Abstract—We propose a new Arithmetic Distribution Neural Network (ADNN) for learning the distributions of temporal pixels during background subtraction. In our ADNN, the arithmetic distribution operations are utilized to propose the arithmetic distribution layers, including the product distribution layer and the sum distribution layer. Furthermore, in order to improve the accuracy of the proposed approach, an improved Bayesian refinement model based on neighboring information, with a GPU implementation, is introduced. In the forward pass and backpropagation of the proposed arithmetic distribution layers, histograms are considered as probability density functions rather than matrices. Thus, the proposed approach is able to utilize the probability information of the histogram and achieve promising results with a very simple architecture compared to traditional convolutional neural networks. Evaluations using standard benchmarks demonstrate the superiority of the proposed approach compared to state-of-the-art traditional and deep learning methods. To the best of our knowledge, this is the first method to propose network layers based on arithmetic distribution operations for learning distributions during background subtraction.

Index Terms—Background Subtraction, Deep Learning, Distribution Learning, Arithmetic Distribution Operations

I. INTRODUCTION

Background subtraction is a fundamental research topic related to motion detection in computer vision, which has attracted increasing attention during a period of explosive growth in video streaming. Traditionally, background subtraction algorithms are closely related to distribution analysis, since the distribution of temporal pixels plays an important role in identifying a pixel as foreground or background. Sophisticated models including deep learning networks have been devised for distribution analysis in background subtraction. Unfortunately, when a convolutional neural network is involved, the histograms, which describe distributions, are considered as feature vectors rather than a probability density function. This is an arbitrary utilization of matrix arithmetic operations since the probability information is ignored, limiting the ability of algorithms based on convolutional neural networks for analyzing complex distributions. Thus, distribution analysis based on convolutional neural networks remains a challenging problem.

In background subtraction, pixels are classified as foreground or background based on comparisons with their historical counterparts. Thus, the distribution of comparisons is a useful feature that can be directly input into the network for classification. However, when distributions are converted into histograms for the convolution operation, they are actually considered as feature vectors in which the probabilistic information is ignored. However, the essence of a histogram is the probability density function which describes the distributions of random variables. In this work, the arithmetic distribution operations [1], which are used to compute the distribution of arithmetic operations of random variables having known distributions, are derived to serve as an alternative to matrix arithmetic operations. This enables histograms to be considered as probability density functions rather than feature vectors. Based on the new operations, we propose the arithmetic distribution layers containing the product and sum distribution layers for learning and classifying distributions. Thereby, a novel Arithmetic Distribution Neural Network (ADNN) including arithmetic distribution layers is devised for background subtraction, as shown in Fig. 1.

The architecture of the proposed ADNN is quite straightforward. Histograms of subtractions between pixels’ current observations and their historical counterparts are used as the input to the arithmetic distribution layers containing the product distribution layer and the sum distribution layer. Specifi-
cally, the learning kernels are also histograms and considered as probability density functions. Both the forward pass and backpropagation of layers are based on arithmetic distribution operations. A classification architecture is attached to label pixels according to the output of the arithmetic distribution layers. In particular, the classification architecture is kept as simple as possible, with only one convolutional layer, one rectified linear unit layer, and one fully connected layer, to demonstrate that the good performance of the proposed approach comes from the proposed arithmetic distribution layers. Unfortunately, since pixels are classified independently, the proposed approach is sensitive to noisy points that can be handled by neighboring information. An improved Bayesian refinement model, with a GPU implementation, is thus proposed for noise compensation. By utilizing the arithmetic distribution operations, histograms can be considered as probability density functions, with the probability information being utilized. This helps the proposed approach achieve promising results compared to even convolutional neural networks. Furthermore, since the histograms of temporal pixels are pixelwise features, a large number of training instances can be captured. Thus, the proposed approach requires fewer than 1% of the ground truth frames during training. In addition, since the distribution information of temporal pixels is independent of scene information, the proposed network does not rely too much on the scenes where training frames are captured. Our ADNN can be trained with video frames obtained from different scenes, and it is valid even when no frame from the scenes of the testing videos is included in the training set.

The main contributions of this paper are:

- We propose the Arithmetic Distribution Neural Network (ADNN) for background subtraction, utilizing the product distribution layer and the sum distribution layer.
- An improved Bayesian refinement model, with a GPU implementation, is proposed to improve the accuracy of the proposed approach. In particular, an approximation of the Gaussian function is utilized to compute the correlation between neighboring pixels.
- Comprehensive experiments are conducted to evaluate the proposed approach, including: a) validating the correctness of the proposed product distribution layer and the sum distribution layer, as shown in Section V-A; b) comparisons between the proposed ADNN and traditional convolutional neural networks on real data are shown in Section V-B; c) a discussion on the generality of the proposed approach under the condition that training frames and testing frames are randomly captured from different scenes, as shown in Section V-C; and, d) a comprehensive comparison between the proposed approach and state-of-the-art methods including traditional and deep learning approaches on standard benchmarks, as shown in Section V-D.

II. RELATED WORK

In this section, we outline background subtraction algorithms briefly into three categories. In Section II-A traditional background subtraction algorithms are introduced. Background subtraction based on deep learning network is discussed in Section II-B. Finally, distribution learning techniques and their relation to background subtraction methods are described in Section II-C.

A. Traditional Algorithms

Traditional algorithms [23–29] have focused on capturing background representation from temporal pixels for subtraction, usually based on artificial models. In particular, the Gaussian mixture model is one of the most popular techniques [21], where the background observations are described by several Gaussian functions [1, 22], with a large number of extensions being proposed [19, 20]. In addition, several excellent publications [17, 24–26] have considered the background as an low-rank component of video frames, given the correlation between background scenes of frames over time. For example, Javed et al. [25], [26] utilized robust principal component analysis [24] to separate the background scenes based on the spatial and temporal subspaces. Yong et al. [17] proposed online matrix factorization for background subtraction. Machine-learning techniques have also been utilized for background subtraction [30–40]. Lin et al. [30] classified the pixels by using a probabilistic support vector machine. Similarly, Han et al. [32] used the density-based features into a classifier based on a support vector machine for classification. Li et al. [31] formulated background subtraction as minimizing a constrained risk function and Culibrk et al. [35] proposed an unsupervised Bayesian classifier using a neural network architecture for background subtraction. Unfortunately, given the complexity and diversity of natural scenes, artificial models are not adequate to generate a perfect classification of pixels for background subtraction. In this work, we propose the arithmetic distribution neural network to automatically learn the distributions of pixels for background subtraction.

