# Foreground Segmentation for Anomaly Detection in Surveillance Videos Using Deep Residual Networks

Lucas P. Cinelli, Lucas A. Thomaz, Allan F. da Silva, Eduardo A. B. da Silva and Sergio L. Netto

***Abstract*— Efficient anomaly detection in surveillance videos across diverse environments represents a major challenge in Computer Vision. This paper proposes a background subtraction approach based on the recent deep learning framework of residual neural networks that is capable of detecting multiple objects of different sizes by pixel-wise foreground segmentation. The proposed algorithm takes as input a reference (anomaly-free) and a target frame, both temporally aligned, and outputs a segmentation map of same spatial resolution where the highlighted pixels denoting the detected anomalies, which should be all the elements not present in the reference frame. Furthermore, we analyze the benefits of different reconstruction methods to the restore original image resolution and demonstrate the improvement of residual architectures over the smaller and simpler models proposed by previous similar works. Experiments show competitive performance in the tested dataset, as well as real-time processing capability.**

***Keywords*— Deep learning, convolutional neural networks, ResNet, residual networks, background subtraction, foreground segmentation, anomaly detection, surveillance, real-time.**

## I. Introduction

Deep convolutional neural networks have been pushing the state-of-the-art in terms of performance in classification tasks [1], [2], [3], as well as in other diverse visual tasks, leveraged by the self-learned representation of data, that progressively builds complex descriptors throughout the model's many layers. Recent works on Residual Networks (ResNets) [3], [4] attempt to overcome the increasing optimization difficulty and performance deterioration brought about by the increasing depth of models by adding skip connections to bypass a small number of convolutional layers at a time. Residual network based models have become the default Convolutional Neural Network (CNN) of choice for many different tasks, not only because they are one of the top performing architectures, but also because they converge for a wider range of hyper-parameters while still keeping a simple modular architecture.

In such a scenario, it is natural to attempt to port these solutions to image segmentation at pixel level. Recently, researchers have started adapting well-known classification-oriented architectures to pixel-wise prediction with successful results [5]. Although this is an active research area [5], [6], [7], few works aim to solve the problem of dense anomaly detection at full-resolution [8], [9].

Background subtraction is an essential part of many computer vision frameworks and can be found in a wide range of image processing applications, hence the importance of this study. Previous works using deep learning [9] rely on patch-wise approaches to construct full-resolution foreground segmentation maps, which leads to massive recalculation of features and waste of both computational resource and time.

The present work investigates how background subtraction and anomaly detection can be efficiently performed pixel-wise using CNNs. For our purpose, we consider as anomaly objects not present in the reference and do not take into account actions nor events. We propose models that only require temporally aligned reference and target frames as input and compute pixel-wise segmentation maps in real-time with competitive performance in the evaluated database.

In the next section, we review the related work on image segmentation and anomaly detection using CNNs. Sections III and IV explain in details the original architecture in which this work was initially based as well as the proposed modifications, respectively. Section V briefly introduces the dataset used and discusses the obtained experimental results. The last section summarize our conclusions.

## II. Related Work

Our system is inspired on the work of Brahams and Van Droogenbroeck [9], who used a modified Lenet5 [10] deep network to predict the probability of each input image pixel belonging to the foreground given its vicinity and the correspondent background model. Their original architecture predicts a single pixel at a time for each image patch under analysis, thus requiring as many runs as there are pixels in the image if one wants to compute a dense segmentation map. Long *et al.* [5] avoided similar efficiency problem in segmentation by transforming the fully-connected layers of their models into the convolutional equivalents and then applying bilinear up-sampling to the coarse output map. By doing so, they were able to recover the original input resolution, thus dispensing with the need for sliding window both during the training and evaluation phases.

The reconstruction problem has been approached in different manners, e.g., in [6], [11] the authors use encoder-decoder networks to learn the representation of data and its reconstruction. They apply unpoolings to obtain sparse up-sampled maps which preserve local spatial information and (de)convolutions to populate these higher resolution maps. Likewise, we also evaluate this kind of approach on our foreground segmentation task and compare it with the bilinear up-sampling one.

