24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Foreground detection by probabilistic mixture models using semantic information from deep networks

Jorge García-González and **Juan M. Ortiz-de-Lazcano-Lobato** and **Rafael M. Luque-Baena** and **Ezequiel López-Rubio**[1]

Abstract. The advent of deep learning networks has led to an improvement in almost every area in the computer vision field. In this work, a foreground detection method is proposed which intends to improve algorithms within video surveillance systems. Specifically, the proposed approach consists of a principled probabilistic model that combines both the output information of a semantic segmentation convolutional neural model and the color value for each pixel. The relevant features are transformed in a nonlinear way so as to enhance the performance of the probabilistic model. In order to determine the method feasibility, a set of experiments based on video sequences that belong to several public repositories have been carried out. The results show that the foreground detection performance of the proposal is higher than that of traditional algorithms in many situations.

1 Introduction

Background modeling is an essential task in video sequence analysis, because it allows moving objects in the scene to be detected, differentiated from stationary background objects and extracted for further processing. It has been studied intensively in the last decades ([3], [4]) due to the difficulty in obtaining a robust and adaptive model which is able to cope with most of the problems that arise in video sequences acquired by real-world cameras such as illumination changes, dynamic backgrounds, camera jitter and camouflage among others.

Many of the presented foreground segmentation methods rely on low-level image features which are relatively fast to compute but also very sensitive to the aforementioned acquisition problems. Normally, methods consider only color components as seen in [23], [32], [35] and [20]. However, some methods attempt to take advantage of edge features ([27], [10]), texture descriptors ([9],[2],[15]), or even to make use of the optical flow in order to achieve a temporally consistent background model ([6]).

Semantic segmentation is a computer vision problem which takes an image and assigns each of its pixels a label corresponding to the type of object which the pixel belongs to. For that purpose, high level features for the objects of interest are computed and used to determine not only whether a pixel belongs to the foreground but also which one of the previously known classes of moving objects it belongs to. Even though it is an extremely difficult task, which involves simultaneous detection, localization, and segmentation of semantic objects and regions, it can be carried out successfully by means of deep neural network models, as shown in [12], [29] and [1].

In spite of having been used for applications so diverse as autonomous driving ([7], [26]), land cover classification for each pixel on a remote sensing image [17] or facial segmentation [28], adaptation and performance of deep neural networks used for semantic segmentation may be improved. A framework in which a traditional and a semantic background modeling methods are combined is presented in [5]. Both methods in conjunction achieve an overall performance which is usually higher than those of the component models when they work in isolation. However, the rules that determine how to combine the component model results are rather ad hoc.

In this paper, a principled foreground segmentation methodology based on a probabilistic model which takes into consideration color as well as semantic information is proposed. The probabilistic formulation integrates both kinds of information naturally by means of a nonlinear transformation of the relevant features, and enables the application of Bayesian inference. The remaining of the paper is divided into three sections. Section 2 presents the probabilistic background modeling method. The experimental tests that have been carried out are described in Section 3 and their results are reported and commented as well. Finally, Section 4 is devoted to conclusions.

2 Methodology

Typical background subtraction models are based on low level visual features such as the RGB values. With the advent of deep learning neural networks which can identify objects and their classes on a video frame, this can be enhanced. Let C be the number of classes. We assume that we have a deep learning neural network which, given an incoming video frame, provides several outputs:

- For each object class, it outputs a binary image with ones for all the pixels which are estimated to belong to an object of that class.
- For each detected object in the scene, the network outputs a region of interest (ROI) which encloses the object (its silhouette), its class and a score, i.e. the confidence that there is an object of that class is really there. The score is a number between 0 and 1.

In what follows, our proposed methodology is detailed. Subsection 2.1 defines the probabilistic model that we propose in order to estimate the probability distribution of the output of the deep learning neural network. Aside from the learning of the probabilistic model, an additional processing is done in order to alleviate the deleterious effects of failures of the deep network (Subsection 2.2).

[1] Department of Computer Languages and Computer Science, University of Málaga, Spain; Biomedic Research Institute of Málaga (IBIMA); email: {jorgegarcia,jmortiz,rmluque,ezeqlr}@lcc.uma.es

2.1 Probabilistic model

The bounding boxes can overlap, for objects of the same or different classes. Therefore, we can use the output of the network in order to obtain a vector of scores $\mathbf{q}_i \in [0,1]^C$ for each pixel i, where each vector component is the score associated to one of the object classes. The observed data for a pixel is the concatenation of $\mathbf{q}_i$ with the observed RGB values $\mathbf{r}_i \in [0,1]^3$:

$$\mathbf{x}_i = (\mathbf{r}_i, \mathbf{q}_i) \in [0,1]^D \tag{1}$$

where $D = C + 3$.

