https://doi.org/10.2352/ISSN.2470-1173.2020.8.IMAWM-114

LambdaNet: A Fully Convolutional Architecture for Directional Change Detection

Bryan Blakeslee & Andreas Savakis; Rochester Institute of Technology; Rochester, NY

Abstract

Change detection in image pairs has traditionally been a binary process, reporting either "Change" or "No Change." In this paper, we present LambdaNet, a novel deep architecture for performing pixel-level directional change detection based on a four class classification scheme. LambdaNet successfully incorporates the notion of "directional change" and identifies differences between two images as "Additive Change" when a new object appears, "Subtractive Change" when an object is removed, "Exchange" when different objects are present in the same location, and "No Change." To obtain pixel annotated change maps for training, we generated directional change class labels for the Change Detection 2014 dataset. Our tests illustrate that LambdaNet would be suitable for situations where the type of change is unstructured, such as change detection scenarios in satellite imagery.

Introduction

Detecting changes in images of the same location observed at different times is important in surveillance and security, disaster management, demographic estimation, crop monitoring, and other important applications. Change detection is a time consuming activity that was once relegated exclusively to humans, as it required a complex set of skills including object detection, contextual analysis, and semantic segmentation. Deep convolutional neural networks (CNNs) [1] [2], [3] have become the backbone of many advanced applications, such as self-driving vehicles [4], image-based medical diagnostics [5], and other tasks [6], including change detection [7], [8], [9]. Existing approaches generally fall into one of two categories: structured and unstructured techniques. Structured methods [9] are designed to detect object-level changes in image pairs. Unstructured methods [7] identify any type of "significant" change of a generic nature that goes beyond object classes. Our LambdaNet architecture attempts to bridge this gap, incorporating structured, object specific change detection, and unstructured, general change detection.

LambdaNet is a new type of Siamese change detection network, that is capable of processing directional change information, i.e., determine additive or subtractive changes. It incorporates change directionality into the output segmentation map to identify areas of an image where objects were added or removed, instead of treating them as a binary change. Furthermore, by training on data that is labelled in a class-agnostic manner, LambdaNet is able to generalize to other significant changes. An example of the LambdaNet performance for directional change detection is illustrated in Figure 1.

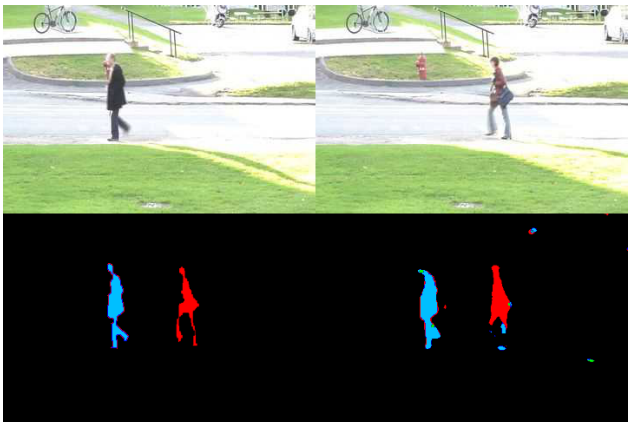


Figure 1. *Example of directional changes detected by LambdaNet. Upper Left: past image; Upper Right: present image; Lower Left: Ground Truth; Lower Right: change prediction. Red indicates additive change and blue indicates subtractive change.*

Related Work

Deep learning techniques have been used for both structured and unstructured change detection. In [7], an AlexNet architecture [1], pretrained on ImageNet [10], was utilized for unstructured change detection. First, the image pair was fed in the network and the activation maps from various layers were extracted as feature representations. Then a Euclidean distance measurement, made between the two activation map stacks, was thresholded to obtain a change map. While simple to implement, this method relies purely on Euclidean distance thresholding to determine a change, and it is void of any semantic content.

The inspiration for LambdaNet is drawn from Siamese networks, semantic segmentation, and multi-scale representations. Semantic segmentation is important to the LambdaNet architecture, as it allows for pixel-level localization of changes between image pairs. One of the first deep learning-based semantic segmentation techniques is the deconvolutional network, introduced by Noh, et al. [11], which created the concept of a fully convolutional autoencoder. The encoder portion of the network is based on the VGG-16 architecture [2], and is responsible for reducing the input image down to a feature representation. Instead of outputting a classification label for the image, the encoder output is fed into a decoder network that mirrors the VGG-16 encoder. The decoder expands its input back to the original dimensionality, assigning a class label at each pixel in the image.

