= \left[\mathbf{y}_\mu^1, \ldots, \mathbf{y}_\mu^K, \mathbf{y}_\sigma^1, \ldots, \mathbf{y}_\sigma^K, \mathbf{y}_\pi^1, \ldots, \mathbf{y}_\pi^K\right]\end{aligned} \tag{4}$$

With our goal of formulating the GMM, we impose a different restriction on threefold outputs from the network:

- First, as the mixing coefficients π_k indicate the proportion of data accounted for by mixture component k, they must be defined as independent and identically distributed probabilities. To achieve this regulation, in principle, we activate the network output with a softmax activation function:

$$\pi_k(\boldsymbol{\chi}_c^T) = \frac{\exp(\mathbf{y}_\pi^k)}{\sum_{l=1}^K \exp(\mathbf{y}_\pi^l)} \tag{5}$$

- Second, in the realistic scenarios, the measured intensity of observed image signals may fluctuate due to a variety of factors, including illumination transformations,

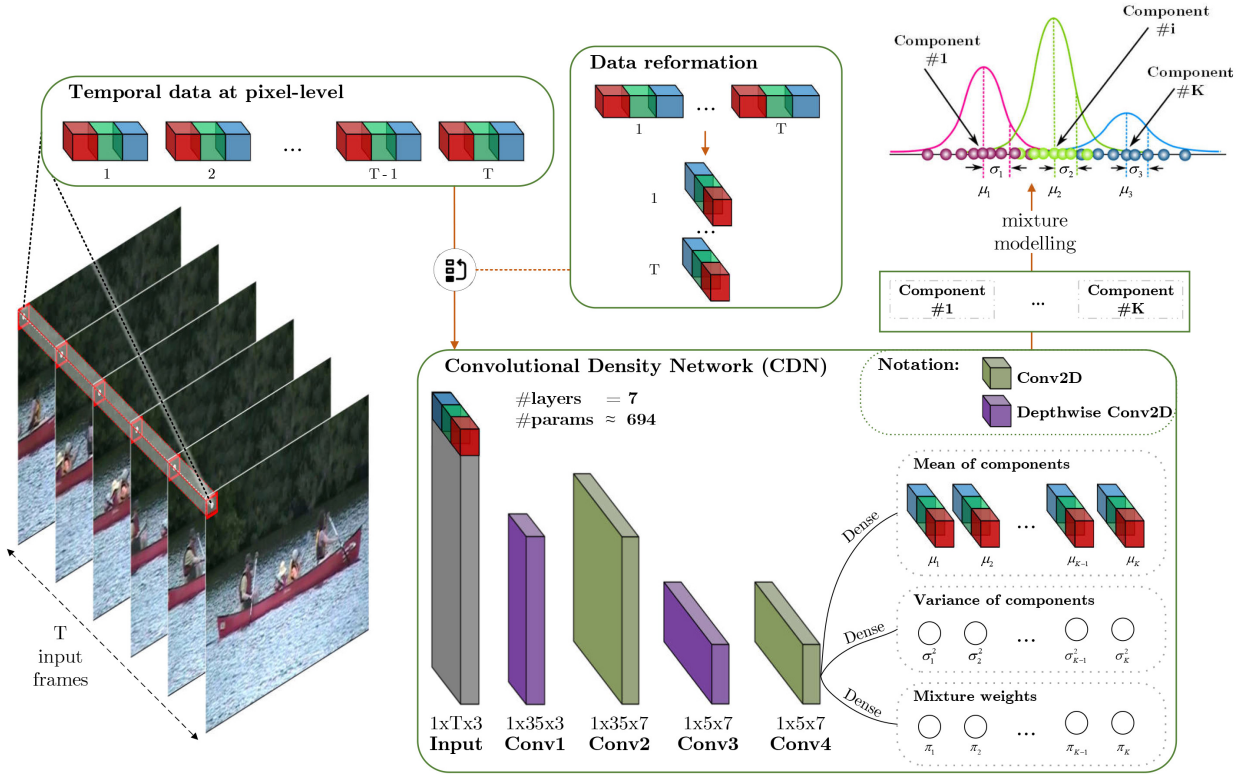


Fig. 2. The proposed architecture of Convolution Density Network of Gaussian Mixture Model

dynamic contexts and bootstrapping. In order to conserve the estimated background, we have to restrict the value of the variance of each component to the range $[\bar{\sigma}_{min}, \bar{\sigma}_{max}]$ so that each component does not span spread the entire color space, and does not focus on one single color cluster:

$$\sigma_k(\boldsymbol{\chi}_c^T) = \frac{\bar{\sigma}_{min} \times (1 - \hat{\sigma}_k) + \bar{\sigma}_{max} \times \hat{\sigma}_k}{255} \tag{6}$$

where $\sigma_k(\boldsymbol{\chi}_c^T)$ is normalized towards a range of $[0, 1]$ over the maximum color intensity value, 255; and $\hat{\sigma}_k$ is the normalized variance that was activated through a hard-sigmoid function from the output neurons $\mathbf{y}_\sigma$ that correspond to the variances:

$$\hat{\sigma}_k(\boldsymbol{\chi}_c^T) = \max\left[0, \min\left(1, \frac{\mathbf{y}_\sigma^k + 1}{2}\right)\right] \tag{7}$$

In this work, we adopt the hard sigmoid function because of the piecewise linear property and correspondence to the bounded form of linear rectifier function (ReLU) of the technique. Furthermore, this was proposed and proved to be more efficient in both in software and specialized hardware implementations by Courbariaux *et al.* [31].

- Third, the mean of the probabilistic mixture is considered on a normalized RGB color space where the intensity values retain in a range of $[0, 1]$ so that they can be approximated correspondingly with the normalized input. Similar to the normalized variance $\hat{\sigma}_k$, the mixture mean is standardized from the corresponding network outputs with a hard-sigmoid function:

$$\mu_k(\boldsymbol{\chi}_c^T) = \max\left[0, \min\left(1, \frac{\mathbf{y}_\mu^k + 1}{2}\right)\right] \tag{8}$$

From the proposed CNN, we extract the periodical background image for each block of pixel-wise time series of data in a period of T. This can be done by selecting the means whose corresponding distributions have the highest degree of high-weighted, low-spread. To have a good grasp of the importance of a component in the mixture, we use a different treatment of weight updates with a ratio of $\pi_{k'}(\boldsymbol{\chi}_c^T)/\sigma_{k'}(\boldsymbol{\chi}_c^T)$. This is the manner of weighting components within a mixture at each pixel by valuing high-weighted, low-spread distributions in the mixture, thereby spotlighting the most significant distribution contributing to the construction of backgrounds.

$$BG(\boldsymbol{\chi}_c^T) = \max(\mu_k \cdot \hat{BG}_{k,T}), \quad \text{for } k \in [1, K] \tag{9}$$

where background mapping is defined at each pixel $\mathbf{x}$ as:

$$\hat{BG}_{k,T}(\boldsymbol{\chi}_c^T) = \begin{cases} 1, & \text{if } \underset{k'}{argmax}[{}^{\pi_{k'}(\boldsymbol{\chi}_c^T)}/_{\sigma_{k'}(\boldsymbol{\chi}_c^T)}] = k \\ & \text{for } k \in [1, K] \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

B. The unsupervised loss function of CDN-GM

In practice, particularly in each real-life scenario, the background model must capture multiple degrees of dynamics, which is more challenging by the fact that scene dynamics may also change gradually under external effects (e.g. lighting deviations). These effects convey the latest information regarding contextual deviations that may constitute new background predictions. Therefore, the modeling of backgrounds must not only take into account the various degrees of dynamics across multiple imaging pixels of the data source, but it must also be able to adaptively update its predictions with respect to semantic changes. Equivalently, in order to approximate a statistical mapping function for background modeling, the proposed neural network function has to be capable of approximating a conditional probability density function, thereby estimating a multi-modular distribution conditioned on its time-wise latest raw imaging inputs. The criteria for the neural statistical function to be instituted can be summarized as follows:

- As a metric for estimating distributions, input data sequences cannot be weighted in terms of order.
- Taking adaptiveness into account, the neural probabilistic density function can continuously interpolate predictions in evolving scenes upon reception of new data.
- The neural network function has to be generalizable such that its model parameters are not dependent on specific learning datasets.