B. Algorithms based on Deep Learning

Recently, several excellent approaches [15, 41–57] have used deep learning networks to learn the background scenes for subtraction. For example, Wang et al. [47] proposed a fully connected network to learn the background scenes, and Zeng et al. [50] utilized a multi-scale strategy to improve the results. Similarly, Lim et al. [48] used a triplet convolutional neural network to extract multi-scale features from background scenes, and Yang et al. [41] improved the robustness of their method by using an end-to-end multi-scale spatio-temporal (MS-ST) method to extract deep features from scenes. Unfortunately, these papers usually assumed a large number of ground truth frames for training, which is very expensive in background subtraction applications. Also, these approaches rely considerably on the scene information, which limits them to the condition that the testing frames and training frames must be captured from the same video. In contrast, Mandal et al. [42], [43] proposed a spatio-temporal feature learning network for background subtraction in unseen videos, but a large number of annotated frames are still assumed during training. In addition, Babaee et al. [52] proposed a robust model in which a network is used to subtract the background
from the current frame and only 5% of the labeled masks are utilized for training. Liang et al. [53] utilized the foreground mask generated by the SubSENSE algorithm [15] rather than manual labeling for training, and Zeng et al. [54] used a convolutional neural network to combine several background subtraction algorithms together. In our work, since the distributions of temporal pixels are captured from every spatial pixel, a large number of training instances can be captured with a limited number of ground truth frames. Thus, our proposed approach requires less than 1% of the ground truth frames for training. Because the distribution information is independent of the scene information, our proposed ADNN is effective even when the training videos and testing videos are completely different.

C. Distribution Learning and Background Subtraction

The distribution of temporal pixels plays an important role [3], [4], [7], [8], [58]–[70] throughout the development of background subtraction algorithms, since it is a good representation of background information. In particular, the Gaussian mixture model proposed by Zivkovic et al. [4] is one of the most popular techniques as we mentioned above. In addition, Lee et al. [3] utilized an adaptive learning rate for each Gaussian function to improve the convergence rate during clustering, and Haines et al. [8] [58] used the Dirichlet processes with Gaussian mixture models to analyze pixel distributions. Recently, Chen et al. [59] [60] used Gaussian mixture models to represent the vertices of spanning trees and Akilan et al. [61] proposed a foreground validation process converted to histograms for convolutions in which the matrix arithmetic operations are used; and, all the objects involved in the operations are considered as vectors. In this condition, the correlation between the entries of a histogram as well as their probability information are ignored. Essentially, histograms are probability density functions that describe the distributions of random variables. When a histogram of a distribution is input into a network for classification, it is actually the random variable that has the input distribution that needs classification. Therefore, a network layer based on the arithmetic of random variables having distributions of the input and the learning kernel is supposed to be a more reasonable substitute for the convolution layer. In addition, the arithmetic distribution operations, which refer to the operations that compute the distribution of arithmetic results of random variables having known distributions, are derived to serve as a better substitute of the matrix arithmetic operations. In this work, the arithmetic distribution layers including the product and sum distribution layers based on arithmetic distribution operations are proposed. In particular, both the input and the learning kernels of layers are considered as probability density functions, which are described by histograms; and, arithmetic distribution operations are used in the forward pass and backpropagation for learning a distribution.

The product distribution layer is used to compute the distribution of the product of random variables having distributions described by the histograms of the input and learning kernel. Assume two independent, continuous random variables $X$ and $W$, which are described by probability density functions $f_X(x)$ and $f_W(w)$ respectively. Both $f_X(x)$ and $f_W(w)$ are represented by histograms. In particular, $f_X(x)$ is assumed to be the input histogram of the product distribution layer, and $f_W(w)$ denotes the histogram of the learning kernel. The output of the product distribution layer $f_Z(z)$ is actually the probability density function of the random variable $Z = XW$, which is the product of $X$ and $W$. In order to capture the expression for $f_Z(z)$, the definition of the cumulative distribution function of $Z$ is proposed first, as shown below:

$$F_Z(z) \overset{def}{=} \mathbb{P}(Z \leq z) = \mathbb{P}(XW \leq z) = \mathbb{P}(XW \leq z, W \geq 0) + \mathbb{P}(XW \leq z, W \leq 0)$$

$$= \mathbb{P}(X \leq \frac{z}{W}, W \geq 0) + \mathbb{P}(X \geq \frac{z}{W}, W \leq 0),$$

where $F_Z(z)$ is the cumulative distribution function of the random variable $Z$. $\mathbb{P}$ is a cumulative distribution under a particular condition. Next, assuming $X$, $W$ and $Z$ are between negative infinity and positive infinity, the expression of $F_Z(z)$ is converted into an expression following the cumulative
distribution function. Mathematically, this can be shown as:

\[
F_Z(z) = \int_{0}^{\infty} f_W(w) \int_{-\infty}^{\infty} f_X(x) dx \, dw + \int_{-\infty}^{0} f_W(w) \int_{-\infty}^{\infty} f_X(x) dx \, dw
\]

\[
= \int_{0}^{\infty} f_W(w) f_X(z) |w| \, dw - \int_{-\infty}^{0} f_W(w) f_X(z) |w| \, dw
\]

\[
= \int_{-\infty}^{\infty} f_W(w) f_X(z) |w| \, dw
\]

\[
\Rightarrow z_j = \sum_{i=-\infty}^{\infty} w_i f_X(z_i) \frac{1}{|i|} \cdot 1 \quad \because dw = 1, \quad f_W(i) = w_i,
\]

where \(w_i\) and \(z_j\) are the entries of histograms that are used to describe the probability density functions \(f_W(w)\) and \(f_Z(z)\) corresponding to random variables \(W\) and \(Z\), respectively. \(i\) and \(j\) are the indices of the entries. The formula for the forward pass of the product distribution layer is thus derived. Then, the gradient of \(f_W(w)\), which is used to update \(w_i\) during backpropagation, is obtained by partial derivatives and the chain rule. Mathematically:

\[
\frac{\partial \text{loss}}{\partial w_i} = \frac{\partial \text{loss}}{\partial Z} \cdot \frac{\partial Z}{\partial w_i} = \sum_{j=\neg}^{\infty} \frac{\partial \text{loss}}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_i}
\]

\[
= \sum_{j=\neg}^{\infty} \frac{\partial \text{loss}}{\partial z_j} \cdot \frac{\partial \left( \sum_{i=\neg}^{\infty} w_i f_X(z_i) \frac{1}{|i|} \right)}{\partial w_i}
\]

\[
= \sum_{j=\neg}^{\infty} \delta z_j f_X(z_j) \frac{1}{|i|}
\]

\[
\Rightarrow \delta w_i = \sum_{j=\neg}^{\infty} \delta z_j f_X(z_j) \frac{1}{|i|},
\]

where \(\delta w_i\) and \(\delta z_j\) are the gradients of entries of histograms which are utilized to update \(f_W(w)\) and \(f_Z(z)\) respectively. \(i\) and \(j\) are indices of entries of histograms, \(\text{loss}\) is the output of the product distribution layer which is used to compare with the target output to compute the gradient \(\partial \text{loss}\). This way, the formula for backpropagation of the proposed product distribution layer is derived.