As seen in (1), the components of $\mathbf{x}_i$ are bounded. This implies that they cannot be adequately modeled by a Gaussian distribution, since the Gaussian distribution has an infinite support. This is unfortunate, since the parameters of a Gaussian distribution are relatively easy to learn. In order to overcome this difficulty, we propose to apply a nonlinear transformation φ to the components of $\mathbf{x}_i$, so that the possible values of the transformed feature vector $\mathbf{y}_i$ matches the support of the Gaussian distribution:

$$y_{ij} = \varphi(x_{ij}) \tag{2}$$

$$\mathbf{y}_i = \boldsymbol{\varphi}(\mathbf{x}_i) \tag{3}$$

where

$$\varphi : [0,1] \to \mathbb{R} \tag{4}$$

such as $\varphi(x) = \arctan\left(\pi x - \frac{\pi}{2}\right)$, $\varphi(x) = \operatorname{atanh}(2x-1)$.

Now the probability density of observing $\mathbf{y}_i$ is given by:

$$p(\mathbf{y}_i) = P(Back)\, p(\mathbf{y}_i|Back) + P(Fore)\, p(\mathbf{y}_i|Fore) \tag{5}$$

The foreground probability density is given by:

$$p(\mathbf{x}_i|Fore) = p(\mathbf{r}_i|Fore)\, p(\mathbf{q}_i|Fore) \tag{6}$$

where the foreground distributions are assumed to be uniform, since incoming objects may be of any color or class:

$$p(\mathbf{r}_i|Fore) = 1 \tag{7}$$

$$p(\mathbf{q}_i|Fore) = 1 \tag{8}$$

This means that the transformed probability density function is given by:

$$p(\mathbf{y}_i|Fore) = \prod_{j=1}^{D} p(y_{ij}|Fore) \tag{9}$$

$$p(y_{ij}|Fore) = p(x_{ij}|Fore)\left|\left(\varphi^{-1}\right)'(y_{ij})\right| \tag{10}$$

where $\left(\varphi^{-1}\right)'$ stands for the derivative of the inverse function of φ, $|\cdot|$ stands for the absolute value of a real number, and $p(x_{ij}|Fore) = 1$ from (6).

Please note that for the arc tangent option we have:

$$\varphi_{arctan}(x_{ij}) = \arctan\left(\pi x_{ij} - \frac{\pi}{2}\right) \tag{11}$$

$$\varphi^{-1}_{arctan}(y_{ij}) = \frac{1}{\pi}\left(\frac{\pi}{2} + \tan y_{ij}\right) \tag{12}$$

$$\left(\varphi^{-1}_{arctan}\right)'(y_{ij}) = \frac{1}{\pi}(\sec y_{ij})^2 \tag{13}$$

Alternatively, for the hyperbolic arc tangent option we have:

$$\varphi_{atanh}(x_{ij}) = \operatorname{atanh}(2x_{ij} - 1) \tag{14}$$

$$\varphi^{-1}_{atanh}(y_{ij}) = \frac{1}{2}(1 + \tanh y_{ij}) \tag{15}$$

$$\left(\varphi^{-1}_{atanh}\right)'(y_{ij}) = \frac{1}{2}(\operatorname{sech} y_{ij})^2 \tag{16}$$

For the background model we can use a Gaussian to model each component of the transformed vector $\mathbf{y}_i$:

$$p(\mathbf{y}_i|Back) = \prod_{j=1}^{D} p(y_{ij}|Back) \tag{17}$$

$$p(y_{ij}|Back) = \frac{1}{\sqrt{2\pi\sigma^2_{ij}}} \exp\left(-\frac{(y_{ij} - \mu_{ij})^2}{2\sigma^2_{ij}}\right) \tag{18}$$

There is a background model for each pixel i. Two vectors $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i \in \mathbb{R}^D$ comprise the background model of each pixel, which store the mean and the variance of the components of $\mathbf{x}_i$ whenever the pixel i belongs to the background, respectively:

$$\mu_{ij} = E[y_{ij}|Back] \tag{19}$$

$$\sigma^2_{ij} = E\left[(y_{ij} - \mu_{ij})^2 | Back\right] \tag{20}$$

We will assume that the a priori probabilities of the background and the foreground are equal:

$$P(Fore) = P(Back) = \frac{1}{2} \tag{21}$$

Therefore, the application of Bayes theorem to (5), (6) and (17) leads to:

$$P(Back|\mathbf{y}_i) = \frac{p(\mathbf{y}_i|Back)}{p(\mathbf{y}_i|Back) + p(\mathbf{y}_i|Fore)} \tag{22}$$

$$P(Fore|\mathbf{y}_i) = \frac{p(\mathbf{y}_i|Fore)}{p(\mathbf{y}_i|Back) + p(\mathbf{y}_i|Fore)} \tag{23}$$