This network architecture is enabled by the use of unpooling and deconvolutional layers. The unpooling layer is not utilized in LambdaNet due to architectural constraints, but the deconvo-

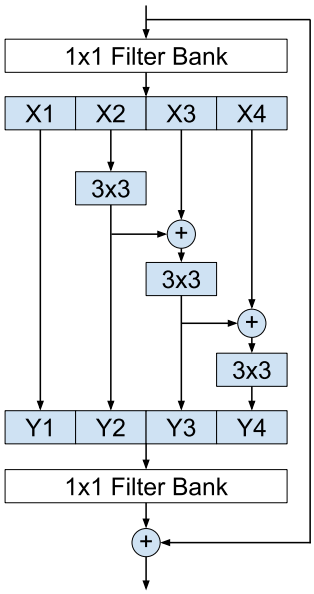


Figure 2. *Architecture of Res2Net module [14].*

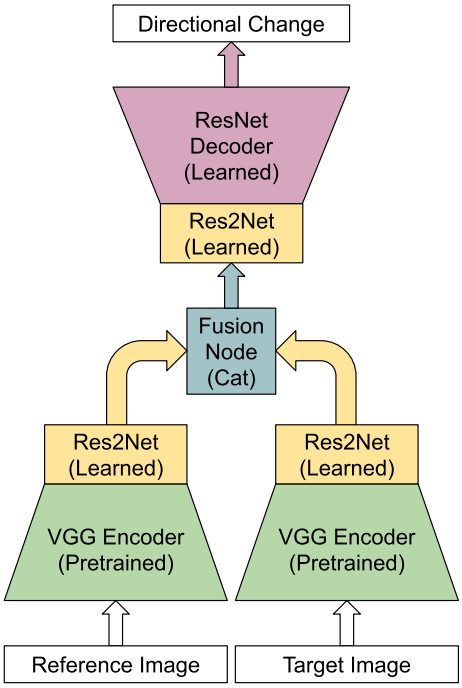


Figure 3. *High level overview of the LambdaNet achitecture.*

lutional layer plays a key role. This layer allows for the network to learn an upsampling function that expands a single pixel into a region of pixels via a learned filter.

Siamese deep neural networks were popularized by Koch, et al. [12] for one-shot learning. Siamese networks take their name from their symmetric channel construction, placing identical network components in parallel. This makes them well-suited for comparison-oriented applications, such as tracking [13]. Operating by passing two data samples to their parallel channels, Siamese networks generate a pair of intermediate outputs that serve as feature representations. These outputs can be compared to each other using a simple Euclidean distance, or a trained decision network [8]. In a more complex architecture, the decision network can be replaced with a deconvolutional decoder [9].

Recently, the Res2Net module [14] was introduced as a promising multi-scale backbone architecture that achieves improved representations. This type of multi-scale representation can be useful for change detection, as changes between image pairs can be of any size. The Res2Net architecture is shown in Figure 2. First, the input activation map is convolved with a 1×1 set of filters to yield activation map $\mathbf{X}$, which is split into $S = 4$ groups, along the channel dimension. The activation map $\mathbf{Y}$ represents an output buffer for the multi-scale convolutions represented by the kernels K. Forking and chaining kernels results in a series of receptive fields of multiple sizes sharing a common center point. Finally, the activation map $\mathbf{Y}$ is convolved with a set of 1×1 filters to yield activation map $\mathbf{Z}$, which is summed with the input map to yield the output activation map.

Methodology

The LambdaNet architecture, shown in Figure 3, incorporates Siamese style encoder channels, a feature fusion node and a decoder network. The Siamese portion of the network, shown in green, consists of a pair of VGG encoders with shared weights. These encoders are pre-trained on ImageNet, which allows for the extraction of a diverse set of features. The outputs of the encoders are passed into a Res2Net module [14], which generates feature representations at multiple scales. At the fusion node, concatenation is used to unify these two feature maps before passing them to the decoder stage. The decoder stage consists of another Res2Net layer, followed by a residual decoder, which generates the final directional change segmentation map. When training the LambdaNet architecture, the learned structures in the network are the Res2Net layers and the residual decoder.

