Combination of Video Change Detection Algorithms by Genetic Programming

Simone Bianco, Gianluigi Ciocca, and Raimondo Schettini

***Abstract*—Within the field of computer vision, change detection algorithms aim at automatically detecting significant changes occurring in a scene by analyzing the sequence of frames in a video stream. In this paper we investigate how state-of-the-art change detection algorithms can be combined and used to create a more robust algorithm leveraging their individual peculiarities. We exploited genetic programming (GP) to automatically select the best algorithms, combine them in different ways, and perform the most suitable post-processing operations on the outputs of the algorithms. In particular, algorithms' combination and post-processing operations are achieved with unary, binary and *n*-ary functions embedded into the GP framework. Using different experimental settings for combining existing algorithms we obtained different GP solutions that we termed *In Unity There Is Strength*. These solutions are then compared against state-of-the-art change detection algorithms on the video sequences and ground truth annotations of the ChangeDetection.net 2014 challenge. Results demonstrate that using GP, our solutions are able to outperform all the considered single state-of-the-art change detection algorithms, as well as other combination strategies. The performance of our algorithm are significantly different from those of the other state-of-the-art algorithms. This fact is supported by the statistical significance analysis conducted with the Friedman test and Wilcoxon rank sum post-hoc tests.**

***Index Terms*—Algorithm combining and selection, change detection, ChangeDetection.net (CDNET), genetic programming (GP).**

I. Introduction

MANY computer vision applications require the detection of significant changing or moving areas within the frames of video streams. In most applications, the changing areas correspond to moving objects or novel objects entering in a scene. In these applications, these objects are considered *foreground* regions in contrast to the, supposedly static, *background* region. Since change detection algorithms need to identify these two types of regions, i.e., by labeling pixels or grouping pixels as belonging to the foreground or background, often the term *background/foreground segmentation* is used. Moreover, since the relevant regions are the foreground ones, the term *background subtraction* is also often used. An example of an application that needs a robust video change detection algorithm as a preprocessing step is video surveillance. These applications need to track moving objects or identify abandoned ones to trigger event-related actions [1] by analyzing and monitoring the video content. Other applications that need video change detection are smart environments and video indexing and retrieval.

Starting from approaches based on a simple difference of pixels, many video change detection algorithms have been proposed in [2]–[5]. Regardless of the rationale of the approach, the goal of any video change detection algorithm is to segment the scene into foreground and background components while trying to cope with the challenges that can be found in real-world videos such as high variation in environmental conditions, illumination changes, shadows, camera-induced distortions, and so on. The output is thus generally noisy, with isolated pixels, holes, and jagged boundaries. Post-processing of the foreground component, ranging from simple noise removal to complex object-level techniques, has been investigated to improve the algorithm's accuracy. Results indicate that significant improvement in performance are possible if a specific post-processing algorithm and the corresponding parameters are set appropriately [6].

Notwithstanding the improvements, change detection algorithms have been demonstrated to perform well on some types of videos but there is no single algorithm that is able to tackle all the challenges in a robust way. This can be clearly seen in the ChangeDetection.net (CDNET) 2014 competition [4], [7] (which follows the CDNET 2012 competition [8]) where change detection algorithms are evaluated on a common dataset composed of different types of videos sequences and classified according to their performance. The video sequences are grouped into categories and each category poses different challenges to the change detection algorithm (e.g., static versus moving camera, day versus night lighting, etc.). There is no single algorithm that is able to successfully manage all the challenges, instead different algorithms are best suited to different problems. For example, most approaches in the literature are designed for a static camera setup, and fail when used with moving cameras such as pan-tilt-zoom (PTZ) ones. To cope with the variability of real-world videos, algorithms are becoming increasingly complex and thus computationally expensive. Parallelization of these algorithms on GPU is a possible way to make them usable in real time applications [9]–[11]. Another way to improve performance limiting the complexity overhead could be to

Manuscript received April 20, 2016; revised September 16, 2016, January 5, 2017, and March 20, 2017; accepted April 4, 2017. Date of publication April 13, 2017; date of current version November 28, 2017. *(Corresponding author: Simone Bianco.)*

The authors are with the Department of Informatic Systems and Communications, University of Milano-Bicocca, 20126 Milan, Italy (e-mail: bianco@disco.unimib.it; ciocca@disco.unimib.it; schettini@disco.unimib.it).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2017.2694160

properly combine state-of-the-art algorithms using simple operators.

The problem is how to choose the suitable algorithms to combine and what combination strategy to apply. The selection and combination should be carried out in an automatic way in order to explore possible solutions (under the given assumptions) and to find a suitable one. Approaches based on evolutionary algorithms (EAs) can be suitable candidates to perform such search [12]–[15].

Inspired by the effectiveness of EAs, to build our change detection algorithm from existing ones, we rely on genetic programming (GP) [16]. As input we feed it the set of the binary foreground/background masks that correspond to the outputs of the change detection algorithms, and a set of unary, binary, and *n*-ary functions to perform both mask's combination (e.g., logical AND and logical OR) as well as mask's post-processing (e.g., filter operators). We base the fitness function to be optimized on a set of standard performance measures, computed on a benchmark dataset of different video sequences. The solution tree obtained by GP will give our change detection algorithm.

The advantage of using GP is threefold. First, we are able to automatically select the algorithms that give the best overall results relative to a set of predefined algorithms. Second, how to combine algorithms to generate intermediate masks, and with which ones, is automatically deduced. Third, which kind of post-processing of the original or intermediate masks to be applied in order to improve the results, is automatically built from the unary, binary and *n*-ary functions. To the best of our knowledge, this is the first work that uses GP to select and combine different video change detection algorithms.

The organization of this paper is as follows. Section II provides an overview of literature works most directly related to this paper. Section III illustrates how GP is used to generate the combined change detection algorithm. In Section IV, we describe the experimental setup used in the evaluation of the proposed solution and we report and discuss the corresponding results along with their statistical significance analysis. Moreover, we analyze the contributions of the selection, combination, and post-processing components of the proposed solution. Finally, Section V concludes this paper.

II. Related Work

A. Change Detection Algorithms

In the last decades, many algorithms have been proposed to solve the problem of video change detection [3], [17]–[20]. The simplest strategy to detect foreground regions in video is to directly subtract the pixels in the current frame from those in a previous or reference one [21]. Although efficient, this approach is sensitive to noise and illumination changes. To limit these issues, temporal or adaptive filters can be applied to build the background model. For example median filter [22]–[24], Kalman filter [25]–[27], and a simplified version of Kalman filter called Wiener filter [28] have been applied. Pixel values can also be analyzed in a given time slot using color histograms and considering the mode [29].

Another way to statistically represent the background is to consider the history over time of the values of the pixels. For example, the background can be modeled as a single Gaussian [30] or a mixture of Gaussians [31]. The latter overcomes the limitation of the unimodal model that cannot handle dynamic background motion. The approaches using the Gaussian model can be also extended by incorporating the generalized Gaussian model [32], [33]. Bayesian approaches have also been proposed to cope with backgrounds having large variations [34], [35]. For example, Li *et al.* [36] proposed a Bayesian framework that incorporates spectral, spatial, and temporal features to characterize the background appearance in complex environments, while Benedek and Szirányi [37] used spatial statistics of the neighboring pixel values to robustly detect the foreground against an object's shadow. A hybrid moving object detection system that uses motion, change, and appearance information for more reliable detections is presented by Wang *et al.* [38]. The method called flux tensor with split Gaussian models (FTSG) uses a split Gaussian method to separately model foreground and background.

The above statistical methods require the definition of the model's parameters. Non parametric methods directly rely on the observed data to statistically model the background [39]. Although these methods can deal with fast changes in the background, they are time consuming, and have an high memory requirement. Improvements have been proposed to overcome these problems [40]–[42].

Sample consensus is another non parametric strategy used to model background pixels. It has been recently used in ViBe [43] and SuBSENSE [44], [45]. Sample consensus determines if a given observation should be considered foreground or background based on its similarity to recently observed samples. ViBe and SuBSENSE use different features (RGB for ViBe and LBP for SubSENSE) and SuBSENSE uses a feedback mechanism to continuously improving the pixel's modeling. In order to reduce the number of samples to be used to model the background, Wang and Dudek [46] exploited a small set of adaptive background templates. The templates are automatically discarded based on their estimated efficacy and new templates are added in their stead.

Subspace learning is another family of background modeling strategies. The frame image is considered as a whole and, by taking into account spatial information, they are more robust to illumination changes. For example, Oliver *et al.* [47] proposed an eigenbackground model. A set of images are used to build a vectorization representation of the scene within a given time frame. This representation is then decomposed via principal component analysis to determine the background via the most descriptive eigenvectors. Using a subset of eigenvectors makes the background more robust to illumination changes. The need to efficiently update the background model within the video sequence has generated many variants such as incremental [48], incremental non-negative matrix factorization [49], and robust matrix factorization [50].

Other methods try to learn information about background or foreground by using machine learning techniques. These techniques conveniently incorporate domain knowledge from available samples. For example, Lin *et al.* [51] used the normalized optical flow and normalized frame differences with a probabilistic SVM to build a background block classifier. Han and Davis [52] integrated color, gradient, and Haar-like features to handle spatio-temporal variations for each pixel within a kernel density approximation framework, while background subtraction is performed using SVM. To overcome the problem of providing large sets of positive and negative examples to the learner, Cheng *et al.* [10] proposed a 1-class SVM method that is able to update the classifier's parameters online.