Recently, Wang *et al.* [12] also presented a CNN model to segment frames in the CDNET database. Their approach, heavily based on [9], consists of a cascade of two 5-layer

Lucas P. Cinelli, Lucas A. Thomaz, Allan F. da Silva, Eduardo A. B. da Silva and Sergio L. Netto¸ Laboratório de Sinais, Sistemas e Telecomunicações (SMT), Coppe, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro-RJ, Brazil, E-mails: lucas.cinelli@smt.ufrj.br, lucas.thomaz@smt.ufrj.br, allan.freitas@smt.ufrj.br, eduardo@smt.ufrj.br, sergioln@smt.ufrj.br.

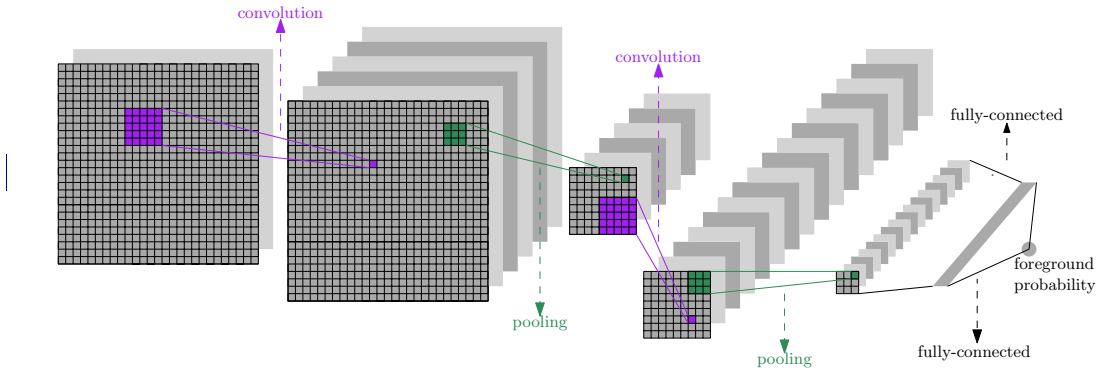


Fig. 1. Original network of [9] for background subtraction. It consists of two feature stages, each composed of one $5 \times 5$ kernel convolution and one $3 \times 3$ non-overlapping max-pooling, followed by two fully-connected layers whose output is the patch-centered pixel foreground probability.

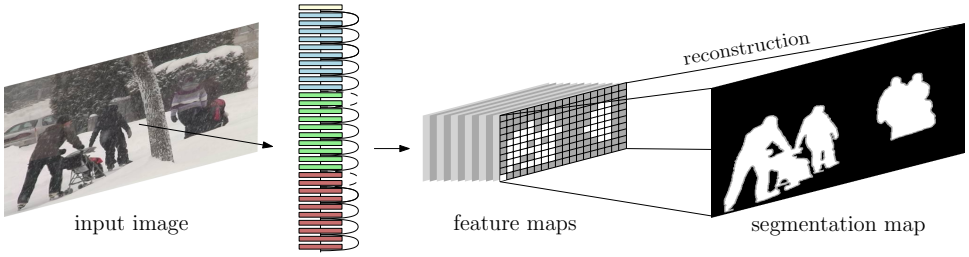


Fig. 2. The pipeline of our foreground segmentation system proposal. The deep neural network processes the input image and outputs a stack of feature maps, onto which we apply one of the studied up-sampling methods and obtain, at the end, the reconstructed foreground probability map.

networks, with the second refining the coarse predictions of the first, whose input are the RGB channels of the image. Even though their reported results are outstanding and the best so far for CDNET, they rely on human intervention to manually pick annotated training frames as well as on video-specific training. These characteristics make their model suited for groundtruth generation, but inadequate for real-time anomaly detection.

DeepAnomaly [8], on the other hand, was proposed with this exact purpose, real-time anomaly detection. It extracts intermediate features from pretrained CNNs and applies Gaussian model variants to represent normal modes and detect anomaly. Its application domain are agricultural fields, which are uncluttered scenarios with mainly periodic background motion.

## III. Original Convolutional Network

The authors in [9] propose a Lenet5-like architecture to perform per-pixel classification as foreground or background. Their system takes as input the grayscale versions of both the target image and its background model, and constructs the segmentation map by independently processing patches around each pixel image. Thus, it constructs an output segmentation map of the same resolution as the input.

