Scene Change Detection Using Multiscale Cascade Residual Convolutional Neural Networks

Daniel F. S. Santos†, Rafael G. Pires†
Department of Computing
São Paulo State University
Bauru, Brazil
{danielfssantos1, rafapires}@gmail.com

Danilo Colombo
Cenpes
Petroleo Brasileiro S.A. - Petrobras
Rio de Janeiro - RJ, Brazil
colombo.danilo@petrobras.com.br

João P. Papa
Department of Computing
São Paulo State University
Bauru, Brazil
joao.papa@unesp.br

***Abstract*—Scene change detection is an image processing problem related to partitioning pixels of a digital image into foreground and background regions. Mostly, visual knowledge-based computer intelligent systems, like traffic monitoring, video surveillance, and anomaly detection, need to use change detection techniques. Amongst the most prominent detection methods, there are the learning-based ones, which besides sharing similar training and testing protocols, differ from each other in terms of their architecture design strategies. Such architecture design directly impacts on the quality of the detection results, and also in the device resources capacity, like memory. In this work, we propose a novel Multiscale Cascade Residual Convolutional Neural Network that integrates multiscale processing strategy through a Residual Processing Module, with a Segmentation Convolutional Neural Network. Experiments conducted on two different datasets support the effectiveness of the proposed approach, achieving average overall $F\text{-}measure$ results of 0.9622 and 0.9664 over Change Detection 2014 and PetrobrasROUTES datasets respectively, besides comprising approximately eight times fewer parameters. Such obtained results place the proposed technique amongst the top four state-of-the-art scene change detection methods.**

I. Introduction

Scene change detection is a specific kind of image processing task, that involves partitioning the digitalized captured scene into foreground and background pixel regions. Such a processing strategy is frequently used in many visual knowledge-based computer intelligent systems, such as traffic monitoring [1], autonomous driving [2], object and people tracking [3], action recognition [4], video surveillance [5], and anomaly detection [6]. Each of those systems presents its challenges for the change detection itself, such as: (a) the shooting environment condition, (b) video capture device quality, and also (c) local computer memory storage capacity.

Concerning some difficulties presented by (a), it can be named a few ones such as shadows, low-light, specular reflections, and blizzard. Regarding (b), it can be noticed problems with the device sensors, mostly due to subtle temperature variations and also issues related to digital noise, mainly generated during analogic to digital signal conversion. Regarding (c), the change detection technique must be adaptable to work in mobile-reduced memory devices such as smartphones, tablets, and drones.

†These authors contributed equally to this paper.

In the last few decades, in an attempt to solve problems (a), (b), and (c), many scene change detection techniques have been developed. They can be classified into two big groups, i.e., the non-learning-based and the learning-based ones. Amongst the non-learning-based group, one can refer to the works of KaewTraKulPong and Bowden [7], Zivkovic [8], and Varadarajan et al. [9], with a strong basis on statistical parametric modeling of the scene changes. Considering the same statistical domain, one can also encounter the works of Bevilaqua et al. [10], and Lanza and Di Stafano [11], that use nonparametric statistics for the scene change modeling. Besides such mentioned techniques, it is possible to find more simple and effective methods, which include SuBSENSE from St-Charles et al. [12], PWCS from St-Charles et al. [13], and IUTIS-5 from Bianco et al. [14].

The second group of change detection techniques includes those methods capable of learning how to differentiate between the foreground and background scene regions, that when properly designed and trained, can easily adapt to difficult change detection scenarios, as demonstrated by the works of Wang et al. [15], that use a multistage and multiscale network named Cascade, Babaee et al. [16], concerning the usage of a multistage convolutional neural network named DeepBS, Santos et al. [17], which use a multistage residual convolutional neural network named CRCNN, Santana et al. [18], which use siamese-based change detection networks named SEU-Nets, and Lim and Keles [19]–[20], that work with autoencoder change detection convolutional neural networks FgSegNet_M, FgSegNet_S, and FgSegNet_v2.

Although the learning-based methods present state-of-the-art results in the literature when compared against the non-learning-based techniques, they are not yet capable at solving, at the same time, the problems (a), (b), and (c). The CascadeCNN, DeepBS, and CRCNN techniques can be low memory consumptive methods, but at the same time, do not achieve FgSegNets results. On the other hand, in the case of the FgSegNets, better detection results implicate high memory consumption.