Similarly, the sum distribution layer is used to compute the distribution of the sum of two random variables \(X\) and \(B\), which are described by \(f_X(x)\) and \(f_B(b)\) respectively. Similar to the product distribution layer, \(f_X(x)\) and \(f_B(b)\) are represented by histograms as well. Utilizing the same mathematical procedure as the product distribution layer, the expression of the probability density function of the sum \(Z = X + B\) of \(X\) and \(B\) is obtained. Mathematically:

\[
f_Z(z) = \int_{-\infty}^{\infty} f_B(b) f_X(z - b) \, db
\]

\[
\Rightarrow z_j = \sum_{i=\neg}^{\infty} b_i f_X(z_j - i) \cdot 1 \quad \because \text{db} = 1, \quad f_B(i) = b_i,
\]

where \(b_i\) and \(z_j\) are the entries of histograms utilized to describe \(f_B(b)\) and \(f_Z(z)\) corresponding to random variables \(B\) and \(Z\), respectively. \(i\) and \(j\) are indices. Also, the formula for backpropagation is:

\[
\frac{\partial \text{loss}}{\partial b_k} = \frac{\partial \text{loss}}{\partial Z} \cdot \frac{\partial Z}{\partial b_k} = \sum_{j=\neg}^{\infty} \frac{\partial \text{loss}}{\partial z_j} \cdot \frac{\partial (\sum_{i=\neg}^{\infty} b_i f_X(z_j - i))}{\partial b_k}
\]

\[
= \sum_{j=\neg}^{\infty} \frac{\partial \text{loss}}{\partial z_j} \cdot f_X(z_j - k)
\]

\[
\Rightarrow \delta b_k = \sum_{j=\neg}^{\infty} \delta z_j f_X(z_j - k),
\]

where \(\delta b_k\) and \(\delta z_j\) are the gradients of entries in histograms corresponding to \(f_B(b)\) and \(f_Z(z)\) respectively, and \(k\) and \(j\) are indices. \(\text{loss}\) is the output of the sum distribution layer which is used to compare with the target output to compute the gradient for updating the sum distribution layer.

With the help of Eqn. [3]–[6] the forward pass and the backpropagation of arithmetic distribution layers can be easily implemented by Pytorch [79]. In particular, the gradient of learning kernels of arithmetic distribution layers is computed and input into the “Autograd package” of PyTorch [79] for backpropagation. Moreover, a validation experiment based on synthetic data is proposed in Section V-A to verify the correctness of the proposed product and sum distribution layers. In the experiments, the arithmetic distribution layers are fed with data from a few synthetic distributions, in order to validate their ability of distribution learning.

IV. ARITHMETIC DISTRIBUTION NEURAL NETWORK FOR BACKGROUND SUBTRACTION

Utilizing the proposed product and sum distribution layers, the arithmetic distribution neural network is devised for background subtraction. Background subtraction is a binary classification of temporal pixels; thus, the distributions of temporal
pixels play an important role. In this work, the distributions of subtractions between pixels and their historical counterparts are used for classification. In particular, histograms are utilized to describe the distributions of subtractions and also directly used as the input of the proposed arithmetic distribution neural network. The network architecture is quite straightforward: histograms are first input into the product distribution layer and the sum distribution layer; then, the outputs of these layers are combined by a convolution followed by a classification architecture which consists of a convolution, a rectified linear unit (Relu) layer, and a fully connected layer. The classification architecture is deliberately kept as simple as possible, with only 3 layers, in order to demonstrate that the good results come from the proposed arithmetic distribution layers.

The components of the proposed arithmetic distribution neural network for background subtraction are illustrated in Fig. 2, with details of the network architecture presented in Table I. Starting with a given frame of a video, denoted as \( \mathcal{I} = \{I_1, I_2, \cdots , I_T\} = \{I_t| t = [1, T] \cap \mathbb{N}\} \), where \( t \) is the frame index, \( T \) the number of frames, and \( \mathbb{N} \) a natural number, to perform background subtraction for a particular pixel located at \((x, y)\) on frame \( t \), the histogram of the subtractions between pixels’ current observation and their historical counterparts is captured for classification. Mathematically:

\[
H_{x,y}(n) = \sum_{i=1}^{T} (I_i(x,y) - I_t(x,y)) \cap n,
\]

where \( H_{x,y} \) is the histogram of subtractions, and \( n \) is the index of entries of the histogram. \( I_t(x,y) \) denotes historical observations of the pixel located at \((x, y)\), and \( I_t(x,y) \) denotes its current observation. The distributions of subtractions are directly used as the input to the product distribution layer and the sum distribution layer for distribution learning. Then, the outputs of these two layers are combined with a convolution procedure which is followed by the classification architecture. Mathematically:

\[
\mathcal{M}(x,y) = \mathcal{L}(\mathcal{C}(\mathcal{F}_p(H_{x,y}) + \mathcal{F}_a(H_{x,y}))),
\]

where \( H_{x,y} \) is the input histogram; \( \mathcal{F}_p \) and \( \mathcal{F}_a \) denote the product distribution and the sum distribution layer; \( \mathcal{C} \) is the convolution procedure; and \( \mathcal{L} \) is the classification architecture consisting of a convolution, a rectified linear unit, and a fully connected layer. \( \mathcal{M}(x,y) \) is the label of a pixel.