Also, neural network-based solutions have received considerable attention. For example, the background segmentation approach proposed by Culibrk *et al.* [53] relies on a Probabilistic Neural Networks combining a neural network for background modeling and a Bayesian classifier for pixel's foreground/background detection. Maddalena and Petrosino [54] designed the SC_SOBS algorithm that models the background with the weights of a neural network. A modified version of the algorithm is proposed by Ferone and Maddalena [55] where the neural network is used to specifically detect moving object for PTZ cameras. A weightless neural network is proposed by Gregorio and Giordano [56]. The approach is called CwisarDH and uses a buffer of pixel values to store previous foreground values in order to make the algorithm more robust against intermittent objects.

The above mentioned approaches are based on visual features computed on the video frames, either at pixel or higher semantic level. A physics-based change detection approach is proposed by Sedky *et al.* [57]. It uses image formation models to computationally estimate a consistent physics-based color descriptor of the spectral reflectance of foreground and background surfaces.

B. Combination and Fusion Techniques

Several attempts to combine the outputs of different change detection algorithms have been investigated. For example the creators of the CDNET challenge have evaluated the performances of the top-3, top-5, and all the 28 reviewed algorithms using a simple majority vote (MV) fusion strategy [4]. The first two fusion schemes obtained the best results in seven performance measures with respect to the top-ranked algorithm and even with respect to the fusion of all the 28 tested algorithms. Also Jodoin *et al.* [58] explored a fusion scheme using MV. In this case, the results of 22 algorithms have been fused as well as subsets of 3, 5, and 7 methods. Results show that the combinations of different algorithms perform better than single ones.

Solutions used for combining classifier could also be used for combining change detection algorithms. By considering the output masks generated by the algorithms as responses of pixel-based classifiers, the problem can be seen as selection and combination of, in our case, binary classifiers (foreground versus background). Several works studied the problem of classifier combining or fusion (e.g., [59]–[64]).

Also in the context of image segmentation, the fusion of different algorithms' outputs is often exploited to obtain a more robust segmentation algorithm. For example, Aljahdali and Zanaty [65] investigated several fusion rules to improve segmentation accuracy. Specifically, the median, mean, product, minimum, and maximum rules are considered, with the mean rule obtaining the overall best performances on the segmentation datasets considered.

If the outputs of the classifier can be expressed as posteriori probabilities, a Bayesian methodology can be used to integrate the belief measure associated with each classifier to provide a combined final belief [61]. For example, Warfield *et al.* [66], [67] used the STAPLE algorithm, an expected maximization strategy, for estimating the "*ground truth*" segmentation from a group of experts' segmentations in the context of medical imaging. STAPLE takes different segmentations and simultaneously estimates the final segmentation and the sensitivity and specificity parameters characterizing the performance of each expert. A similar approach is also used by Rohlfing *et al.* [68] to estimate the final segmentation from atlas-based segmentations of 3-D con-focal microscopy images of bee brains. Mignotte [69] designed the probabilistic rand index-based algorithm (PRIF) fusion scheme. This scheme is based on a Markovian Bayesian fusion procedure, and the fusion is guided by the PRIF [70]. This index measures the agreement of one segmentation result to multiple ground-truth segmentations, in a quantitative and perceptual way.

Recently Wang *et al.* [71] formalized the problem of image segmentation fusion as a combinatorial optimization problem in terms of information theory. To reduce the computational complexity required with respect to previous optimization techniques a generative Bayesian image segmentation fusion model is proposed.

C. Image-Related Genetic Programming

EAs attempt to solve complex problems by finding the optimal solution mimicking aspects of the natural evolution of biological systems. Specifically, individuals in a population evolve and compete with each other toward a defined goal. Different EA approaches have been proposed, such as genetic algorithm (GA), evolutionary strategy, GP, and memetic algorithm. All these approaches have been successfully applied to a wide range of problem domains: optimization [72], parameters' estimation [73], classification [74], feature selection [75], and in image processing and computer vision applications [76]–[78].

With respect to image-related applications, GP has been widely used for image segmentation, enhancement, classification, feature extraction, and object recognition.

For example, Song and Ciesielski [79] used GP to segment texture images under a supervised learning approach. Overlapping image regions are processed and labeled as belonging to one of the defined texture classes. A MV strategy on the multiple pixel labels assign the final class for each pixel. Singh *et al.* [80] tackled the segmentation as a recognition problem. Segmentation programs are evolved by GP from a pool of low level image analysis operators in order to obtain the one able to perform the most accurate segmentation.

Existing segmentation approaches can be improved by GP as done, for example, by Amelio and Pizzuti [81], where segmentation is performed using a normalized cut algorithm [82]. Images are represented as weighted undirected graphs and GP is used to find an optimal partitioning of the graphs corresponding to the final segmentation.

For the task of image enhancement, GP has been used to create pseudo-colored images for visualization purposes as done by Poli and Cagnoni [83]. Gray-scale images are optimally colored in such a way to enhance the readability of magnetic resonance images. Image enhancement can be achieved by appropriate linear and non linear image filters. An approach for the automatic construction of image filters using GP for different image analysis tasks is proposed by Pedrino *et al.* [84]. By combining input images, goal images, and a set of image processing operators, the developed algorithm searches for the best solution that can be directly used for hardware control.

One of the most important task in image processing and analysis pertains the classification of image contents. An early example of application of GP to image classification tasks is the work presented by Agnelli *et al.* [85]. Simple arithmetic operations, along with exponential function, conditional function and constants are used to construct binary classification trees on a set of domain-specific features detecting image primitives. Zhang and Smart [86] investigated GP for multiclass object classification. Instead of relying on fixed thresholds to separate the class boundaries, a multiclass classifier is build using GP to dynamically determine a set of boundaries to distinguish between different classes. Muni *et al.* [87] used GP to build a multitree classifier consisting of evolved trees, where each tree represents a classifier for a particular class. Trees with poor performances are given more chances to evolve and improve their performances exploiting a tree unfitness concept. Usually, GP image classifiers are based on selected domain-specific image features. Differently, Al-Sahaf *et al.* [88] used GP to classify raw images directly. A two-tier GP is used for both image feature extraction and image classification. The features are self-constructed by GP along the evolutionary process in the first tier. The second tier makes classification decisions. In order to build reliable classifiers, many labeled data are required during the training phase. To overcome this issue, several GP image classification strategies have been recently proposed that use few labeled instances of each class to evolve classifiers capable of generalizing to unseen data [89], [90].

Using the appropriate image features is essential to successfully approach a given task. Often these features are manually chosen or crafted by domain experts. GP has been used to automatically derive the most effective features for the problem that must be solved. For example, Krawiec and Bhanu [91] used features for object recognition that are automatically coevoluted along with image processing operators. Trujillo and Olague [92] used GP to synthesize low-level image operators that detect interesting points on digital images. The algorithm generates improved versions of existing image processing operators as well as new ones. Transform-based evolvable features are introduced by Kowaliw *et al.* [93]. A Cartesian GP algorithm is used to generate them for image classification. The idea at the base of these features has been extended for the problem of object recognition [94]. The authors introduced a network superstructure that co-evolves with the low-level GP representations, and that is able to generate improved image features. Recently, GP has been used to synthesize rotation-invariant texture image descriptors using few image samples [95]. The approach is able to automatically discover rotation-invariant image keypoints that can be effectively used for texture classification. GP has also been used together with transfer learning to solve complex image classification problems by extracting and reusing blocks of knowledge/information, which are automatically discovered from similar as well as different image classification tasks during the evolutionary process [96].

Closely related to the problem of video change detection, is that of motion detection in a scene. Pinto and Song [97] used a multiframe accumulate representation to describe the video frames. GP is exploited to generate a motion detector (i.e., learn a classifier) which can differentiate "*motion*" versus "*no-motion*" areas. Shi and Song [98] presented a GP-evolved motion detector. The algorithm is able to detect a target motion ignoring irrelevant ones, and is capable of distinguishing different kinds of motions. Action recognition requires the temporal analysis of a sequence of images by using suitable image descriptors able to capture motion trajectories. Liu *et al.* [99] used spatio-temporal motion features automatically evolved via GP on a population of primitive 3-D operators. The approach outperforms state-of-the-art hand-crafted and machine learned techniques.

III. Proposed Approach

As stated in the introduction, our idea is to combine together different change detection algorithms. We named our proposed approach IUTIS quoting the Greek fabulist Aesop (620 BC–560 BC): "*In Unity There Is Strength.*"

Since there is no clear procedure to obtain a robust change detection algorithm by combining existing ones, a possible solution could be that of designing it by a trial-and-error process as done by Goyette *et al.* [4] and Jodoin *et al.* [58] where different algorithms selections are tested. Instead, we propose an approach to automatically determine a good selection and combination of algorithms using GP [16].

The major difference between GP and the other EAs is that GP is a domain-independent evolutionary method that genetically breeds a population of functions, or more generally, computer programs to solve a problem. In our case the solutions correspond to fusion strategies.

The solutions can be represented as trees, lines of code, expressions in prefix or postfix notations, strings of variable length, etc. We use the representation first introduced by Koza [16]: potential solutions are represented as LISP-like tree structures built using a set of terminal symbols $\mathcal{T}$ and a set of nonterminal or functional symbols $\mathcal{F}$. The iterative process of a generic GP is given in Algorithm 1.

Algorithm 1: GP

begin
  Generate a population P composed of an even number N of individuals ;
  Generation $\leftarrow 0$;
  repeat
    Calculate the fitness f of all the individuals in population P ;
    Create a new empty population P' ;
    repeat
      Select two individuals $C_i, C_j \in P$ using the chosen selection operator ;
      Perform the crossover between C_i and C_j with probability p_c, and let $\widetilde{C}_i$ and $\widetilde{C}_j$ be the offspring. If crossover is not applied, let $\widetilde{C}_i = C_i$ and $\widetilde{C}_j = C_j$;
      Mutate each node and leaf in $\widetilde{C}_i$ and $\widetilde{C}_j$ with certain probability p_m, and let $\widehat{C}_i$ and $\widehat{C}_j$ be the offspring;
      Insert $\widehat{C}_i$ and $\widehat{C}_j$ into population P' ;
    until *until population P' is composed of exactly N individuals*;
    Perform the copy $P \leftarrow P'$ and delete P ;
    Generation $\leftarrow$ *Generation* $+ 1$;
  until *a termination condition is satisfied*;

Before running GP, we need to set a number of parameters, which are as follows.