The CNN model is fairly simple and small: two feature stages having each one a single convolution operation followed by rectified linear units (ReLU) and max-pooling for subsampling; and two fully-connected layers at the end playing the role of a classical feed-forward neural network for classification (Figure 1). The convolutions have $5 \times 5$ kernels, stride of one and feature map size of 6 and 16, respectively; the poolings, $3 \times 3$ non-overlapping receptive fields; the fully-connected layers, 120 and 6 units, in that order. The input is a patch of $27 \times 27$ pixels in size, and the output is a single sigmoid neuron that yields the confidence (probability) of the central pixel belonging to the foreground class. The soft segmentation map generated by recursively applying the model to all image pixels is then thresholded at 0.5 to generate the equivalent binary segmentation mask.

The major drawback of this system is the wasteful usage of computational resources, since there is considerable overlap between neighboring pixels. The authors do not explore this property and if properly handled could considerably reduce computation time and/or reduce energy consumption during evaluation. Although it may be highly parallelizable to separately compute each patch, this calculation could be avoided altogether.

## IV. Proposed Convolutional Networks

Contrary to the original patch-wise approach, we calculate the whole segmentation map at once (Figure 2), taking advantage of the convolutional layers to perform the fully-connected role, i.e., we replace the latter by $1 \times 1$ convolutions. Consequently, our architecture becomes fully convolutional and able to efficiently process images of any given size in one single pass while still retaining its highly parallel nature. Furthermore, each output pixel is evaluated based on a larger region of the input, hence aggregating more information than the rather small $27 \times 27$ patch for the classification.

This modification, however, implies the reduction of the output due to down-sampling operations done throughout the network, generally either from pooling operations (Figure 1) or

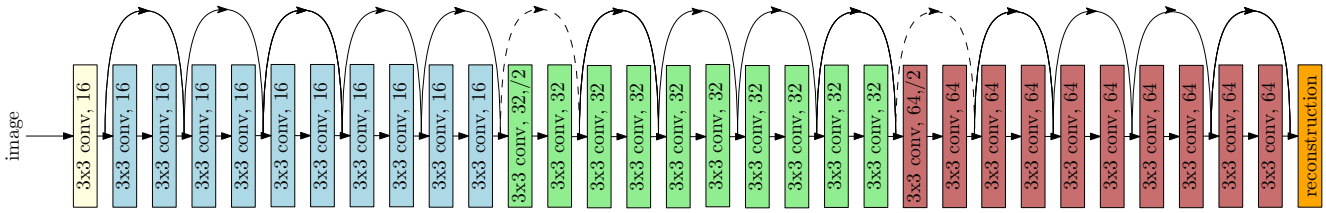


Fig. 3. Base residual network [3] from which we add the reconstruction network. There are only $3 \times 3$ convolutions and two 2-strided convolutions for spatial dimension reduction. The dashed shortcuts increase the number of feature maps at the same time they reduces resolution.

strided convolutions (Figure 3), that is, convolutions where the kernel shifts by more than one input position. In order to revert this issue and recover the original resolution, we investigate three different up-sampling options:

- Simple bilinear interpolation of the final feature map;
- Deconvolutional network with (linear/non-linear) deconvolutional layers to rebuild the feature map;
- Decoder network with interleaved bilinear interpolations and convolutions to restore the compressed feature map.

We consider two reconstruction methods with learnable parameters, each inspired by different previous works. The first, based on the DeconvNet network of Noh *et al.* [6], interleaves unpoolings, which up-samples the feature maps and places their values according to the stored pooling indexes of the max locations, with a series of stacked deconvolutions to populate those sparse maps. The second, derived from the SegNet [11], discards the intermediate fully-connected layers present in [6] and employs regular convolutions, rather than deconvolutions, to fill in the sparse maps. According to the authors in [11], the latter approach considerably reduces the network complexity (number of parameters), and eases down the training process, dispensing with additional assistance such as region proposals, which we do not resort to in any of our methods.

One may argue that the bilinear interpolation method is a mere special case of the deconvolutional network and thus could be achieved by the latter learning the appropriate set of filter values. However, the complex architecture imposed by the stack of deconvolutions makes this phenomenon difficult to happen. Hence, it is worthwhile to separately analyze the bilinear case, which is the reconstruction method of choice for the basic network in [5].