Directional Change Detection

In our experiments, we used the Change Detection 2014 (CD2014) dataset [15], which consists of 53 videos, split across 11 categories. These videos are from RGB or thermal sensors, captured at either street level or from an elevated position in a variety of resolutions and conditions, including heavy snowfall, day and night scenes, and locations with high levels of background noise. The ground truth of CD2014 takes the form of segmentation maps, with one map corresponding to each frame of the video. The labels are class-agnostic, in the sense that each pixel is labelled as either a Change or No Change, without explicitly labelling each pixel with a class name. Secondary information is labeled in the ground truth change masks, such as a shadow region, unknown region (due to motion blur around moving objects), or "Don't Care" regions.

Figure 4 shows sample images along with their corresponding ground truth masks. In each of the truth frames, the changes are labelled with white pixels. Surrounding the target pixels is a thin border of light grey, indicating a "Don't Care" region to account for motion blur. The bottom truth image contains a large block of dark grey pixels, indicating a portion of the image outside the region of interest. In the second row truth image, the darkest

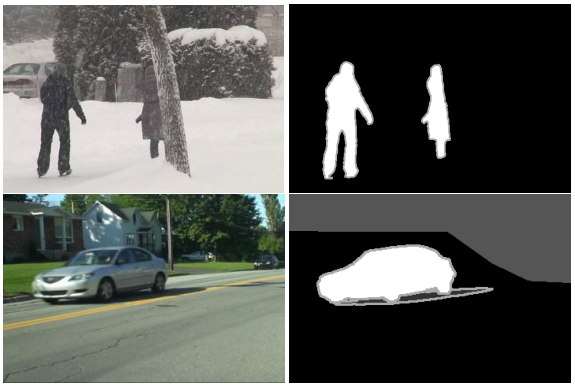


Figure 4. *Sample images from the Change Detection 2014 dataset [15]. The left column contains sample images, while the right column consists of pixel level ground truth.*

grey region marks a shadow cast by the target object. This shadow also has a thin border of "Don't Care" light grey pixels. These examples illustrate the intricacies of ground truth labeling for this dataset.

In the CD2014 dataset, change detection is considered a binary classification problem, i.e. pixels are labeled as change or no change, which does not capture directional change. Under the directional change paradigm, four classes are used to denote change at the pixel level: Additive Change, Subtractive Change, Exchange, and No Change. These classes are derived from both the per-frame pixel classification, with consideration to which frame, either past or present, the change occurs. An Additive Change class label means that the given pixel did not belong to a target in the past frame, but belongs to a target in the present frame. For a Subtractive Change class label, the situation is reversed. For the Exchange class, a different target is present at the same location. A No Change class indicates that there is no change between the past and present frame.

The generation of new ground truth for directional change detection was accomplished via a logical-OR-like operation between pairs of binary change masks. One mask was designated as the past mask, while the other was taken as the present mask. The operation described above was then applied to the image pair to obtain the fused multi-class ground truth. The source masks were then reversed and the same operation was applied.

Training

LambdaNet is trained in a similar manner to semantic segmentation networks. The primary hyperparameters of interest are the learning rate, that was set empirically to 10^{-5}, and the gradient scalars G_c for change class c, corresponding to Additive Change, Subtractive Change, and Exchange. The No Change class weight was set to one. These scalars were computed using:

$$G_c(x) = \ln\left(\frac{\sum_c \sum_p x}{\sum_p x[c]}\right) \tag{1}$$

In Equ. (1) the numerator consists of the total number of pixels in the dataset for all classes, while the denominator is the total number of pixels in change class c. A natural log is applied to this ratio to constrain the upper bound of the per-class scale factors, preventing the positive classes of highly imbalanced datasets from overwhelming the negative class. These scale factors were used in the cross-entropy loss function to enhance the gradients of the change classes and equalize the extreme imbalance of change classes with respect to the No Change class. The scaled cross-entropy loss function is shown in Equation (2).

$$L_{CE}(x,c) = G_c(-x[c] + \log(\sum_p \exp(x[p]))) \tag{2}$$

During training, the input image pairs are applied to the network in both "directions" to avoid bias during training. This means that first, the past image is shown to the left input and the present image shown to the right input, and then the inputs are reversed. This type of training was done to prevent the decoder from overfitting to changes in one direction. It also allowed the network to observe both change "directions" equally, while doubling the number of Exchange samples, which is the least common class. Sample training triplets are shown in Figure 5.

