CDN-MEDAL: Dual Convolutional Networks on Gaussian Mixture Density for Motion Segmentation

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

***Abstract*—Background modeling has become an emerging 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 insufficient properties of the observed scenes 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, in combination with a foreground extraction step. In this work, we propose a novel, two-stage method of change detection with two convolutional neural networks. One of the architectures simulates the statistical learning of a Gaussian mixture to obtain a statistical inference on scene analysis, while the other implements a 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 regarding background construction and foreground segmentation.**

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

I. Introduction

With the swift progress in computer vision, vision-based surveillance systems with static cameras have become promising technologies that are carried out advanced tasks such as behavior analysis [1], object segmentation and classification [2]. In pursuing these tasks, background subtraction methods are the most widely used approaches for motion detection. However, it is unavoidable that a deployed system has to deal with the visual multimodality under various conditions and cameras' shaking. Regarding this issue, background modeling is a pivotal component when it comes to proper analysis of these input signals to enable a fundamental understanding of a video sequence. Literally, a background is simply a scene containing stationary objects and components that are not of interest to the system (e.g., streets, houses, trees). From the background image, desired objects (e.g., cars, pedestrians) are segregated and localized via considering a set of visual signals (pixels) against its background in binary masks, called foregrounds. The construction of background depends on the algorithmic handling of various situations [3], including illumination changes, scene dynamics, bootstrapping, camouflage, and foreground aperture.

Extensive research has been conducted to address the multimodality of background modeling. With the increasing advancement of specialized processing units for large-scale data, Deep Neural Networks (DNNs) have emerged as a prominent pattern matching and a visual prediction mechanism. Studies employing DNNs for background modeling to conveniently limit search spaces are dramatically few compared to directly localizing objects with architectures of massive nonlinear approximation. However, regarding the problem of background modeling and foreground detection, DNNs have experienced two primary shortcomings as follows:

A compulsory requirement of a huge-scale dataset of labelled images: DNNs-based models for motion detection exploit weak statistical regularities between input sequences of images and annotated background scenes. Thus, in order to generalize to all practical scenes in real life, a prohibitively large dataset consisting of all practical scenarios and effects is needed. Because there are very few training labels in video sequences for building generalized background models [4], there is nothing but data-driven experiments assure that the scenes' true properties are appropriately presented from the spurious regularities gained by the sampling peculiarities of the training set.

A prevailing fail on contextual variation: Recently, foreground segmentation has been considered from the perspective of binary classification schemes, which have been proposed to minimize a sum-of-squares or a cross-entropy error function in DNNs-based approaches to reflect the true objective of the problem as closely as possible. Accordingly, models are usually trained to optimize the statistical properties on the training data when the actual aim is to generalize well to the target scenes. This conditional average will be inadequate for many contextual dynamics [5], [6]. Hence, DNNs-based methods perform well on experimental datasets of background modeling and change detection but mostly fail on unseen situations in real-world scenarios.

To address the above issues, in this article, we introduce a novel, two-stage framework employing two convolutional neural networks (CNNs). The first module presents the conditional probability distribution of observed scenes grounded on a Gaussian Mixture Model (GMM), called Convolutional Density Network of Gaussian Mixture (CDN-GM). The second one calculates foreground via Motion Estimation with Differencing Approximation via Learning on a network of convolutional autoencoder (MEDAL-net). Notably, the CDN-

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

Corresponding email: hvusynh@hcmiu.edu.vn

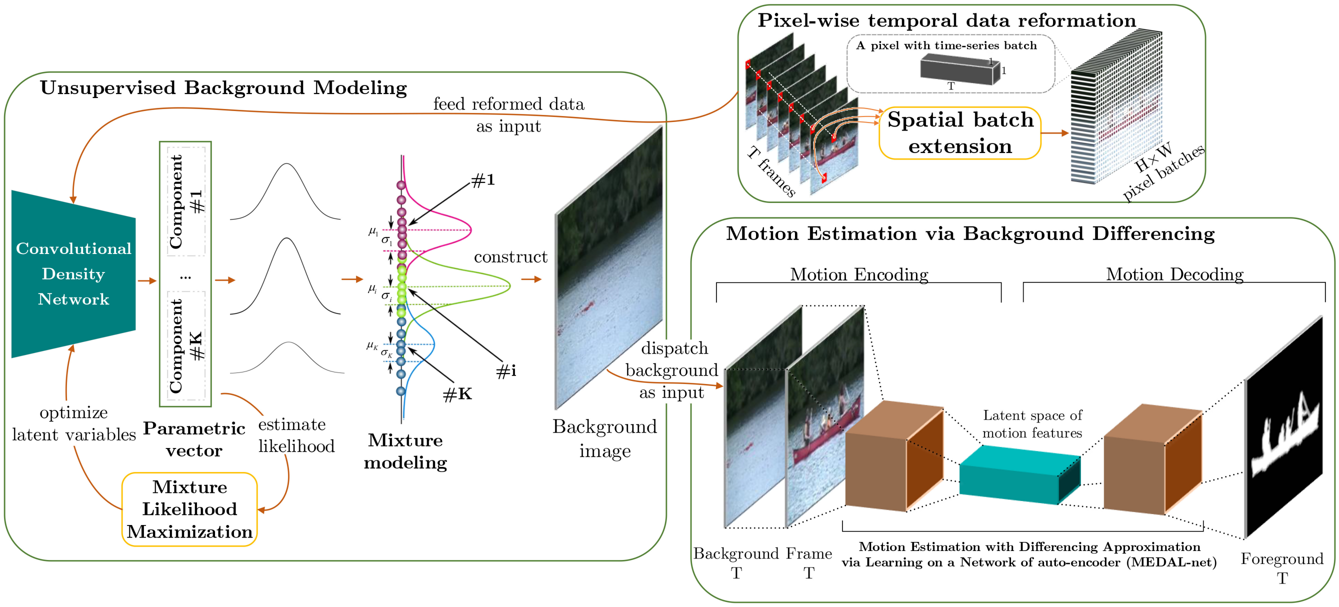
Unsupervised Background Modeling
feed reformed data as input
Convolutional Density Network
Component #1
...
Component #K
Parametric vector
optimize latent variables
estimate likelihood
Mixture Likelihood Maximization
#1
#i
#K
μ_1
σ_1
μ_i
σ_i
μ_K
σ_K
Mixture modeling
construct
Background image
Pixel-wise temporal data reformation
A pixel with time-series batch
T
1
T frames
Spatial batch extension
H×W pixel batches
Motion Estimation via Background Differencing
Motion Encoding
Motion Decoding
dispatch background as input
Latent space of motion features
Background T
Frame T
Motion Estimation with Differencing Approximation via Learning on a Network of auto-encoder (MEDAL-net)
Foreground T