Hence, satisfying the prescribed criteria, we propose a powerful loss function capable of directing the model's parameters towards adaptively capturing the conditional distribution of data inputs, thereby approximating a statistical mapping function in a technologically parallelizable form. At every single pixel, the proposed CNN estimates the probabilistic density function on the provided data using its GMM parameters. Specifically, given the set $\boldsymbol{\chi}_c^T$ randomly selected, vectorized data points, it is possible to retrieve the continuous conditional distribution of the data target $\mathbf{x}$ with the following functions:

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{\chi}_c^T) \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) \tag{11}$$

where the general disposition of this distribution is approximated by a finite mixture of Gaussians, whose values are dependent on our learnable neural variables:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) = \frac{1}{\sqrt{2\pi \cdot \sigma_k(\boldsymbol{\chi}_c^T)^2}} \cdot \exp\left\{-\frac{\left\|\mathbf{x} - \mu_k(\boldsymbol{\chi}_c^T)\right\|^2}{2\sigma_k(\boldsymbol{\chi}_c^T)^2}\right\} \tag{12}$$

In our proposed loss function, the data distribution to be approximated is the set of data points relevant to background construction. This is rationalized by the proposed loss function's purpose, which is to direct the neural network's variables towards generalizing universal statistical mapping functions. Furthermore, even with constantly evolving scenes where the batches of data values also vary, this loss measure can constitute fair weighting on the sequence of inputs. Our proposed loss measure is designed to capture various pixel-wise dynamics over a video scene and to encompass even

unseen perspectives via exploiting the huge coverage of multiple scenarios across more than one case with data. In other words, the order of the network's input does not matter upon loading, which is proper for any statistical function on estimating distribution. For modeling tasks, we seek to establish a universal multi-modular statistical mapping function on the RGB color space, which would require optimizing the loss not just on any single pixel, but for b block of time-series image intensity data fairly into a summation value.

$$\mathcal{L} = \sum_{i}^{b} \sum_{j}^{T} \mathcal{L}_j^{(i)} \tag{13}$$

$$\text{where} \qquad \mathcal{L}_j^{(i)} = -\ln \left(\sum_{k=1}^{K} \pi_k^{(i)} \mathcal{N}(\mathbf{x}_j | \mu_k^{(i)}, \sigma_k^{(i)}) \right) \tag{14}$$

where $\mathbf{x}_j$ is the j^{th} element of the i^{th} time-series data $\boldsymbol{\chi}_c^{T,(i)}$ of pixel values; $\pi^{(i)}$, $\mu^{(i)}$, and $\sigma^{(i)}$ are respectively the desired mixing coefficients, means, and variances that commonly model the distribution of $\boldsymbol{\chi}_c^{T,(i)}$ in GMM.

We define $\mathcal{L}_j^{(i)}$ as the error function for our learned estimation on an observed data point $\mathbf{x}_j$, given the locally relevant dataset $\boldsymbol{\chi}_c^{T,(i)}$ for the neural function. $\mathcal{L}_j^{(i)}$ is based on the statistical log-likelihood function and is equal to the negative of its magnitude. Hence, by minimizing this loss measure, we will essentially be maximizing the expectation value of the GMM-based neural probabilistic density function $P(\mathbf{x})$, from the history of pixel intensities at a pixel position. Employing stochastic gradient descent on the negative logarithmic function $\mathcal{L}_j^{(i)}$ involves not only monotonic decreases, which are steep when close to zero, but also upon convergence it also leads to the proposed neural function approaching an optimized mixture of Gaussians probability density function.

In addition, since our loss function depends entirely on the input and the output of the network (i.e., without external data labels), the proposed work can be considered an unsupervised approach. This is because the objective of our network is to maximize the likelihood of the output on the data itself, not to any external labels. With this loss function, the optimization of the network to generalize on new data is available on the fly without needing any data labeled manually by humans. The key thing here is that whether the neural network can learn to optimize the loss function with the standard stochastic gradient descent algorithm with *back-propagation*. This can only be achieved if we can obtain suitable equations of the partial derivatives of the error $\mathcal{L}$ with respect to the outputs of the network. As we describe in the previous section, $\mathbf{y}_\mu$, $\mathbf{y}_\sigma$, and $\mathbf{y}_\pi$ present the proposed CDN-GM's outputs that formulate to the latent variables of GMM model. The partial derivative $\partial \mathcal{L}_j^{(i)} \big/ \partial \mathbf{y}^{(k)}$ can be evaluated for a particular pattern and then summed up to produce the derivative of the error function $\mathcal{L}$. To simplify the further analysis of the derivatives, it is convenient to introduce the following notation that presents the posterior probabilities of the component k in the mixture, using Bayes theorem:

$$\Pi_k^{(i)} = \frac{\pi_k^{(i)} \mathcal{N}(\mathbf{x}_j | \boldsymbol{\mu}_k^{(i)}, \sigma_k^{(i)})}{\sum_{l=1}^{K} \pi_l^{(i)} \mathcal{N}(\mathbf{x}_j | \boldsymbol{\mu}_l^{(i)}, \sigma_l^{(i)})} \tag{15}$$

First, we need to consider the derivatives of the loss function with respect to network outputs $\mathbf{y}_\pi$ that correspond to the mixing coefficients π_k. Using Eq. (14) and (15), we obtain:

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \pi_k^{(i)}} = \frac{\Pi_k^{(i)}}{\pi_k^{(i)}} \tag{16}$$

From this expression, we perceive that the value of $\pi_k^{(i)}$ explicitly depends on $\mathbf{y}_\pi^{(l)}$ for $l = 1, 2, ..., K$ as $\pi_k^{(i)}$ is the result of the softmax mapping from $\mathbf{y}_\pi^{(l)}$ as indicated in Eq. (5). We continue to examine the partial derivative of $\pi_k^{(i)}$ with respect to a particular network output $\mathbf{y}_\pi^{(l)}$, which is

$$\frac{\partial \pi_k^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} = \begin{cases} \pi_k^{(i)}(1 - \pi_l^{(i)}), & \text{if } k = l \\ -\pi_l^{(i)} \pi_k^{(i)}, & \text{otherwise.} \end{cases} \tag{17}$$

By chain rule, we have

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} = \sum_k \frac{\partial \mathcal{L}_j^{(i)}}{\partial \pi_k^{(i)}} \frac{\partial \pi_k^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} \tag{18}$$

From Eq. (15), (16), (17), and (18), we then obtain

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} = \pi_l^{(i)} - \Pi_l^{(i)} \tag{19}$$

For $\mathbf{y}_\sigma^{(k)}$, we make use of Eq. (2), (6), (7), (14), and (15), by differentiation, to obtain

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\sigma^{(k)}} = \frac{3.2}{255} \Pi_k^{(i)} \left(\frac{c}{2} \sqrt{(2\pi)^c (\sigma_k^{(i)})^{c+2}} - \frac{\| \mathbf{x}_j - \boldsymbol{\mu}_k \|^2}{2(2\pi)^c (\sigma_k^{(i)})^{c+2}} \right) \tag{20}$$

for $-2.5 < \mathbf{y}_\sigma^{(k)} < 2.5$. This is because the piece-wise property in the definition of the hard-sigmoid activation function.