1) The sets $\mathcal{F}$ and $\mathcal{T}$ of functional (or nonterminal) and terminal symbols that are used to build the potential solutions.
2) The fitness function $f(\cdot)$.
3) The population size N.
4) The maximum size of the individuals, typically expressed as the maximum number of tree nodes or the maximum tree depth.
5) The maximum number of generations.
6) The algorithm used to initialize the population. A set of initialization algorithms can be found in [16]).
7) The selection operator.
8) The crossover rate.
9) The mutation rate.
10) Presence or absence of elitism (i.e., preserving unaltered the best solutions to the next iteration).

Given a set of n primitive change detection algorithms $\mathcal{C} = \{C_k\}_{k=1}^{n}$, the solutions evolved by GP are built using the set of functional (or nonterminal) symbols $\mathcal{F}$ and the set of terminal symbols $\mathcal{T} = \mathcal{C}$. The functional symbols correspond to operations performed on the inputs. We explicitly incorporate into the GP framework the list of operations given in Table I along with their functional symbols. They operate in the spatial neighborhood of the image pixel, or combine (stack) the information at the same pixel location but across different change detection algorithms.

TABLE I
SET OF FUNCTIONAL SYMBOLS USED IN GP AND THEIR CORRESPONDING OPERATORS

Function	Inputs	Domain	Effect
ERO	1	spatial	Morphological erosion with a 3×3 square structuring element
DIL	1	spatial	Morphological dilation with a 3×3 square structuring element
MF	1	spatial	Median filter with a 5×5 kernel
OR	2	stack	Logical OR operation
AND	2	stack	Logical AND operation
MV	>2	stack	Majority vote

We define the fitness function used in GP taking inspiration from the CDNET website, where change detection algorithms are evaluated using different performance measures and ranked accordingly. More specifically, the performance measures used are: recall, precision, specificity, false positive ratio (FPR), false negative ratio (FNR), percentage of wrong (pixels) classifications (PWCs), and F-measure [4], [7]. Each measure is averaged across the video sequences. Given a set of video sequences $\mathcal{V} = \{V_1, \ldots, V_S\}$, and a set of performance measures $\mathcal{M} = \{m_1, \ldots, m_M\}$, the fitness of a candidate solution C_0, $f(C_0)$, is based on the average ranking of the solution across all performance metrics, averaged across al frames in all training video sequences $\mathcal{V}$. For each measure there are three components, that are weighted by $[w_0, w_1, w_2]$. Then the weighted sum is averaged across all measures. The per-measure components are as follows.

1) The individual's integer rank, when compared against the primitive algorithms, on measure m_j; since the different measure might be uncommensurable with each other, and since finding a single metric to accurately measure the ability of a method to detect motion or change without producing excessive false positives and false negatives is not trivial [4], the role of this component is to solve both issues by producing a single number.
2) The individual's value on measure m_j, shifted by the best value achieved by any primitive algorithm for that measure computed on the frames of the video sequences; the role of this component is that of adding gradient to the integer rank and to continue improving a solution when its rank is 1.
3) A size penalty equal to the proportion of primitive algorithms used by the individual. The role of this component is to force GP to select a small number of algorithms in $\mathcal{C}$ to build the candidate solutions.

Formally, the fitness function is defined as follows:

$$f(C_0) = \frac{1}{M}\sum_{j=1}^{M}\Bigg(w_0 \cdot \operatorname{rank}\big(C_0; \{m_j(C_k(\mathcal{V}))\}_{k=1}^{n}\big) + w_1 \cdot \sum_{j=1}^{M} P_1^j(C_0) + w_2 \cdot P_2(C_0)\Bigg) \quad (1)$$

where $\operatorname{rank}(C_0; \cdot)$ computes the rank of the candidate solution C_0 with respect to the set of algorithms $\mathcal{C}$ according to the measure m_j. $P_1^j(C_0)$ is defined as the signed distance between

TABLE II
TOP NINE CHANGE DETECTION ALGORITHMS LISTED ON THE CDNET 2014 CHALLENGE WEBSITE (AS OF JULY 2014) RANKED BY THEIR AVERAGE RANKING ACROSS THE 11 VIDEO CATEGORIES

Rank	Method	Abbrev.	Description	Reference
1	FTSG	FTS	Flux Tensor with Split Gaussian models	[38]
2	SuBSENSE	SBS	Self-Balanced SENsitivity SEgmenter	[45]
3	CwisarDH	CWS	Change Detection with Weightless Neural Networks	[56]
4	Spectral-360	SPC	Change Detection based on Spectral Reflectaces	[57]
5	AMBER	AMB	Extension of the Adapting Multi-resolution Background ExtractoR	[46]
6	KNN	KNN	Adaptive Gaussian Mixture Model	[40]
7	SC_SOBS	SCS	Spatial Coherence Self-Organizing Background Subtraction	[54]
8	RMoG	RMG	Region-based Mixture of Gaussians	[100]
9	KDE	KDE	Change detection based on Kernel Density Estimation	[101]

the candidate solution C_0 and the best algorithm in $\mathcal{C}$ according to the measure m_j

$$P_1^j(C_0) = \begin{cases} -m_j(C_0(\mathcal{V})) + \max\limits_{C_k:k=1\ldots n} m_j(C_k(\mathcal{V})) \\ \qquad\qquad \text{if the higher } m_j \text{ the better} \\ m_j(C_0(\mathcal{V})) - \min\limits_{C_k:k=1\ldots n} m_j(C_k(\mathcal{V})) \\ \qquad\qquad \text{if the lower } m_j \text{ the better} \end{cases} \tag{2}$$

and $P_2(C_0)$ is a penalty term corresponding to the number of different algorithms selected for the candidate solution C_0

$$P_2(C_0) = \frac{\text{\# of algorithms selected in } C_0}{\text{\# of algorithms in } \mathcal{C}}. \tag{3}$$

The relative importance of the three components of the fitness is regulated by the weights w_0, w_1, and w_2, respectively. Since we want the fitness function to be driven by the first component, i.e., the average rank, and since both P_1 and P_2 output values in the interval $[0, 1]$, we set the weights $[w_0, w_1, w_2] = [1, 0.01, 0.01]$. In this way the contribution of P_1 and P_2 are encoded starting from the hundredths, thus avoiding any risk of rank inversion.

Our proposed combination strategy is able to simultaneously achieve three goals: algorithm selection, combination and processing. The functional symbols in Table I, act both as aggregation functions for the combination, as well as image processing functions (specifically the local functions such as the morphological ones and the median filter). Moreover, the nature of the GP algorithm coupled with the penalty factor P_2, allow us to perform automatic algorithm selection in a seamless manner during the generation of the intermediate solutions. This is an advantage of our proposed strategy compared to other combination algorithms where the selection has to be performed in advance.

IV. EXPERIMENTS

In this section, we describe the experimental setup used in combining state-of-the art algorithms and the results. We based our experiments on the CDNET 2014 challenge [7] that has received great attention in the evaluation of change detection algorithms. It provides a set of video sequences of various categories that can be used to test the algorithms on different environment conditions. Moreover, it provides an evaluation protocol that can be used to compare the performances of change detection algorithms against each other.

A. Setup

As the set of algorithms to be combined we considered the top ranked ones that have been evaluated within the CDNET 2014 challenge as of July 2014. We have chosen the top ranked ones because we are interested to investigate how far can we get by combining the top performing change detection algorithms. Table II shows the top nine algorithms listed according to their average ranking across video categories. The outputs of the various algorithms are available on the website allowing us to perform our combination experiment on "*certified*" data. In our experiments, we selected a subset of the top ranked algorithms and executed the GP algorithm using a training set of video sequences of the CDNET 2014 challenge. In particular, the training set was created by extracting from each category the shortest video sequence. Specifically, these sequences are (category/sequence): baseline/pedestrians, dynamicBackground/canoe, cameraJitter/badminton, intermittentObjectMotion/parking, shadow/peopleInShade, thermal/park, badWeather/wetSnow, lowFramerate/tramCrossroad_1fps, nightVideos/winterStreet, PTZ/zoomInZoomOut, and turbulence/turbulence3.

As for performance measures we computed them using the framework of the challenge that evaluates the seven different measures listed in the previous section. A ranking of the tested algorithms is also computed starting from the partial ranks on these measures.

We evaluate our proposed combining strategy against three different fusion algorithms: 1) a simple MV scheme; 2) the STAPLE method [66], [67] (STAPLE); and 3) a PRIF [69]. Following the challenge's rules, each algorithm uses a single set of parameters. The STAPLE algorithm estimates its parameters on-line (i.e., the sensitivities and specificities of each expert) while for the PRIF algorithm we used the default parameters. The parameters used to initialize the GP algorithm are reported in Table III.

B. Results

We applied GP on different sets $\mathcal{C}$ constructed using the top n algorithms in Table II with $n \in \{3, 5, 7, 9\}$. For each setting,

TABLE III
SET OF PARAMETERS USED IN GP

Parameter name	Setting
Functional symbols	$\mathcal{F}$ (see Sec. III)
Terminal symbols	$\mathcal{T}$ (see Sec. III)
Fitness function	$f(\cdot)$ defined in Eq. 1
Population size	50
Max tree depth	Dynamic
Max number of gen.	100
Initialization algorithm	Ramped half-and-half [16]
Selection operator	Tournament, with tournament size 5
Crossover and Mutation rates	Adaptive: each operator probability value is adapted to reflect its performance. A percentage of the probability value is replaced by a value proportional to the operator's performance [102].
Elitism	Yes
Number of runs	25

the shortest best solution across the 25 independent runs is considered. The resulting algorithms, that we named IUTIS-3 and IUTIS-5, are shown in Figs. 1 and 2, respectively. In the same figures, for each solution tree, an example of the output at each node on a sample frame is also reported.