It must be noted that, in order to avoid numerical loss of precision, (17) is best implemented in practice as follows:

$$p(\mathbf{y}_i|Back) = \exp\left(\sum_{j=1}^{D} \log p(y_{ij}|Back)\right) \tag{24}$$

where the componentwise log densities can be derived from (18):

$$\log p(y_{ij}|Back) = -\frac{1}{2}\log 2\pi - \log\sigma_{ij} - \frac{(y_{ij} - \mu_{ij})^2}{2\sigma^2_{ij}} \tag{25}$$

Next the stochastic approximation method [25] is employed to estimate the background model:

$$\mu_{ij}(0) = y_{ij}(0) \tag{26}$$

$$\sigma^2_{ij}(0) = \epsilon \tag{27}$$

$$\mu_{ij}(t+1) = \eta P(Back|\mathbf{y}_i(t))\, y_{ij}(t) + (1 - \eta P(Back|\mathbf{y}_i(t)))\, \mu_{ij}(t) \quad (28)$$

$$\sigma^2_{ij}(t+1) = \eta P(Back|\mathbf{y}_i(t))\, (y_{ij}(t) - \mu_{ij}(t))^2 + (1 - \eta P(Back|\mathbf{y}_i(t)))\, \sigma^2_{ij}(t) \quad (29)$$

where $\epsilon > 0$ is a small constant, $\eta > 0$ is a suitable step size and t is the time step, i.e. the video frame index.

2.2 Time window analysis

The semantic segmentation method may commit some errors sometimes. For instance, it could detect an object where there is not anyone actually, or some foreground objects may be assigned momentarily to an incorrect class or even to no class in the case of foreground object camouflage. In order to counter these eventual errors, we have used a time window analysis criterion to ensure that the segmentation of pixel i at time t is consistent with the segmentation of that pixel during the time interval from $t-k$ to $t+k$, in relation to a value l that, given a foreground probability, is regarded as the threshold between foreground and background. This value will be typically $l = 0.5$.

Let s_{it} be the foreground probability of pixel i at time instant t. Then our time window analysis consists in replacing s_{it} by:

$$s'_{it} = \begin{cases} s_{it} & \text{if } s_{it} \geq l \ \wedge\ \overline{s_i} \geq l \\ s_{it} & \text{if } s_{it} < l \ \wedge\ \overline{s_i} < l \\ \overline{s_i} & \text{otherwise} \end{cases} \quad (30)$$

with

$$\overline{s_i} = \frac{1}{2k+1} \sum_{h \in \{t-k,\ldots,t+k\}} s_{ih} \quad (31)$$

3 Experimental Results

3.1 Method Configuration

In order to work properly, the proposed methodology needs a model that provides a suitable image semantic segmentation. Among the different proposals found in the literature [33, 16], the Mask-RCNN deep neural model [1][2], which had been trained with Microsoft COCO dataset [18], is the most appropriate since it provides a segmentation mask of each detected object. This model is an evolution of previous models of object detection such as Fast R-CNN [13] or Faster R-CNN [24], which provide as output the bounding box of the detected object together with its class. It is important to point out that we have not trained the network nor applied a fine tuning so that it can adapt better to the experimental data sets, as our intention was to develop a framework in which any model able to provide semantic information, i.e. class membership likelihood and region of interest (ROI) for each semantically segmented object in the frame, could be used without any change. Only semantic information with 0.5 or greater likelihood has been used in order to avoid dubious semantic segmentation data.

Regarding the probabilistic model parameters, they were set to the same values for all the experiments: $\eta = 0.005$, $\epsilon = 0.1$ and time window size $k = 1$. $\sigma^2 = \epsilon$ was chosen as the variance minimum value in order to avoid numeric errors while updating the background model.

As nonlinear transformation φ we have used equation (11), therefore $(\varphi^{-1})'$ is defined by equation (13).

Once the segmentation images are generated, we have applied to them a closure morphological transformation with window size $(9, 9)$ pixels.