Finally, for $\mathbf{y}_\mu^{(k)}$, let $\mu_{k,l}^{(i)}$ be the l^{th} element of the mean vector where l is an integer lies in $[0, c)$ and suppose that $\mu_{k,l}^{(i)}$ corresponds to an output o_k^μ of the network. We can get derivative of $\mu_{k,l}^{(i)}$ by taking Eq. (2), (8), (14), (15) into the differentiation process:

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\mu^{(k)}} = 0.2 \times \Pi_k^{(i)} \frac{x_{j,l} - \mu_{k,l}^{(i)}}{\sigma_k^{(i)}} \tag{21}$$

for $-2.5 < \mathbf{y}_\mu^{(k)} < 2.5$.

From Eq. (19), (20), and (21), when CDN-GM is performed data-driven learning individually on each video sequence using Adam optimizer with a learning rate of α, the process tries to regulate the values of laten parameters in the mixture model via minimizing the negative of log likelihood function. Hence, once the proposed model has been trained on video sequences, it is obviously seen that the network can predict the conditional density function of the target background, which is a statistical description of time-series data of each image point, so far, the foreground mask is then segmented correspondingly. The primary conceptualization in the model is to address the problems of DNNs as we mentioned above via online adaptively acquiring the underlying properties of a sequence of images to construct corresponding background scenes at concrete moments rather than memorizing the single-valued mapping between input frames and labelled backgrounds.

C. Foreground Segmentation with Non-linearity Differencing

In this section, we present the description of our proposed convolutional auto-encoder, called MEDAL-net, which simulates non-linear frame-background differencing for foreground detection. Traditionally, thresholding schemes are employed to find the highlighted difference between an imaging input and its corresponding static view in order to segment motion. For example, Stauffer and Grimson [6] employed variance thresholding on background - input pairs by modeling the static view with the Gaussian Mixture Model. While the experimental results suggest certain degrees of applicability due to its simplicity, the approach lacks in flexibility as the background model is usually not static and may contain various motion effects such as occlusions, stopped objects, shadow effects, etc.

In practice, a good design of a difference function between the current frame and its background must be capable of facilitating motion segmentation across a plethora of scenarios and effects. However, for the countless scenarios in real life, where there are unique image features and motion behaviors to each, there is yet any explicit mathematical model that is general enough to cover them all. Because effective subtraction requires high-degreed non-linearity in order to compose a model for the underlying mathematical framework of many scenarios, following the Universal approximation theorem [32], we design the technologically parallelizable neural function for an approximation of such framework. Specifically, we make use of a CNN to construct a foreground segmentation network. The motive is further complemented by two folds:

- Convolutional Neural Networks have long been known for their effectiveness in approximating nonlinear functions with arbitrary accuracy.
- Convolutional Neural Networks are capable of balancing between both speed and generalization accuracy, especially when given an effective design and enough representative training data.

However, recent works exploiting CNN in motion estimation are still generating heavy-weighted models which are computationally expensive and not suitable for real-world deployment. In our proposed work, we exploit the use of a pair of the current video frame and its corresponding background as the input to the neural function and extract motion estimation. By combining this with a suitable learning objective, we explicitly provide the neural function with enough information to mold itself into a context-driven non-linear difference function, thereby restricting model behavior and its search directions. This also allows us to scale down the network's parameter size, width, and depth to focus on learning representations while maintaining generalization for unseen cases. As empirically shown in the experiments, the proposed architecture is light-weighted in terms of the number of parameters, and is also extremely resource-efficient, e.g. compared to FgSegNet [29].

1) Architectural design: The overall flow of the MEDAL-net is shown in Fig. 3. We employ the encoder-decoder design approach for our segmentation function. With this approach, data inputs are compressed into a low-dimensional latent space of learned informative variables in the encoder, and the encoded feature map is then passed into the decoder, thereby generating foreground masks.

In our design, we fully utilize the use of depthwise separable convolution introduced in MobileNets [33] so that our method can be suitable for mobile vision applications. Because this type of layer significantly scales down the number of convolutional parameters, we reduced the number of parameters of our network by approximately 81.7% compared to using only standard 2D convolution, rendering a light-weighted network of around 2,800 parameters. Interestingly, even with such a small set of parameters, the network still does not lose its ability to generalize predictions at high accuracy. Our architecture also employs normalization layers, but only for the decoder. This design choice is to avoid the loss of information in projecting the contextual differences of background-input pairs into the latent space via the encoder, while formulating normalization to boost the decoder's learning. The architecture of the proposed model is described in Table II.

a) Encoder: The encoder can be thought of as a folding function that projects the loaded data into an information-rich low-dimensional feature space. In our architecture, the encoder takes in pairs of video frames and their corresponding backgrounds concatenated along the depth dimension as its inputs. Specifically, the background image estimated by CDN-GM is concatenated with imaging signals such that raw information can be preserved for the neural network to freely learn to manipulate. Moreover, with the background image also in its raw form, context-specific scene dynamics (e.g. moving waves, camera jittering, intermittent objects) are also captured. Thus, as backgrounds are combined with input images to formulate

TABLE II
BODY ARCHITECTURE OF MEDAL-NET

Type / Stride	Filter shape	Ouput size
Input	-	N x H x W x 6
DW conv / s1	3 x 3 x 1	N x H x W x 6
Conv / s1 / ReLU	1 x 1 x 6 x 16	N x H x W x 16
DW conv / s1	3 x 3 x 1	N x H x W x 16
Conv / s1 / ReLU	1 x 1 x 16 x 16	N x H x W x 16
Max pool / s2	2 x 2 x 1	N x (H / 2) x (W / 2) x 16
DW conv / s1	3 x 3 x 1	N x (H / 2) x (W / 2) x 16
Conv / s1 / ReLU	1 x 1 x 6 x 16	N x (H / 2) x (W / 2) x 16
DW conv / s1	3 x 3 x 1	N x (H / 2) x (W / 2) x 16
Conv / s1 / ReLU	1 x 1 x 16 x 16	N x (H / 2) x (W / 2) x 16
Max pool / s2	2 x 2 x 1	N x (H / 4) x (W / 4) x 16
DW conv / s1	3 x 3 x 1	N x (H / 4) x (W / 4) x 16
Conv / s1	1 x 1 x 16 x 16	N x (H / 4) x (W / 4) x 16
InstanceNorm / ReLU	-	N x (H / 4) x (W / 4) x 16
Upsampling	-	N x (H / 2) x (W / 2) x 16
DW conv / s1	3 x 3 x 1	N x (H / 2) x (W / 2) x 16
Conv / s1	1 x 1 x 16 x 16	N x (H / 2) x (W / 2) x 16
InstanceNorm / ReLU	-	N x (H / 2) x (W / 2) x 16
Upsampling	-	N x H x W x 16
DW conv / s1	3 x 3 x 1	N x H x W x 16
Conv / s1	1 x 1 x 16 x 16	N x H x W x 16
InstanceNorm / ReLU	-	N x H x W x 16
DW conv / s1	3 x 3 x 1	N x H x W x 16
Conv / s1 / Hard Sigmoid	1 x 1 x 16 x 1	N x H x W x 1

predictions, MEDAL-net may further learn to recognize motions that are innate to a scene, thereby selectively segmenting motions of interest based on the context.