We apply the discussed modifications to the network of [9], obtaining three different models termed: **Bilinear**, **Deconv**, and **Decoder**. We further construct a variant of the **Deconv** where we do not use activation functions in-between deconvolutions of the reconstruction network (**Deconv-linear**) in order to infer how much of the deconvolution potential we are actually able to explore.

Besides the aforementioned study, we also analyze the benefits of a more modern and deeper architecture, i.e, the ResNet [13], to extract the underlying input image descriptors fed to our up-sampling structures. Residual networks have proven to be powerful architectures and won first place in several tracks in both ILSVRC [14] and COCO 2015 [15] competitions. By relying on residual models rather than on the Lenet5-based models, network representational power ought to be greater as well as more general, thus allowing better performances. As is common in classification-adapted models, we drop the final average pooling present in the original ResNet proposition since it completely removes all remaining spatial information of the already small feature map [5]. As far as our knowledge goes, this is the first work to propose a residual deep learning network for the foreground segmentation task.

Similarly to the previous case, we analyze the same combinations of reconstruction in the residual models and observe how the behavior carries over. We train direct bilinear interpolation (**Residual-Bilinear**), deconvolutional network with both non-linear activations (**Residual-Deconv**) and no activation whatsoever (**Residual-Deconv-linear**), and variants of the decoder network. Our base ResNet model (Figure 3) has 32 layers and no bottleneck architecture.

A fundamental difference in our SegNet-based residual architecture is that the encoder is implemented with strided convolutions instead of max-poolings, therefore no unpoolings can be done. This is evident since there are no indexes to be kept and later used as reference to sparsely up-sample the feature map. We tackle this issue by progressively performing bilinear interpolations instead, and setting the up-sampling factors equal to the size of the stride in down-sampling convolutions. Therefore, our residual decoder architecture (**Residual-Decoder-Deep**) interleaves bilinear up-sampling layers, which progressively recover the original resolution, with residual blocks, which are the residual equivalents of the trainable decoder filter banks in [11]. The main drawback of a decoder of same size as the encoder is the computational burden imposed, after all, network depth is doubled. Hence, we evaluate a smaller variant with single convolutions rather than residual blocks (**Residual-Decoder-Shallow**).

## V. Experimental Results

Following our inspirational work [9], we train and validate our models in the CDNET 2014 database [16], which targets change and motion detection applications. This dataset is a compilation of fixed surveillance camera videos from different sources covering a wide range of detection challenges, such as camera jitter, intermittent object motion, shadows, etc. Furthermore, it disposes of accurate pixel-level foreground segmentation, exactly what we intend the networks to learn.

We define our training and validation sets as follows: the first 70% of each video's annotated frames are for training, and the last 30% for validation. Aiming to diminish the class imbalance problem, we suppress all training frames that do not have any foreground pixel whilst preserving the validation set intact, thus keeping its representativeness.

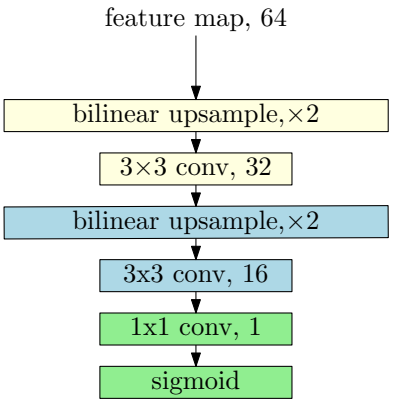


Fig. 4. Modification of the decoder network of the SegNet-derived model to employ bilinear interpolations between convolutional blocks rather than unpoolings.

We express performance of models in terms of their F1 metric, which correlates well to the average ranking of methods in CDNET according to [17], and of their false negative rate (FNR), which tends to be high due to the class imbalance and is a delicate matter in an anomaly detection system. The F1 measure represents the compromise between precision (`Prec`) and recall (`Rec`) measurements, and it is defined as their harmonic mean

$$F1 = 2\,\frac{\texttt{Prec}\times\texttt{Rec}}{\texttt{Prec}+\texttt{Rec}}\ ,$$

or, equivalently

$$F1 = \frac{2\times\texttt{true pos}}{2\times\texttt{true pos}+\texttt{false neg}+\texttt{false pos}}\ .$$

Before training, we perform some necessary preprocessing steps. We apply a median filter to the first 150 frames of each video, which do not have groundtruth mask, and are not used neither for training nor validation, to obtain the static background model employed as reference in our pipeline. Next, we extract the imposed region of interest, isolate the grayscale images of both target and reference frames, scale them down to $256\times192$ pixels, and randomly apply horizontal flip to artificially enlarge the training set.