3.2 Sequences

Videos in four categories from the ChangeDetection.net (CDnet) 2014 data set [14], which can be downloaded from its website [3], were chosen as a basis for experimental test and comparison between the proposed method and other state-of-the-art ones. The selected categories are: *baseline* (*B*) with four videos showing common situations with mild challenges typical of other categories, *dynamicBackground* (*D*) with six videos exhibiting environments where background presents natural noise such as moving water and foliage, *shadow* (*S*) with six videos exhibiting many shadows of different nature (strong and faint) caused by objects of various kinds (moving ones, trees, buildings, etc) and *turbulence* (*T*) with four videos presenting air turbulence caused by rising heat.

3.3 Methods

In order to make the comparison, other unsupervised segmentation methods from bibliography have been considered. Some of them have been obtained from BGS library ([31])[4]: *GA - Wren (Wren)* [34], *GMM - Zivkovic (Zivkovic)* [35], *SOBS* [21], *SOBS_CF* [22], *KDE* ([11]) . Other methods can be found on their authors' websites: *FSOM* ([19]) and *CL-VID* ([20]).

SemanticBGS [5] has been also used to compare. It consists of a framework that allows the user to combine a traditional foreground segmentation method and a semantic segmentation one which is expected to provide a semantic mask with the probability of an object to be part of each one of the foreground classes. Therefore, the results of both segmentation methods are taken into account during the segmentation process, which finally yields a more accurate segmented image. This framework piece of code has been obtained from its developers' web [5].

With the aim of being as fair as possible in the experimental tests, the semantic information that SemanticBGS is provided is the same as the one supplied to the proposed method, that is to say, both segmentation methods receive the same class assignment expressed as an array of likelihood values for each one of the frame pixels. The probabilistic information was generated by means of the deep neural network Mask-RCNN, which was initially set to yield semantic information of only those classes with a likelihood greater than 0.5. The probability values for the remaining classes, which very often correspond to a false detection and, thus, should be ignored, are set to 0 by Mask-RCNN. In the case of SemanticBGS, the semantic information had to be translated to an only grayscale mask and was done as indicated in [5] with *R* = {*person, bicycle, car, motorcycle, bus, truck, boat, handbag, suitcase, bottle, couch, book*} as classes of foreground objects that were liable to be detected in the video sequences that are part of CDNet 2014.

[2] https://github.com/matterport/Mask_RCNN

[3] http://changedetection.net/

[4] https://github.com/andrewssobral/bgslibrary

[5] http://www.telecom.ulg.ac.be/semantic/

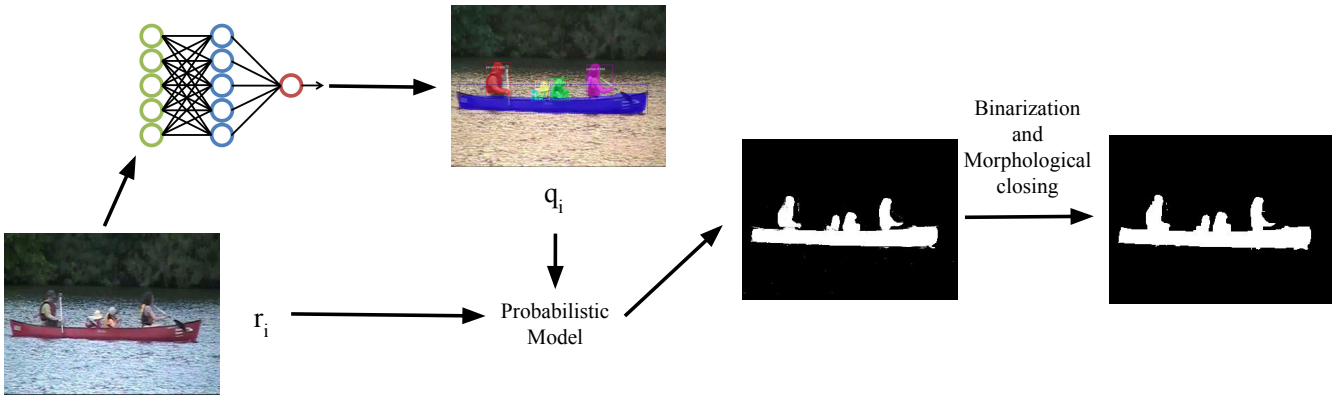

Figure 1: Methodology. Each pixel of frame t is represented by a color vector r_i and semantic segmentation vector q_i. The probabilistic model integrates q_i and r_i, and a grayscale image is obtained. Finally, binarization and morphological closing operations are performed on that image.

In addition to semantic information, SemanticBGS was also provided with the foreground segmentation frames created by the recent CL-VID foreground segmentation method.