The solutions obtained with $n > 5$ are not reported since in these cases the GP algorithm found a solution identical to IUTIS-5. This is an evidence that GP is able to automatically select the best set of algorithms. This is one of the major differences between our method and the other fusion-based algorithms considered, which are not able to perform automatic algorithm selection, and thus use all the given algorithms. From the solution trees it is possible to notice that IUTIS-3 automatically created MV-3 in its right branch. IUTIS-5, instead, automatically created as part of the solution MV-3 and MV-5 in its left and right branch, respectively.

Fig. 3 shows the average rank and F-measure of our solutions compared with the fusion-based algorithms considered (i.e., MV, PRIF, and STAPLE) varying the number of algorithms available in $\mathcal{C}$. For sake of comparison with the other methods in the state of the art performance are measured on the whole CDNET 2014 challenge video sequences. Note that the results on the CDNET website are computed on the whole dataset and the algorithms compared on the website are often tuned on it. Given that the training set contains less than 10% of the total number of frames, the influence of the training set on the performances is negligible. In fact, if we consider for example the overall F-measure, IUTIS-3 obtains 0.7694 on the whole CDNET 2014 dataset, 0.7781 excluding the video sequences used for training, and 0.7413 on the training set alone. IUTIS-5, respectively, obtains 0.7821, 0.7896, and 0.7573. Similar differences are observed for the other performance measures. These numbers permit us to say that no overfitting occurred during training. For completeness, the separate train/test results are reported in Supplementary material.

From the plots reported in Fig. 3 we can see that the average rank of IUTIS-3 is 3.41 points lower than that of MV-3, while IUTIS-5 is 4.42 points lower than that

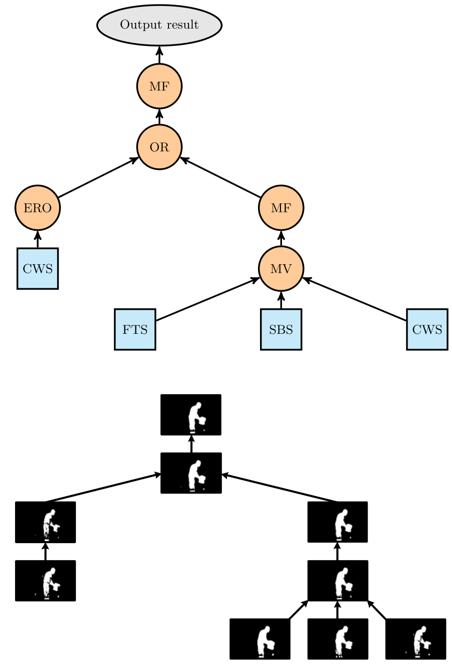

Fig. 1. IUTIS-3 solution tree and its example masks. SBS, FTS, and CWS refer to SuBSENSE, FTSG, and CwisarDH algorithms, respectively.

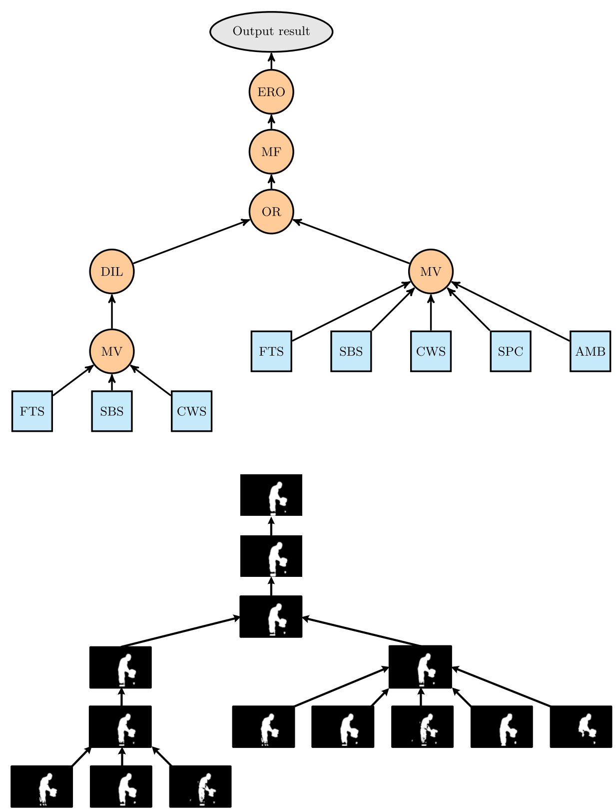

Fig. 2. IUTIS-5 solution tree and its example masks. SBS, FTS, CWS, SPC, and AMB refer to SuBSENSE, FTSG, CwisarDH, Spectral-360, and AMBER algorithms, respectively.

of MV-5. Since IUTIS-3 and IUTIS-5 contain MV-3 and MV-5 as part of them, this improvement is due to the additional filtering and post-processing operations automatically selected by GP.

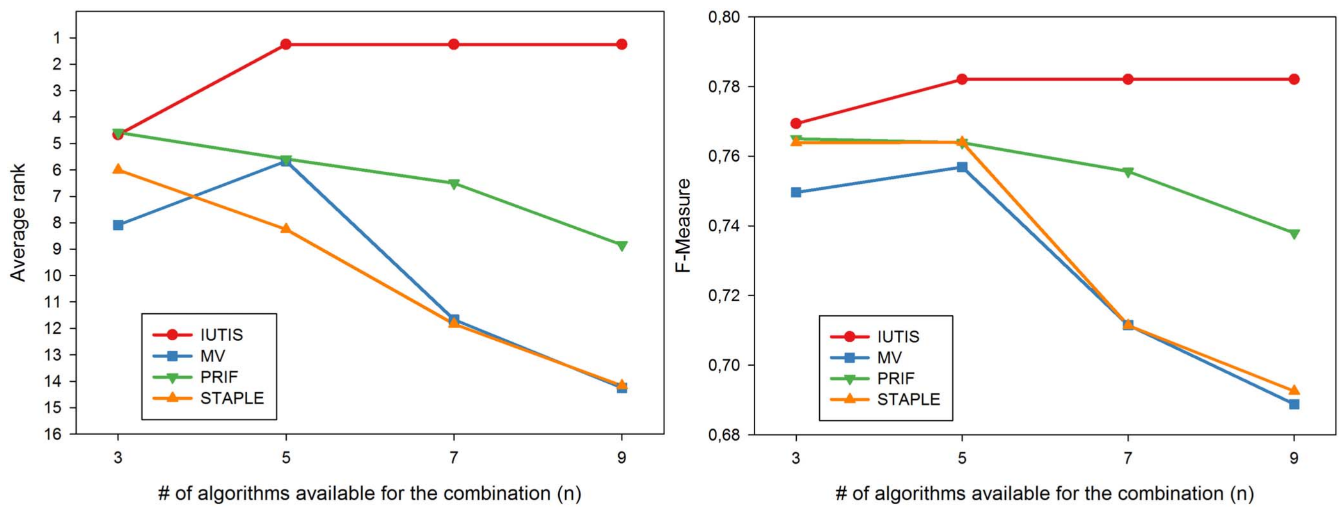

Fig. 3. Plots of the average rank (left) and F-measure (right) for the different fusion-based algorithms considered by varying the number of algorithms available in the combination, i.e., $n = 3, 5, 7, 9$.

TABLE IV
COMPARISON OF OUR PROPOSED SOLUTIONS TO SINGLE AND FUSION-BASED ALGORITHMS IN THE STATE-OF-THE-ART IN TERMS OF RANK IN EACH VIDEO CATEGORY AND AVERAGE RANK

Method ID	Avg rank	Overall	Bad Weath.	Low F.rate	Night Videos	PTZ	Turb.	Base.	Dynam. Backg.	Camera Jitter	Interm. Obj. M.	Shadow	Therm.
IUTIS-5	1.25	1	1	1	1	1	2	1	2	1	2	1	1
IUTIS-7	1.25	1	1	1	1	1	2	1	2	1	2	1	1
IUTIS-9	1.25	1	1	1	1	1	2	1	2	1	2	1	1
PRIF-3	4.58	2	3	2	3	3	5	3	9	7	6	6	6
IUTIS-3	4.67	3	2	3	8	5	9	4	4	4	8	3	3
PRIF-5	5.58	4	7	6	5	4	1	10	3	8	7	8	4
MV-5	5.67	5	9	8	2	6	10	2	10	3	3	6	4
STAPLE-3	6.00	6	4	5	4	8	6	5	5	8	12	2	7
PRIF-7	6.50	8	10	4	9	2	3	9	1	6	16	5	5
MV-3	8.08	7	11	11	6	13	8	6	12	5	4	5	9
STAPLE-5	8.25	9	6	9	13	12	4	13	7	9	5	4	8
PRIF-9	8.83	12	12	7	12	9	5	12	6	2	14	8	7
FTSG	9.67	11	8	14	10	11	12	17	8	16	1	6	2
SuBSENSE	10.67	10	5	10	7	10	7	14	15	15	17	7	11
MV-7	11.67	13	14	8	14	14	13	7	15	12	9	11	10
STAPLE-7	11.83	15	11	13	15	16	11	9	13	10	10	9	10
STAPLE-9	14.17	16	15	15	17	17	15	8	16	13	15	10	13
MV-9	14.25	17	13	12	17	15	14	11	17	14	13	16	12
CwisarDH	14.42	14	19	17	12	7	17	16	14	11	20	12	14
Spectral-360	17.00	18	18	16	11	18	18	19	17	18	21	14	16
AMBER	17.33	19	16	22	21	23	16	22	11	13	11	15	19
RMoG	18.58	20	18	21	18	21	21	21	19	19	18	13	14
SC_SOBS	18.75	21	21	20	16	20	22	15	18	17	19	18	18
KNN	19.00	22	17	18	19	19	20	20	20	20	22	16	15
KDE	20.00	23	20	19	20	22	19	18	21	21	23	17	17

We observe also how the performance of PRIF and STAPLE decrease by increasing the number of algorithms available. MV-5 instead outperforms MV-3, but also in this case the performance decrease for $n = 7$ and 9. A different trend can be observed for IUTIS, which being able to perform automatic algorithm selection, is able to remove from the final solution those algorithms that could degrade the performance. The complete comparison of our proposed solutions with respect to the single algorithms of Table II and fusion-based algorithms in the state-of-the-art is reported in Table IV. It is possible to notice that all fusion-based algorithms with $n \leq 5$ outperform all the single algorithm, while this is true only for IUTIS and PRIF for higher values of n. In particular we can observe that there are three categories on which IUTIS-5 is not ranked first, i.e., turbulence, dynamic background and intermittent object motion.