In this work, we attempt to improve the effectiveness of the CRCNN method in dealing with problems (a) and (b), trying to maintain the technique already good compromise with the problem (c). In that sense, we propose four modifications to

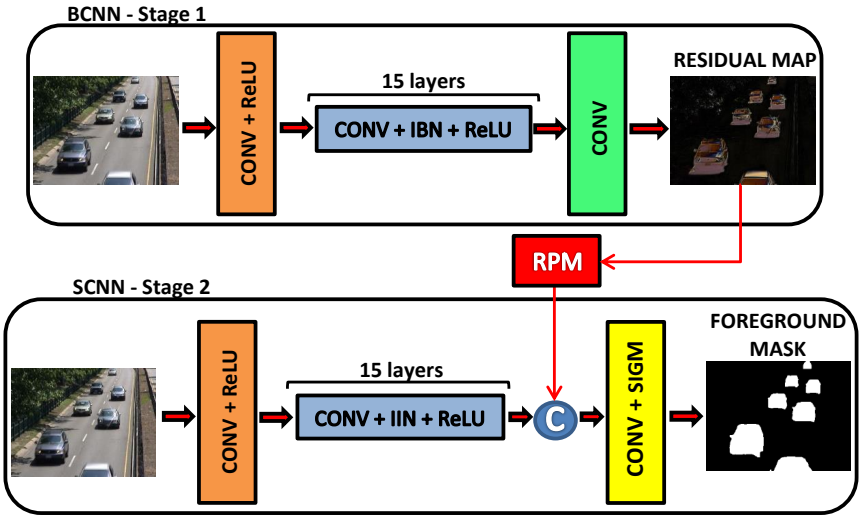

Fig. 1. Architecture of the proposed MCRCNN model, where the output of the residual processing module **RPM** is depth-wise concatenated to the 15th SCNN convolutional layer feature maps. **CONV** stands for convolutional layers, while **IBN** and **IIN** indicate, respectively, the interleaved batch normalization and instance normalization layers, which applied at every three subsequent convolutional layers.

improving the CRCNN method. Such changes include: (i) Multiscale residual map processing (ii) Multistage training using high-level feature aggregation policy (iii) Interleaved and hybrid intralayer feature normalization using batch [21] and instance [22] normalization strategies, and (iv) color image processing.

The remaining of this manuscript is divided into Section II, describing the theoretical basis of the proposal, Section III, presenting the proposal training and evaluation methodology, Section IV, presenting and discussing the quantitative and qualitative obtained results, and Section V, showing the conclusion of this work and pointing towards future research directions.

II. Proposed Approach

In this work, we propose a learning-based scene change detection technique named Multiscale Cascade Residual Convolutional Neural Network (MCRCNN). Such a proposal is based on the work of Zhang et al. [23], concerning the usage of residual learning and on the work of Santos et al. [17], regarding the usage of a multistage cascaded convolutional neural network for scene change detection. Figure 1 summarizes the MCRCNN proposal, which consists of a two-stage deep convolutional neural network composed of 20 layers and a multiscale Residual Processing Module (RPM).

The first stage of the MCRCNN model consists of learning how to generate the so-called residual map, as described in more detail in Subsection II-A. In the second stage, the multiscale processed residual map, as described by Subsection II-B, is integrated into the change detection network, whose functionality is described by Subsection II-C.

A. Background Convolutional Neural Network

The first change detection stage of the MCRCNN model, named Background Convolutional Neural Network (BCNN), is responsible for generating a foreground highlighted image, such as the vehicles in Figure 1. The BCNN architecture is very similar to the Denoising Convolutional Neural Network (DnCNN) proposed by Zhang et al. [23]. As shown by Figure 1, it starts with a single convolutional layer, shown in orange color, gets deeper with the insertion of 15 more convolutional layers, represented by the blue-colored rectangle, and ends with a single convolutional layer, shown in green color.

Blue-colored and orange-colored layers in Figure 1 are locally activated by Rectified Linear Unity (ReLU) functions [24], use kernels of size 3×3, and output 64 feature maps each. The green-colored layer is linearly activated, uses kernels of size 3×3, and outputs the residual map color image. One particularity of the blue-colored layers is the Interleaved Batch Normalizations (IBNs), which are batch normalization [21] operations applied at intervals of three layers, just before the ReLU activation procedure. Such a strategy tries to equally distribute the normalization procedure along the entire network avoiding processing overhead, also diminishing the network memory consumption.

The BCNN training procedure follows the same principles of the CRCNN work [17]. It consists of two phases: the first one takes an interval $I = \{S_1, S_2, ..., S_m\}$ of consecutive frames from the video and uses it to calculate the *deterministic background image*, which stands for an image s that represents the median of such an interval[1]. The second phase consists of minimizing the accumulated[2] square error between the deterministic background image and the *approximated background image* b, which is represented as follows:

[1]The same procedure was adopted by Lanza et al. [10]. Other alternatives would be using auxiliary non-learning-based segmentation techniques, like performed by Babaee et al. [16], or even manually selection.