3.4 Evaluation

The evaluation measure to compare each method performance from a quantitative point of view was the *F-score* (also denoted as F_1 or *F-measure*). It was first introduced to evaluate information extraction methods in [8] and since then it has been widely used in segmentation and tracking tasks [30]. It has been calculated for each frame within temporal region of interest indicated by changedetection.net. The average value for all frames in a video is the evaluation assigned to that video. In addition, the average value for all videos in a category is the evaluation assigned to that category.

3.5 Experimental Results

Some qualitative examples from our experimental results for all methods are shown in figures 2 and 3 on pages 5 and 6 respectively.

According to the quantitative results for each category, which can be seen in table 1 on page 7, the proposed method outperforms the other competing ones in half of the categories and is runner-up in the remaining categories. In table 2 on page 7, the *F-measure* values that each method obtains for each analyzed video sequence are presented. It can be noted that the proposed method achieves the best performance score on average.

It can be noticed that the presented method is able to deal with intrinsic noise of low intensity since the semantic information which is provided seems to compensate for pixel level colour noise. It is also noticeable the average good result of our proposal, as compared to other methods, with respect to the *turbulence* category. This category contains grayscale sequences and our method is able to discriminate even with little color difference while semantic segmentation network is able to identify objects.

Our method also tends to filter shadows since they are generally not recognized by the semantic segmentation method so they are not likely to be segmented as foreground even if there is a color change.

Even if our method is able to deal with some segmentation errors, the existence of foreground objects that are not recognized by the semantic segmentation method for long periods of time could be a problem because even if their color is different from the background, the absence of semantic assignation could turn the probability model to output background. As we can observe on canoe images on figure 2 and 3 on pages 5 and 6, our method does not mark the row as foreground since the neural network has problems to assign it a class and its color is not significantly different from background. Because of this, the improvement of the underlying semantic segmentation method should lead to a direct improvement on our method.

4 Conclusions

Semantic segmentation is vital to truly understand what happens in a video sequence. Thanks to convolutional neural networks, now we are able to know not only local information as the color of a pixel but also to which object it belongs. Foreground segmentation is a problem with a strong component of human interpretation and we understand a scene not as pixels with changing colors but as a set of object in a 3D space even if we are looking into a 2D image from a camera. If we are able to provide our systems the information about the existence of those objects to compensate for its spatial-temporal reasoning limitations, we should do it in order to get a foreground segmentation more likely to fit our interpretation.

This paper presents a foreground detection method that combines basic pixel information as color with semantic segmentation information into a probabilistic model. Tests on four different categories with a total of twenty video sequences have been carried out. The proposed method has been compared with other seven foreground segmentation algorithms based on color information and another method that combines semantic segmentation information and the segmentation provided by another foreground detection method. Tables according to *F-measure* value for each video and category have been provided and the proposed method shows a high performance on average in addition to various advantages as robustness to low noise and shadow resistance. Therefore, it can be accepted that our proposed approach has been experimentally validated.

ACKNOWLEDGEMENTS

This work is partially supported by the Ministry of Science, Innovation and Universities of Spain [grant number RTI2018-094645-B-I00], project name Automated detection with low cost hardware of unusual activities in video sequences. It is also partially supported by the Autonomous Government of Andalusia (Spain) [grant number P12-TIC-657], project name Self-organizing systems and robust estimators for video surveillance. Both of them include funds from the

Figure 2: First row shows original frames: 940 from pedestrians from baseline category, 958 from canoe sequence from dynamicBackground category, 7009 from cubicle sequence from shadows category and 1797 from turbulence0 from turbulence category. Second row shows their ground-truth provided by changedetection.net. Followings rows show their segmentation created by our method, SemanticBGS, CL-VID, FSOM and SOBS.

