Pattern Recognition 000 (2017) 1–15

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A deep convolutional neural network for video sequence background subtraction

Mohammadreza Babaee[a,*], Duc Tung Dinh[a], Gerhard Rigoll[a]

Institute for Human-Machine Communication, Technical University of Munich, Germany

ARTICLE INFO

Article history:
Received 23 December 2016
Revised 25 June 2017
Accepted 27 September 2017
Available online xxx

Keywords:
Background subtraction
Video segmentation
CNN
Deep learning

ABSTRACT

In this work, we present a novel background subtraction from video sequences algorithm that uses a deep Convolutional Neural Network (CNN) to perform the segmentation. With this approach, feature engineering and parameter tuning become unnecessary since the network parameters can be learned from data by training a single CNN that can handle various video scenes. Additionally, we propose a new approach to estimate background model from video sequences. For the training of the CNN, we employed randomly 5% video frames and their ground truth segmentations taken from the *Change Detection* challenge 2014 (CDnet 2014). We also utilized spatial-median filtering as the post-processing of the network outputs. Our method is evaluated with different data-sets, and it (so-called *DeepBS*) outperforms the existing algorithms with respect to the average ranking over different evaluation metrics announced in CDnet 2014. Furthermore, due to the network architecture, our CNN is capable of real time processing.

1. Introduction

With the tremendous amount of available video data, it is important to maintain the efficiency of video based applications to process only relevant information. Most video files contain redundant information such as background scenery, which costs a huge amount of storage and computing resources. Hence, it is necessary to extract the meaningful information, e.g. vehicles or pedestrians, to deploy those resources more efficiently. Background subtraction is a binary classification task that assigns each pixel in a video sequence with a label, for either belonging to the background or foreground scene [1,21,25].

Background subtraction, which is also called change detection, is applied to many advanced video applications as a pre-processing step to remove redundant data, for instance in tracking or automated video surveillance [17]. In addition, for real-time applications, like tracking, the algorithm should be capable of processing the video frames in real-time.

One simple example of the application of a background subtraction method is the pixel-wise subtraction of a video frame from its corresponding background image. After being compared with the difference threshold, pixels with a larger difference than a certain threshold value are labeled as foreground pixels, otherwise as background pixels. Unfortunately, this strategy will yield poor segmentation due to the dynamic nature of the background, that is induced by noise or illumination changes. For example, due to lighting changes, it is common that even pixels belonging to the background scene can have intensities very different from their other pixels in the background image and they will be falsely classified as foreground pixels as a consequence. Thus, sophisticated background subtraction algorithms that assure robust background subtraction under various conditions must be employed.

In the following sections, the difficulties in this area, our proposed solution for background subtraction and our contributions will be illustrated.

1.1. Challenges

The main difficulties that complicate the background subtraction process are:

Illumination changes: When scene lighting changes gradually (e.g. moving clouds in the sky) or instantly (e.g. when the light in a room is switched on), the background model usually has a illumination different from the current video frame and therefore yields false classification.

Dynamic background: The background scene is rarely static due to movement in the background (e.g. waves, swaying tree leaves), especially in outdoor scenes. As a consequence, parts of the background in the video frame do not overlap with the corresponding parts in the background image, hence, the pixel-wise correspondence between image and background is no longer existent.

* Corresponding author.
E-mail address: reza.babaee@tum.de (M. Babaee).

https://doi.org/10.1016/j.patcog.2017.09.040
0031-3203/

Camera jitter: In some cases, instead of being static, it is possible that the camera itself is frequently in movement due to physical influence. Similar to the dynamic background case, the pixel locations between the video and background frame do not overlap anymore. The difference in this case is that it also applies to non-moving background regions.

Camouflage: Most background subtraction algorithms work with pixel or color intensities. When foreground objects and background scene have a similar color, the foreground objects are more likely to be (falsely) labeled as background.

Night Videos: As most pixels have a similar color in a night scene, recognition of foreground objects and their contours is difficult, especially when color information is the only feature in use for segmentation.

Ghosts/intermittent object motion: Foreground objects that are embedded into the background scene and start moving after background initialization are the so-called *ghosts*. The exposed scene, that was covered by the ghost, should be considered as background. In contrast, foreground objects that stop moving for a certain amount of time, fall into the category of intermittent object motion. Whether the object should be labeled as foreground or background is strongly application dependent. As an example, in the automated video surveillance case, abandoned objects should be labeled as foreground.

Hard shadows: Dark, moving shadows that do not fall under the illumination change category should not be labeled as foreground.