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

GM was inspired by a conventional fully-connected neural network regulating a one-to-many or multi-valued mapping in a problem of robot inverse kinematics [7]. The key idea of our proposed framework is to train a CNN with a loss function of minimizing the dissimilarity between the empirical distribution to obtain the statistical formalism in temporal dimension at each image point, and to use the extracted statistical dynamics of the current scene to limit generalization search space of the foreground extraction module, i.e., of MEDAL-net.

The primary goal of this model is to address the problems of DNNs as we mentioned above via adaptively acquiring the underlying properties of a sequence of images to construct corresponding background scenes and foreground images at concrete moments rather than memorize the single-valued mapping between input frames and labeled foregrounds. The overview of our proposed framework is presented in Fig. 1. In summary, our contributions are summarized as follows:

Firstly, with an assumption that the background scene of a video sequence contains the most commonly seen intensity value at each image point, we propose a feed-forward CNNs representing a conditional probability density function that models the time-related history of every pixel location in the first pipeline of the proposed framework. In this architecture, conditioned on pixel-wise vectors of intensity values over a time period, the network's outputs determine the kernels of a Mixture of Gaussians. Accordingly, the mixture is accentuated by the frequency of pixel-wise values with which each of Gaussian components characterizes the background.

Secondly, with the goal of modeling the underlying generator of the data, we propose a loss function in the manner of unsupervised learning so that the most likely background description of actually observed data can be made when the trained network is subsequently presented with a new value of input. The proposed model not only achieves higher degrees of interpretability compared to the idea of estimating an implicit hidden function in previous neural network methods but also gains better capability of adaptation in contextual dynamics with statistical learning in the proposed DNNs.

Thirdly, to come up with a more generalizable and light-weighted foreground extraction module, we design a compact convolutional auto-encoder which simulates a difference mapping between input frames and corresponding background scenes. This network makes use of features of static components in images from the first module of background modeling. Although trained with scene-specific data, the architecture maintains good generalization to different sequences including unseen situations with 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 [5], [8], [9]. Among published methods that meet the requirements of robustness, adaptation to scene dynamics, memory efficiency, and real-time processing, two unique 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 the input sequence of observed scenes and the hand-labeled background/foreground images on non-linear regularities.

In statistical approaches, the visual features of image points are modeled with an explainable probabilistic foundation regarding either pixel-level or region-level in the perspective of temporal and spatial resolutions. In the last decades, there have been a variety of literature-inspired statistical models that were proposed to resolve the problem of background initialization. Stauffer and Grimson [10] proposed a pioneering work that handles gradual changes in outdoor scenes using pixel-level Mixtures of Gaussians (MOG) with a sequential framework of K-means-based distribution matching. To enhance the foreground/background discrimination ability regarding scene dynamics, Pulgarin-Giraldo *et al.* [11] improved GMM with a contextual sensitivity that used Euclidean-based Least Mean Square to update the framework of parameters estimation. Another modification on CIE L*a*b* color space is Boosted Gaussian Mixture Model (BMOG) [12] which was introduced to investigate the adaptation of GMM with different color schemes. To enhance the performance, Lu *et al.* [13] applied a median filter on the input frame to reduce the spatial dimension of the image before performing the background initialization. Validating the robustness of background modeling in a high amount of dynamic scene changes, Ha *et al.* [14] proposed a GMM with a paradigm of high variation removal by incorporating entropy estimation. In addition to GMM, Cauchy Mixture Models (CMM) was exploited to detect foreground objects via eliminating noise and capturing periodical perturbations in illumination-change situations and dynamic scenarios [15]. There are also other developments in background modeling with fuzzy concepts recently. A post-processing scheme that utilizes Gibb's Markov Random Field fuzzy clustering and gray level co-occurrence matrix features [16] was proposed to more effectively remove the moving shadows out of the moving objects. Zeng *et al.* [17] introduced an adaptive histogram learning method where histogram accumulation is monitored by a fuzzy controller to address the susceptibility to outliers and the randomness of histogram partitioning. To improve the quality of foreground segmentation, Yu *et al.* [18] adopted a fuzzy adaptive background maintenance method with a dynamic fuzzy nearness degree threshold update. Subudhi *et al.* [19] were one of the groundbreakers who used higher space of kernelized fuzzy set-theoretic method via the projection of the highly non-linear 3-dimensional feature space to classify foreground-background image points. Considering the varying degree of participation of pixels in a particular region instead of trying to capture spatial information of each region, a new fuzzy-based feature descriptor on each image pixel was proposed by Giveki *et al.* [20] with an assumption that the participation is the same for all pixels in a neighborhood. Overall, statistical models were developed with explicit probabilistic hypotheses to present the correlation of history observation at each image point or a pixel block. The evaluation of these methods usually leads to a compromise between the segregation of slow-moving objects and rapid adaptation to sudden dynamics in scene changes within short-term measurement. In practice, the intervention in the model's update process is a sensitive concern in multi-contextual scenarios.