To ensure that the performance of IUTIS-5 are statistically different from those of the other algorithms, we conducted a statistical significance analysis on the results in Table VIII using the Friedman test [103], [104]. The analysis shows that there is a statistically significant difference in the performance of the algorithms with a $\chi^2(24) = 240.7$

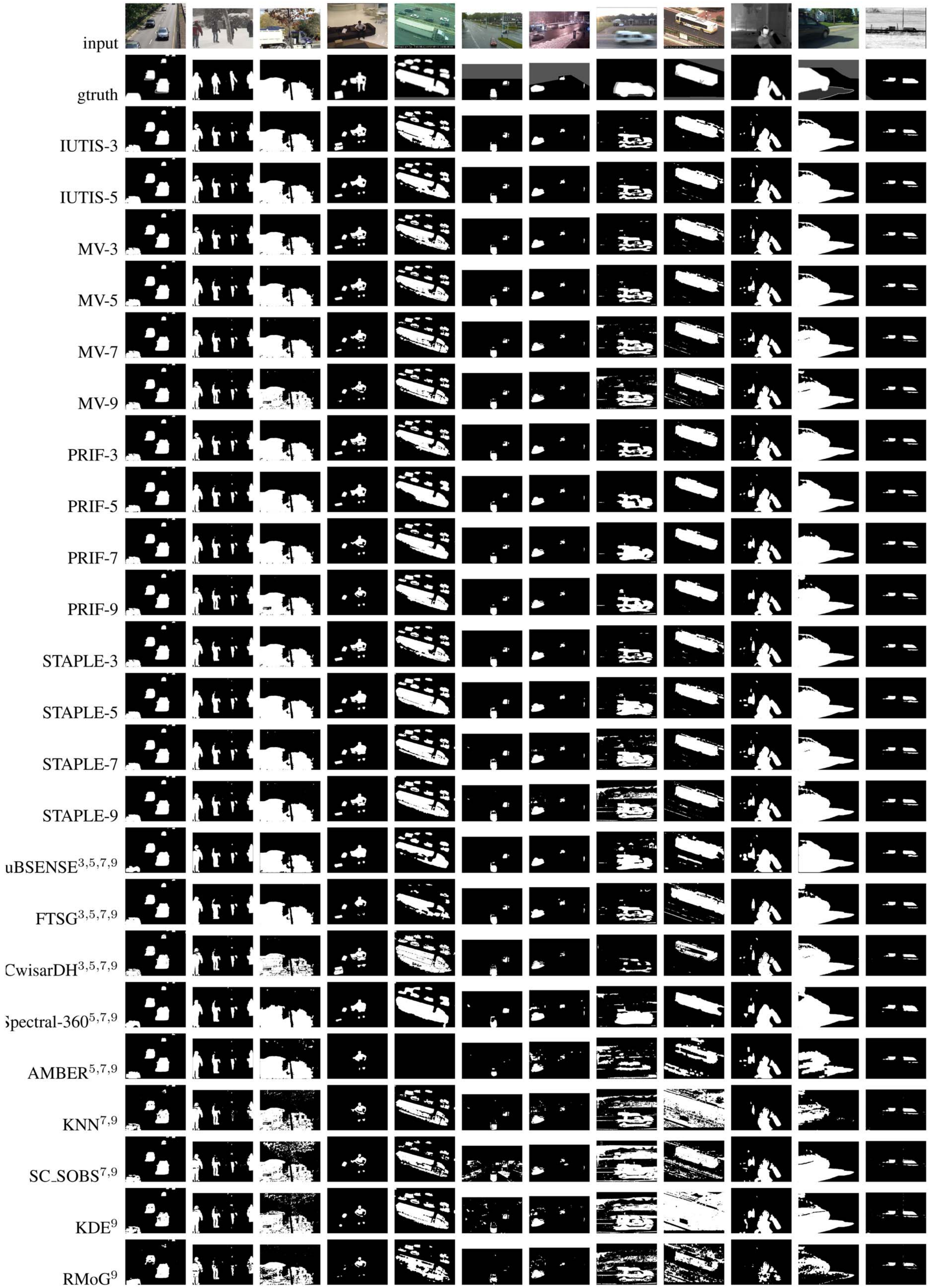

Fig. 4. Examples of binary masks created by the tested algorithms. The superscripts indicate in what fusion set $\mathcal{C}$ the algorithm is used (e.g., SuBSENSE, FTSG, and CwisarDH are used to build IUTIS-3, MV-3, PROF-3, and STAPLE-3).

and $p < 0.01$. We subsequently performed a post-hoc test using pairwise Wilcoxon rank sum test [105] at the significance level $\alpha = 0.05$. Table V shows the p-values of the 25 algorithms against each other. As it can be seen, apart from IUTIS-7 and IUTIS-9, IUTIS-5 compared against the other algorithms, obtains p-values lower than α also considering the

TABLE V
p-VALUES OF THE PAIRWISE WILCOXON RANK SUM TESTS OF THE 25 ALGORITHMS AT $\alpha = 0.05$. SIGNIFICANT DIFFERENCES, I.E., p-VALUES BELOW THE BONFERRONI-CORRECTED CRITICAL VALUE, ARE HIGHLIGHTED IN GRAY

	IUTIS-7	IUTIS-9	PRIF-3	IUTIS-3	PRIF-5	MV-5	STAPLE-3	PRIF-7	MV-3	STAPLE-5	PRIF-9	FTSG	SuBSENSE	MV-7	STAPLE-7	STAPLE-9	MV-9	CwisarDH	Spect.-360	AMBER	RMoG	SC_SOBS	KNN	KDE
IUTIS-5	1.0e+00	1.0e+00	7.4e-06	3.0e-06	2.3e-05	7.4e-06	3.0e-06	5.0e-05	7.4e-07	7.4e-07	3.0e-06	5.3e-05	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
IUTIS-7	-	1.0e+00	7.4e-06	3.0e-06	2.3e-05	7.4e-06	3.0e-06	5.0e-05	7.4e-07	7.4e-07	3.0e-06	5.3e-05	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
IUTIS-9	-	-	7.4e-06	3.0e-06	2.3e-05	7.4e-06	3.0e-06	5.0e-05	7.4e-07	7.4e-07	3.0e-06	5.3e-05	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
PRIF-3	-	-	-	8.7e-01	2.5e-01	4.3e-01	2.1e-01	2.9e-01	7.3e-03	5.0e-03	4.9e-03	1.0e-02	8.0e-05	5.2e-06	2.2e-06	1.5e-06	7.4e-07	2.2e-06	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
IUTIS-3	-	-	-	-	3.1e-01	4.5e-01	1.4e-01	2.3e-01	4.1e-03	4.0e-03	6.9e-03	1.4e-02	2.2e-04	1.5e-05	2.2e-06	3.0e-06	7.4e-07	4.4e-06	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
PRIF-5	-	-	-	-	-	9.7e-01	7.6e-01	6.8e-01	6.0e-02	4.8e-02	2.6e-02	2.1e-02	1.6e-03	3.8e-05	5.9e-06	5.2e-06	7.4e-07	9.6e-06	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
MV-5	-	-	-	-	-	-	7.6e-01	8.0e-01	6.8e-02	7.9e-02	3.9e-02	3.6e-02	2.3e-03	1.2e-04	5.0e-05	1.0e-05	7.4e-07	7.4e-06	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07	7.4e-07
STAPLE-3	-	-	-	-	-	-	-	9.0e-01	1.1e-01	7.8e-02	3.8e-02	3.6e-02	2.8e-03	6.4e-05	2.5e-05	5.2e-06	3.0e-06	1.6e-05	1.5e-06	2.2e-06	7.4e-07	7.4e-07	7.4e-07	7.4e-07
PRIF-7	-	-	-	-	-	-	-	-	2.1e-01	2.3e-01	1.3e-01	9.1e-02	1.5e-02	2.7e-03	7.4e-04	2.3e-04	5.5e-05	1.6e-04	3.7e-06	1.4e-05	3.0e-06	2.2e-06	2.2e-06	7.4e-07
MV-3	-	-	-	-	-	-	-	-	-	8.8e-01	4.6e-01	3.2e-01	1.2e-01	6.3e-03	9.1e-03	1.1e-04	3.9e-05	2.1e-04	5.2e-06	2.4e-05	1.5e-06	7.4e-07	7.4e-07	7.4e-07
STAPLE-5	-	-	-	-	-	-	-	-	-	-	7.4e-01	4.0e-01	1.1e-01	1.1e-02	6.3e-03	1.3e-04	8.4e-05	5.0e-04	4.4e-06	2.7e-05	2.2e-06	7.4e-07	7.4e-07	7.4e-07
PRIF-9	-	-	-	-	-	-	-	-	-	-	-	6.8e-01	3.4e-01	3.8e-02	4.9e-02	2.6e-04	2.6e-04	1.7e-03	1.1e-05	1.3e-04	2.2e-06	7.4e-07	7.4e-07	7.4e-07
FTSG	-	-	-	-	-	-	-	-	-	-	-	-	8.5e-01	3.2e-01	3.4e-01	2.1e-02	9.2e-03	1.6e-02	9.5e-05	9.9e-04	1.5e-05	8.1e-06	8.9e-06	2.2e-06
SuBSENSE	-	-	-	-	-	-	-	-	-	-	-	-	-	5.4e-01	4.2e-01	2.3e-02	1.9e-02	2.9e-02	9.2e-05	7.5e-04	3.7e-05	7.4e-06	7.4e-06	2.2e-06
MV-7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9.4e-01	1.6e-02	3.6e-02	6.0e-02	9.9e-05	1.9e-03	3.3e-05	1.5e-06	1.5e-06	7.4e-07
STAPLE-7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4.4e-02	2.2e-02	4.5e-02	7.5e-05	1.4e-03	2.5e-05	5.2e-06	5.2e-06	7.4e-07
STAPLE-9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8.3e-01	9.6e-01	3.2e-03	7.3e-02	8.0e-04	1.0e-04	9.8e-05	4.4e-06
MV-9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8.3e-01	3.9e-03	9.1e-02	3.2e-04	6.7e-05	7.3e-05	7.4e-06
CwisarDH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5.4e-02	1.4e-01	4.1e-03	2.2e-03	1.5e-03	1.9e-04
Spect.-360	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7.6e-01	6.8e-02	1.2e-01	5.4e-02	6.7e-03
AMBER	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6.8e-01	5.6e-01	4.1e-01	1.4e-01
RMoG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8.9e-01	8.5e-01	2.9e-01
SC_SOBS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7.9e-01	2.1e-01
KNN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3.2e-01