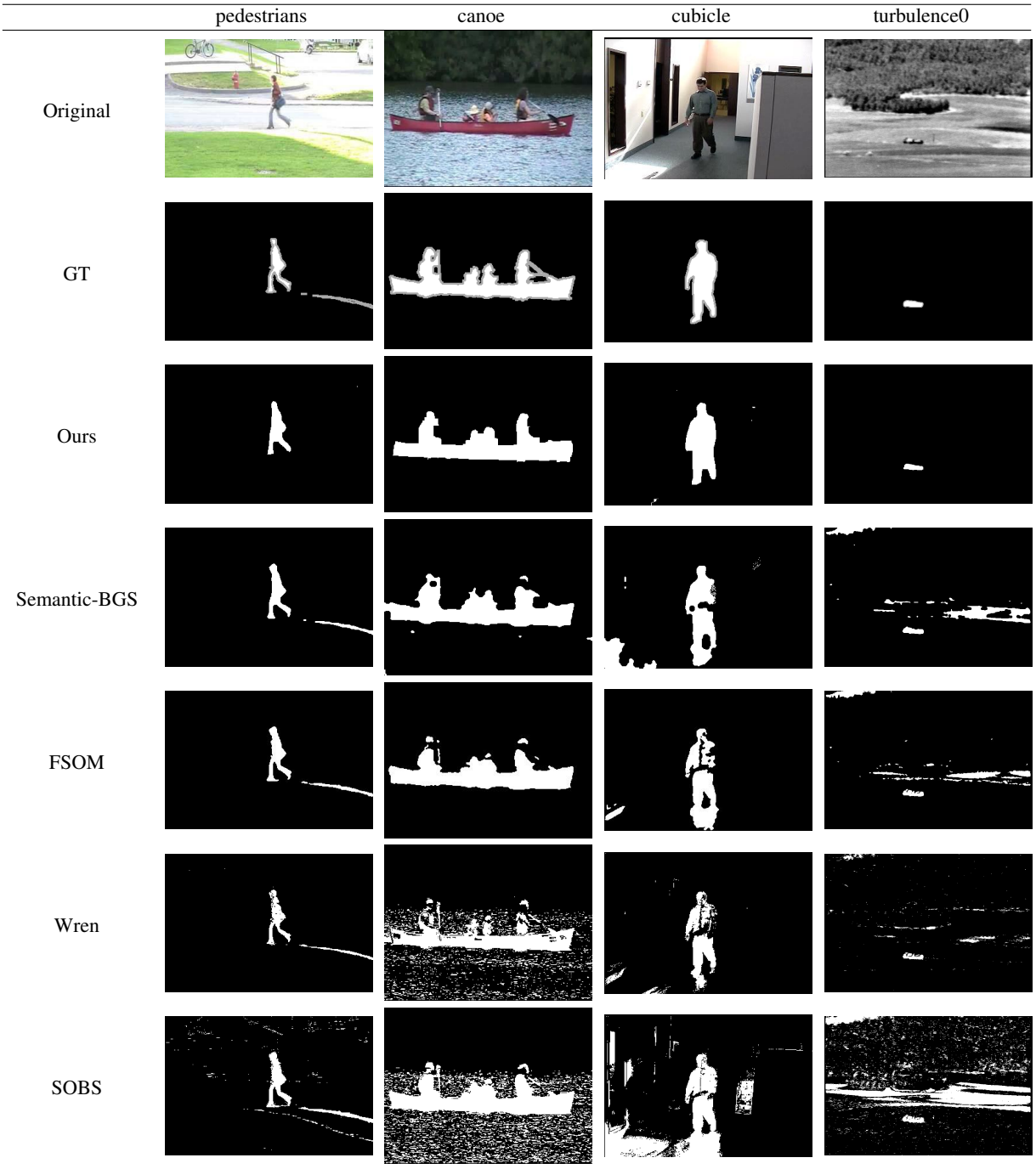

Figure 3: Continues figure 2 on page 5. First row shows original frames: 940 from pedestrians from baseline category, 958 from canoe sequence from dynamicBackground category, 7009 from cubicle sequence from shadows category and 1797 from turbulence0 from turbulence category. Second row shows their ground-truth provided by changedetection.net. Followings rows show their segmentation created by our method, SOBS_CF, Wren, KDE and Zivkovic.

Table 1: Average *F-measure* (the higher, the better) and standard deviation between parentheses for each category and method. Two best values for each category are printed in black (first) and gray (second).

	baseline	dynamicBackground	shadow	turbulence
Ours	0.841(0.080)	0.249(0.173)	**0.563(0.230)**	**0.202(0.096)**
SemanticBGS	0.869(0.094)	0.214(0.191)	0.533(0.226)	0.142(0.102)
CL-VID	**0.886(0.087)**	0.216(0.191)	0.537(0.229)	0.161(0.109)
FSOM	0.827(0.091)	**0.253(0.178)**	0.499(0.195)	0.164(0.099)
KDE	0.596(0.071)	0.111(0.064)	0.397(0.176)	0.171(0.073)
Wren	0.668(0.102)	0.130(0.097)	0.401(0.127)	0.137(0.079)
SOBS	0.727(0.121)	0.103(0.085)	0.455(0.221)	0.067(0.073)
SOBS_CF	0.589(0.214)	0.103(0.100)	0.401(0.230)	0.063(0.068)
Zivkovic	0.725(0.112)	0.168(0.130)	0.440(0.154)	0.172(0.101)

Table 2: Average *F-measure* (the higher, the better) for each video and method. Letter between parentheses indicates each video category with (*B*) as *baseline*, (*D*) as *dynamicBackground*, (*S*) as *shadow* and (*T*) as *turbulence*. Last row shows average value for all videos for each method. Bold value is the average best value, gray value is the second one.