[2]Minimizing the sum rather than the mean cost value imposes to the optimization an even bigger penalization.

$$b = f - BCNN(f; \Theta_1), \tag{1}$$

where f denotes the input image normalized between $[0, 1]$, Θ_1 refers to the BCNN trainable parameters, and $BCNN(\cdot)$ refers to the residual map learned during the training process. In light of that, the BCNN training process aims at minimizing the following equation:

$$L_B(b, f; \Theta_1) = \frac{1}{2} \sum_{i=1}^{n} ||b_i - s_i||_F^2, \tag{2}$$

where n stands for the number of training samples and $||\cdot||_F^2$ represents the Frobenius norm. Notice that we employed a patch-based methodology, where b_i and s_i denote the i^{th} patch extracted from images b and s, respectively.

B. Residual Processing Module

The Residual Processing Module (RPM) design was inspired by the Feature Processing Modules FPM [19] and FPM_M [20]. It serves mainly to improve the BCNN residual map quality treating undesirable spatial coherence problems. As shown by Figure 2, RPM starts by applying, over the residual map, a Spatial Dropout (SD) pre-processing technique, which according to Hinton et al. [25] is a very efficient strategy to prevent the network parameters from overspecialization.

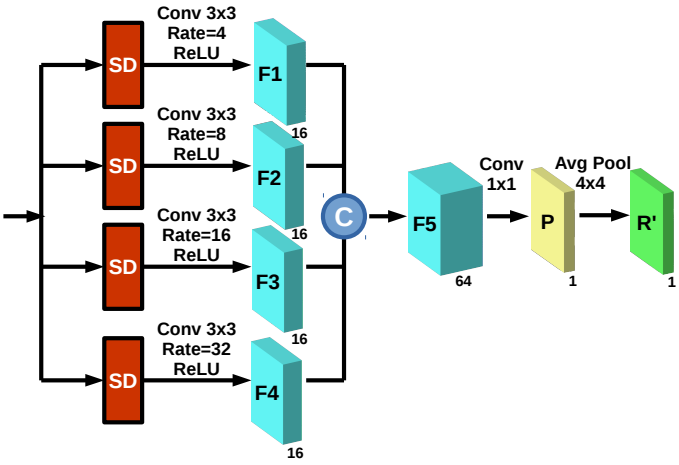

Fig. 2. Residual Processing Module architecture.

After the SD regularization, the residual map is conducted to the multiscale processing stage, where it is convolved by dilated[3] filters at rates of 4, 8, 16, and 32. The dilation results are then activated by ReLU functions, which generate the feature maps **F1** to **F4**.

Such generated feature maps are next depth-wise concatenated into **F5** and convolved by a single 1×1 filter, producing the local linear-combined map **P**. In the last RPM processing step, **P** is smoothed by an average pooling layer of window size 4×4, generating the refined residual map **R'**, that after has been properly normalized[4], is used in the second stage of the MCRCNN proposed model.

C. Segmentation Convolutional Neural Network

The second stage of the MCRCNN model is named Segmentation Convolutional Neural Network (SCNN) and it is responsible to generate the probability map identifying, with real values between $[0, 1]$, the image change locations, also called foreground regions. In this multiscale version of the CRCNN proposed by Santos et al. [17], the RPM output (see Subsection II-B for more details) is depth-wise concatenated with the 15th SCNN convolutional layer output[5]. The resultant block of 65 feature maps is then convolved by a single filter of size 3×3 and activated by a sigmoid function, been such convolutional process represented in Figure 1 by the yellow rectangle.

The SCNN normalization policy follows the BCNN one, but in such case, the IBNs are substituted[6] by Interleaved Instance Normalizations (IINs) [22]. The training process follows the work by [17], which aims at minimizing the average binary cross-entropy measured between the network output and the ground-truth binary detection mask. Such an image corresponds to the pre-annotated true foreground regions present in the grayscale input image. Therefore, the SCNN training process aims at minimizing the following equation:

$$L_S(t, f; \Theta_2) = -\sum_{i=1}^{k1} \sum_{j=1}^{k2} [t_{i,j} \log(\hat{t}_{i,j}) + (1 - t_{i,j}) \log(1 - \hat{t}_{i,j})], \tag{3}$$

where

$$\hat{t} = SCNN(f; \Theta_2), \tag{4}$$

notice that t is the ground-truth pre-annotated binary mask, Θ_2 stands for the SCNN trainable parameters, f indicates the SCNN input color image, the same BCNN input image, and $k1$ and $k2$ denote the maximum image height and width, respectively.

III. Methodology

In this section, we present the methodology used to train and evaluate the proposed MCRCNN model. To simplify the explanation we structured it into Subsection III-A, which presents the relevant information about the datasets used in this work, Subsection III-B, that describes the proposal training procedures, and Subsection III-C, which discuss the MCRCNN evaluation protocol.

[3]Such a strategy tries to simulate the usage of kernel sizes of respectively 7×7, 11×11, 18×18, and 35×35.

[4]The values of the output of the RPM module are normalized between $[0, 1]$ using min-max normalization.

[5]Such output comprehends a set of 64 feature maps activated by ReLU function.

[6]Since the SCNN optimization consists in using the full-sized images, IN processing adapts better than BN ones.