Recently, there have also been many attempts to apply different types of deep neural networks into background subtraction and background modeling problems by virtue of self-organized mapping in supervised learning. Inspired from LeNet-5 [21], a network architecture for handwritten digit classification, one of the earliest efforts to subtract the background from the input image frame using in this approach was done by Braham *et al.* [22]. The work mainly concentrated on exploring the potential of visual features learned by hidden layers to construct foreground images with a binary classification scheme. Similarly, Wang *et al.* proposed a deep convolutional neural network that was trained on only a small subset of frames as there is a large redundancy in a video taken by surveillance systems [23]. The model requires a hand-labeled segmentation of motion masks as a subjective indicator in observed scenes. Lim *et al.* [24] constructed an encoder-decoder architecture with the encoder inherited from VGG-16 [25]. The network takes the current video frame, the previous frame, and the grayscale background model as the input to estimate the compressed representation of the input and to deconvolve the compressed feature map into a foreground segmentation map. Another method is DeepBS [26] which was proposed by Babaee *et al.* to compute the background model using both SuBSENSE [27] and Flux Tensor method [28]. The authors used a small patch of the input scenes and the corresponding background images as input for visual feature maps extraction, and then, the output layers are combined and post-processed to give the final segmentation map. Another method that used an autoencoder in terms of a triplet CNN to learn multi-scale hidden representations of observed scenes was proposed as FgSegNet [29]. The final output of the decoder in FgSegNet is the extracted foreground binary map from the latent visual features. Chen *et al.* were one of the first who introduced a pixel-wise deep sequence learning architecture with attention mechanism and long short-term memory (LSTM) to incorporate into an architecture of ConvLSTM [30]. This technique aims to exploit the high-level features of spatio-temporal information at pixel level. Recently, Akilan *et al.* [31] demonstrate the effectiveness of ConvLSTM via the consideration of operations in 3D space. The model employed both 3D convolution and ConvLSTM operations to extract both short-term and long-term temporal features with double-encoding and slow-decoding. Another approach that adopts generative adversarial networks (GAN) was introduced by Sultana *et al.* [32]. The authors utilized two CNNs, which are a context prediction network to estimate the background in regions with moving objects removed after a pre-processing step and a texture optimizing network to enhance the predicted context. Particularly, the whole method is performed in an unsupervised way. An architecture of background modeling with a self-organizing neural network was investigated in RGBD-SOBS [33]. The work modeled the video frame's color and depth information separately to give the foreground mask in each model.

All things considered, most neural-network-based methods are benefited 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. These models take advantage of the parallel mechanism of highly-optimized processing units (e.g., TPU and GPU) for extensive data learning. Nevertheless, nothing in this supervised learning approach proves that deep neural networks possess true properties of the scene from the sampling peculiarities of the training set. Furthermore, the training time and the performance in deployment of these methods are usually costly. In other words, they did not ensure real-time performance, which is a crucial requirement for any practical system. Therefore, adaptability and real-time learning capability regarding supervised deep learning methods is a significant challenge in video analysis. Fundamentally, regarding the problems of predicting the target foreground and background images, however, the conditional average represents a very limited description of the statistical properties of the target scenes, and for many contextual dynamics, this will be wholly inadequate. In this work, we propose a novel compact framework of CNN-based unsupervised background construction that exploits temporal information with light-weighted convolutional non-linear difference filtering to overcome the above drawbacks of previous studies in background subtraction, focusing on both background and foreground modeling. Comprehensively, the introduced convolutional background model captures attention towards static distributions while the CNN-based encoder-decoder filter focuses on representing the more general subtraction function, which is an idea commonly employed in post-processing steps of statistical methods to extract moving regions. The conditional density functions represented by CDN are modeled in a completely general framework by combining the convention of CNN and a mixture of probabilistic functions.

III. The Proposed Method

A. Convolution Density Network of Gaussian Mixture

According to Zivkovic's study [34], 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 $\boldsymbol{\theta}_k$:

$$\sum_{k=1}^{K} \pi_k = 1 \tag{2}$$

Because of the modality of observed scenes, the intensity of target pixels is assumed to be distributed normally. 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 [10]. Hence, the multivariate Gaussian distribution can be re-formulated with restricted attention as:

$$\begin{aligned} P(\mathbf{x}|\boldsymbol{\theta}_k) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^c \sigma_k^c}} \exp\left(-\frac{\| \mathbf{x} - \boldsymbol{\mu}_k \|^2}{2\sigma_k}\right) \end{aligned} \tag{3}$$

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 that formulates a conditional formalism of GMM density function of $\mathbf{x}$ given a set of randomly selected, vectorized data points $\boldsymbol{\chi}_T$, which is called Convolutional Density Network (CDN):

$$\mathbf{y}_T = f_\theta(\boldsymbol{\chi}_c^T) \sim P(\mathbf{x}|\boldsymbol{\chi}_\mathbf{c}^\mathbf{T}) \tag{4}$$

where $f_\theta\left(\cdot\right)$ is a set of non-linearity transformations

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 [7]. In the proposed scheme, the network performs learning in the feature extractor itself to formulate the statistical inference on time series of intensity values. First, regarding the recently proposed CNNs, the characteristics of local connectivity in convolution layers motivate the network to learn the common visual patterns in a local region of image. Literally, the background image contains most frequently presented intensities in the sequence of observed scenes. Hence, in CDN, 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. But the main deficiency of unstructured nets is that they have no built-in invariance with respect to translations or local distortions of the inputs. 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 lighter and exploiting the parallel processing of a set of multiple pixel-wise analysis within a batch of images.