Bonferroni correction. This indicates that the performance of IUTIS-5 are significantly different with respect to the other algorithms examined.

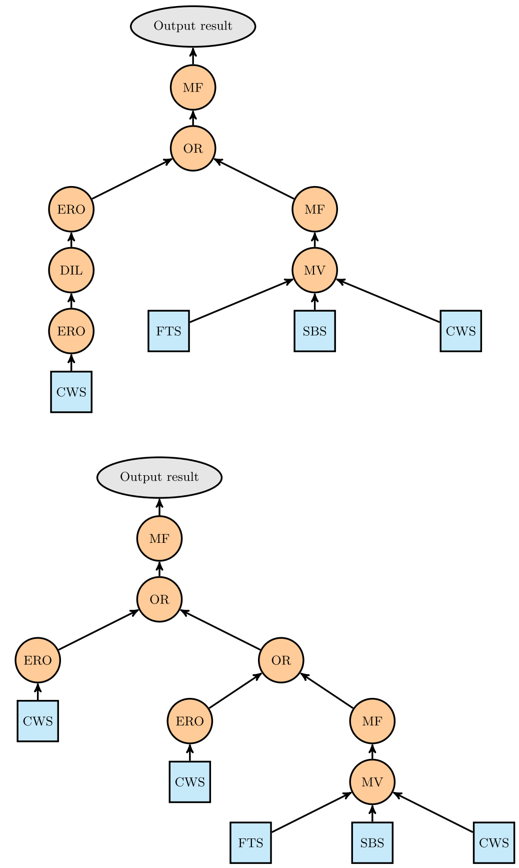

Fig. 5. Two variant solutions of IUTIS-3 found by GP. These were discarded having size greater than IUTIS-3. SBS, FTS, and CWS refer to SuBSENSE, FTSG, and CwisarDH algorithms, respectively.

Outputs of some of the tested algorithms on sample frames in the CDNET 2014 dataset are shown in Fig. 4 together with input images and ground truth masks. Detailed evaluation results of the IUTIS-3 and IUTIS-5 algorithms in terms of all the seven performance measures and for each category of the evaluation dataset are reported in Tables VI and VII, respectively.

Since the proposed algorithm is stochastic by nature, we report in Fig. 5 two variant solutions found in different runs by our GP fusion scheme using the top three algorithms in Table II. In most of the runs the solutions found are identical to IUTIS-3. In the rest of cases they have the same overall results. The variants of IUTIS-3 reported in Fig. 5 are taken among the latter ones. These solutions trees are similar to IUTIS-3, have the same overall results, and contain semantic introns [106]. In fact they possess nonfunctional branches: the sequence of erosions and dilations in the left branch of the top tree in Fig. 5 is equivalent to a single erosion; in the bottom tree, the top OR operation is uninfluential on the final result. We observe the same behavior for IUTIS-5.

Finally, Table VIII reports the complete official ranking on the CDNET 2014 website at the moment of the submission (September 14, 2016). As it can be seen, both our solutions outperform all the evaluated change detection algorithms including the most recent ones.

TABLE VI
DETAILED EVALUATION RESULTS OF THE IUTIS-3 ALGORITHM FOR EACH CATEGORY OF THE EVALUATION DATASET

Scenarios	Recall	Specificity	FPR	FNR	PWC	Precision	FMeasure
Overall	0.7896	0.9944	0.0056	0.2104	1.1813	0.7951	0.7694
Bad Weather	0.7502	0.9993	0.0007	0.2498	0.5010	0.9280	0.8246
Low Framerate	0.8183	0.9964	0.0036	0.1817	0.8224	0.7813	0.7949
Night Videos	0.6243	0.9839	0.0161	0.3757	2.4354	0.4312	0.4814
PTZ	0.6508	0.9885	0.0115	0.3492	1.4869	0.3886	0.4230
Turbulence	0.7708	0.9998	0.0002	0.2292	0.1823	0.9368	0.8416
Baseline	0.9712	0.9981	0.0019	0.0288	0.3002	0.9393	0.9546
Dynamic Background	0.8778	0.9993	0.0007	0.1222	0.1985	0.9239	0.8960
Camera Jitter	0.7923	0.9924	0.0076	0.2077	1.5231	0.8520	0.8139
Intermittent Object Motion	0.6987	0.9946	0.0054	0.3013	3.2481	0.8146	0.7136
Shadow	0.9478	0.9914	0.0086	0.0521	1.0410	0.8585	0.8984
Thermal	0.7832	0.9945	0.0055	0.2168	1.2552	0.8922	0.8210

TABLE VII
DETAILED EVALUATION RESULTS OF THE IUTIS-5 ALGORITHM FOR EACH CATEGORY OF THE EVALUATION DATASET

Scenarios	Recall	Specificity	FPR	FNR	PWC	Precision	FMeasure
Overall	0.7972	0.9952	0.0048	0.2028	1.0863	0.8105	0.7821
Bad Weather	0.7503	0.9994	0.0006	0.2497	0.4977	0.9349	0.8289
Low Framerate	0.8376	0.9974	0.0026	0.1624	0.7452	0.7724	0.7911
Night Videos	0.6333	0.9848	0.0152	0.3667	2.3252	0.4578	0.5132
PTZ	0.6687	0.9917	0.0083	0.3313	1.1465	0.4348	0.4703
Turbulence	0.7730	0.9999	0.0001	0.2270	0.1713	0.9624	0.8507
Baseline	0.9680	0.9983	0.0017	0.0320	0.3053	0.9464	0.9567
Dynamic Background	0.8636	0.9996	0.0004	0.1364	0.1808	0.9324	0.8902
Camera Jitter	0.8220	0.9925	0.0075	0.1780	1.4389	0.8511	0.8332
Intermittent Object Motion	0.7047	0.9963	0.0037	0.2953	3.0420	0.8501	0.7296
Shadow	0.9492	0.9923	0.0077	0.0508	0.9484	0.8766	0.9084
Thermal	0.7990	0.9952	0.0048	0.2010	1.1484	0.8969	0.8303

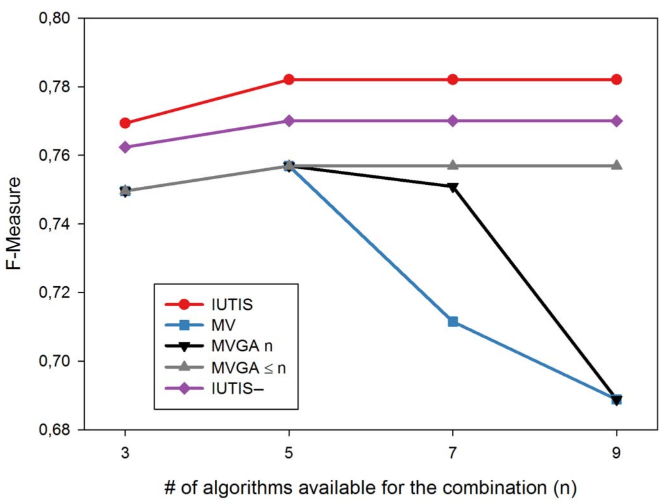

Fig. 6. Plots of the F-measure for the different baseline algorithms considered in the ablation study by varying the number of algorithms available in the combination, i.e., $n = 3, 5, 7, 9$.