A. Datasets

1) Change Detection Dataset 2014: The Change Detection Dataset 2014 (CD2014) is a large and freely available dataset of videos collected by Wang et al [26] from different realistic, camera-captured, and challenging scenarios. Such a dataset contains 11 video categories with 4 to 6 video sequences each, subdivided into:

- **Baseline**: combines mild challenges present in Dynamic Background, Camera Jitter, Intermittent Object Motion, and Shadow categories into four different videos named highway, office, pedestrians, and PETS2006.
- **Dynamic Background** (Dyn. Bg.): includes scenes from six different videos with so much background motion, e.g., cars and trucks passing in front of a tree shaken. Such video names are boats, canoe, fall, fountain01, fountain02, and overpass.
- **Camera Jitter** (C. Jitter): contains four indoor and outdoor videos captured by unstable video devices, for example vibrating cameras. Those video names are badminton, boulevard, sidewalk, and traffic.
- **Intermittent Object Motion** (Int. Obj.): contains six videos with objects that move and then stop for a short while producing "ghosting" artifacts. Such video names are abandonedBox, parking, sofa, streetLight, tramstop, and winterDriveway.
- **Shadow**: six indoor and outdoor videos containing objects surrounding by a strong shadow that could be miss detected as real moving objects. Such video names are backdoor, bungalows, busStation, copyMachine, cubicle, and peopleInShade.
- **Thermal**: five videos that have been captured by far-infrared cameras named corridor, diningRoom, lakeSide, library, and park.
- **Bad Weather** (B. Weat.): includes four outdoor videos captured from challenging winter weather conditions, e.g., snowstorms, and fog. Such video names are blizzard, skating, snowFall, and wetSnow.
- **Low Framerate** (L. Frame.): four videos captured varying frame-rates between 0.17fps and 1fps. Such video names are port_0_17fps, tramCrossroad_1fps, tunnelExit_0_35fps, and turnpike_0_5fps.
- **PTZ** (PanTZ): four videos captured by pan-tilt-zoom cameras and named continuousPan, IntermittentPan, twoPositionPTZCam, and zoomInZoomOut.
- **Turbulence** (Turbul.): four outdoor videos that show air turbulence caused by rising heat. Which are named turbulence0, turbulence1, turbulence2, and turbulence3.

2) PetrobrasROUTES: The PetrobrasROUTES is a private dataset which consists of 281 high-resolution color images collected from an indoor Petrobras[7] workspace. The main challenge of such a dataset regards the detection of objects obstructing escape routes.

[7]Petrobras is a publicly-held company on an integrated basis and specialized in the oil, natural gas, and energy industry [27].

B. Training procedure

The training procedure methodology follows basically the same protocols of [17], where for the CD2014 dataset consist of:

1) to select 300 color images[8] and their 300 correspondent binary images, which were ground-truth manually annotated.
2) to calculate the deterministic background over the first 100 images.
3) to train the BCNN network using batches of randomly extracted patches of size 40×40, like in [23], from the input and output background images to minimize the cost of Equation (2). The patches were augmented using geometric transformations, such as rotation and reflection.
4) to freeze all BCNN network trainable parameters and just train the second MCRCNN part, the SCNN network, and also the RPM module using the full-sized images to minimize the cost of Equation (3).

For the PetrobrasROUTES the training procedure consists in:

1) to select 51 color images and their 51 correspondent binary images, which were ground-truth manually annotated.
2) to manually select one of the 51 color images to be the deterministic background.
3) to follow the same steps 3) and 4) from the CD2014 dataset training protocol.

The BCNN, RPM, and SCNN parameters were trained using the Adam method [28] by a maximum of 100 epochs[9], with 500 gradient updates per epoch, using a learning rate[10] of 0.001 and batches of size 128 for the BCNN training process. We trained the MCRCNN parameters with 80% of the input images and used the remaining 20% to evaluate the convergence of the training process.

C. Evaluation procedure

The evaluation process consists in to apply the trained MCRCNN model over each video test image following the protocol:

- **Deep Segmentation**: first forward propagating the test images through the trained BCNN model, generating the residual image counterpart, and through the trained SCNN model. Before the last SCNN convolution, we concatenate the residual image to the 15th SCNN convolutional layer outputs. Later, we binarized[11] the SCNN probabilistic output.

[8]We used the same set of training images from [20] to train the proposed MCRCNN model.

[9]Depending on the training video sequence, convergence can be achieved in less than 100 epochs.

[10]The initial value is reduced by a factor of 0.1 every time the loss function hits a plateau.

[11]In the majority of the experiments, the best threshold value was 0.7, except for the categories B. Weat, Dyn. Bg., Int. Obj., and N. Videos, which used values of respectively 0.8, 0.9, 0.6, and 0.9.