Unfortunately, the histograms utilized for classification are captured from independent pixels; thus, the correlation between pixels is ignored. In order to improve the accuracy of the proposed approach, an improved Bayesian refinement model is proposed. For completeness, we briefly introduce the Bayesian refinement model; please check [78] for more details. In the Bayesian refinement model, the labels of pixels are re-inferred according to the correlations with their neighborhoods, and the Bayesian theory is utilized during inference. In particular, Euclidean distance is used to compute the correlation. In contrast, we utilize a mixture of Gaussian approximation functions to capture the correlation. This is the main difference compared
to the original Bayesian refinement model. Mathematically:

\[
\mathcal{F}(I(x, y), \mathcal{M}) = \operatorname{argmax}_{a_i} \frac{P(I(x, y)|a_i)P(a_i)}{P(I(x, y))} \\
= \operatorname{argmax}_{a_i} P(a_i) \sum_{k=1}^{K} \pi_k \mathcal{N}_p(v_k|\mu_{k,i}, \Sigma_{k,i})
\]

\[
\therefore P(I(x, y)|a_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}_p(v_k|\mu_{k,i}, \Sigma_{k,i}),
\]

where \( \mathcal{F} \) is denoted as the proposed improved Bayesian refinement model, \( a_i \in \{0, 1\} \) denotes the labels of foreground or background; \( I(x, y) \) is a pixel located at \((x, y)\); and, \( P(I(x, y)|a_i) \) is the probability that the label of this pixel is \( a_i \), which is captured through a mixture of Gaussian approximation functions \( \mathcal{N}_p(v_k|\mu_{k,i}, \Sigma_{k,i}) \). In particular, \( v_k \) denotes the feature vector consisting of the Lab color and spatial position of the pixel \( I(x, y) \), and \( k \) is the index of entries in a vector. \( \mu_k \) and \( \Sigma_k \) denote the mean and variance of features of a pixel in a local rectangular range with center at \((x, y)\) and radius \( R = 4 \). \( \pi_k \) is the weight to mix the Gaussian approximation functions \( \mathcal{N}_p \) which is mathematically expressed as:

\[
\mathcal{N}_p(x|\mu, \sigma) = \begin{cases} 
1 + \frac{x - \mu}{n\sigma} & |x - u| \leq n\sigma \\
0 & \text{otherwise}
\end{cases}
\]

where \( \mu \) and \( \sigma \) denote the mean and variance, and \( n \) is a user parameter. During experiments, \( n = 2 \) gives us the best results. As shown in Fig. 3, the Gaussian approximation function is actually a rough estimate of the Gaussian function. We use a piecewise function to approximate the waveform of the Gaussian function considering the computational cost. Also, it is more convenient for a GPU implementation, which significantly accelerates the refinement procedure.

Finally, the output binary mask is used in the input again to generate better results iteratively. The Bayesian refinement model is utilized to iteratively refine the foreground mask. Mathematically:

\[
\mathcal{M}_n(x, y) = \mathcal{F}(I(x, y), \mathcal{M}_{n-1}),
\]

where \( n \) is the iteration number and \( \mathcal{M}_{n-1} \) is the binary mask from the last iteration. Using a GPU implementation, with the number of iterations set to 30, the entire refinement procedure only takes a few second.

The improved Bayesian refinement model (IBRM) runs much faster than the Bayesian refinement model (BRM) with almost no loss in accuracy. A comparison between them on a few frames from videos with different resolutions is shown in Table II. In particular, the running time of BRM and IBRM with iteration numbers 1, 20 and 50 are presented, as well as the Fm value of their corresponding output masks after refinement. As shown in Table II when the number of iterations is 50, although the Fm value of output mask has obvious improvement, the run time also increases to 52s, which is too long for real applications. In contrast, IBRM needs only 3.5s in processing time, and the Fm value of the output mask is still close to the one for BRM. Thus, the superiority of the proposed IBRM is demonstrated.

### V. Experiments

#### A. Verification of Arithmetic Distribution Layers

In this section, we verify the correctness of the proposed arithmetic distribution layers including the product distribution layer and sum distribution layer, which are the implementations of Eqns. 4 and 5 respectively. In the experiments, synthetic data is used for verification. In particular, two continuous independent random variables \( X \) and \( W \), which are described by two different probability density functions representing two histograms, are generated. In addition, the product \( Z_p = XW \) and the sum \( Z_s = X + W \) of the two variables \( X \) and \( W \) are computed for use as the target output. The values of \( X \) and \( W \) as well their product and sum are generated over one million times to capture the target histograms of \( Z_p \) and \( Z_s \). The verification experiment is quite straightforward: the histogram of \( X \) is input into the product distribution layer and the sum distribution layer, respectively, to compute with the histogram of \( W' \) in layers to output the histograms of \( Z'_p \) and \( Z'_s \). The output histograms are then compared with the target histograms of \( Z_p \) and \( Z_s \) to capture the gradients to train the arithmetic distribution layers. Finally, the correlation values between the output histograms and target histograms

| Video       | Resolution | NI | BRM Time/s | IBRM Time/s | BRM Fm value | IBRM Fm value |
|-------------|------------|----|------------|-------------|--------------|---------------|
| highway     | 320 × 240  | 1  | 1.2533     | 1.4429      | 0.9554       | 0.9612        |
|             | 20         | 50 | 21.1842    | 2.2524      | 0.9826       | 0.9828        |
|             | 1          | 20 | 52.7545    | 3.5712      | 0.9911       | 0.9905        |
| canoe       | 320 × 240  | 1  | 1.1496     | 1.4215      | 0.9528       | 0.9528        |
|             | 20         | 50 | 19.9488    | 2.2968      | 0.9535       | 0.9482        |
|             | 1          | 20 | 50.4685    | 3.5780      | 0.9534       | 0.9453        |
| wetSnow     | 720 × 540  | 1  | 5.2601     | 1.6945      | 0.7252       | 0.7279        |
|             | 20         | 93 | 16.9302    | 5.8695      | 0.7731       | 0.7646        |
|             | 50         | 234| 12.5385    | 0.7750      | 0.7732       | 0.7732        |

NI: Number of iterations; BRM: Bayesian refinement model; IBRM: Improved Bayesian refinement model.
Correlation value = 0.999

III. In particular, the last three layers and the size of input possible. The details of architectures are shown in Table [III]. In particular, the last three layers and the size of input are kept the same in our ADNN and the compared CNNs. During comparisons several videos from the CDnet [80] dataset are extracted. In the experiments, one ground truth frame from a particular video is used as the output label of CNNs and ADNN for training. Furthermore, the distributions of differences between historical and current observations of pixels corresponding to the ground truth frame are used as the input for training. After training, the remaining frames are used for testing to demonstrate the superiority of ADNN in distribution analysis compared to CNNs. For comparison, the Re (Recall), Pr (Precision) and Fm (F-measure) metrics are used. Quantitative comparisons are shown in Table [IV].