C. Ablation Study

In the introduction we stated that the advantage of using GP is threefold, namely: 1) algorithm selection; 2) algorithm combination; and 3) post-processing. In this section, we conduct an ablation study to separate their contribution. Two new baselines are therefore evaluated. The first one isolates the influence of the choice of primitive algorithms from the other two contributions of the GP, and consists of the MV of n algorithms among the nine available algorithms, where the set of n members is chosen by GA. The relevant GA parameters are the same used for GP (see Table III). The second baseline isolates the influence of post-processing from the other contributions, and consists in a GP instance without the post-processing operators. The comparison of IUTIS with these baselines is reported in Fig. 6. Two different lines are present for the GA baseline: the black one (MVGA n) consists of the MV of a set of exactly n algorithms, while the gray one (MVGA $\leq n$) considers up to n algorithms. The GP instance without the post-processing operators (IUTIS−) is represented by a purple line. From the plot it is possible to see that when $n \leq 5$, algorithm selection is not able to find a solution better than MV; in these cases, the higher performance of IUTIS is due to the algorithms combination and post-processing. When $n \geq 7$ instead, most of the performance gain over the simple MV is due to algorithm selection. Even for the GA baseline, when forced to select n algorithms (i.e., MVGA n), performance start to drop. The plot of the baseline IUTIS− is always higher than that of MVGA $\leq n$ by almost a constant amount, indicating that the logical operators are useful to fuse the outputs of the different algorithms. Concerning the post-processing, its importance is evinced from the fact that the line of IUTIS is always higher than that of IUTIS−. Its influence is larger when a larger number of algorithms is considered (i.e., $n > 3$). Moreover, MVGA $\leq n$ often ends up using the same primitive algorithms as are used by IUTIS and IUTIS−.

TABLE VIII
AVERAGE RANKING OF THE ALGORITHMS OF THE CDNET WEBSITE AS PER SEPTEMBER 14, 2016

Method	Average ranking across categories
IUTIS-5	2.18
IUTIS-3	5.45
PAWCS [111]	6.36
SuBSENSE [45]	7.82
SharedModel [112]	8.64
FTSG [38]	8.82
SaliencySubsense [*]	9.45
M4CD Version 2.0 [*]	9.73
Superpixel Strengthen Backgr. Subtr. [*]	9.82
CwisarDRP [*]	10.36
M4CD Version 1.0 [*]	12.27
Multimode Backgr. Subtr. [*]	12.36
C-EFIC [113]	12.91
Multimode Backgr. Subtr. V.0 (MBS V0) [114]	14.55
EFIC [115]	14.73
CwisarDH [56]	14.82
Spectral-360 [57]	17.73
Sample based background subtractor (SBBS) [*]	18.55
AMBER [46]	19.36
AAPSA [116]	21.18
GraphCutDiff [117]	23.00
KNN [40]	23.55
SC_SOBS [54]	23.55
Mahalanobis distance [21]	24.00
SOBS_CF [118]	24.09
RMoG [100]	24.27
KDE [101]	25.73
GMM Stauffer & Grimson [31]	27.73
CP3-online [119]	27.73
GMM Zivkovic [120]	28.91
Multiscale Spatio-Temporal BG Model [121]	30.36
Euclidean distance [21]	32.00

Note: Methods with reference[*] have been submitted to journals or conferences. See the changedetection.net website for current status.

V. CONCLUSION

In this paper, we have presented an evolutionary approach, based on GP, to combine video change detection algorithms to create a more robust algorithm. The solutions provided by GP allow us to automatically select the best subset of the input algorithms. This is one of the major differences between our method and the other fusion-based algorithms considered, which are not able to perform automatic algorithm selection, and thus use all the given algorithms. Moreover, we are able to automatically combine them in different ways, and perform post-processing on their outputs using unary, binary and n-ary operators embedded into the GP framework.

We have shown that applying our approach on the best algorithms in the state-of-the-art, we are able to create the IUTIS-5 algorithm that ranks first by a large margin among a total of 32 change detection algorithms on the CDNET 2014. The statistical significance analysis performed using the Friedman test and the Wilcoxon rank sum post-hoc tests show that the performance of IUTIS-5 are significantly different from the ones of the other state-of-the-art algorithms.

As a future work we plan to investigate if the same approach, applied on very simple change detection algorithms, is able to create solutions with comparable results to more complex algorithm. Moreover, it would be interesting to use our framework to create new algorithms from scratch using atomic image processing operators as building blocks. We also plan to investigate the improvement that recent contributions in GP could give such as Pareto multiobjective [107], implicit fitness sharing [108], [109], and the use of novelty as an objective [110].

REFERENCES

[1] M. H. Sedky, M. Moniri, and C. C. Chibelushi, "Classification of smart video surveillance systems for commercial applications," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance (AVSS)*, 2005, pp. 638–643.

[2] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 4. The Hague, The Netherlands, 2004, pp. 3099–3104.

[3] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[4] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4663–4679, Nov. 2014.

[5] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: A review and comparative evaluation," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 43–60, 2016.

[6] D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proc. IEEE 5th Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Santa Fe, NM, USA, 2008, pp. 192–199.

[7] Y. Wang *et al.*, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2014, pp. 393–400.

[8] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Providence, RI, USA, 2012, pp. 1–8.

[9] V. Pham, P. Vo, V. T. Hung, and L. H. Bac, "GPU implementation of extended Gaussian mixture model for background subtraction," in *Proc. IEEE Int. Conf. Comput. Commun. Technol. Res. Innov. Vis. Future (RIVF)*, Nov. 2010, pp. 1–4.

[10] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, "Real-time discriminative background subtraction," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1401–1414, May 2011.

[11] G. Szwoch, D. Ellwart, and A. Czyżewski, "Parallel implementation of background subtraction algorithms for real-time video processing on a supercomputer platform," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 111–125, 2016.

[12] P. Ross, S. Schulenburg, J. G. Marín-Blázquez, and E. Hart, "Hyper-heuristics: Learning to combine simple heuristics in bin-packing problems," in *Proc. GECCO*, 2002, pp. 942–948.

[13] E. K. Burke, M. Hyde, G. Kendall, and J. Woodward, "A genetic programming hyper-heuristic approach for evolving 2-d strip packing heuristics," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 942–958, Dec. 2010.

[14] R. C. Barros, A. C. de Carvalho, and A. A. Freitas, "Evolutionary algorithms and hyper-heuristics," in *Automatic Design of Decision-Tree Induction Algorithms*. Cham, Switzerland: Springer, 2015, pp. 47–58.

[15] F. Rogai, C. Manfredi, and L. Bocchi, "Metaheuristics for specialization of a segmentation algorithm for ultrasound images," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 730–741, Oct. 2016.

[16] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, vol. 1. Cambridge, MA, USA: MIT Press, 1992.

[17] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[18] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent Patents Comput. Sci.*, vol. 1, no. 1, pp. 32–54, 2008.

[19] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino, "Background subtraction for automated multisensor surveillance: A comprehensive review," *EURASIP J. Adv. Signal Process.*, vol. 1, p. 43, 2010.

[20] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection—A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, 2011.

[21] Y. Benezeth, P.-M. Jodoin, B. Emile, C. Rosenberger, and H. Laurent, "Comparative study of background subtraction algorithms," *J. Electron. Imag.*, vol. 19, no. 3, 2010, Art. no. 033003.

[22] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vis. Appl.*, vol. 8, no. 3, pp. 187–193, 1995.

[23] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[24] M.-H. Hung, J.-S. Pan, and C.-H. Hsieh, "Speed up temporal median filter for background subtraction," in *Proc. 1st Int. Conf. Pervasive Comput. Signal Process. Appl. (PCSPA)*, Harbin, China, 2010, pp. 297–300.

[25] K. Karman and A. Brandt, "Moving object recognition using an adaptive background memory," in *Time-Varying Image Processing and Moving Object Recognition*, vol. 2. Amsterdam, The Netherlands, Elsevier, 1990, pp. 297–307.

[26] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using Kalman-filtering," in *Proc. Int. Conf. Recent Adv. Mechatronics*, 1995, pp. 193–199.

[27] S.-C. Sen-Ching and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Proc. Electron. Imag. SPIE*, San Jose, CA, USA, 2004, pp. 881–892.

[28] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1. 1999, pp. 255–261.

[29] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transp. Res. Record J. Transp. Res. Board*, vol. 1944, no. 1, pp. 82–88, 2006.

[30] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[31] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Fort Collins, CO, USA, 1999, pp. 246–252.

[32] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure, "Robust foreground extraction technique using background subtraction with multiple thresholds," *Opt. Eng.*, vol. 46, no. 9, 2007, Art. no. 097004.

[33] M. S. Allili, N. Bouguila, and D. Ziou, "Finite general Gaussian mixture modeling and application to image and video foreground segmentation," *J. Electron. Imag.*, vol. 17, no. 1, 2008, Art. no. 013005.

[34] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.

[35] O. Tuzel, F. Porikli, and P. Meer, "A Bayesian approach to background modeling," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, 2005, p. 58.

[36] L. Li, W. Huang, I.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[37] C. Benedek and T. Szirányi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 608–621, Apr. 2008.

[38] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split Gaussian models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, 2014, pp. 420–424.

[39] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[40] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[41] T. Tanaka, A. Shimada, D. Arita, and R.-I. Taniguchi, "A fast algorithm for adaptive background model construction using Parzen density estimation," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance (AVSS)*, London, U.K., 2007, pp. 528–533.

[42] Y. Nonaka, A. Shimada, H. Nagahara, and R.-I. Taniguchi, "Evaluation report of integrated background modeling based on spatio-temporal features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Providence, RI, USA, 2012, pp. 9–14.

[43] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[44] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Flexible background subtraction with self-balanced local sensitivity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, 2014, pp. 414–419.

[45] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[46] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, 2014, pp. 401–404.

[47] N. M. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[48] Y. Li, "On incremental and robust subspace learning," *Pattern Recognit.*, vol. 37, no. 7, pp. 1509–1518, 2004.

[49] S. S. Bucak and B. Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognit.*, vol. 42, no. 5, pp. 788–797, 2009.