TABLE I
COMPARISON OF F-MEASURE RESULTS OF 11 CATEGORIES FROM CD2014 DATASET

Methods	Baseline	C.Jitter	B.Weat	Dyn.Bg.	Int.Obj.	L.Frame.	N.Videos	PanTZ	Shadow	Thermal	Turbul.	Overall
FgSegNet_v2 [20]	**0.9980**	**0.9961**	0.9900	**0.9950**	0.9939	**0.9579**	0.9816	**0.9936**	0.9966	0.9942	**0.9815**	**0.9890**
FgSegNet_S [19]	**0.9980**	0.9951	**0.9902**	0.9902	**0.9942**	0.9511	**0.9837**	0.9837	**0.9967**	**0.9945**	0.9796	0.9878
FgSegNet_M [19]	0.9975	0.9945	0.9838	0.9838	0.9933	0.9558	0.9779	0.9779	0.9954	0.9923	0.9776	0.9865
MCRCNN	0.9938	0.9889	0.9632	0.9811	0.9893	0.8619	0.9428	0.9344	0.9906	0.9765	0.9635	0.9622
CRCNN [17]	0.9919	0.9799	0.9569	0.9687	0.9755	0.8498	0.9388	0.8967	0.9852	0.9818	0.9637	0.9535
Cascade [15]	0.9786	0.9758	0.9451	0.9451	0.8505	0.8804	0.8926	0.8926	0.9593	0.8958	0.9215	0.9272
DeepBS [16]	0.9580	0.8990	0.8647	0.8647	0.6097	0.5900	0.6359	0.6359	0.9304	0.7583	0.8993	0.7593
IUTIS-5 [14]	0.9567	0.8332	0.8289	0.8289	0.7296	0.7911	0.5132	0.5132	0.9084	0.8303	0.8507	0.7820
PAWCS [13]	0.9397	0.8137	0.8059	0.8059	0.7764	0.6433	0.4171	0.4171	0.8934	0.8324	0.7667	0.7477
SuBSENSE [12]	0.9503	0.8152	0.8594	0.8594	0.6569	0.6594	0.4918	0.4918	0.8986	0.8171	0.8423	0.7453

- **Misclassification Rate**: in such a step, we calculated the number of correct and incorrect detections encoded by the True Positives (TPs), i.e, the number of pixels correctly classified as foreground, the True Negatives (TNs), i.e., the number of pixels correctly classified as background, the False Positives (FPs), i.e., the number of background pixels incorrectly classified as foreground, and the False Negatives (FNs), i.e., the number of foreground pixels incorrectly classified as background.
- **Detection Measurements**: in such a step, (TPs), (TNs), (FPs), and (FNs) are combined into four different measures used to evaluate the robustness of the proposed MCRCNN model. Those measures are computed as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

$$F - measure = 2.0 \times \frac{Recall \times Precision}{Recall + Precision}, \tag{7}$$

and

$$PWC = 100.0 \times \frac{FN + FP}{TP + FP + FN + TN} \tag{8}$$

where PWC denotes the percentage of wrong classifications.

IV. EXPERIMENTAL RESULTS

In this section, we present the results of the proposed MCRCNN method regarding the comparison against the non-learning-based change detection techniques, IUTIS-5 [14], PAWCS [13], SuBSENSE [12], and the learning-based ones FgSegNet_v2 [20], FgSegNet_S [19], FgSegNet_M [19], Cascade [15], DeepBS [16], and CRCNN [17].

For the sake of clarity, the discussion is subdivided into Subsection IV-A, which presents the quantitative and qualitative results related to the CD2014 dataset, and Subsection IV-B, which presents the results regarding Petrobras-ROUTES dataset.

A. CD2014 Dataset Results

According to Table I, the MCRCNN proposal, in comparison against SuBSENSE, PAWCS, and IUTIS-5 techniques, shows average overall $F\text{-}measure$ improvements of 0.2169, 0.2145, and 0.1802, respectively. In the worst-case scenario, considering the comparison against the learning-based techniques, MCRCNN average overall $F\text{-}measure$ results were lower than FgSegNet_v2, FgSegNet_S, and FgSegNet_M by respectively 0.0268, 0.0256, and 0.0243. Table I also shows that MCRCNN average overall $F\text{-}measure$ results overcome DeepBS, Cascade, and CRCNN ones. In such cases, the results were improved by respectively 0.2029, 0.035, and 0.0090, respectively.

Analyzing Table II, one can see that the proposed technique, in the best-case scenario, achieved improvements in $Precision$, $Recall$, and PWC of respectively 0.2196, 0.1969, and 1.8883, regarding the comparisons against SuBSENSE and DeepBS techniques. Table II also shows that even so MCRCNN was not capable to overcome the FgSegnets, it gets close $Precision$ results of 0.0046 and 0.0053 in comparisons against FgSegNet_S and FgSegNet_M, respectively.