As shown in Table [IV] the proposed approach achieves better Fm value for most of the videos. In the comparison between the proposed ADNN and CNN1, since the product and sum distribution layers are the unique differences, any superiority of ADNN is attributable to our distribution arithmetic layers. In the comparison one may doubt that the number of parameters in the ADNN is greater than CNN1, since more parameters are supposed to generate better results. However, the proposed ADNN also achieves better results compared to CNN2 which has over 10 times the number of parameters than the proposed approach, while the depth of the proposed ADNN and CNN2 are the same. In addition, the proposed approach achieves better results even compared to CNN3 which contains almost 100 times the number of parameters and deeper network architecture. Hence, in a few complex videos such as “fountain02,” “backdoor,” and “bungalows,” the proposed ADNN has clear improvements compared to all three CNNs. Thus, it is fair to conclude that the proposed arithmetic distribution neural network is better for distribution classification compared to a traditional convolutional neural network.

C. Generality of Distribution Learning

The distributions of temporal pixels are relatively independent of the scene information, demonstrating that the proposed approach is generalizable. Compared to previous scene information based algorithms that need to be trained with a particular network for every video, the proposed approach is not limited by the scene information and can be trained with groups of frames obtained from different videos simultaneously. Our approach is effective even when no frame from the testing video is included in the training set. The generality of distribution information demonstrates the potential applications of the proposed approach in real applications, since a single well-trained network can be used in several different scenes with limited ground truth frames required for training. In order to demonstrate generality, we randomly select two videos as the training set in which 4 ground truth frames from each video are extracted for training, and the remaining frames are utilized for testing. Note that the ground truth frames utilized for training take less than 1% of the total ground truth frames for a video. Besides, two more new videos are also randomly selected from the dataset to test the proposed approach, under the condition that no frame from the testing video is included in the training set. In addition, in order to test the proposed approach under a more
TABLE III
DETAILS OF THE NETWORK ARCHITECTURE OF THE ARITHMETIC DISTRIBUTION NEURAL NETWORK (ADNN) AND THE CONVOLUTIONAL NEURAL NETWORKS (CNNs), USED FOR COMPARISON TO DEMONSTRATE THE SUPERIORITY OF THE PROPOSED APPROACH.

| Type  | Filters | Size   | Type  | Filters | Size   | Type  | Filters | Size   | Data Size         |
|-------|---------|--------|-------|---------|--------|-------|---------|--------|------------------|
| Input |         |        | Conv  | 8       | 3 × 1 × 1 | Conv  | 16     | 8 × 1 × 1 |                  |
|       |         |        | Conv  | 32      | 16 × 201 × 1 | ProDis | 1      | 3 × 201 × 1 | B × 3 × 201 × 1 |
|       |         |        | Conv  | 32      | 128 × 1 × 1 | SumDis | 1      | 3 × 201 × 1 |                  |
|       |         |        |       |         |         |       |         |         | B × 1 × 201 × 1 |
|       |         |        |       |         |         |       |         |         |                  |
| NTP   | 405     | NTP    | 29370 | NTP     | 133290 | NTP   | 1611    |        |                  |

ProDis: product distribution layer  SumDis: sum distribution layer  Conv: Convolutional  Relu: Rectified Linear Unit  NTP: Number of total parameters  B: batch size

TABLE IV
QUANTITATIVE COMPARISON BETWEEN CNNs AND ADNN USING RE, PR, AND Fm METRICS BASED ON REAL DATA.

| Videos                  | CNN1    | CNN2    | CNN3    | ADNN    |
|-------------------------|---------|---------|---------|---------|
| highway                 | 0.9086  | 0.8133  | 0.8583  | 0.8732  |
| pedestrians             | 0.9370  | 0.9404  | 0.9387  | 0.9536  |
| fountain01              | 0.1404  | 0.1988  | 0.1646  | 0.6434  |
| canoe                   | 0.8134  | 0.8398  | 0.8264  | 0.9464  |
| fountain02              | 0.7289  | 0.8812  | 0.7978  | 0.7850  |
| peopleInShade           | 0.9993  | 0.2478  | 0.3963  | 0.9054  |
| backdoor                | 0.7665  | 0.5195  | 0.6192  | 0.9275  |
| traffic                 | 0.6881  | 0.9008  | 0.7802  | 0.7009  |
| sidewalk                | 0.5490  | 0.3675  | 0.4403  | 0.8392  |
| busStation              | 0.8887  | 0.7870  | 0.8212  | 0.8846  |
| bungalows               | 0.8636  | 0.9064  | 0.9094  | 0.8535  |
| library                 | 0.9101  | 0.9702  | 0.9392  | 0.9075  |
| Average                 | 0.7628  | 0.7022  | 0.7076  | 0.8517  |

As shown in Table [V], the proposed approach is trained well when frames from different scenes are combined as the training set and achieves good performance during the testing on the remaining frames. For example, the proposed ADNN is trained by images captured from videos “highway” and “pedestrians,” and achieves 0.9572 in Fm values. It is effective even when the unseen videos “office” and “PETS2006” are used for testing. Furthermore, the proposed approach still achieves good results when the training frames are captured from four videos, which are randomly selected from different categories. As shown in Fig. [5] when the scene information of four videos that produce the training frames are completely different, the proposed approach is still well-trained with the input using the ground truth frames from these different videos. Also, the proposed approach achieves good results when another four unseen videos from different categories are used for testing. In conclusion, the proposed approach can be trained with image frames with different scene information, and it is effective even when unseen videos are utilized for testing. Thus, the generality of the proposed approach is demonstrated.