The architecture of CDN contains seven learned layers, not counting the input – two depthwise convolutional, two convolutional and three dense layers. The our network is summarized in Fig. 2. The input of our rudimentary architecture of proposed network is a time series of color intensity at each image point, 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

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	$(H * W) \times K$

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 is to construct an architecture of CNN which presents multivariate mapping in forms of Gaussian Mixture Model with the mechanism of offline learning. Regarding this perspective, it is critical to ensure the generalization of the proposed probabilistic formulation inside the CNN. In other words, the regularities in the proposed CNN should cover generalized presentation of observed arrangement of image intensity at pixel level with a degree of sufficiency. 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 architecture is extended through a spatial extension of temporal data at image points. The extensive architecture is 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:

$$\begin{aligned} \mathbf{y}_T &= \left[\mathbf{y}_\mu\left(\boldsymbol{\chi}_c^T\right), \mathbf{y}_\sigma\left(\boldsymbol{\chi}_c^T\right), \mathbf{y}_\pi\left(\boldsymbol{\chi}_c^T\right)\right] \\ &= \left[\mathbf{y}_\mu^1, \ldots, \mathbf{y}_\mu^K, \mathbf{y}_\sigma^1, \ldots, \mathbf{y}_\sigma^K, \mathbf{y}_\pi^1, \ldots, \mathbf{y}_\pi^K\right] \end{aligned} \tag{5}$$

With our goal of formulating the GMM, we impose 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, it is crucial that they are required to be defined as independent and identically distributed probabilities. To achieve this regulation, in principle, we activate the network output with a softmax activation function:

$$\pi_k(\boldsymbol{\chi}_c^T) = \frac{\exp(\mathbf{y}_\pi^k)}{\sum_{l=1}^K \exp(\mathbf{y}_\pi^l)} \tag{6}$$

- Second, in the realistic scenarios, the measure intensity of observed image signals may fluctuate due to a variety of factors including illumination transformations, 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 the components do not span the entire color space.

$$\sigma_k(\boldsymbol{\chi}_c^T) = \frac{\bar{\sigma}_{min} \times (1 - \hat{\sigma}_k) + \bar{\sigma}_{max} \times \hat{\sigma}_k}{255} \tag{7}$$

where $\hat{\sigma}_k$ is the normalized variance that was activated through a hard-sigmoid function from the output neurons $\mathbf{y}_\sigma$ that correspond to the variances:

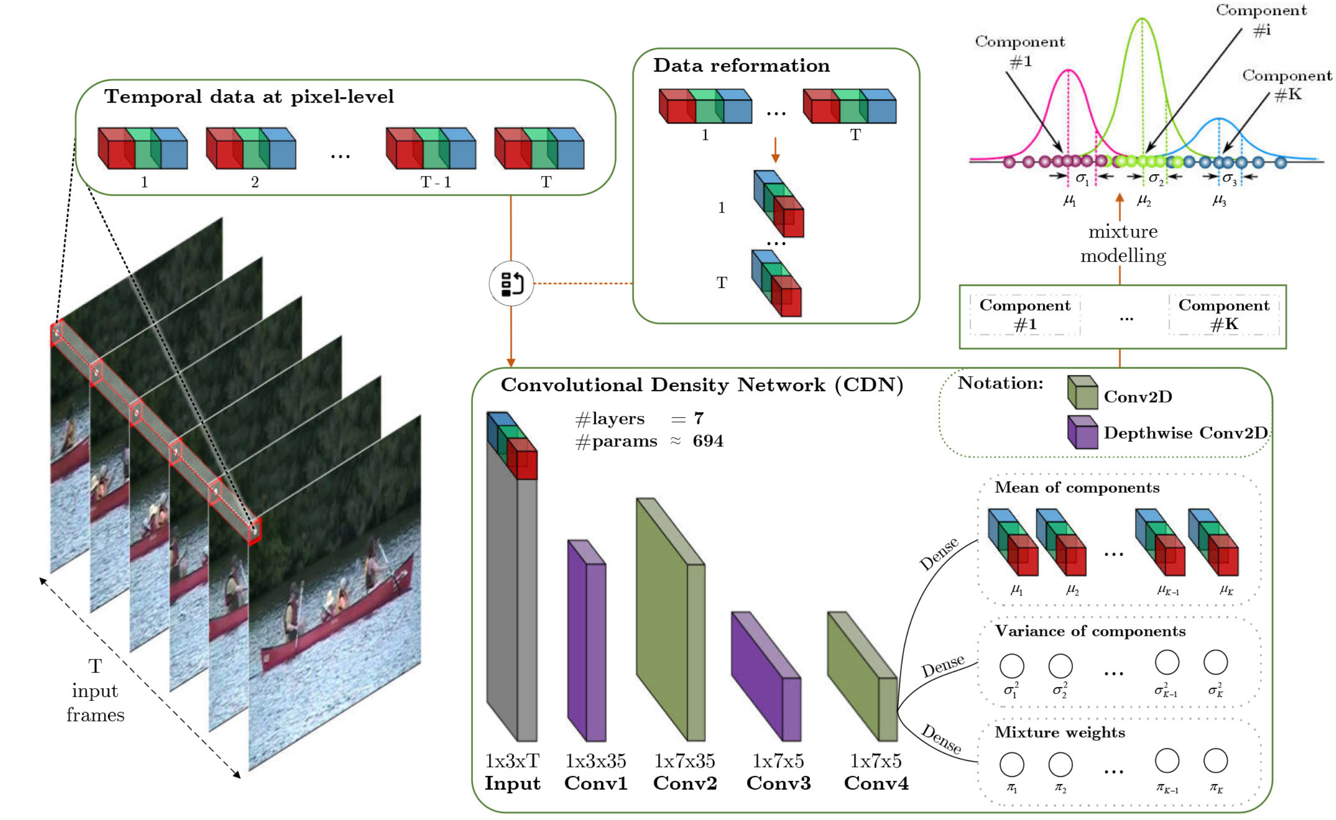

Fig. 2. The proposed architecture of Convolution Density Network of Gaussian Mixture Model

$$\hat{\sigma}_k(\boldsymbol{\chi}_c^T) = \begin{cases} 0, & \text{if } \mathbf{y}_\sigma^k < -2.5 \\ 0.2 \times \mathbf{y}_\sigma^k + 0.5, & \text{if } -2.5 \leq \mathbf{y}_\sigma^k \leq 2.5 \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

- Third, the mean of 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:

$$\mu_k(\boldsymbol{\chi}_c^T) = \begin{cases} 0, & \text{if } \mathbf{y}_\mu^k < -2.5 \\ 0.2 \times \mathbf{y}_\mu^k + 0.5, & \text{if } -2.5 \leq \mathbf{y}_\mu^k \leq 2.5 \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

We choose the hard-sigmoid function for the means and the variances because of the piecewise linear property and correspondence to the bounded form of linear rectifier function (ReLU) of the technique [35].