In addition, by explicitly providing a pair of the current input frame and its background image to segment foregrounds, our designed network essentially constructs a simple difference function that is capable of extending its behaviors to accommodate contextual effects. Thus, we theorize that approximating this neural difference function would not require an enormous number of parameters. In other words, it is possible to reduce the number of layers and the weights' size of the foreground extraction network to accomplish the task. Hence, the encoder only consists of a few convolutional layers, with 2 max-pooling layers for downsampling contextual attributes into a feature-rich latent space.

b) Decoder: The decoder of our network serves to unfold the encoded feature map into the foreground space using convolutional layers with two upsampling layers to restore the original resolution of its input data.

In order to facilitate faster training and better estimation of the final output, we engineered the decoder to include instance normalization, which is apparently more efficient than batch normalization [34]. Using upsampling to essentially expand the latent tensors, the decoder also employs convolutional layers to induce non-linearity like the encoder.

The final output of the decoder is a grayscale probability map where each pixel's value represents the chance that it is a component of a foreground object. This map is the learned motion segmentation results with pixel-wise confidence scores determined on account of its neighborhood and scene-specific variations. In our design, we use the hard sigmoid activation function because of its property that allows faster gradient propagation, which results in less training time.

At inference time, the final segmentation result is a binary image obtained by placing a constant threshold on the generated probability map. Specifically, suppose $\mathbf{X}$ is a probability map of size $N \times H \times W \times 1$, and let the set F be defined as:

$$F = \{(x, y, z) | \mathbf{X}_{x,y,z,0} \geq \epsilon\} \tag{22}$$

where $x \in [0, N]$, $y \in [0, H]$, $z \in [0, W]$, and ϵ is an experimentally determined parameter. In other words, F is a set of indices of $\mathbf{X}$ that satisfy the threshold ϵ. The segmentation map $\hat{\mathbf{Y}}$ of size $N \times H \times W$ is obtained by:

$$\hat{\mathbf{Y}}_{i,j,k} = \begin{cases} 1, & (i,j,k) \in F \\ 0, & otherwise \end{cases} \tag{23}$$

where 1 represents indices classified as foreground, and 0 represents background indices.

2) Training:

a) Data preparation: The training dataset for MEDAL-net is carefully chosen by hand so that the data maintains the balance between background labels and foreground labels since imbalance data will increase the model's likelihood of being overfitted. We choose just 200 labeled ground truths to train the model. This is only up to 20% of the number of labeled frames for some sequences in CDnet, and 8.7% of CDnet's labeled data in overall. During training, the associated background of each chosen frame is directly generated using CDN-GM as MEDAL-net is trained separately from CDN-GM because of the manually chosen input-label pairs.

b) Training procedure: We penalize the output of the network using the cross-entropy loss function commonly used for segmentation tasks $[x, y, z]$, as the goal of the model is to learn a Dirac delta function for each pixel. The description of the loss function is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W}[\mathbf{Y}_{i,j,k}\log(\hat{\mathbf{Y}}_{i,j,k}) + (1-\mathbf{Y}_{i,j,k})\log(1-\hat{\mathbf{Y}}_{i,j,k})] \tag{24}$$

where $\mathbf{Y}$ is the corresponding target set of foreground binary masks for $\hat{\mathbf{Y}}$, the batch of predicted foreground probability maps. The network is trained for about 1000 epochs for each

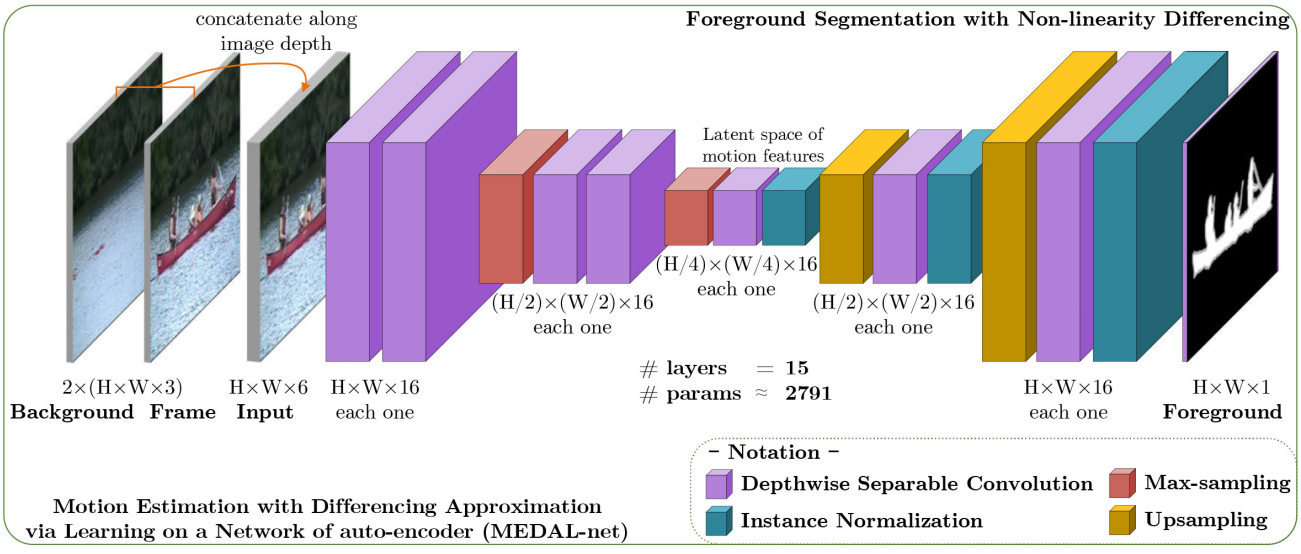


Fig. 3. The proposed architecture of MEDAL-net grounded on convolutional autoencoder for foreground detection

sequence in CDnet using Adam optimizer with the learning rate = 0.005.

With this straightforward learning objective applied on our CNN, the designed architecture is enabled to learn not only pixel-wise motion estimates of the training set, but it also is taught to recognize inherent dynamics in its data, and perform as a context-driven neural difference function to accurately interpolate region-wise foreground predictions of unseen perspectives.

IV. Experiments and Discussion

A. Experimental Setup

In this section, we proceed to verify experimentally the capabilities of the proposed method via comparative evaluations in capturing motion attributes. This is in order to evaluate the effectiveness of CDN-MEDAL-net in foreground detection. Our proposed scheme is designed to explicitly incorporate the probabilistic density properties into the architecture to achieve accurate adaptiveness, while taking advantage of parallel computing technologies often used with DNNs to compete with state-of-the-art works in speed given its light structure. Therefore, we compare the accuracy of the proposed framework not only with unsupervised approaches that are light-weighted and generalizable without pretraining: GMM – Stauffer & Grimson [6], GMM – Zivkovic [7], SuBSENSE [26], PAWCS [35], TensorMoG [9], BMOG [8], FTSG [27], SWCD [36], but also with the data-driven, supervised models which trade computational expenses for high accuracy performance: FgSegNet_S [29], FgSegNet [37], FgSegNet_v2 [38], Cascade CNN [22], DeepBS [25], STAM [39].

In terms of chosen metrics for measuring motion features, we employ quantitative analysis on values that can be appraised from confusion matrices, i.e. Precision, Recall, F-Measure, False-Negative Rate (FNR), False-Positive Rate (FPR) and Percentage of Wrong Classification (PWC). With the overall results being drawn from the combination of all confusion matrices across given scenarios, the benchmarks on CDnet-2014 [40] were performed by comparing foreground predictions against provided ground-truths. Then, we evaluate the proposed framework trained with CDnet-2014 on Wallflower [41] without any tuning or retraining laten parameters to examine the capability of our proposed approach in unseen scenarios having similar dynamics. Finally, we will also analyze all methods in terms of processing speed with the image resolution of 320×240 and draw final conclusions.