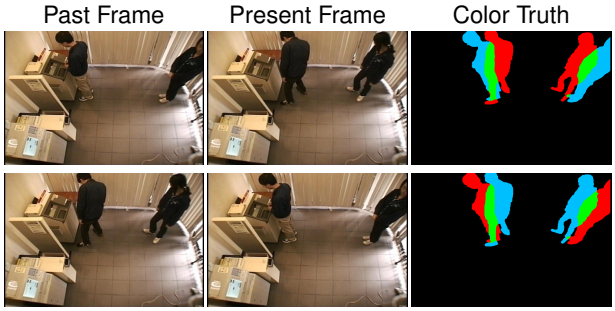


Figure 5. *Sample colorized triplet entries for multi-class change detection. The left column is the past frame, the center column is the present frame, and the right column is the fused ground truth map. Best viewed in color. Red indicates Additive Change, blue indicates Subtractive Change, green indicates Exchange and Black indicates No Change.*

Results and Discussion

Evaluation of LambdaNet focused on directional change detection by training via cross-validation on random data splits. Table 1 shows results for LambdaNet trained for directional change detection and LambdaNet operating in binary change detection mode, trained on the same random dataset splits.

Directional change detection results are shown in Table 1. These results were generated by treating each directional change class as a one versus all binary classification problem. For each case, the chosen metrics were recall, precision, F1-Score, and the mean intersection over union (mIOU). Recall measures the fraction of correctly marked change pixels, while precision calculates what fraction of pixels that were predicted as changes were actually true changes. The F1-Score combines the precision and recall metrics as a harmonic mean to give a single measure for the change class of interest. The mean intersection over union gives a raw number of the quality of the segmentation regions compared to the ground truth regions.

Inspection of Table 1 shows that the Additive and Substractive Change metrics compare favorably with the binary LambdaNet's version of change detection. This is especially true for the F1 score and precision. Furthermore, the similarity of the AC and SC metric scores indicate that LambdaNet has learned both

Table 1. Results comparing the directional change LambdaNet to the binary change LambdaNet. The Binary (BC) class row shows results for the binary version of LambdaNet. The Average row shows the average of the directional LambdaNet results for Additive (AC), Subtractive (SC), and Exchange (EC) classes.

Change Class	Recall	Precision	F1-Score	Mean IoU
Binary (BC)	0.7292	0.6395	0.6682	0.5137
Additive (AC)	0.5940	0.6447	0.6150	0.4504
Subtractive (SC)	0.6329	0.6399	0.6288	0.4660
Exchange (EC)	0.4217	0.1955	0.2452	0.1556
Average	0.5495	0.4934	0.4963	0.3573

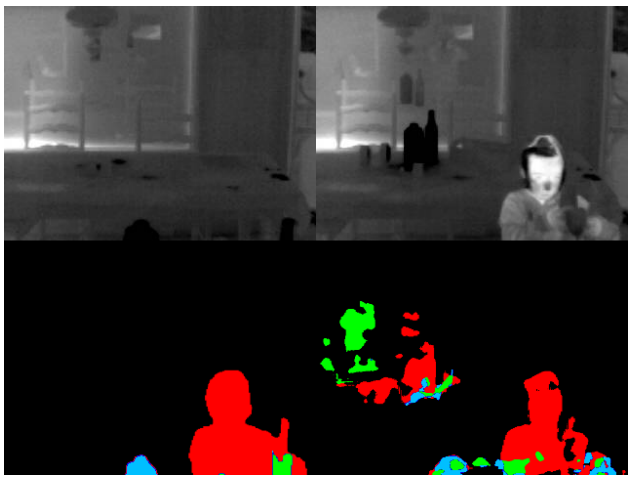


Figure 6. *Sample frames from video of a child removing bottles from a table. Upper left is past frame; Upper right is present frame; Lower left is Ground Truth; Lower right is change prediction.*

of these directional classes equally well. On the other hand, the EC class performs poorly, indicating that the directional change scheme has difficulty identifying object swaps. An exchange can be a replacement of a change target with another target. However, this type of replacement can also occur in the background No Change class. As a result, the network is left to "guess" at the nature of the change, and this results in errors. The network may occasionally identify an unstructured change in the background as an Exchange, and thus, the primary source of accuracy degradation is the EC class, due to these unstructured changes.