	Ours	SemanticBGS	CL-VID	FSOM	KDE	Wren	SOBS	SOBS_CF	Zivkovic
highway (*B*)	0.828749	0.940988	0.943907	0.921312	0.622140	0.762715	0.767758	0.698266	0.802163
office (*B*)	0.931589	0.928479	0.932838	0.749874	0.618598	0.530915	0.726116	0.412643	0.567023
pedestrians (*B*)	0.739974	0.734835	0.756860	0.747140	0.649859	0.726393	0.562909	0.407119	0.726203
PETS2006 (*B*)	0.864925	0.873412	0.908863	0.887757	0.491696	0.653631	0.850752	0.837594	0.805085
boats (*D*)	0.150797	0.151241	0.152595	0.138354	0.073748	0.081331	0.061521	0.067156	0.102002
canoe (*D*)	0.578069	0.573720	0.575833	0.588397	0.238354	0.309995	0.260740	0.296403	0.412855
fall (*D*)	0.211489	0.105460	0.105934	0.214744	0.097288	0.085562	0.075799	0.066850	0.119922
fountain01 (*D*)	0.107105	0.040496	0.040685	0.102710	0.066212	0.029518	0.024361	0.022542	0.039552
fountain02 (*D*)	0.157106	0.146129	0.151980	0.176627	0.086425	0.135330	0.061795	0.047570	0.146592
overpass (*D*)	0.291425	0.266735	0.266396	0.298223	0.103959	0.137408	0.136331	0.120290	0.189534
backdoor (*S*)	0.323040	0.330634	0.335016	0.322733	0.264062	0.292699	0.279401	0.253846	0.285554
bungalows (*S*)	0.355006	0.348728	0.348604	0.350031	0.296853	0.294753	0.313930	0.277698	0.323200
busStation (*S*)	0.821878	0.747286	0.747708	0.760233	0.541208	0.527617	0.588551	0.428011	0.605519
copyMachine (*S*)	0.854165	0.849871	0.859561	0.670139	0.633333	0.531186	0.784228	0.770267	0.586198
cubicle (*S*)	0.451595	0.352536	0.347116	0.315518	0.182222	0.269942	0.213037	0.136173	0.293703
peopleInShade (*S*)	0.570993	0.569824	0.584975	0.576425	0.463492	0.490060	0.553205	0.540308	0.542942
turbulence0 (*T*)	0.253907	0.027538	0.028045	0.034375	0.142572	0.051208	0.009184	0.007248	0.049175
turbulence1 (*T*)	0.201001	0.135060	0.204579	0.221015	0.146865	0.151725	0.063723	0.066017	0.232693
turbulence2 (*T*)	0.068367	0.129663	0.129665	0.142016	0.115056	0.108755	0.025126	0.020353	0.131598
turbulence3 (*T*)	0.285614	0.276373	0.283517	0.256747	0.278925	0.237902	0.171241	0.157080	0.272668
Average	**0.452340**	0.426450	0.435234	0.423719	0.305644	0.320432	0.326485	0.281672	0.361709

European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They have also been supported by the Biomedic Research Institute of Málaga (IBIMA). They also gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan X GPUs.

REFERENCES

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[2] N. Armanfard, M. Komeili, and E. Kabir, 'Ted: A texture-edge descriptor for pedestrian detection in video sequences', *Pattern Recognition*, **45**(3), 983–992, (2012).

[3] Thierry Bouwmans, 'Traditional and recent approaches in background modeling for foreground detection: An overview', *Computer Science Review*, **11-12**, 31 – 66, (2014).

[4] Thierry Bouwmans, Sajid Javed, Maryam Sultana, and Soon Ki Jung, 'Deep neural network concepts for background subtraction: A systematic review and comparative evaluation', *Neural Networks*, **117**, 8 – 66, (2019).

[5] M. Braham, S. Piérard, and M. Van Droogenbroeck, 'Semantic background subtraction', in *IEEE International Conference on Image Processing (ICIP)*, pp. 4552–4556, Beijing, China, (September 2017).

[6] Mingliang Chen, Qingxiong Yang, Qing Li, Gang Wang, and Ming-Hsuan Yang, 'Spatiotemporal background subtraction using minimum spanning tree and optical flow', in *Computer Vision – ECCV 2014*, eds.,

David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, pp. 521–534, Cham, (2014). Springer International Publishing.

[7] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, 'Monocular 3d object detection for autonomous driving', volume 2016-December, pp. 2147–2156, (2016).