In our experiment, the number of Gaussians K is empirically and heuristically to balance the CDN-GM's capability of modeling constantly evolving contexts (e.g. moving body of water) under many effects of potentially corruptive noises. With K too big, many GMM components many be unused or they simply capture the various noises within contextual dynamics. As the Gaussian component corresponding to the background intensity revolves around the most frequently occurring color subspaces to draw predictions, the extra components serve only as either placeholders for abrupt changes in backgrounds, be empty or capture intermittent noises of various degrees. In practice, noise Gaussian components in GMM are pulse-like as they would appear for short durations, and low-weighted because they are not as often matched as background components. Nevertheless, they still present corruptive effects to our model. Our proposed CDN-GM model was set up with the number of Gaussian components $K = 3$ for all experimented sequences, and was trained on CDnet-2014 dataset with Adam optimizer using a learning rate of $\alpha = 1e^{-4}$.

In addition, the constants $\bar{\sigma}_{min}$ and $\bar{\sigma}_{max}$ were chosen such that no Gaussian components span the whole color space while not contracting to a single point that represents noises. If the $[\bar{\sigma}_{min}, \bar{\sigma}_{\max}]$ interval is too small, all of the Gaussian components will be likely to focus on one single color cluster. Otherwise, if the interval is too large, some of the components might still cover all intensity values, making it hard to find the true background intensity. Based on this assumption and experimental observations, we find that the difference between color clusters usually does not exceed approximately 16 at minimum and 32 at maximum.

Regarding MEDAL-net, the value of ϵ was emperically chosen to be 0.3 in order to extract the foreground effectively even under high color similarity between objects and background.

B. Results on CDnet 2014 Benchmarks

Using the large-scale CDnet-2014 dataset, we demonstrate empirically the effectiveness of our proposed approach across a plethora of scenarios and effects. For each thousands-frame sequence of a scenario, we sample only 200 foreground images for training our foreground estimator. This strategy of sampling for supervised learning is the same as that of FgSegNet's and Cascade CNN. The experimental results are summarized in Table III, which highlights the F-measure quantitative results of our approach compared against several existing state-of-the-art approaches, along with Fig. 4 that provides qualitative illustrations. Despite its compact architecture, the proposed approach is shown to be capable of significantly outperforming unsupervised methods, and competing with complex deep-learning-based, supervised approaches in terms of accuracy on all but only the *PTZ* scenario. In this experimental dataset, we pass over the *PTZ* subdivision where our approach of CDN-GM is unsustainable to model the underlying description of the most likely background because of the fluctuation of actually observed data sequences when the recording camera rotates continuously. Accordingly, our MEDAL-net scheme of foreground segmentation encounters difficulty in estimating difference between input frames and corresponding background scenes.

In comparison with unsupervised models built on the GMM background modeling framework like GMM – Stauffer & Grimson, GMM – Zivkovic, BMOG and TensorMoG, the proposed approach is better augmented by the context-driven motion estimation plugin, without being constrained by simple thresholding schemes. Thus, it is able to provide remarkably superior F-measure results across the scenarios, especially on those where there are high degrees of noises or background dynamics like *LFR*, *NVD*, *IOM*, *CJT*, *DBG* and *TBL*. However, it is apparently a little worse than TensorMoG

TABLE III
F - MEASURE COMPARISONS OVER ALL OF ELEVEN CATEGORIES IN THE CDNET 2014 DATASET

	Method	*BDW*	*LFR*	*NVD*	*PTZ*	*THM*	*SHD*	*IOM*	*CJT*	*DBG*	*BSL*	*TBL*
Unsupervised	GMM – S & G	0.7380	0.5373	0.4097	0.1522	0.6621	0.7156	0.5207	0.5969	0.6330	0.8245	0.4663
	GMM – Zivkovic	0.7406	0.5065	0.3960	0.1046	0.6548	0.7232	0.5325	0.5670	0.6328	0.8382	0.4169
	SuBSENSE	$0.8619_{(2)}$	0.6445	$0.5599_{(3)}$	$0.3476_{(3)}$	$0.8171_{(3)}$	$0.8646_{(3)}$	0.6569	$0.8152_{(2)}$	0.8177	$0.9503_{(1)}$	$0.7792_{(2)}$
	PAWCS	0.8152	$0.6588_{(3)}$	0.4152	$0.4615_{(1)}$	$0.9921_{(1)}$	$0.8710_{(2)}$	$0.7764_{(3)}$	$0.8137_{(3)}$	$0.8938_{(1)}$	$0.9397_{(3)}$	0.6450
	TensorMoG	$0.9298_{(1)}$	$0.6852_{(2)}$	$0.5604_{(2)}$	0.2626	0.7993	$0.9738_{(1)}$	$0.9325_{(1)}$	$0.9325_{(1)}$	0.6493	$0.9488_{(2)}$	$0.8380_{(1)}$
	BMOG	0.7836	0.6102	0.4982	0.2350	0.6348	0.8396	0.5291	0.7493	0.7928	0.8301	0.6932
	FTSG	0.8228	0.6259	0.5130	0.3241	0.7768	0.8535	$0.7891_{(2)}$	0.7513	$0.8792_{(2)}$	0.9330	0.7127
	SWCD	$0.8233_{(3)}$	$0.7374_{(1)}$	$0.5807_{(1)}$	$0.4545_{(2)}$	$0.8581_{(2)}$	0.8302	0.7092	0.7411	$0.8645_{(3)}$	0.9214	$0.7735_{(3)}$
*	**CDN-MEDAL-net**	**0.9045**	**0.9561**	**0.8450**	-	**0.9129**	**0.8683**	**0.8249**	**0.8427**	**0.9372**	**0.9615**	**0.9187**
Supervised	FgSegNet_S	$0.9897_{(2)}$	$0.8972_{(2)}$	$0.9713_{(2)}$	$0.9879_{(1)}$	$0.9921_{(1)}$	$0.9937_{(3)}$	$0.9940_{(3)}$	$0.9957_{(2)}$	$0.9958_{(2)}$	$0.9977_{(1)}$	0.9681
	FgSegNet	$0.9845_{(3)}$	$0.8786_{(3)}$	$0.9655_{(3)}$	$0.9843_{(3)}$	$0.9648_{(3)}$	$0.9973_{(2)}$	$0.9958_{(1)}$	$0.9954_{(3)}$	$0.9951_{(3)}$	$0.9944_{(3)}$	$0.9921_{(2)}$
	FgSegNet_v2	$0.9904_{(1)}$	$0.9336_{(1)}$	$0.9739_{(1)}$	$0.9862_{(2)}$	$0.9727_{(2)}$	$0.9978_{(1)}$	$0.9951_{(2)}$	$0.9971_{(1)}$	$0.9961_{(1)}$	$0.9952_{(2)}$	$0.9938_{(1)}$
	Cascade CNN	0.9431	0.8370	0.8965	0.9168	0.8958	0.9414	0.8505	$0.9758_{(3)}$	0.9658	0.9786	0.9108
	DeepBS	0.8301	0.6002	0.5835	0.3133	0.7583	0.9092	0.6098	0.8990	0.8761	0.9580	0.8455
	STAM	0.9703	0.6683	0.7102	0.8648	0.9328	0.9885	0.9483	0.8989	0.9155	0.9663	$0.9907_{(3)}$

*Semi-Unsupervised; Experimented scenarios include bad weather (*BDW*), low frame rate (*LFR*), night videos (*NVD*), pan-tilt-zoom (*PTZ*), turbulence (*TBL*), baseline (*BSL*), dynamic background (*DBG*), camera jitter (*CJT*), intermittent object motion (*IOM*), shadow (*SHD*), and thermal (*THM*). In each column, $Red_{(1)}$ is for the best, $Green_{(2)}$ is for the second best, and $Blue_{(3)}$ is for the third best.