Figure 6 illustrates unstructured directional changes in images analyzed with LambdaNet. Since the change marks indicate differences between the image pairs, the LambdaNet decoder learns a separation function that ignores common elements between images and only highlights their differences. Figure 6 shows a video of a child clearing a table, with unstructured changes for the bottles that are detected by LambdaNet even though they are not present in the ground truth. It also appears that while the desired unstructured behavior is present, it is not strongly associated with a particular change class, as evidenced by the mix of classes used to mark the moved bottles.

Directional change may be viewed as a refined version of the traditional binary change detection. As can be seen in Figure 7, when the three color directional change mask is flattened into a single channel, the result is very similar to the binary change case. Quantitatively, the mean square error between the flattened directional change mask and the binary mask is 1.21.

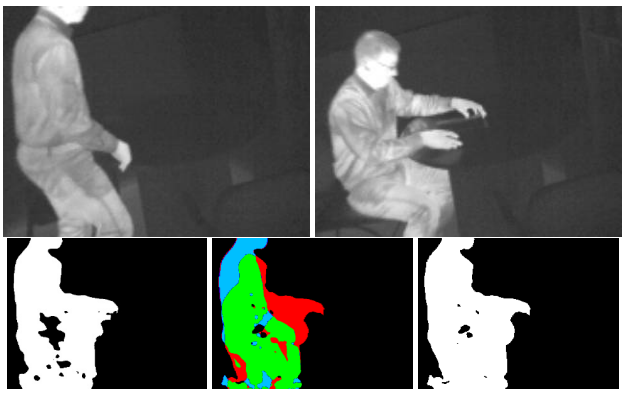


Figure 7. *Comparison of binary and directional change masks. Top row consists of past and present frames. Bottom row shows, from left to right, the binary change, directional change, and binary change after flattening the directional change mask.*

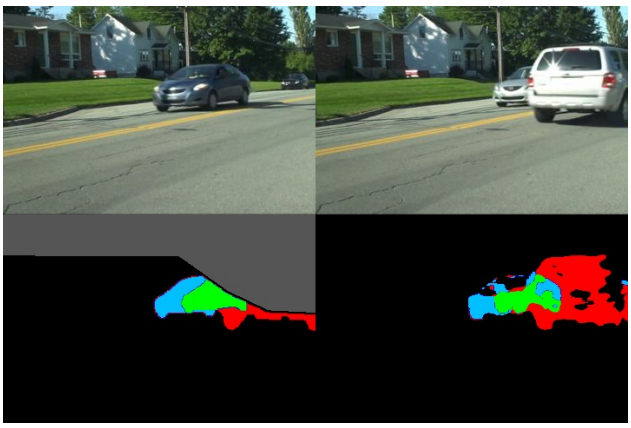


Figure 8. *Example of directional change where the ground truth contains a significant "Don't Care" region shown in gray. Upper left is past frame; Upper right is present frame; Lower left is Ground Truth; Lower right is change prediction.*

The most significant issue in evaluating LambdaNet is that its unstructured behavior is often penalized in the metric scores as a false positive. In many instances, this is due to the way the ground truth is labelled in the CD2014 dataset. Several videos have large "Don't Care" regions where otherwise valid change targets are masked as "No Change" areas. In other cases, some change targets are only partially segmented, leading to incomplete learned representations. This is particularly visible in the ground truth image in Figure 8 and yields a lower quality segmentation

Figure 9. *Example with strong illumination and small targets, showing cars driving off a highway exit at night. Upper left is past frame; Upper right is present frame; Lower left is Ground Truth; Lower right is change prediction.*

for the structured change of the vehicles.

Limitations of LambdaNet include that it is sensitive to intense changes in illumination and prone to errors for very small targets. The sensitivity to illumination changes is likely related to the network's ability to detect unstructured changes. Figure 9, shows an failure example where change detection is marked due to bright illumination. This occurs frequently in the "Night Videos" category, due to extreme glare from car headlights or streetlights. LambdaNet also struggles to identify changes between objects that are extremely small and/or occluded, also shown in Figure 9. These are the situations that a human would also find the most difficult.