It is worth noting that even so the FgSegNets quantitative results, presented by Tables I and II, overcome the MCRCNN method, our proposal network architecture is much more compact. It has a total of $1,116,618$ parameters, while the top two ranked techniques, i.e., FgSegNet_v2 and FgSegNet_S, comprise an amount of $9,225,161$ and $7,622,465$ parameters, respectively. Besides, even considering the RPM module size, the MCRCNN almost preserves the same CRCNN size of $1,112,720$ parameters.

TABLE II
COMPARISON OF PRECISION, RECALL AND PWC OVERALL RESULTS FROM CD2014 DATASET.

Methods	Avg. Precision	Avg. Recall	Avg. PWC
FgSegNet_v2 [20]	**0.9823**	0.9891	**0.0402**
FgSegNet_S [19]	0.9751	**0.9896**	0.0461
FgSegNet_M [19]	0.9758	0.9836	0.0559
MCRCNN	0.9705	0.9514	0.1037
CRCNN [17]	0.9604	0.9602	0.1348
Cascade [15]	0.8997	0.9506	0.4052
DeepBS [16]	0.8332	0.7545	1.9920
IUTIS-5 [14]	0.8087	0.7849	1.1986
PAWCS [13]	0.7857	0.7718	1.1992
SuBSENSE [12]	0.7509	0.8124	1.6780

According to Figure 3, when comparing the MCRCNN foreground detection masks in row (d) with the CRCNN masks in row (e), it can be noticed that MCRCNN exhibit more problems related to false negatives, been those problems more pronounced in Bad Weather and Shadow category scenes. The first one regards the incomplete detection of the truck body, and the second one concerns the middle person foot and the people heads. Such observations corroborate with the average overall MCRCNN $Recall$ results presented by Table III.

According to Figure 3, the images from row (f) show that the Cascade technique also has some difficulties to detect changes in the Bad Weather and Shadow categories. It presents even worst false negative issues, as it can be seen by the barely detected truck body in the Bad Weather scene, and by the undetected person's head in the Shadow scene. Also, regarding the Shadow category, different from MCRCNN, CRCNN, and FgSegNet_v2 techniques, Cascade was not able to avoid the false positive shadow regions.

Besides the foreground masks, row (b) of Figure 3 shows us the BCNN normalized residual map. As it can be noticed, the miss detected foreground regions are pretty much related to the dark residual map regions. In such cases, we argue that the RPM dilation processing strategy was not capable of properly fill those map holes, which could contribute to the SCNN paying less attention to such regions during its training procedure.

B. PetrobrasROUTES Dataset Results

Considering the experiments conducted over the PetrobrasROUTES dataset, Table III shows that the MCRCNN results overcome learning-based state-of-the-art change detection techniques like FgSegNet_v2, FgSegNet_S, and CRCNN in terms of at least three of the four used detection measurements.

According to Table III, in the best case scenario, regarding the comparison against the FgSegNet_v2 technique, the MCRCNN method exhibit improvements of 0.0524, 0.1084, and 0.3535 in terms of $F\text{-}measure$, $Recall$, and PWC measurements, respectively. In the worst case scenarios, the MCRCNN comparisons against FgSegNet_S and CRCNN exhibit worsen results of 0.0109 and 0.0078 in terms of respectively $Precision$ and PWC measurements.

TABLE III
COMPARISON OF PRECISION, RECALL AND PWC OVERALL RESULTS FROM PETROBRASROUTES DATASET.

Methods	F-measure	Precision	Recall	PWC
FgSegNet_v2 [20]	0.9095	0.9672	0.8583	0.5831
FgSegNet_S [19]	0.9221	**0.9770**	0.8732	0.4287
MCRCNN	**0.9664**	0.9661	**0.9667**	0.2296
CRCNN [17]	0.9619	0.9611	0.9627	**0.2218**

Concerning the detection quality analysis, Figure 4(c) shows that MCRCNN was able to produce a much more precise foreground object detection mask in comparison against FgSegNet_v2, whose results were severed affected by false negatives, as shown by Figure 4(e). On the other hand, even so in Figure 4(c) most of the object shape was recovered, in comparison against Figure 4(d), which shows the CRCNN results, and against Figure 4(b), which shows the reference ground-truth mask, the MCRCNN technique presents more false positive areas around the detected foreground object.

V. CONCLUSION

In this work, we proposed a novel Cascade Residual Convolutional Neural Network that integrates a multiscale processing strategy (through a developed residual processing module) with a learning-based segmentation mechanism in an attempt to solve the scene change detection problems. Regarding tests conducted over CD2014 dataset, the proposed MCRCNN model achieved results close to the state-of-the-art change detection techniques. The proposal was capable of overcoming three supervised learning-based change detection methods and three other non-learning based ones. Even so the MCRCNN did not overcome the FgSegNet_v2, FgSegNet_S, and FgSegNet_M techniques regarding CD2014 dataset, it proven to be much more compact, i.e., around $8\times$ smaller than the best scored FgSegNet_v2 technique in the number of network parameters. Regarding the test conducted over PetrobrasROUTES dataset, the proposed MCRCNN model outperformed the top two state-of-the-art techniques FgSegNet_v2 and FgSegNet_S, and also the CRCNN method. Regarding future works, we pretend to focus our investigation in the MCRCNN false negative problem, conducting a more careful analysis of the RPM filter. We also intend to search for other possible ways to improve the residual learning process and also explore different ways of integrating the residual learned map with the second stage MCRCNN segmentation network.