TABLE V
QUANTITATIVE GENERALITY EVALUATION OF THE PROPOSED APPROACH USING RE, PR AND Fm METRICS.

| Training Sets                  | Testing Sets                  | Performance |
|--------------------------------|--------------------------------|-------------|
| Category Video                 | Category Video                | Re   | Pr   | Fm   |
| Baseline highway               | Baseline highway              | 0.9968 | 0.9814 | 0.9981 |
| Baseline pedestrians           | Baseline pedestrians          | 0.9891 | 0.9274 | 0.9572 |
| Dyn. Bg. fountain01            | Dyn. Bg. founatin01           | 0.8435 | 0.8256 | 0.8345 |
| Dyn. Bg. overpass              | Dyn. Bg. overpass             | 0.9395 | 0.8367 | 0.8851 |
| Dyn. Bg. canoe                 | Dyn. Bg. canoe                | 0.8268 | 0.8102 | 0.8184 |
| Dyn. Bg. peopleInShade         | Dyn. Bg. peopleInShade        | 0.9961 | 0.9398 | 0.9671 |
| Dyn. Bg. fountain01            | Dyn. Bg. overpass             | 0.9890 | 0.9766 | 0.9924 |
| Dyn. Bg. canoe                 | Dyn. Bg. canoe                | 0.9794 | 0.7536 | 0.8518 |
| Dyn. Bg. bungalows             | Dyn. Bg. bungalows            | 0.8404 | 0.8400 | 0.8874 |
| Dyn. Bg. peopleInShade         | Dyn. Bg. peopleInShade        | 0.9835 | 0.9794 | 0.9814 |
| Dyn. Bg. overpass              | Dyn. Bg. overpass             | 0.9864 | 0.7913 | 0.8781 |
| Dyn. Bg. boats                 | Dyn. Bg. boats                | 0.9700 | 0.9361 | 0.9527 |
| Dyn. Bg. Low Fr. tramCross.    | Dyn. Bg. Low Fr. tramCross.   | 0.8476 | 0.8122 | 0.8295 |
| Ther. library                  | Ther. library                 | 0.9467 | 0.8902 | 0.9675 |
| Bad Wea. skating               | Bad Wea. skating              | 0.9639 | 0.9831 | 0.9734 |
| Baseline pedestrians           | Baseline pedestrians          | 0.9093 | 0.8473 | 0.9133 |
| Baseline canoe                 | Baseline canoe                | 0.9775 | 0.8674 | 0.9191 |
| Baseline corridor              | Baseline corridor             | 0.7651 | 0.9077 | 0.8303 |
| Shadow bungalows               | Shadow bungalows              | 0.9725 | 0.9802 | 0.9764 |
In this section, a comprehensive evaluation of the proposed approach is presented through comparisons with several state-of-the-art methods including deep learning networks on the LASIESTA [81] and CDNet2014 [80] datasets. To the best of our knowledge, after the rise of deep learning networks in the background subtraction field, the fairness of comparisons between deep learning methods has been a concern. It is commonly accepted that the quantity of training data and the number of parameters in a network have significant and direct contribution to the performance of various methods [97], but, the assumptions of training data, numbers of parameters in the network and the utilization of pre-trained networks in these methods are completely different. A few methods have generated almost perfect results with the assumption of a large number of ground truth frames for training. For example, FgSegNet model [92] achieves over 99% in Fm value on the CDNet Dataset [80], when 200 frames of ground-truth mask from each video are extracted for training and over hundreds of millions of parameters are used in the network. Such algorithms did achieve better results than the proposed approach. However, the comparison between FgSegNet and the proposed approach is unfair, since we only use 20 ground truth frames for training and the number of parameters in our network is much less than their network. In addition, there are a few semi-supervised algorithms (e.g., GuidedBS [53], BSUV-Net [90] and GraphMOS [59]) which did not utilize any ground truth frames from testing videos for training. However, these methods assumed a large number of binary masks from another video for training, and used several pre-trained networks. For example, VGG-16 [93] or Mask R-CNN [94] are combined into their network. It should be noted that these pre-trained networks still need a significant amount of data for training. For instance, GraphMOS [56] employed Mask R-CNN network [94] and DeepLab network [95] in their network, and both of these two networks were well trained with a large number of masks for objects segmentation or semantic segmentation which are similar to the ground masks of background subtraction. In contrast, the proposed approach does not use any pre-trained models or extra training data from others datasets. The comparison between the proposed approach and GraphMOS model is also questionable. GraphMOS is devised for unseen videos but many training frames may already contain information from testing videos; and although the proposed approach only assumes less than 1 % of ground truth frames for training, such frames are still extracted from testing videos. Thus, before the comparison between the proposed approach and other compared algorithms, the training data and pre-trained networks utilized in algorithms are discussed. The proposed approach is compared

**TABLE VI**

| Videos | ADNN-IBRM | D-DPDL [78] | CueV2 [82] | Hai [8] | Cue [83] |
|--------|-----------|-------------|------------|--------|--------|
| LSL_01 | 0.9764    | 0.9596      | 0.9208     | 0.9622 | 0.8143 |
| LSL_02 | 0.9309    | 0.8687      | 0.8403     | 0.8130 | 0.7576 |
| LCA_01 | 0.9807    | 0.9309      | 0.9062     | 0.9220 | 0.8424 |
| LCA_02 | 0.9201    | 0.8850      | 0.7826     | 0.8656 | 0.6296 |
| LOCA_1 | 0.9783    | 0.9710      | 0.7013     | 0.8920 | 0.8274 |
| LOC_02  | 0.9735    | 0.9677      | 0.8600     | 0.9526 | 0.8781 |
| LIL_01  | 0.7702    | 0.7161      | 0.6452     | 0.8861 | 0.7966 |
| LIL_02  | 0.7620    | 0.8972      | 0.6523     | 0.8122 | 0.7864 |
| LMB_01  | 0.9874    | 0.9699      | 0.9543     | 0.9816 | 0.7779 |
| LMB_02  | 0.9731    | 0.9195      | 0.9204     | 0.7064 | 0.6797 |
| LBS_01  | 0.9787    | 0.8371      | 0.7132     | 0.6285 | 0.5065 |
| LBS_02  | 0.9626    | 0.6718      | 0.6156     | 0.7333 | 0.6607 |
| O_CL_01 | 0.9840    | 0.9792      | 0.9508     | 0.6946 | 0.9280 |
| O_CL_02 | 0.9788    | 0.9800      | 0.9045     | 0.9588 | 0.8995 |
| O_RA_01 | 0.9896    | 0.9072      | 0.8453     | 0.8225 | 0.7462 |
| O_RA_02 | 0.9839    | 0.9083      | 0.8886     | 0.9590 | 0.8699 |
| O_SN_01 | 0.9733    | 0.9690      | 0.9317     | 0.3054 | 0.8214 |
| O_SN_02 | 0.9562    | 0.9341      | 0.6256     | 0.0426 | 0.0895 |
| O_SU_01 | 0.9186    | 0.9065      | 0.6774     | 0.8115 | 0.6527 |
| O_SU_02 | 0.9413    | 0.9388      | 0.7669     | 0.9021 | 0.8074 |
| Average | 0.9460    | 0.9068      | 0.8031     | 0.7826 | 0.7386 |