Lastly, LambdaNet's performance (in binary mode) was compared with that of a purely unstructured version of change detection that relied exclusively on thresholding the encoder network representations. Under this paradigm, a pair of images were passed through the dual encoders and their activation maps were subtracted from each other, after bilinear interpolation to the original image size. The difference map was summed in the channel dimension and thresholded to produce a binary change map. Sample outputs are shown as Figures 10 and 11.

In Figure 10, the threshold was set manually to obtain the best result. In this case, both segmentations, obtained using threshold and the binary LambdaNet, are qualitatively very similar. However, the thresholding method fails when the background is dynamic, as shown in Figure 11. The threshold-based technique marks background changes as significant changes, while LambdaNet rejects such background noise. This is because LambdaNet not only learns what a valid change consists of, it also learns what changes are insignificant and rejects those in the final prediction.

Conclusion

We presented LambdaNet, a fully convolutional network consisting of Siamese encoders with multi-scale modules, a fusion node and a decoder network, that was able to successfully identify additions and removals of target objects from pairs of input images. Additionally, LambdaNet displays ability to identify unstructured changes without explicit labels. This allows the network to identify changes in image pairs that were not originally

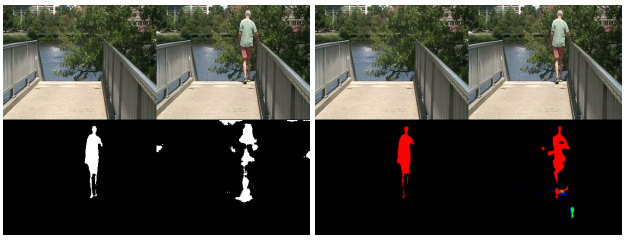


Figure 10. *Comparison of threshold results (left) and LambdaNet results (right) for binary change detection with a change target present. For each side: Upper left is past frame; Upper right is present frame; Lower left is Ground Truth; Lower right is change prediction.*

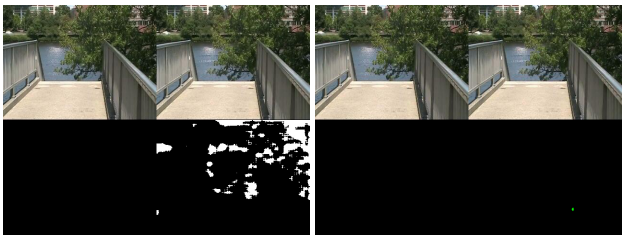


Figure 11. *Comparison of threshold results (left) and LambdaNet results (right) for binary change detection with dynamic background. For each side: Upper left is past frame; Upper right is present frame; Lower left is Ground Truth; Lower right is change prediction.*

labelled in the ground truth, making LambdaNet suitable for applications where prior knowledge of the environment may be limited.

Acknowledgement

This research was supported in part by the Air Force Office of Scientific Research (AFOSR) under Dynamic Data Driven Applications Systems (DDDAS) grant FA9550-18-1-0121, Kitware Inc., and the Center for Emerging and Innovative Sciences (CEIS), an Empire State Development-designated Center for Advanced Technology. The authors would also like to thank Sid Pendelberry and RIT Research Computing for their support in the completion of this work.

References

[1] Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton, ImageNet Classification with Deep Convolutional Neural Networks, "Proceedings of the 25th International Conference on Neural Information Processing Systems," Volume 1, pg. 1097, 2012.

[2] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," International Conference on Learning Representations, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition," CoRR, abs/1512.03385, 2015.

[4] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving," IEEE Intelligent Vehicles Symposium (IV), 2018.

[5] Olaf Ronneberger, Philipp Fischer and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," International Conference on Medical Image Computing and Computer-Assisted Intervention, pg. 234, 2015.