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.

$$BG(\boldsymbol{\chi}_c^T) = \max(\mu_k \cdot \hat{BG}_{k,T}), \quad \text{for } k \in [1, K] \tag{10}$$

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

$$\hat{BG}_{k,T}(\boldsymbol{\chi}_c^T) = \begin{cases} 1, & \text{if } \underset{k'}{argmax}[{}^{\pi_{k'}(\boldsymbol{\chi}_c^T)}/_{\sigma_{k'}(\boldsymbol{\chi}_c^T)}] = k \\ & \qquad\qquad \text{for } k \in [1, K] \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

B. The unsupervised loss function of CDN-GM

In practice, particularly to each real-life scenario, there are multiple degrees of dynamics that the background model is required to capture. However, the background model is further challenged by the fact that scene dynamics may also change gradually under the influences of external effects such as lighting deviations, weather conditions, or object displacements. These effects convey the latest information regarding contextual deviations that may constitute new background predictions. Therefore, a functional background model must not only be capable of capturing the multi-modular distribution from the dynamics of its data source, but it must also be able to adaptively include into the static view new information for the distribution of interest. In order to serve its part as a statistical mapping function that can be used for background modeling, the proposed neural network function has to be capable of approximating a probability density function and estimating the relevant data distribution. The criteria can be summarized as follows for the neural statistical function to be instituted:

- 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 that is 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 neural function provides an estimate of the probabilistic density function on the provided data using its MOG 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:

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{\chi}_c^T) \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) \tag{12}$$

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

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k) = \frac{1}{\sqrt{2\pi \cdot \sigma_k(\boldsymbol{\chi}_c^T)^2}} \cdot \exp\left\{ -\frac{\left\|\mathbf{x} - \mu_k(\boldsymbol{\chi}_c^T)\right\|^2}{2\sigma_k(\boldsymbol{\chi}_c^T)^2} \right\} \tag{13}$$

In our proposed loss function, the data distributions to be approximated are the sets of data points that are relevant to background construction themselves. 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 for capturing the various pixel-wise dynamics over a complete scene, and for encompassing even unseen perspectives via exploiting the huge coverage of multiple scenarios across more than one case with data. In other words, the input data do not matter in terms of order 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 color space, which would require optimizing the loss not just on any single pixel, but for b block of time-series image data fairly into a summation value.

$$\mathcal{L} = \sum_{i}^{b} \sum_{j}^{T} \mathcal{L}_j^{(i)} \tag{14}$$

where

$$\mathcal{L}_j^{(i)} = -\ln \left(\sum_{k=1}^{K} \pi_k^{(i)} \mathcal{N}(\mathbf{x}_j | \mu_k^{(i)}, \sigma_k^{(i)}) \right) \tag{15}$$

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 MoG probability density function. In this way, the neural function's accuracy performance is founded on bases of virtually complete datasets, whilst being completely unsupervised, for learning multi-modular statistical mapping, and not just for learning to model backgrounds which may lead to context-dependent parameters limited in terms of generalization capability. As we predict the multi-modular form of the input data via the Mixture of Gaussians, the background estimation can be trivially extracted as the most probable component in the mixture.

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'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:

$$\Pi_k^{(i)} = \frac{\pi_k^{(i)} \mathcal{N}(\mathbf{x}_j | \boldsymbol{\mu}_k^{(i)}, \sigma_k^{(i)})}{\sum\limits_{l=1}^{K} \pi_l^{(i)} \mathcal{N}(\mathbf{x}_j | \boldsymbol{\mu}_l^{(i)}, \sigma_l^{(i)})} \tag{16}$$

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. (15) and (16), we obtain:

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \pi_k^{(i)}} = \frac{\Pi_k^{(i)}}{\pi_k^{(i)}} \tag{17}$$

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. (6). We continue to examine the partial derivative of $\pi_k^{(i)}$ with respect to a particular network output $\mathbf{y}_\pi^{(l)}$, which is

$$\frac{\partial \pi_k^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} = \begin{cases} \pi_k^{(i)}(1 - \pi_l^{(i)}), & \text{if } k = l \\ -\pi_l^{(i)} \pi_k^{(i)}, & \text{otherwise.} \end{cases} \tag{18}$$

By chain rule, we have

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

From Eq. (16), (17), (18), and (19), we then obtain

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\pi^{(l)}} = \pi_l^{(i)} - \Pi_l^{(i)} \tag{20}$$

For $\mathbf{y}_\sigma^{(k)}$, we make use of Eq. (3), (7), (8), (15), and (16), by differentiation, to obtain

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\sigma^{(k)}} = \frac{3.2}{255} \Pi_k^{(i)} \left(\frac{c}{2} \sqrt{(2\pi)^c \left(\sigma_k^{(i)}\right)^{c+2}} - \frac{\| \mathbf{x}_j - \boldsymbol{\mu}_k \|^2}{2(2\pi)^c \left(\sigma_k^{(i)}\right)^{c+2}} \right) \tag{21}$$

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. (3), (9), (15), (16) into the differentiation process:

$$\frac{\partial \mathcal{L}_j^{(i)}}{\partial \mathbf{y}_\mu^{(k)}} = 0.2 \times \Pi_k^{(i)} \frac{x_{j,l} - \mu_{k,l}^{(i)}}{\sigma_k^{(i)}} \tag{22}$$

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

From Eq. (20), (21), and (22), when CDN-BM 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 train 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 DNN 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 [10] employed variance thresholding on background - input pairs by modeling the static view with the Gaussian Mixture Model. Whilst 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 an input - background subtraction function 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 the many scenarios, following the Universal approximation theorem [36] and existing literature, we design the technologically parallelizable neural function for an approximation of such framework. Specifically, we make use of a convolutional architecture to construct the Foreground Detection Network. The motive is also 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 CNNs 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 background - input pairs as inputs to the neural function and extract motion estimations. 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 parameters size, width, and depth to focus on learning representations whilst maintaining generalization for unseen cases. As empirically shown in the experiments, the proposed architecture is very 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 FDNet 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 extraction function represented by the decoder, thereby generating foreground masks.