**D. Evaluation of Arithmetic Distribution for Background Subtraction**

In this section, a comprehensive evaluation of the proposed approach is presented through comparisons with several state-of-the-art methods including deep learning networks on the LASIESTA [81] and CDNet Dataset [80]. The best of our knowledge, after the rise of deep learning networks in the background subtraction field, the fairness of comparisons between deep learning methods has been a concern. It is commonly accepted that the quantity of training data and the number of parameters in a network have significant and direct contribution to the performance of various methods [91]; but, the assumptions of training data, numbers of parameters in the network and the utilization of pre-trained networks in these methods are completely different. A few methods have generated almost perfect results with the assumption of a large number of ground truth frames for training. For example, FgSegNet model [92] achieves over 99% in Fm value on the CDNet Dataset [80], when 200 frames of ground-truth mask from each video are extracted for training and over hundreds of millions of parameters are used in the network. Such algorithms did achieve better results than the proposed approach. However, the comparison between FgSegNet and the proposed approach is unfair, since we only use 20 ground truth frames for training and the number of parameters in our network is much less than their network. In addition, there are a few semi-supervised algorithms (e.g., GuidedBS [53], BSUV-Net [90] and GraphMOS [59]) which did not utilize any ground truth frames from testing videos for training. However, these methods assumed a large number of binary masks from another video for training, and used several pre-trained networks. For example, VGG-16 [93] or Mask R-CNN [94] are combined into their network. It should be noted that these pre-trained networks still need a significant amount of data for training. For instance, GraphMOS [56] employed Mask R-CNN network [94] and DeepLab network [95] in their network, and both of these two networks were well trained with a large number of masks for objects segmentation or semantic segmentation which are similar to the ground masks of background subtraction. In contrast, the proposed approach does not use any pre-trained models or extra training data from others datasets. The comparison between the proposed approach and GraphMOS model is also questionable. GraphMOS is devised for unseen videos but many training frames may already contain information from testing videos; and although the proposed approach only assumes less than 1 % of ground truth frames for training, such frames are still extracted from testing videos. Thus, before the comparison between the proposed approach and other compared algorithms, the training data and pre-trained networks utilized in algorithms are discussed. The proposed approach is compared
The quantitative evaluation of the proposed ADNN incorporating the improved Bayesian refinement model (ADNN-IBRM) on the LASIESTA [81] dataset is shown in Table VII. During the evaluation, both the proposed approach and the D-DPDL model randomly extracted 3 ground truth frames of each video for training. This takes only 0.72% of the ground truth frames available in the LASIESTA [81] dataset. As shown in Table VII, the proposed approach achieves better results in almost all these videos compared to D-DPDL [78], since the proposed approach learns the entire histogram rather than an expected value of the histogram. Unfortunately, the proposed ADNN-IBRM does not work very well for the videos “I_II_01” and “I_II_02.” In these videos, the illumination varies over time, but only 3 ground truth frames are randomly selected during training. The quantity of training frames is very small. Thus, it is possible that the time intervals of training frames are very short, and the information on global illumination change is not included in the training frames. This results in the poor results of the proposed approach.

Quantitative and qualitative evaluations of the proposed approach on the CDNet2014 [80] dataset are shown in Table VII and Fig. 6, respectively. In particular, ADNN demonstrates the results of the proposed approach without improved Bayesian refinement model, and ADNN-IBRM presents the results of the complete proposed approach. The number of parameters of our ADNN is around 1 million, which is much less than the ones for other compared methods based on deep learning networks. In addition, no pre-trained network or extra training data is utilized in the proposed approach. The details of network architecture of the proposed approach is shown in Table VI and Fig. 6, respectively. In particular, ADNN demonstrates the results of the proposed approach without improved Bayesian refinement model, and ADNN-IBRM presents the results of the complete proposed approach. The number of parameters of our ADNN is around 1 million, which is much less than the ones for other compared methods based on deep learning networks.
As shown in Table VII, the proposed ADNN-IBRM has the best overall performance compared to both unsupervised and supervised methods. The ADNN excluding the improved Bayesian refinement model also achieves promising results. Actually, most methods based on deep learning networks are promising. However, it should be noted again that all of these methods assume much more ground truth masks than the proposed approach. Also, the number of parameters in these other networks are much greater than the proposed approach. Unfortunately, the proposed approach does not produce good results for videos in the “nightVideos” (Nig. Vid.) and “PTZ” categories. The intensity of pixels in the “nightVideos” category is very low; the pixels are too dark and do not generate enough distribution information for learning. Thus, the proposed approach only achieves 0.69 on the Fm metric. Videos in the “PTZ” category are obtained by a moving camera, but the motion of the camera is not large. Hence, the distributions of temporal pixels can be used in a short time interval, which can be learned. Thus, the proposed approach achieves a 0.74 Fm value.

The proposed approach is implemented in Pytoch [79], and the source code will be made available following acceptance of our paper. Experiments are run on a GeForce GTX 1080 GPU processor with 8 GB memory. During training 60 epochs are set as the maximum, the learning rate is set to 0.0001 and the Adam method [99] with default parameters is used for training.

VI. CONCLUSION

We proposed the Arithmetic Distribution Neural Network (ADNN) for background subtraction. Specifically, the arithmetic distribution layers, including the product and sum distribution layers, are included in our ADNN. The proposed approach improves the robustness and accuracy of the proposed approach. Utilizing the arithmetic distribution layers, histograms are considered as probability density functions. This probability information is used during the learning procedure of the proposed approach. Moreover, since the distribution of temporal pixels is relatively independent of the scene information, the proposed approach is effective even when the training and testing frames are acquired from different videos. Comprehensive evaluations comparing with
state-of-the-art methods showed the superior performance of the proposed approach, and demonstrated its potential for use in practical applications.