[6] Justin Johnson and Alexandre Alahi and Fei-Fei Li, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," European Conference on Computer Vision, 2016.
[7] Arabi Mohammed El Amin, Qingjie Liu, and Yunhong Wang, "Convolutional neural network features based change detection in satellite images," SPIE First International Workshop on Pattern Recognition, pg. 181, 2016.
[8] F. Rahman, B. Vasu, J. V. Cor, J. Kerekes and A. Savakis, "Siamese network with multi-level features for patch-based change detection in satellite imagery," IEEE Global Conference on Signal and Information Processing, pg. 958, 2018.
[9] Caye Daudt,B. Le Saux and A. Boulch, "Fully Convolutional Siamese Networks for Change Detection," IEEE International Conference on Image Processing (ICIP), 2018.
[10] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and Fei-Fei Li, "ImageNet: A Large-Scale Hierarchical Image Database," Computer Vision and Pattern Recognition, 2009.
[11] Hyeonwoo Noh, Seunghoon Hong and Bohyung Han, "Learning Deconvolution Network for Semantic Segmentation," International Conference on Computer Vision, pg. 1520, 2015.
[12] Gregory Koch, Richard Zemel and Ruslan Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," International Conference on Machine Learning, 2015.
[13] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional Siamese networks for object tracking," European Conference on Computer Vision, pp. 850–865, Springer, 2016.
[14] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xinyu Zhang, Ming-Hsuan Yang, and Philip H. S. Torr, "Res2Net: A New Multi-scale Backbone Architecture, IEEE Transactions on Pattern Analysis and Machine Intelligence," 2019.
[15] N. Goyette, P. Jodoin, F. Porikli, J. Konrad and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012.

Author Biography

Bryan Blakeslee received dual BS degrees in Electrical Engineering and Computer Engineering from the Rochester Institute of Technology (2014). He worked at both SpaceX (Hawthorne, CA) and HRL Laboratories (Malibu, CA), before obtaining an MS in Computer Engineering from RIT (2019). His interests are in the development of novel deep learning architectures and their applications in the embedded systems domain.

Andreas Savakis is Professor of Computer Engineering and Director of the Center for Human-aware Artificial Intelligence (CHAI) at Rochester Institute of Technology (RIT). He received his Ph.D. in Electrical and Computer Engineering from North Carolina State University. Prior to joining RIT, he was with Kodak Research Labs. His research interests include computer vision, deep learning, machine learning, domain adaptation, object tracking, human pose estimation, and scene analysis. Dr.Savakis has coauthored over 120 publications and is co-inventor on 12 U.S. patents.

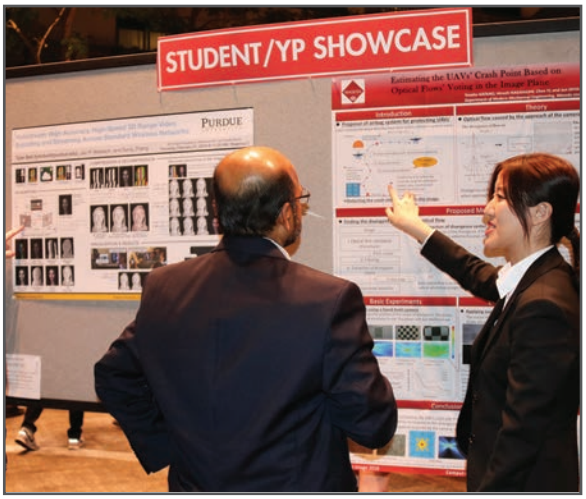
JOIN US AT THE NEXT EI!

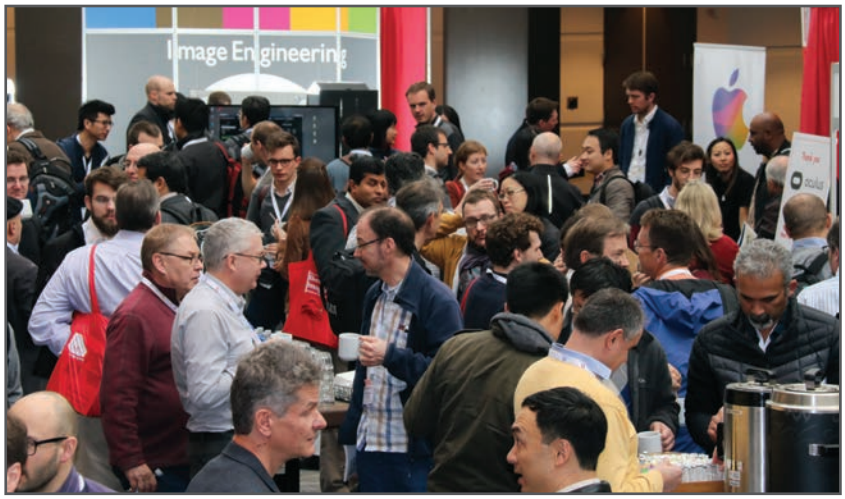
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org