In our design, we fully utilize the use of depthwise separable convolution introduced in MobileNets [37] 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 about 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, whilst formulating normalization to boost the decoder's learning.

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

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 the image-background pairs concatenated along the depth dimension as its inputs. Specifically, the background model estimated by CDN 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 model 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 predictions, the neural network may further learn to recognize motions and changes that are innate to a scene, thereby selectively segmenting motions of interest based on the context.

In addition, by explicitly providing image-background pairs 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 size of the weights of the 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 the 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 [38]. 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:

$$F = \{(x, y, z) | \mathbf{X}_{x,y,z,0} \geq \epsilon\} \tag{23}$$

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:

$$\hat{\mathbf{Y}}_{i,j,k} = \begin{cases} 1, & (i, j, k) \in F \\ 0, & otherwise \end{cases} \tag{24}$$

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

2) Training:

a) Data preparation: The training data for the network is carefully chosen by hand so that the data maintains the balance between the background labels and foreground labels since imbalance data will increase the model 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 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 commonly used for segmentation tasks $[x, y, z]$, as the goal of the model is to

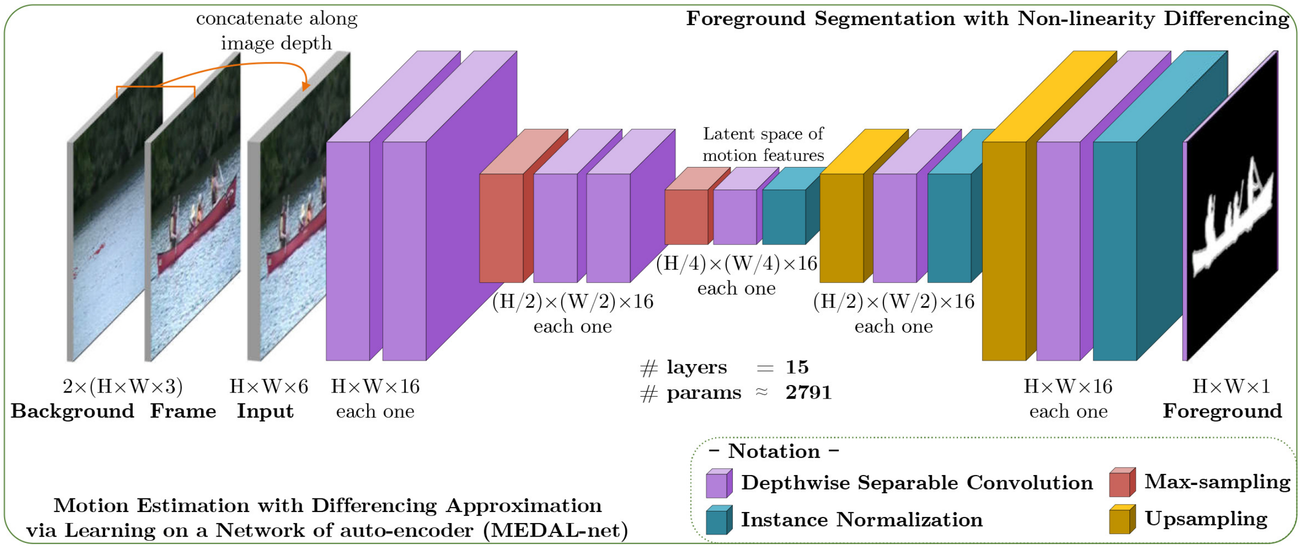

Fig. 3. The proposed architecture of MEDAL-net grounded on convolutional autoencoder for foreground detection

learn a Dirac delta function for each pixel. The description of the loss function is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W}[\mathbf{Y}_{i,j,k}\log(\hat{\mathbf{Y}}_{i,j,k}) + (1-\mathbf{Y}_{i,j,k})\log(1-\hat{\mathbf{Y}}_{i,j,k})] \tag{25}$$

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

With this straightforward learning objective applied on our convolutional neural network, the designed architecture is enabled to learn not only the pixel-wise motion estimates of the training data, but it also is taught to recognize the 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 demonstrate the effectiveness of the proposed CDN-GM and MEDAL-net, which are designed to explicitly incorporate the probabilistic density properties into the architecture to achieve accurate adaptiveness, whilst 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 results of the proposed framework not only with unsupervised approaches that are light-weighted and generalizable without pretraining: GMM – Stauffer & Grimson [10], GMM – Zivkovic [34], SuBSENSE [27], PAWCS [39], TensorMoG [40], BMOG [12], FTSG [28], SWCD [41], but also with the data-driven, supervised models which trade computational expenses for high accuracy performance: FgSegNet_S [29], FgSegNet [42], FgSegNet_v2 [43], Cascade CNN [23], DeepBS [26], STAM [44].

In terms of chosen metrics for measuring the features of motion and spatial changes, 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 [45] were performed by comparing foreground predictions against provided ground-truths. Detailed analysis is illustrated in the next subsection, where we will mix quantitative analysis with qualitative linking. Then, we evaluate the proposed framework trained with CDNet-2014 on Wallflower [46] 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 to draw final conclusions.

In our experiment, K is empirically and heuristically determined by the CDN-GM's capability of modeling constantly evolving contexts (e.g. moving body of water) under the effects of potentially corruptive noises such that the framework utilizes reasonable space and time complexity where K is a scaling factor. As K is greater, there are many GMM components that are either unused or they simply capture the various noises presenting within the imaging sequences, especially contextual dynamics. This is because the background component would only revolve around the most frequently occuring color subspaces to draw predictions; so the extra components would serve as either placeholders for abrupt changes in backgrounds, be empty or capture intermittent noises of various degrees. In addition, using a greater value of K would increase the probability that observed intensiy values match with the Gaussian mixture. In practice, noise Gaussian components in GMMs are pulse-like as they would appear for short durations, and low-weighted because they are not as often matched as background components. Hence, when K is great enough, the cumulative combination of non-background Gaussian components in the mixture would greatly reduce the non-matching probability if certain minimum variances are introduced. Our proposed CDN-GM model was set up with the number of Gaussian components $K = 4$ 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 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 Benchmark

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

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.