In this work, we follow the trend of Deep Learning and apply its concepts to background subtraction by proposing a CNN to perform this task. We justify this approach with the fact that background subtraction can be performed without temporal information, given a sufficiently good background image. With such a background image, the task itself breaks down into a comparison of a image-background pair. Hence, the input samples can be independent among each other, enabling a CNN to perform the task with only spatial information. The CNN is responsible for extracting the relevant features from a given image-background pair and performs segmentation by feeding the extracted features into a classifier. In order to get more spatially consistent segmentation, post-processing of the network output is done by spatial-median filtering and or a fully connected CRF framework. Due to the use of a CNN, no parameter tuning or descriptor engineering is needed.

To train the network, a large amount of labeled data is required. Fortunately, due to the process of background subtraction, by comparing an image with its background, it is not necessary to use images of a full scene for training. It is also possible to train the network via subsets of a scene, i.e. with patches of image-background pairs, since the procedure also holds for image patches. As a consequence, we can extract enough training samples from a limited amount of labeled data.

To the best of our knowledge, background subtraction algorithms that use CNN are scene specific to this day, i.e. a CNN can only perform satisfying background subtraction on a single scene (that was trained with scene specific data) and also lacks the ability to perform the segmentation in real time. Our proposed approach yields a universal network that can handle various scenes without having to retrain it every time the scene changes. As a consequence, one can train the network with data from multiple scenes and hence increase the amount of training data for a network. Also, by using the proposed network architecture, it is possible to process video frames in real-time with conventional computing resources. Therefore, our approach can be considered for real-time applications.

The outline of this paper is as follows: In Section 2, early and recent algorithms for background subtraction are presented. In Section 3, we explain our proposed approach for background subtraction. Here, we first describe our proposed approach to estimate background image and next we illustrate our CNN for background subtraction. In Sections 4–6, we describe the experimental evaluation of the algorithm including the used datasets, the evaluation metrics and the obtained results followed by detailed discussion and analysis. Finally, in Section 7, we conclude our work and provide future work with some ideas.

2. Overview approaches

The majority of background subtraction algorithms are composed of several processing modules which are explained in the following sections (see Fig. 1).

Background Model: The background model is essential for the background subtraction algorithm. In general, the background model is used as a reference to compare with the incoming video frames. Furthermore, the initialization of the background model plays an important role since video sequences are normally not completely free of foreground objects during the bootstrapping phase. As a consequence, the model gets corrupted by including foreground objects into the background model, which leads to false classifications.

Background Model Maintenance: In reality, background is never completely static but changes over time. There are many strategies to adapt the background model to these changes by using the latest video frames and/or previous segmentation results. Trade-offs must be found in the adaption rate, which regulates how fast the background model is updated. High adaption rate leads to noisy segmentation due to the sensitivity to small or temporary changes. Slow adaption rate, however, yields an outdated background model and therefore false segmentation. Selective update adapts the background model with pixels that were classified as background. In that case, deadlock situations can occur by not incorporating misclassified pixels into the background model, i.e. once a background pixel is falsely classified as foreground, it would never be used to update the background and would always be considered as a foreground pixel. On the other hand, by using all pixels as in the blind update strategy, such deadlock situations can be prevented but will also distort the background model since foreground pixels are incorporated into the background model.

Feature extraction: In order to compare the background image with the video frames, adequate features that represent relevant information must be selected. Most algorithms use gray scale or RGB intensities as features. In some cases, pixel intensities along with other hand engineered features (e.g. [11] or [2]) are combined. Also, the choice of the feature region is important. One can extract the features over pixels, blocks or patterns. Pixel-wise features often yield noisy segmentation results since they do not encode local correlation, while block-wise or pattern-wise features tend to be insensitive to slight changes.

Segmentation: With the help of a background model, the respective video frames can be processed. Background segmentation is performed by extracting the features from corresponding pixels or pixel regions of both frames and using a distance measure, e.g. the Euclidean distance, to measure the similarity between those pixels. After being compared with the similarity threshold, each pixel is either labeled as background or foreground. The combination of those building blocks forms an overall background subtraction system. The robustness of the system is always dependent and limited by the performance of each individual block, i.e. it can not be expected to perform well if one module delivers poor performance. Background subtraction is a well studied field, therefore there exists a vast number of algorithms for this purpose (see Fig. 2). Since most of the top performing methods at present are based on the early proposed algorithms, some of which are outlined in the beginning. Subsequently a few of the current methods for background subtraction will be introduced.

Please cite this article as: M. Babaee et al., A deep convolutional neural network for video sequence background subtraction, Pattern Recognition (2017), https://doi.org/10.1016/j.patcog.2017.09.040