The cost function we employ in optimization is the average binary cross-entropy over all pixel positions $i$

$$-\sum_i (t_i \ln(y_i) + (t_i - 1)\ln(1 - y_i))\ ,$$

between the target class label $t_i$ and the foreground class probability $y_i$. The solver is the Adam algorithm [18] with Torch's default moment coefficients. We train for 80 epochs with an initial learning rate of 0.01, learning rate step decay of $50\%$ each 10 epochs, regularization of 0.0002 and mini-batch size of 16. We adopt standard initialization practices for our models: the He initialization [13] for convolutional layers, bilinear interpolation filter values for deconvolutional layers, and unitary scale factor with null offset for batch normalization layers. For the original network, however, we follow the guidelines given in [9].

First, as a control test, we evaluate the performance of the architecture defined in [9] on the whole CDNET database, since the authors of [9] do not take into account some key video categories in their reported results. Because there are no references to the learning rate decay schedule nor the regularization used, we experiment with all six combinations of either no regularization or 0.0002, and no learning rate decay, $50\%$ at each 10 epochs or $90\%$ at each 30 epochs. All unmodified implementations of the original model oscillate at about 62% of F1 score in the validation set towards the end of training. We show the performance curve for one such model (**Original** in Figure 5).

TABLE I
FALSE NEGATIVE RATE AND F1-MEASURE OF THE BEST RESIDUAL RECONSTRUCTION MODELS FOR THE VALIDATION SET.

| Residual model | FNR | F1 |
|---|---|---|
| Bilinear | 11.9% | 83.9% |
| Deconv | 13.0% | 84.2% |
| Deconv-Linear | 13.7% | 84.1% |
| Decode-Deep | 16.2% | 82.4% |
| Decode-Shallow | 11.4% | 84.9% |

Not all modified versions perform better than the original model, the **Decoder** variant exhibits $\sim 8\%$ higher FNR (Figure 6) and similar F1. This behavior is explained by the removal of the intermediate fully-connected layer in that model, which has a rather harmful effect because the network was already small (6 layers). On the other hand, the deconvolutional versions converged to considerably higher F1 ($\sim 69\%$) and are the best reconstruction schemes for the original network. T he non-linear deconvolution (**Deconv**) oscillates more than its linear counterpart (**Deconv-linear**), confirming that despite having greater representational capacity, learning non-linear deconvolutions is a more difficult task. **Bilinear** interpolation yields $\sim 65\%$ of F1 score, not very different from the **Decoder**.

Next, we carry these approaches to residual architectures and observe not only a much steadier behavior, but also much better scores (Figure 7). With exception of the **Residual-Decoder-Deep**, whose best score is 82.4%, all other methods converge to values around 84% with the difference among them inferior to 1%. Then, we analyze the FNR (Table I) to determine whether there is a best model and which one it is. Even though the competition is tight, **Residual-Decoder-Shallow** achieves the highest F1 (84.9%) while having the lowest FNR (11.4%), 1% lower than the second best FNR.

Our shallow enconder-decoder network, which employs bilinear interpolation between convolutional blocks (Figure 4) rather than using the pooling indexes of the encoder network to sparsely up-sample the feature map in the decoder, has only two additional learnable layers on top of the base network and is the best performing model among the tested architectures, running under 5.1 ms per preprocessed frame on a 6-core Intel® i7 6850K with NVIDIA® GeForce® GTX 1080 at a frame resolution of $256 \times 192$ pixels. Not only the **Residual-Decoder-Shallow**, but virtually all residual reconstruction networks studied have state-of-the-art results in the CDNET2014 database, losing only to the CNN proposed in [12], whose aim is groundtruth generation of videos and not anomaly detection.