[50] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung, "A probabilistic approach to robust matrix factorization," in *Computer Vision—ECCV 2012* (LNCS 7578), A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Heidelberg, Germany: Springer, 2012, pp. 126–139.

[51] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "Learning a scene background model via classification," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1641–1654, May 2009.

[52] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.

[53] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1614–1627, Nov. 2007.

[54] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?" in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Providence, RI, USA, 2012, pp. 21–26.

[55] A. Ferone and L. Maddalena, "Neural background subtraction for pan-tilt-zoom cameras," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 571–579, May 2014.

[56] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, 2014, pp. 409–413.

[57] M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: A physics-based technique for change detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, 2014, pp. 405–408.

[58] P.-M. Jodoin, S. Pierard, Y. Wang, and M. Van Droogenbroeck, "Overview and benchmarking of motion detection methods," in *Background Modeling and Foreground Detection for Video Surveillance*. Boca Raton, FL, USA: CRC Press, 2014.

[59] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[60] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2004.

[61] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Comput. Inf. Syst.*, vol. 7, no. 1, pp. 1–10, 2000.

[62] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[63] H. Wang, Y. Zhang, and G. Qian, "Multiple binary classifiers fusion using induced intuitionistic fuzzy ordered weighted average operator," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, 2011, pp. 230–235.

[64] N. García-Pedrajas and D. Ortiz-Boyer, "An empirical study of binary classifier fusion methods for multiclass classification," *Inf. Fusion*, vol. 12, no. 2, pp. 111–130, 2011.

[65] S. Aljahdali and E. A. Zanaty, "Combining multiple segmentation methods for improving the segmentation accuracy," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Marrakesh, Morocco, 2008, pp. 649–653.

[66] S. K. Warfield, K. H. Zou, and W. M. Wells, "Validation of image segmentation and expert quality with an expectation-maximization algorithm," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2002*. Heidelberg, Germany: Springer, 2002, pp. 298–306.

[67] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation," *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 903–921, Jul. 2004.

[68] T. Rohlfing, D. B. Russakoff, and C. R. Maurer, Jr., "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 983–994, Aug. 2004.

[69] M. Mignotte, "A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1610–1624, Jun. 2010.

[70] M. Mignotte, "Segmentation by fusion of histogram-based k-means clusters in different color spaces," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 780–787, May 2008.

[71] H. Wang *et al.*, "Bayesian image segmentation fusion," *Knowl. Based Syst.*, vol. 71, pp. 162–168, Nov. 2014.

[72] I. Fister, Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *arXiv preprint arXiv:1307.4186*, 2013.

[73] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993.

[74] M. I. Heywood, "Evolutionary model building under streaming data for classification tasks: Opportunities and challenges," *Genet. Program. Evol. Machines*, vol. 16, no. 3, pp. 283–326, 2015.

[75] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[76] S. M. Bhandarkar and H. Zhang, "Image segmentation using evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 1–21, Apr. 1999.

[77] M. Paulinas and A. Ušinskas, "A survey of genetic algorithms applications for image enhancement and segmentation," *Inf. Technol. Control*, vol. 36, no. 3, pp. 278–284, 2007.

[78] N. E. A. Khalid, N. M. Ariff, S. Yahya, and N. M. Noor, "A review of bio-inspired algorithms as image processing techniques," in *Proc. Int. Conf. Softw. Eng. Comput. Syst.*, Kuantan, Malaysia, 2011, pp. 660–673.

[79] A. Song and V. Ciesielski, "Texture segmentation by genetic programming," *Evol. Comput.*, vol. 16, no. 4, pp. 461–481, 2008.

[80] T. Singh, N. Kharma, M. Daoud, and R. Ward, "Genetic programming based image segmentation with applications to biomedical object detection," in *Proc. 11th Annu. Conf. Genet. Evol. Comput.*, Montreal, QC, Canada, 2009, pp. 1123–1130.

[81] A. Amelio and C. Pizzuti, "An evolutionary approach for image segmentation," *Evol. Comput.*, vol. 22, no. 4, pp. 525–557, 2014.

[82] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[83] R. Poli and S. Cagnoni, "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement," in *Proc. Genet. Program.*, Stanford, CA, USA, 1997, pp. 269–277.

[84] E. C. Pedrino *et al.*, "A genetic programming based system for the automatic construction of image filters," *Integr. Comput.-Aided Eng.*, vol. 20, no. 3, pp. 275–287, 2013.

[85] D. Agnelli, A. Bollini, and L. Lombardi, "Image classification: An evolutionary approach," *Pattern Recognit. Lett.*, vol. 23, nos. 1–3, pp. 303–309, 2002.

[86] M. Zhang and W. Smart, "Multiclass object classification using genetic programming," in *Proc. Workshops Appl. Evol. Comput.*, Coimbra, Portugal, 2004, pp. 369–378.

[87] D. P. Muni, N. R. Pal, and J. Das, "A novel approach to design classifiers using genetic programming," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 183–196, Apr. 2004.

[88] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Two-tier genetic programming: Towards raw pixel-based image classification," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12291–12301, 2012.

[89] H. Al-Sahaf, M. Zhang, and M. Johnston, "A one-shot learning approach to image classification using genetic programming," in *Proc. Aust. Joint Conf. Artif. Intell.*, Dunedin, New Zealand, 2013, pp. 110–122.

[90] H. Al-Sahaf, M. Zhang, and M. Johnston, "Binary image classification: A genetic programming approach to the problem of limited training instances," *Evol. Comput.*, vol. 24, no. 1, pp. 143–182, 2016.

[91] K. Krawiec and B. Bhanu, "Visual learning by evolutionary and coevolutionary feature synthesis," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 635–650, Oct. 2007.

[92] L. Trujillo and G. Olague, "Automated design of image operators that detect interest points," *Evol. Comput.*, vol. 16, no. 4, pp. 483–507, 2008.

[93] T. Kowaliw, W. Banzhaf, N. Kharma, and S. Harding, "Evolving novel image features using genetic programming-based image transforms," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 2502–2507.

[94] T. Kowaliw, W. Banzhaf, and R. Doursat, "Networks of transform-based evolvable features for object recognition," in *Proc. 15th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Amsterdam, The Netherlands, 2013, pp. 1077–1084.

[95] H. Al-Sahaf, A. Al-Sahaf, B. Xue, M. Johnston, and M. Zhang, "Automatically evolving rotation-invariant texture image descriptors by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 83–101, Feb. 2017.

[96] M. Iqbal, B. Xue, H. Al-Sahaf, and M. Zhang, "Cross-domain reuse of extracted knowledge in genetic programming for image classification," *IEEE Trans. Evol. Comput.*, to be published, doi: 10.1109/TEVC.2017.2657556.

[97] B. Pinto and A. Song, "Learning motion detectors by genetic programming," in *Proc. Aust. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, 2009, pp. 160–169.

[98] Q. Shi and A. Song, "Selective motion detection by genetic programming," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 496–503.

[99] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 158–170, Jan. 2016.

[100] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of Gaussians for dynamic background modelling," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Kraków, Poland, 2013, pp. 63–68.

[101] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Computer Vision ECCV 2000*. Heidelberg, Germany: Springer, 2000, pp. 751–767.

[102] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proc. Int. Conf. Genet. Algorithms*, Fairfax, VA, USA, 1989, pp. 61–69.

[103] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[104] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[105] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[106] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 17–26, Feb. 2001.

[107] H. Zhao, "A multi-objective genetic programming approach to developing Pareto optimal decision trees," *Decis. Support Syst.*, vol. 43, no. 3, pp. 809–826, 2007.

[108] P. Darwen and X. Yao, "Every niching method has its niche: Fitness sharing and implicit sharing compared," in *Proc. Int. Conf. Parallel Problem Solving Nature*, Berlin, Germany, 1996, pp. 398–407.

[109] B. Sareni and L. Krahenbuhl, "Fitness sharing and niching methods revisited," *IEEE Trans. Evol. Comput.*, vol. 2, no. 3, pp. 97–106, Sep. 1998.

[110] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evol. Comput.*, vol. 19, no. 2, pp. 189–223, 2011.

[111] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2015, pp. 990–997.

[112] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Turin, Italy, 2015, pp. 1–6.

[113] G. Allebosch, D. Van Hamme, F. Deboeverie, P. Veelaert, and W. Philips, "C-EFIC: Color and edge based foreground background segmentation with interior classification," in *Proc. Int. Joint Conf. Comput. Vis. Imag. Comput. Graphics*, Berlin, Germany, 2015, pp. 433–454.
[114] H. Sajid and S.-C. S. Cheung, "Background subtraction for static & moving camera," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, 2015, pp. 4530–4534.
[115] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips, "EFIC: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints," in *Advanced Concepts for Intelligent Vision Systems* (LNCS 9386). Cham, Switzerland: Springer, 2015, pp. 130–141.
[116] G. Ramírez-Alonso and M. I. Chacón-Murguía, "Auto-adaptive parallel SOM architecture with a modular analysis for dynamic object segmentation in videos," *Neurocomputing*, vol. 175, pp. 990–1000, Jan. 2016.
[117] A. Miron and A. Badii, "Change detection based on graph cuts," in *Proc. IWSSIP*, London, U.K., 2015, pp. 273–276.
[118] L. Maddalena and A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 179–186, 2010.
[119] D. Liang and S. Kaneko, "Improvements and experiments of a compact statistical background model," *arXiv preprint arXiv:1405.6275*, 2014.
[120] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. Cambridge, U.K., Aug. 2004, pp. 28–31.
[121] X. Lu, "A multiscale spatio-temporal background model for motion detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, 2014, pp. 3268–3271.

Authors' photographs and biographies not available at the time of publication.