ACKNOWLEDGMENT

The authors are grateful to CNPq grants 307066/2017-7 and 427968/2018-6, FAPESP grants 2013/07375-0 and 2014/12236-1, as well as Petrobras grant 2017/00285-6.

REFERENCES

[1] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1291–1296, 2002.

[2] X. Dai, "Hybridnet: A fast vehicle detection system for autonomous driving," *Signal Processing: Image Communication*, vol. 70, pp. 79–88, 2019.

[3] J. Zhou and J. Hoang, "Real time robust human detection and tracking system," in *Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2005, pp. 149–149.

[4] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6202–6211.

[5] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1937–1944.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *Computing Surveys*, vol. 41, no. 3, p. 15, 2009.

[7] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

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

[9] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of gaussians for dynamic background modelling," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2013, pp. 63–68.

[10] A. Bevilacqua, L. Di Stefano, and A. Lanza, "A simple self-calibration method to infer a non-parametric model of the imaging system noise," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (wacv/motion'05)-Volume 1*, vol. 2. IEEE, 2005, pp. 229–234.

[11] A. Lanza and L. Di Stefano, "Statistical change detection by the pool adjacent violators algorithm," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1894–1910, 2011.

[12] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, pp. 359–373, 2014.

[13] St-Charles, Pierre-Luc, Bilodeau, Guillaume-Alexandre, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *2015 IEEE winter conference on applications of computer vision*. IEEE, 2015, pp. 990–997.

[14] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 914–928, 2017.

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

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

[17] D. F. Santos, R. G. Pires, D. Colombo, and J. P. Papa, "Video segmentation learning using cascade residual convolutional neural network," in *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2019, pp. 1–7.

[18] M. C. Santana, L. A. P. Júnior, T. P. Moreira, D. Colombo, V. H. C. de Albuquerque, and J. P. Papa, "A novel siamese-based approach for scene change detection with applications to obstructed routes in hazardous environments," *IEEE Intelligent Systems*, vol. 35, no. 1, pp. 44–53, 2019.

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

[20] Lim, L. Ang, Keles, and H. Yalim, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, pp. 1–12, 2019.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[23] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Madison, WI, USA: ACM, 2010, pp. 807–814.

[25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[26] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 387–394.

[27] "Petrobras," http://www.petrobras.com.br/en, accessed: 2019-06-04.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