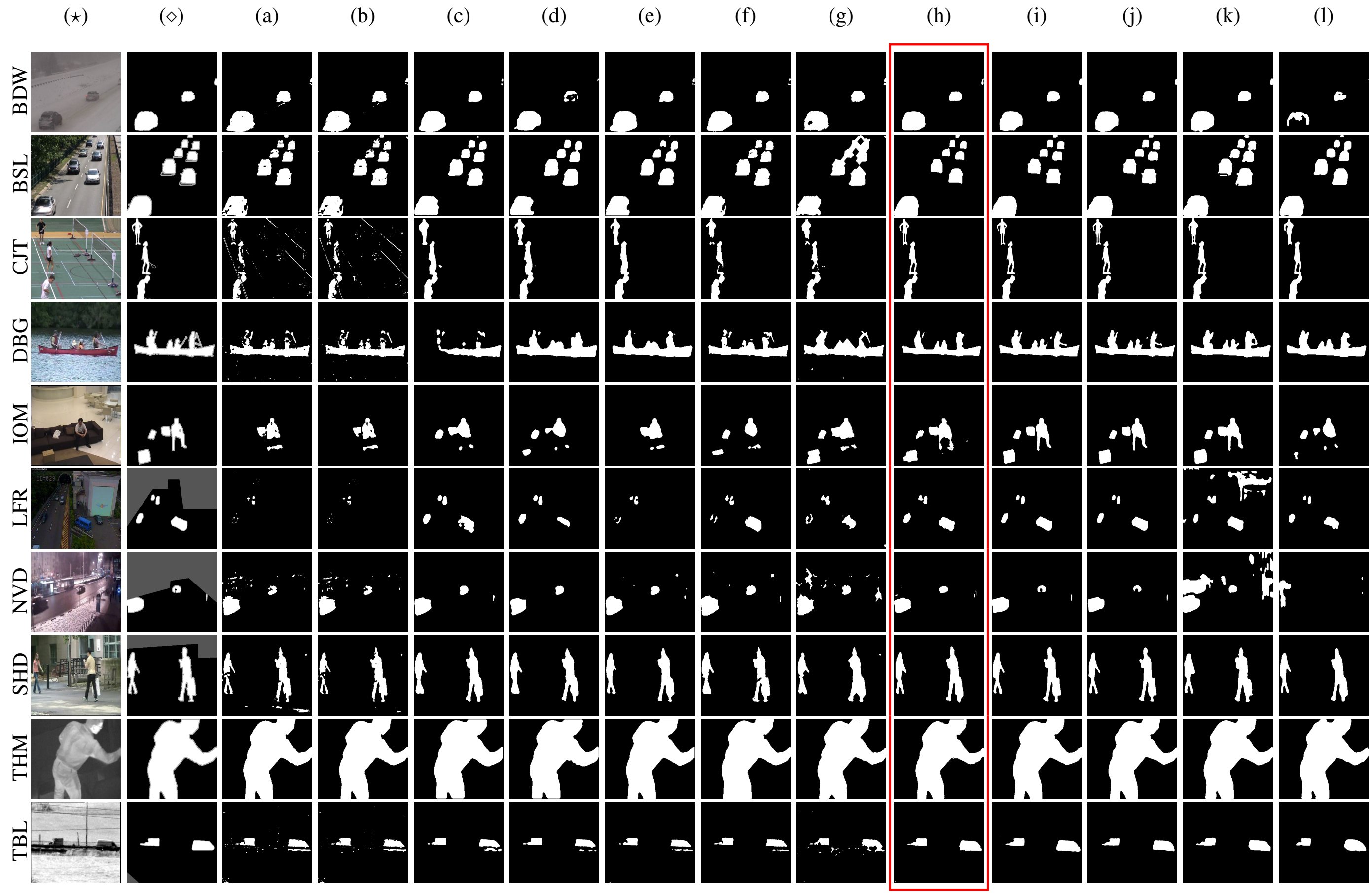

Fig. 4. Visual quality comparison for foreground detection on all video sequences in eleven categories in CDnet 2014. The columns include: (⋆) input frame, (⋄) corresponding groundtruth foreground, (a) GMM – S & G, (b) GMM – Zivkovic, (c) SuBSENSE, (d) PAWCS, (e) BMOG, (f) FTSG, (g) SWCD, (h) CDN-MEDAL-net,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

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

TABLE V

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.

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 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*, *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 shown to be 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. Whilst 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 having been trained on all scenarios, their behaviors may 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, a highlight is where our method surpasses even these computationally expensive to be at the top of the *LFR* scenario. 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, whilst cementing its practical use in its ability to compete with supervised ones despite its light-weighted structure. Table V presents evaluation metrics of a confusion matrix.

C. Results on Wallflower Benchmark without Tuning

Using the Wallflower dataset, we aim to empirically determine the effectiveness of our proposed approach on unseen sequences, using only trained weights from CDNet-2014 from scenarios that we consider having similar dynamics. The results are mixed but they tend towards suggesting good degrees of generalization from trained scenarios over to those unseen. The experimental evaluation are presented in Table IV, which highlights the F-measure quantitative results of our approach compared against some of state-of-the-art methods in both of approaches: 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

TABLE IV

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, **Black** is for the best within each scenario.