REFERENCES

[1] M. D. Springer, “The algebra of random variables,” Tech. Rep., 1979.
[2] O. Barnich and M. Van Droogenbroeck, “Vibe: A universal background subtraction algorithm for video sequences,” IEEE Trans. Image Process., 2011.
[3] D.-S. Lee, “Effective gaussian mixture learning for video background subtraction,” IEEE Trans. Pattern Anal. Mach. Intell., 2005.
[4] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in Int. Conf. Pattern Recognit. (ICPR), 2004.
[5] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” IEEE Trans. Image Process., 2004.
[6] C. R. Jung, “Efficient background subtraction and shadow removal for monochromatic video sequences,” IEEE Trans. Multimedia, no. 3, pp. 571–577, April 2009.
[7] S. Varadarajan, D. Miller, and H. Zhou, “Spatial mixture of Gaussians for dynamic background modelling,” in Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance, 2013, pp. 63–68.
[8] T. S. F. Haines and T. Xiang, “Background subtraction with dirichlet-process mixture models,” IEEE Trans. Pattern Anal. Mach. Intell., no. 4, pp. 670–683, April 2014.
[9] P. Tiefenbacher, M. Hofmann, D. Merget, and G. Rigoll, “Pid-based regulation of background dynamics for foreground segmentation,” in IEEE Int. Conf. Image Process. (ICIP), Oct 2014, pp. 3282–3286.
[10] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models,” in IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), June 2014, pp. 420–424.
[11] G. Ramírez-Alonso and M. I. Chacón-Murguía, “Auto-adaptive parallel svm architecture with a modular analysis for dynamic object segmentation in videos,” Neurocomputing, pp. 990 – 1000, 2016.
[12] D. Liang, S. Kaneko, M. Hashimoto, S. Lee, and T. Le-Tien, “Nic: A robust background extraction algorithm for foreground detection in dynamic scenes,” Pattern Recognit., no. 4, pp. 1374 – 1390, 2015.
[13] S. Bianco, G. Ciocca, and R. Schettini, “How far can you get by combining change detection algorithms?” in Int. Conf. Image Analysis and Process., 2017.
[14] Y. Chen, J. Wang, and H. Lu, “Learning sharable models for robust background subtraction,” in IEEE Int. Conf. Multimedia and Expo (ICME), June 2015, pp. 1–6.
[15] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, “Subsense: A universal change detection method with local adaptive sensitivity,” IEEE Trans. Image Process., no. 1, pp. 359–373, Jan 2015.
[16] X. Liu, J. Yao, X. HONG, X. Huang, Z. Zhou, C. Qi, and G. Zhao, “Background subtraction using spatio-temporal group sparsity recovery,” IEEE Trans. Circuits Syst. Video Technol., no. 99, pp. 1–1, 2017.
[17] H. Yong, D. Meng, W. Zuo, and L. Zhang, “Robust online matrix factorization for dynamic background subtraction,” IEEE Trans. Pattern Anal. Mach. Intell., July 2018.
[18] S. Minnema and Y. Wang, “An admm approach to masked signal decomposition using subspace representation,” IEEE Trans. Image Process., July 2019.
[19] T. Bouwmans, F. El Baf, and B. Vachon, “Statistical background modeling for foreground detection: A survey,” in Handbook of pattern recognition and computer vision. World Scientific, 2010.
[20] K. Goyal and J. Singhai, “Review of background subtraction methods using gaussian mixture model for video surveillance systems,” Artificial Intelligence Review, 2018.
[21] “Review of background subtraction methods using gaussian mixture model for video surveillance systems,” Artificial Intelligence Review, Jan 2017.
[22] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1999, pp. −252 Vol. 2.
[23] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, “Background modeling and subtraction by codebook construction,” in Int. Conf. Image Process. (ICIP), Oct 2004, pp. 3061–3064.
[24] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” J. ACM, no. 3, Jun. 2011.
[25] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, “Background–foreground modeling based on spatiotemporal sparse subspace clustering,” IEEE Trans. Image Process., Dec 2017.
[26] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. K. Jung, “Moving object detection in complex scene using spatiotemporal structured-sparse rpa,” IEEE Trans. Image Process., Feb 2019.
[27] T. Huynh-The, O. Banos, S. Lee, B. H. Kang, E. S. Kim, and T. Le-Tien, “Nic: A robust background extraction algorithm for foreground detection in dynamic scenes,” IEEE Trans. Circuits Syst. Video Technol., no. 7, pp. 1478–1490, July 2017.
[28] S. Jiang and X. Lu, “Wesame: A weight-sample-based method for background subtraction,” IEEE Trans. Circuits Syst. Video Technol., no. 99, pp. 1–1, 2017.
[29] 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, 2020.
[30] H.-H. Lin, T.-L. Liu, and J.-H. Chiang, “A probabilistic svm approach for background scene initialization,” in Int. Conf. Image Process. (ICIP), June 2002, pp. 893–896 vol.3.
X. Jia, W. Li, J. Liu, and Y. Zhang, “Label distribution learning by
IEEE Trans. Knowledge and
F. Cheng, S. Huang, and S. Ruan, “Advanced background subtraction ap-
T. Akilan, Q. J. Wu, and Y. Yang, “Fusion-based foreground en-
P. Hou, X. Geng, Z.-W. Huo, and J. Lv, “Semi-supervised adaptive label
ting for video sequence background subtraction,” Pattern Recognit.,
X. Liang, S. Liao, X. Wang, W. Liu, Y. Chen, and S. Z. Li, “Deep back-
ground subtraction with guided learning,” in IEEE Int. Conf. Multimedia
A. K. Dongdng Zeng, Ming Zhu, “Combining background subtraction
IEEE Transactions on Pattern Analysis and Machine
Y. Ge, J. Zhang, X. Ren, C. Zhao, J. Yang, and A. Basu, “Deep variation
T. S. F. Haines and T. Xiang, “Background subtraction with dirichlet
M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M. Yang, “Spatiotem-
M. Chen, Q. Yang, Q. Li, G. Wang, and M.-H. Yang, “Spatiotem-
T. Akilan, Q. J. Wu, and Y. Yang, “Fusion-based foreground en-
F. Cheng, S. Huang, and S. Ruan, “Advanced background subtraction ap-
G. Gemignani and A. Rozza, “A robust approach for the background
Z. Zeng, J. Jia, D. Yu, Y. Chen, and Z. Zhu, “Pixel modeling using his-
B. Han, D. Comaniciu, and L. Davis, “Sequential kernel density ap-
zidual layer self-organizing maps,” IEEE Trans-
Z. Liu, K. Huang, and T. Tan, “Foreground object detection using
top-down information based on em framework,” IEEE Trans. Image
D. Li, L. Xu, and E. Goodman, “A fast foreground object detection
B. Gao, C. Xing, C. Xie, J. Wu, and X. Geng, “Deep label distribution
Z. He, X. Li, Z. Zhang, F. Wu, X. Geng, Y. Zhang, M. Yang, and
Z. Huo, X. Yang, C. Xing, Y. Zhou, P. Hou, J. Lv, and X. Geng, “Deep
gage distribution learning for apparent age estimation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2016.