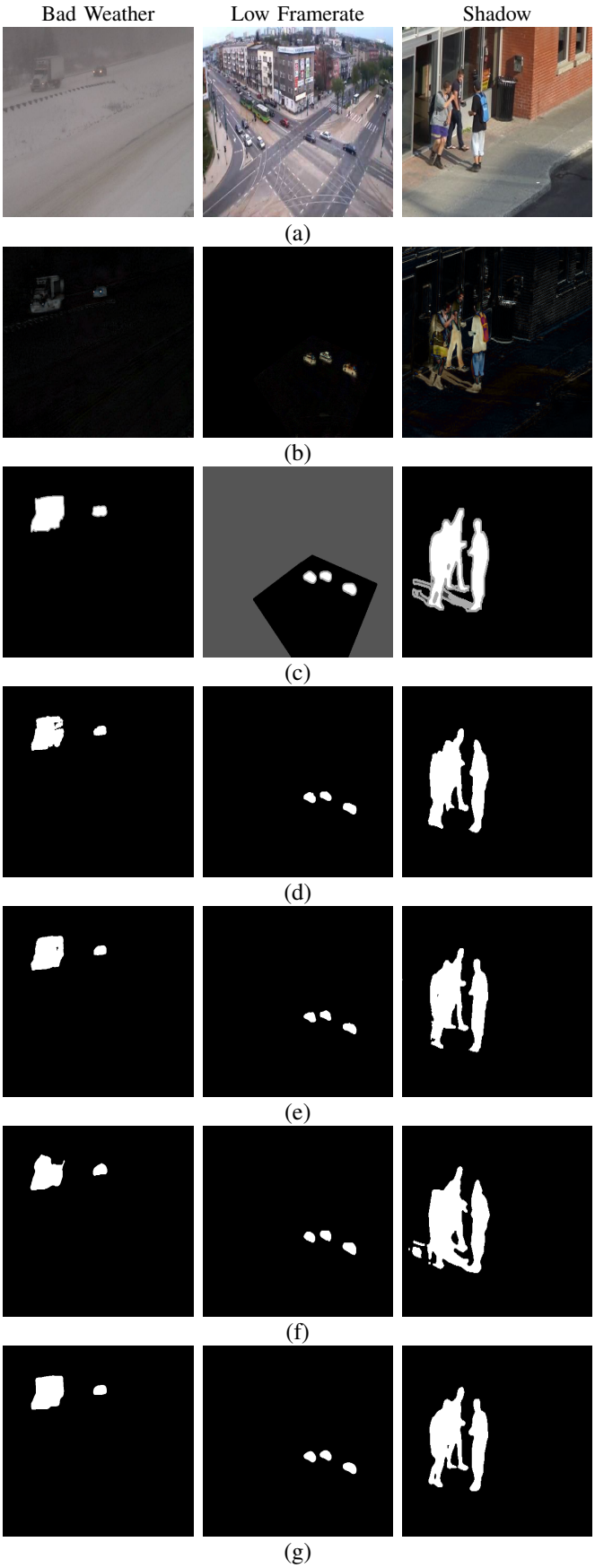

Fig. 3. Qualitative results considering the categories "Bad Weather", "Low Framerate", and "Shadow" from CD2014 dataset: (a) input RGB frame, (b) MCRCNN residual maps, (c) ground-truth detection masks, results concerning (d) proposed MCRCNN, (e) CRCNN, (f) Cascade and, (g) FgSegNet_v2.

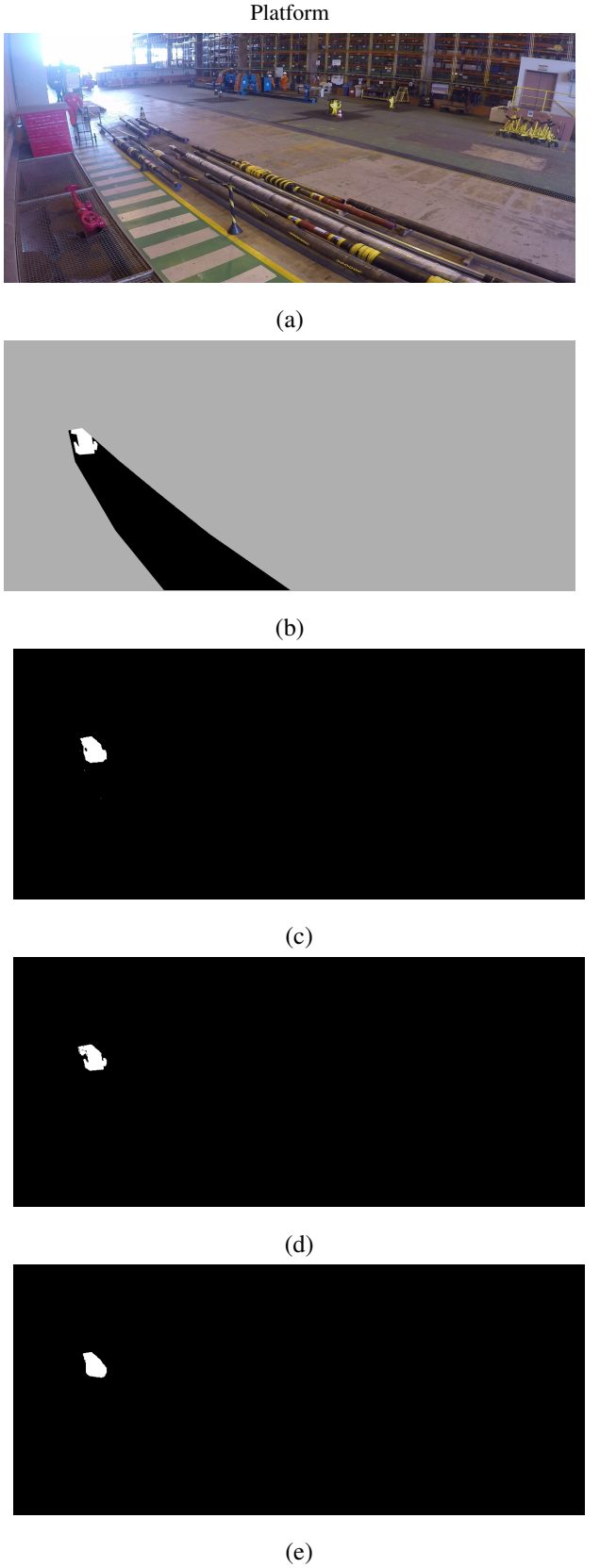

Fig. 4. Qualitative results considering an obstructed route video scene from PetrobrasROUTES dataset: (a) input RGB frame, (b) ground-truth detection mask, and results concerning (c) MCRCNN, (d) CRCNN and (e) FgSegNet_v2 techniques.