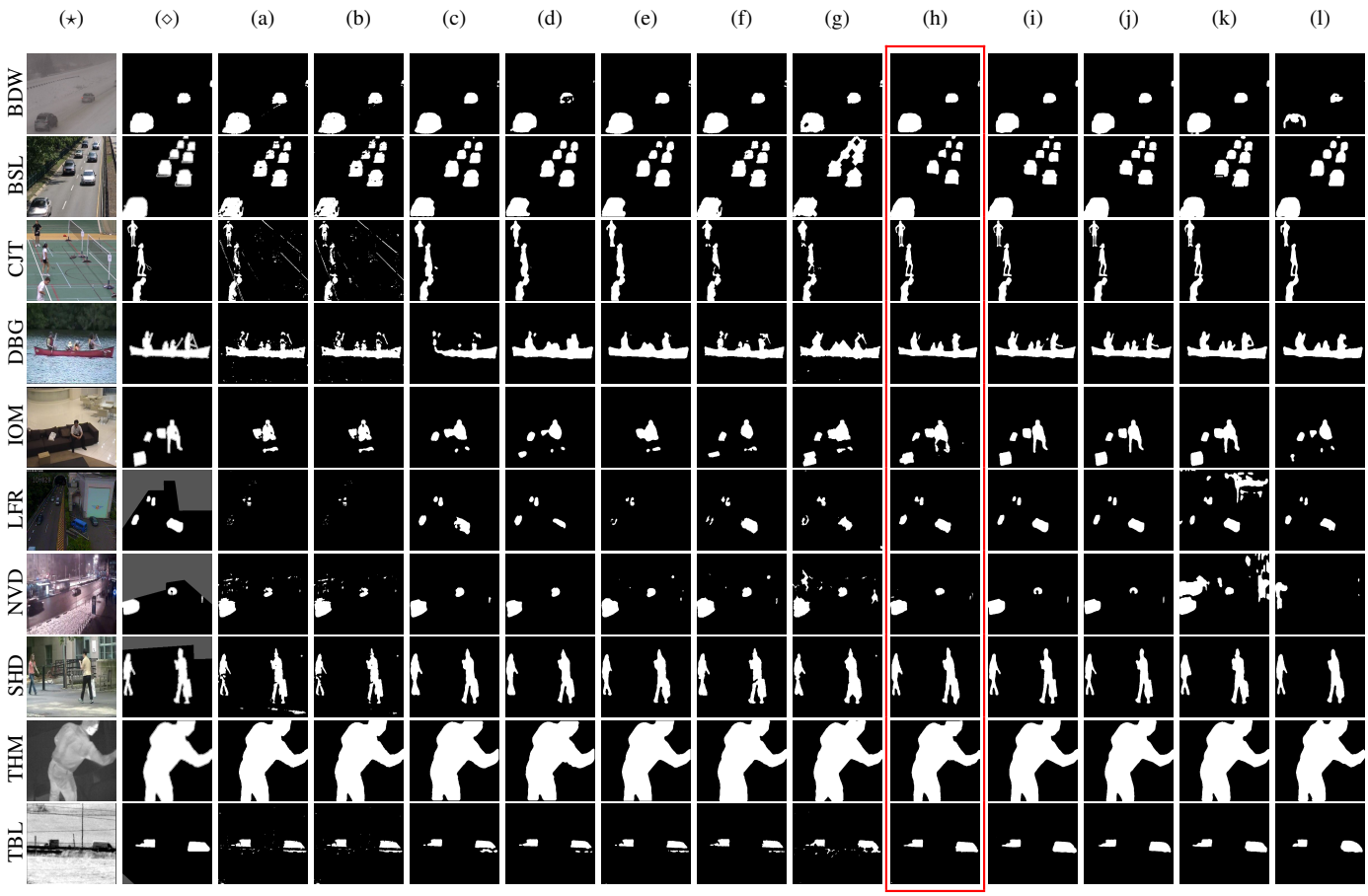


Fig. 4. Visual quality comparison for foreground detection on all video sequences in eleven categories in CDnet 2014. The columns include: (⋆) input frame, (⋄) corresponding groundtruth foreground, (a) GMM – S & G, (b) GMM – Zivkovic, (c) SuBSENSE, (d) PAWCS, (e) BMOG, (f) FTSG, (g) SWCD, (h) CDN-MEDAL-net, (i) FgSegNet_S, (j) FgSegNet_v2 (k) Cascade CNN, (l) DeepBS.

on *BDW*, *SHD*, *IOM* and *CJT*, which may be attributed to TensorMoG carefully tuned hyperparameters on segmenting foreground, thereby suggesting that the proposed method is still limited possibly by its architectural size and training data. Comparison with other unsupervised methods is also conducted, using mathematically rigorous approaches such as SuBSENSE, PAWCS, FTSG, SWCD that are designed to tackle scenarios commonly seen in real life (i.e. *BSL*,

TABLE IV
RESULT OF QUANTITATIVE EVALUATION ON CDNET 2014 DATASET

	Method	*Average Recall*	*Average FPR*	*Average FNR*	*Average PWC*	*Average Precision*
Unsupervised	GMM – S & G	0.6846	0.0250	0.3154	3.7667	0.6025
	GMM – Zivkovic	0.6604	0.0275	0.3396	3.9953	0.5973
	SuBSENSE	0.8124	0.0096	$0.1876_{(1)}$	1.6780	0.7509
	PAWCS	$0.7718_{(3)}$	$0.0051_{(1)}$	0.2282	$1.1992_{(1)}$	$0.7857_{(2)}$
	TensorMoG	$0.7772_{(2)}$	0.0107	$0.2228_{(3)}$	2.3315	$0.8215_{(1)}$
	BMOG	0.7265	0.0187	0.2735	2.9757	0.6981
	FTSG	0.7657	$0.0078_{(3)}$	0.2343	$1.3763_{(3)}$	$0.7696_{(3)}$
	SWCD	$0.7839_{(1)}$	$0.0070_{(2)}$	$0.2161_{(2)}$	$1.3414_{(2)}$	0.7527
*	**CDN-MEDAL-net**	**0.9232**	**0.0039**	**0.0768**	**0.5965**	**0.8724**
Supervised	FgSegNet_S	$0.9896_{(1)}$	$0.0003_{(2)}$	$0.0104_{(1)}$	$0.0461_{(2)}$	0.9751
	FgSegNet	$0.9836_{(3)}$	$0.0002_{(1)}$	$0.0164_{(3)}$	$0.0559_{(3)}$	0.9758
	FgSegNet_v2	$0.9891_{(2)}$	$0.0002_{(1)}$	$0.0109_{(2)}$	$0.0402_{(1)}$	$0.9823_{(2)}$
	Cascade CNN	0.9506	0.0032	0.0494	0.4052	0.8997
	DeepBS	0.7545	0.0095	0.2455	1.9920	0.8332
	STAM	0.9458	$0.0005_{(3)}$	0.0542	0.2293	$0.9851_{(1)}$

*Semi-Unsupervised; In each column, $Red_{(1)}$ is for the best, $Green_{(2)}$ is for the second best, and $Blue_{(3)}$ is for the third best.

DBG, *SHD*, and *BDW*). Nevertheless, F-measure results of the proposed approach around 0.90 suggests that it is still able to outperform these complex unsupervised approaches, possibly ascribing to its use of hand-labeled data for explicitly enabling context capturing.