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 yields very good results on *Camouflage* and *WavingTrees*, with even relatively better results on *Bootstrap*, *ForegroundAperture* and *TimeOfDay*. Whilst 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 the relatively poor accuracy of it 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 the unsupervised approaches, we conducted our speed tests on the configuration of an Intel Core i7 with 16 GB RAM and recorded the qualitative results of execution performance (in frame-per-seconds) into Table VI together with supervised ones. At the speed of 129.4510 fps, it is apparent that CDN-MEDAL-net is much faster than other supervised deep learning approaches, i.e., DeepBS, STAM, Cascade CNN, and FgSegNet's variants, of which the fastest runs at 23.1275 fps with FgSegNet_S. Our approach makes such efficient use of hardware resources because of its lightweight architecture and the latent-space-limitation approach by incorporating a parallelized unsupervised estimation of background scenes as a pre-processing step. In contrast, other DNNs architectures are burdened with a large number of trainable parameters due to their intention towards achieving accurate input-target mapping. Furthermore, the proposed scheme dominates most of the unsupervised methods with rigorous mathematics-based frameworks in terms of speed and accuracy such as SuBSENSE, SWCD, and PAWCS because their paradigms of sequential processing lead to 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 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 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% in the rest.

TABLE VI
PROCESSING SPEED EVALUATION RELATED METHODS WITH RESOLUTION 320×240 (UNIT: FPS – FRAMES PER SECOND)

GMM - Zivkovic 419.9497$_{(1)}$	TensorMoG 302.5261$_{(2)}$	**CDN-MEDAL-net** 129.451$_{(3)}$	GMM - S & G 119.6968	BMOG 102.0245
FgSegNet_S 23.1275	FgSegNet 21.5431	SWCD 20.0608	FgSegNet_v2 18.0146	SuBSENSE 15.7168
Cascade CNN 12.5214	PAWCS 12.1585	STAM 10.8122	FTSG 10.1912	DeepBS 10.0149

Note: *Red*$_{(1)}$ is for the best, *Green*$_{(2)}$ for the 2^{nd} best, and *Blue*$_{(3)}$ for the 3^{rd} best.

V. CONCLUSION

This paper has proposed a novel, two-stage framework with a GMM-based CNN for background modeling to extract motion features and a convolutional auto-encoder simulating frame-background subtraction for foreground detection, which are considered as a search space limitation approach to compress CNNs models. Our most significant contributions for this paper include a pixel-wise, light-weighted, feed-forward CNNs representing a multi-modular conditional probability density function of the temporal history of data and a corresponding loss function for the CNNs 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, but it is also designed in the tensor form to exploit technologically parallelizing modern hardware. Furthermore, we proved that incorporating such statistical features into the motion-region extraction phase promises more efficient use of powerful hardware with a dominant high-speed performance and a better generalization ability with only a few thousands of latent parameters and a small-scale set of training labels in a deep non-linear scheme.

On the other hand, in its pure form, CDN's predictions can be riddled with false positives due to its sensitivity to spatial changes concerning the background, which leads to inaccurate prediction of the foreground extraction phase. This is obvious given the scheme of utilizing only pixel-wise temporal data for the task of motion analysis. Our future work seeks to address these effects by extending the convolutional receptive fields for spatio-temporal filtering or find ways to prune existing well-performed pre-trained CNNs model to improve the unconstrained domain of effects in segmentation while maintaining real-time execution.

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 Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 1s, 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, June 2020.

[3] 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.

[4] R. Kalsotra and S. Arora, "A comprehensive survey of video datasets for background subtraction," *IEEE Access*, vol. 7, pp. 59 143–59 171, 2019.

[5] 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, 2019.

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

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

[8] 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.

[9] 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.

[10] C. Stauffer and W. E. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[11] J. D. Pulgarin-Giraldo, A. Alvarez-Meza, D. Insuasti-Ceballos, T. Bouwmans, and G. Castellanos-Dominguez, "GMM background modeling using divergence-based weight updating," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[12] 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*, 2018.

[13] 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, 2018.

[14] 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, 2018.

[15] 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*, 2019.

[16] B. N. Subudhi, S. Ghosh, S. B. Cho, and A. Ghosh, "Integration of fuzzy Markov random field and local information for separation of moving objects and shadows," *Information Sciences*, 2016.

[17] Z. Zeng, J. Jia, D. Yu, Y. Chen, and Z. Zhu, "Pixel modeling using histograms based on fuzzy partitions for dynamic background subtraction," *IEEE Transactions on Fuzzy Systems*, 2017.

[18] T. Yu, J. Yang, and W. Lu, "Dynamic Background Subtraction Using Histograms Based on Fuzzy C-Means Clustering and Fuzzy Nearness Degree," *IEEE Access*, 2019.

[19] B. N. Subudhi, T. Veerakumar, S. Esakkirajan, and A. Ghosh, "Kernelized Fuzzy Modal Variation for Local Change Detection From Video Scenes," *IEEE Transactions on Multimedia*, 2019.

[20] D. Giveki, M. A. Soltanshahi, and M. Yousefvand, "Proposing a new feature descriptor for moving object detection," *Optik*, 2020.

[21] 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.

[22] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *International Conference on Systems, Signals, and Image Processing*, 2016.

[23] Y. Wang, Z. Luo, and P. M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, 2017.

[24] 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 2017*, 2017.

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

[26] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, 2018.

[27] 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*, 2015.

[28] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[29] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, 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*, 2019.

[31] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-Based image-to-image foreground segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[32] M. Sultana, A. Mahmood, S. Javed, and S. K. Jung, "Unsupervised deep context prediction for background estimation and foreground segmentation," *Machine Vision and Applications*, 2019.

[33] L. Maddalena and A. Petrosino, "Self-organizing background subtraction using color and depth data," *Multimedia Tools and Applications*, 2019.

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

[35] 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.

[36] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

[37] A. 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," *ArXiv*, vol. abs/1704.04861, 2017.

[38] 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)*, 2017, pp. 4105–4113.

[39] P. St-Charles, G. 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*, 2015, pp. 990–997.

[40] 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, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6973

[41] 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. [Online]. Available: https://doi.org/10.1117/1.JEI.27.2.023002

[42] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, p. 256–262, Sep 2018. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2018.08.002

[43] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, vol. 23, pp. 1369–1380, 2019.

[44] 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, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/23/5142

[45] "CDnet 2014: An expanded change detection benchmark dataset," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[46] 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*, vol. 1, 1999, pp. 255–261 vol.1.