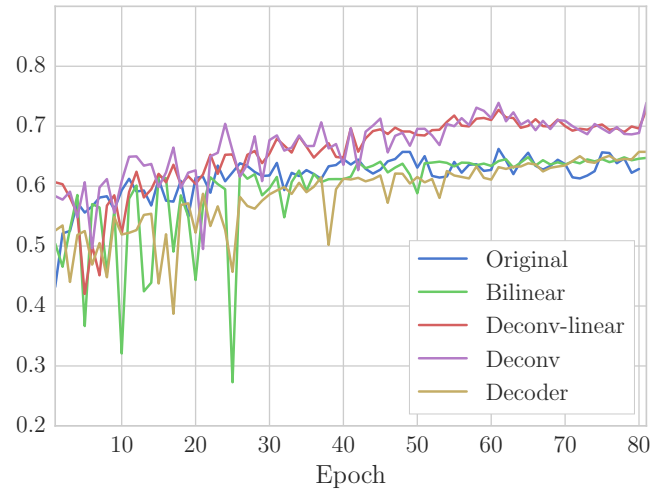


Fig. 5. F1-score of the original architecture [9] and our variants for the validation set.

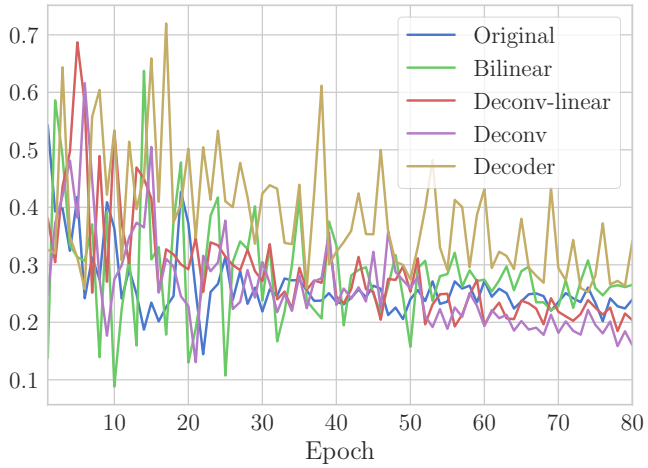


Fig. 6. False negative rate of the original architecture [9] and our variants for the validation set.

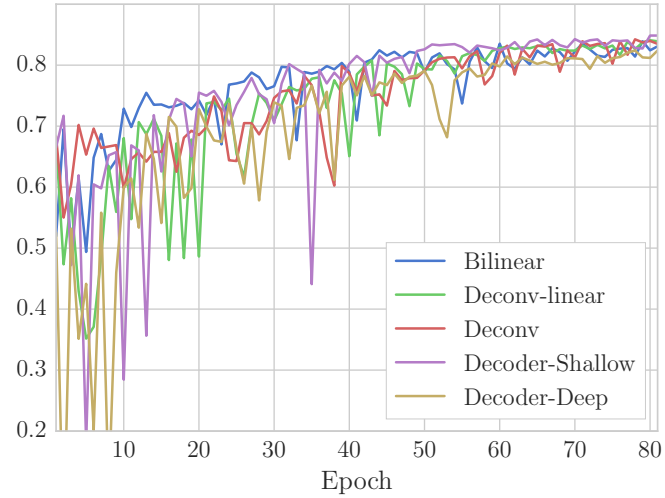


Fig. 7. F1-score of our residual models for the validation set.

## VI. Conclusions

Throughout this work we studied different deep convolutional networks to segment the foreground, regarded as anomaly with respect to the background, in videos from surveillance cameras. We proposed fast and efficient models that compute the pixel-wise segmentation map in real-time taking as input temporally aligned reference and target frames. The proposed techniques are competitive with current state-of-the-art methods for the CDNET database [16] and, according to the CDNET online ranking, seconds only to [12], which has a distinct purpose as discussed in Section II.

## References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, January 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, May 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[4] ——, "Identity mappings in deep residual networks," *preprint arXiv:1603.05027*, October 2016.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 3431–3440.

[6] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *preprint arXiv:1606.00915*, June 2016.

[8] P. Christiansen, L. N. Nielsen, K. A. Steen, R. N. Jørgensen, and H. Karstoft, "DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field," *Sensors*, vol. 16, no. 11, p. 1904, November 2016.

[9] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *International Conference on Systems, Signals and Image Processing*, Bratislava, Slovakia, May 2016, pp. 1–4. [Online]. Available: http://hdl.handle.net/2268/195180

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *preprint arXiv:1511.00561*, October 2016.

[12] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *The IEEE International Conference on Computer Vision*, December 2015.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, September 2014, pp. 740–755.

[16] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," pp. 387–394, June 2014.

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

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, July 2015.