In comparison with supervised approaches, the proposed approach is apparently very competitive against the more computationally expensive state-of-the-arts. For instance, our approach considerably surpasses the generalistic methods of STAM and DeepBS on *LFR* and *NVD*, but it loses against both of these methods on *SHD* and *CMJ*, and especially is outperformed by STAM on many scenarios. While STAM and DeepBS are constructed using only 5% of CDnet-2014, they demonstrate good generalization capability across multiple scenarios by capturing the holistic features of their training dataset. However, despite being trained on all scenarios, their behaviors showcase higher degrees of instability (e.g. with *LFR*, *NVD*) than our proposed approach on scenarios that deviate from common features of the dataset. Finally, as our proposed method is compared against similarly scene-specific approaches like FgSegNet's, Cascade CNN, the results were within expectations for almost all scenarios that ours would not be significantly outperformed, as the compared models could accommodate various features of each sequence in their big architectures. However, surprisingly, our method surpasses even these computationally expensive to be at the top of the *LFR* scenarios. This suggests that, with a background for facilitating motion segmentation from an input, our trained model can better tackle scenarios where objects are constantly changing and moving than even existing state-of-the-arts.

Overall, these comparisons serve to illustrate the superiority of the proposed approach in terms of accuracy over unsupervised approaches using only small training datasets, while cementing its practical use in its ability to compete with supervised ones despite its light-weighted structure. Table IV presents evaluation metrics of a confusion matrix.

C. Results on Wallflower Benchmarks without Tuning

Using the Wallflower dataset, we aim to empirically determine our proposed approach's effectiveness on unseen sequences, using only trained weights from scenarios of similar dynamics in CDnet-2014. The results apparently tend towards suggesting good degrees of our generalization from trained scenarios over to those unseen. Experimental evaluations are presented in Table V, highlighting the F-measure quantitative results of our approach compared against some state-of-the-art methods in supervised, and unsupervised learning.

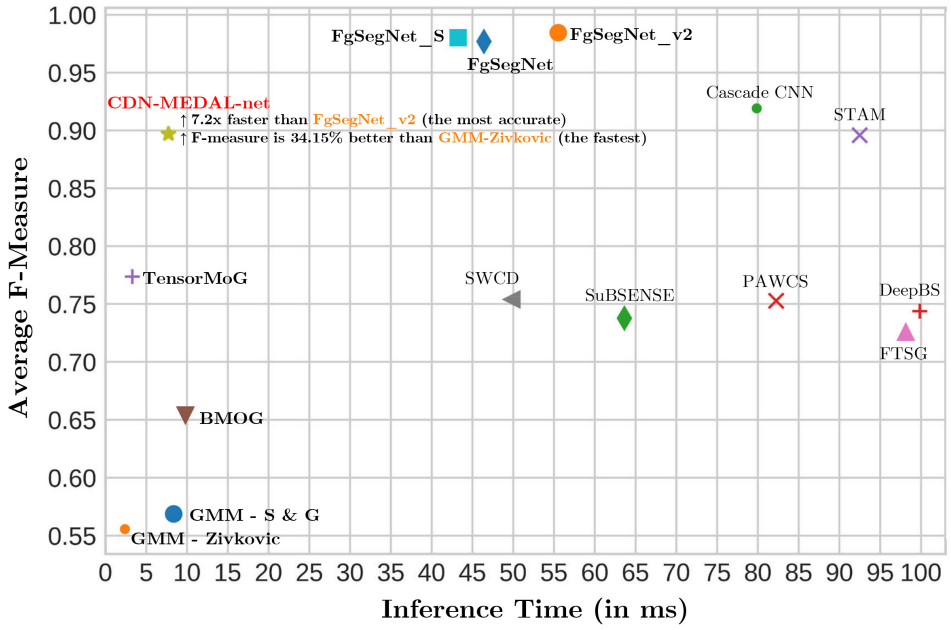
Specifically, on the *Camouflage* scenario, our approach presents a very high score of 0.97 in terms of F-measure using the *copyMachine* sequence of the *SHD* scenario in CDnet-2014. As the model learns to distinguish between object motions and the shadow effects of *copyMachine*, it even extends to recognizing object motions of similar colors. Under Bootstrap where motions are present throughout the sequence, we employ the straight-forward background subtraction function learned via the clear features of static-view-versus-motion of *highway* in *BSL*, giving an F-score of 0.768. Likewise, the model's capture of scene dynamics with *office* of *BSL*, *backdoor* of *SHD* and *fountain02* of *DBG* are extended towards respective views of similar features: *ForegroundAperture* of clear motions against background, *TimeOfDay* where there are gradual illumination changes and *WavingTrees* of dynamic background motions, providing decently accurate results. On the other hand, the *LightSwitch* scenario presents a big challenge where lightings are abruptly changed. As there is no scenario with this effect on the CDnet-2014 dataset, we chose the *SHD* simply for its ability to distinguish objects clearly but the F-measure result is quite poor.

In comparison with existing methods whose aim are towards generalization like some unsupervised approaches GMM – Stauffer & Grimson, SuBSENSE, and CDnet-pretrained supervised approaches STAM, DeepBS, our proposed method

TABLE V
F - MEASURE COMPARISONS OVER THE SIX SEQUENCES OF WALLFLOWER DATASET WITH MODEL PARAMETERS TUNED ON CDNET-2014

	Method	*Bootstrap*	*LightSwitch*	*WavingTrees*	*Camouflage*	*ForegroundAperture*	*TimeOfDay*
∓UnS.	GMM – Stauffer & Grimson	0.5306	0.2296	**0.9767**	0.8307	0.5778	0.7203
	SuBSENSE	0.4192	0.3201	0.9597	0.9535	0.6635	0.7107
*	**CDN-MEDAL-net**	**0.7680**	0.5400	0.8156	0.9700	**0.8401**	**0.7429**
∓Sup.	DeepBS [33]	0.7479	0.6114	0.9546	**0.9857**	0.6583	0.5494
	STAM	0.7414	**0.9090**	0.5325	0.7369	0.8292	0.3429

*Semi-Unsupervised; ∓UnS. = Unsupervised and Sup. = Supervised; In each column, **Bold** is for the best within each scenario.



	Method	Time‡	Avg. FM∓
Unsupervised	GMM – S & G	119.697$_{(3)}$	0.5688
	GMM – Zivkovic	419.950$_{(1)}$	0.5557
	SuBSENSE	15.717	0.7377
	PAWCS	12.159	0.7529$_{(3)}$
	TensorMoG	302.526$_{(2)}$	0.7738$_{(1)}$
	BMOG	102.025	0.6542
	FTSG	10.191	0.7256
	SWCD	20.061	0.7540$_{(2)}$
*	**CDN-MEDAL-net**	**129.451**	**0.8972**
Supervised	FgSegNet_S	23.128$_{(1)}$	0.9803$_{(2)}$
	FgSegNet	21.543$_{(2)}$	0.9771$_{(3)}$
	FgSegNet_v2	18.015$_{(3)}$	0.9847$_{(1)}$
	Cascade CNN	12.521	0.9193
	DeepBS	10.015	0.7439
	STAM	10.812	0.8959

In each column, $Red_{(1)}$ is for the best, $Green_{(2)}$ is for the 2^{nd} best, and $Blue_{(3)}$ is for the 3^{rd} best. *Semi-Unsupervised; ‡The inference time is measured in frame-per-second unit. ∓The average F-measure scores are evaluated on CDnet-2014 dataset

Fig. 5. Computational speed and average F-measure comparison with state-of-the-art methods.

yields very good results on *Camouflage* and *WavingTrees*, with even relatively better results on *Bootstrap*, *ForegroundAperture* and *TimeOfDay*. While obviously this does not evidence that our approach is capable of completely better generalization from training than others, it does suggest that the proposed framework is able to excellently generalize to scenarios with dynamics similar to those learned, as supported by its relatively poor accuracy on *LightSwitch*.