[8] Nancy Chinchor, 'Muc-4 evaluation metrics', in *Proceedings of the 4th Conference on Message Understanding*, MUC4 '92, pp. 22–29, Stroudsburg, PA, USA, (1992). Association for Computational Linguistics.

[9] P. Chiranjeevi and S. Sengupta, 'Neighborhood supported model level fuzzy aggregation for moving object segmentation', *IEEE Transactions on Image Processing*, **23**(2), 645–657, (Feb 2014).

[10] I. Dahi, M. Chikr El Mezouar, N. Taleb, and M. Elbahri, 'An edge-based method for effective abandoned luggage detection in complex surveillance videos', *Computer Vision and Image Understanding*, **158**, 141–151, (2017).

[11] Ahmed Elgammal, David Harwood, and Larry Davis, 'Non-parametric model for background subtraction', in *Computer Vision (ECCV)*, pp. 751–767. Springer, (2000).

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, 'Region-based convolutional networks for accurate object detection and segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(1), 142–158, (2016).

[13] Ross Girshick, 'Fast R-CNN', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, (2015).

[14] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Prakash Ishwar, et al., 'Changedetection. net: A new change detection benchmark dataset.', in *CVPR Workshops*, number 2012, pp. 1–8, (2012).

[15] M. Heikkilä and M. Pietikäinen, 'A texture-based method for modeling the background and detecting moving objects', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(4), 657–662, (2006).

[16] R. Kalsotra and S. Arora, 'A comprehensive survey of video datasets for background subtraction', *IEEE Access*, **7**, 59143–59171, (2019).

[17] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, 'Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks', pp. 680–688, (2016).

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, 'Microsoft coco: Common objects in context', in *Computer Vision – ECCV 2014*, eds., David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, pp. 740–755, Cham, (2014). Springer International Publishing.

[19] E. López-Rubio, R.M. Luque-Baena, and E. Domínguez, 'Foreground detection in video sequences with probabilistic self-organizing maps', *International Journal of Neural Systems*, **21**(3), 225–246, (2011).

[20] Ezequiel López-Rubio, Miguel A. Molina-Cabello, Rafael Marcos Luque-Baena, and Enrique Domínguez, 'Foreground detection by competitive learning for varying input distributions', *International Journal of Neural Systems*, **28**(05), 1750056, (2018).

[21] L. Maddalena and A. Petrosino, 'A self-organizing approach to background subtraction for visual surveillance applications', *Trans. Img. Proc.*, **17**(7), 1168–1177, (July 2008).

[22] Lucia Maddalena and Alfredo Petrosino, 'A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection', *Neural Computing and Applications*, **19**(2), 179–186, (2010).

[23] Lucia Maddalena and Alfredo Petrosino, 'The SOBS algorithm: What are the limits?', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 21–26, (2012).

[24] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster r-cnn: Towards real-time object detection with region proposal networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(6), 1137–1149, (2017).

[25] H. Robbins and S. Monro, 'A stochastic approximation method', *The Annals of Mathematical Statistics*, **22**(3), 400–407, (1951).

[26] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A.M. Lopez, 'The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes', volume 2016-December, pp. 3234–3243, (2016).

[27] K. Roy, J. Kim, M. T. B. Iqbal, F. Makhmudkhujaev, B. Ryu, and O. Chae, 'An adaptive fusion scheme of color and edge features for background subtraction', in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, (Aug 2017).

[28] S. Saito, T. Li, and H. Li, 'Real-time facial segmentation and performance capture from rgb input', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **9912 LNCS**, 244–261, (2016).

[29] E. Shelhamer, J. Long, and T. Darrell, 'Fully convolutional networks for semantic segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(4), 640–651, (2017).

[30] A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, 'Visual tracking: An experimental survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(7), 1442–1468, (2014).

[31] Andrews Sobral and Thierry Bouwmans, 'Bgs library: A library framework for algorithm's evaluation in foreground/background segmentation', in *Background Modeling and Foreground Detection for Video Surveillance*, CRC Press, Taylor and Francis, (2014).

[32] C. Stauffer and W. E. L. Grimson, 'Adaptive background mixture models for real-time tracking', in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pp. 246–252 Vol. 2, (June 1999).

[33] X. Wang, 'Deep learning in object recognition, detection, and segmentation', *Foundations and Trends in Signal Processing*, **8**(4), 217–382, (2016).

[34] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland, 'Pfinder: Real-time tracking of the human body', *IEEE Transactions on pattern analysis and machine intelligence*, **19**(7), 780–785, (1997).

[35] Zoran Zivkovic, 'Improved adaptive gaussian mixture model for background subtraction', in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pp. 28–31. IEEE, (2004).