D. Computational Speed Comparison

The proposed framework was implemented on a CUDA-capable machine with an NVIDIA GTX 1070 Ti GPU or similar, along with the methods that require CUDA runtime, i.e., TensorMoG, DeepBS, STAM, FgSegNet, and Cascade CNN. For unsupervised approaches, we conducted our speed tests on the configuration of an Intel Core i7 with 16 GB RAM. Our results are recorded quantitatively with execution performance in frame-per-seconds (FPS), and time (miliseconds) versus accuracy in Fig. 5. At the speed of 129.4510 fps, it is apparent that CDN-MEDAL-net is much faster than other supervised deep learning approaches, of which the fastest - FgSegNet_S - runs at 23.1275 fps. By concatenating estimations of background scenes with raw signals for foreground extraction, our approach makes such efficient use of hardware resources due of its completely lightweight architecture and the latent-space-limitation approach. In contrast, other DNNs architectures are burdened with a large number of trainable parameters to achieve accurate input-target mapping. Furthermore, the proposed scheme dominates the mathematically rigorous unsupervised methods frameworks in terms of speed and accuracy such as SuBSENSE, SWCD, and PAWCS, as their paradigms of sequential processing is penalized by significant penalties in execution. Significantly, the average speeds of the top three methods dramatically disparate. With the objective of parallelizing the traditional imperative outline of rough statistical learning on GMM, TensorMoG reformulates a tensor-based framework that surpasses our duo architectures at 302.5261 fps. On the other hand, GMM - Zivkovic's design focuses on optimizing its mixture components, thereby significantly trading off its accuracy to attain the highest performance. Notwithstanding, our proposed framework gives the most balanced trade-off (top-left-most) in addressing the speed-and-accuracy dilemma. Our model outperforms other approaches of top accuracy ranking when processing at exceptionally high speed, while obtaining good accuracy scores, at over 90% on more than half of CDnet's categories and at least 84%.

V. Conclusion

This paper has proposed a novel, two-stage framework with a GMM-based CNN for background modeling, and a convolutional auto-encoder MEDAL-net to simulate input-background subtraction for foreground detection, thus being considered as a search space limitation approach to compress a model of DNNs, while keeping its high accuracy. Our first and second contributions in this paper include a pixel-wise, light-weighted, feed-forward CNN representing a multi-modular conditional probability density function of the temporal history of data, and a corresponding loss function for the CNN to learn from virtually inexhaustible datasets for approximating the mixture of Gaussian density function. In such a way, the proposed CDN-GM not only gains better capability of adaptation in contextual dynamics with humanly interpretable statistical learning for extension, but it is also designed in the tensor form to exploit technologically parallelizing modern hardware. Secondly, we showed that incorporating such statistical features into MEDAL-net's motion-region extraction phase promises more efficient use of powerful hardware, with prominent speed performance and high accuracy, along a decent generalization ability using a small-scale set of training

labels, in a deep non-linear scheme of only a few thousands of latent parameters.

REFERENCES

[1] S. Zhang, H. Zhou, D. Xu, M. E. Celebi, and T. Bouwmans, "Introduction to the Special Issue on Multimodal Machine Learning for Human Behavior Analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1s, pp. 1–2, apr 2020.

[2] S. Ammar, T. Bouwmans, N. Zaghden, and M. Neji, "Deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance," *IET Image Processing*, vol. 14, no. 8, pp. 1490–1501, jun 2020.

[3] S. Park, M. Ji, and J. Chun, "2D human pose estimation based on object detection using RGB-D information," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 800–816, feb 2018.

[4] Y. C. Bilge, F. Kaya, N. I. Cinbis, U. Celikcan, and H. Sever, "Anomaly detection using improved background subtraction," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, may 2017, pp. 1–4.

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

[6] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. IEEE Comput. Soc, pp. 246–252.

[7] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* IEEE, 2004, pp. 28–31 Vol.2.

[8] I. Martins, P. Carvalho, L. Corte-Real, and J. L. Alba-Castro, "BMOG: boosted Gaussian Mixture Model with controlled complexity for background subtraction," *Pattern Analysis and Applications*, vol. 21, no. 3, pp. 641–654, aug 2018.

[9] S. V.-U. Ha, N. M. Chung, H. N. Phan, and C. T. Nguyen, "TensorMoG: A Tensor-Driven Gaussian Mixture Model with Dynamic Scene Adaptation for Background Modelling," *Sensors*, vol. 20, no. 23, p. 6973, dec 2020.

[10] R. Kalsotra and S. Arora, "A Comprehensive Survey of Video Datasets for Background Subtraction," *IEEE Access*, vol. 7, pp. 59 143–59 171, 2019.

[11] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction:A systematic review and comparative evaluation," *Neural Networks*, vol. 117, pp. 8–66, sep 2019.

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[13] C. Bishop, "Mixture density networks," Tech. Rep. NCRG/94/004, January 1994.

[14] T. Bouwmans, "Traditional Approaches in Background Modeling for Static Cameras," in *Background Modeling and Foreground Detection for Video Surveillance*. Chapman and Hall/CRC, aug 2014, pp. 1–1–1–54.

[15] B. Garcia-Garcia, T. Bouwmans, and A. J. Rosales Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, p. 100204, feb 2020.

[16] J. D. Pulgarin-Giraldo, A. Alvarez-Meza, D. Insuasti-Ceballos, T. Bouwmans, and G. Castellanos-Dominguez, "GMM Background Modeling Using Divergence-Based Weight Updating," 2017, pp. 282–290.

[17] S. Viet-Uyen Ha, D. Nguyen-Ngoc Tran, T. P. Nguyen, and S. Vu-Truong Dao, "High variation removal for background subtraction in traffic surveillance systems," *IET Computer Vision*, vol. 12, no. 8, pp. 1163–1170, dec 2018.

[18] X. Lu, C. Xu, L. Wang, and L. Teng, "Improved background subtraction method for detecting moving objects based on GMM," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 11, pp. 1540–1550, nov 2018.

[19] D. Sowmiya and P. Anandhakumar, "Cauchy Mixture Model-based Foreground Object Detection with New Dynamic Learning Rate Using Spatial and Statistical information for Video Surveillance Applications," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, vol. 90, no. 5, pp. 911–924, dec 2020.

[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, may 2016, pp. 1–4.

[22] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, sep 2017.

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

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[25] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635–649, apr 2018.

[26] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, jan 2015.

[27] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, jun 2014, pp. 420–424.

[28] T. P. Nguyen, C. C. Pham, S. V.-U. Ha, and J. W. Jeon, "Change Detection by Training a Triplet Network for Motion Feature Extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 433–446, feb 2019.

[29] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, sep 2018.

[30] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise Deep Sequence Learning for Moving Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2567–2579, sep 2019.

[31] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 3123–3131.

[32] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, jan 1989.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," apr 2017.

[34] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 4105–4113.

[35] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A Self-Adjusting Approach to Change Detection Based on Background Word Consensus," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, jan 2015, pp. 990–997.

[36] S. Isik, K. Özkan, S. Günal, and Ömer Nezih Gerek, "SWCD: a sliding window and self-regulated learning-based background updating method for change detection in videos," *Journal of Electronic Imaging*, vol. 27, no. 2, pp. 1 – 11, 2018.

[37] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, sep 2018.

[38] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, vol. 23, no. 3, pp. 1369–1380, aug 2020.

[39] D. Liang, J. Pan, H. Sun, and H. Zhou, "Spatio-Temporal Attention Model for Foreground Detection in Cross-Scene Surveillance Videos," *Sensors*, vol. 19, no. 23, p. 5142, nov 2019.

[40] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, jun 2014, pp. 393–400.

[41] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999, pp. 255–261 vol.1.