https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-213

Evaluation of semi-frozen semi-fixed neural network for efficient computer vision inference

Chyuan-Tyng Wu, Peter van Beek, Phillip Schmidt, Joao Peralta Moreira, and Thomas R. Gardos
Intel Corporation; Santa Clara, California, USA

Abstract

Deep neural networks have been utilized in an increasing number of computer vision tasks, demonstrating superior performance. Much research has been focused on making deep networks more suitable for efficient hardware implementation, for low-power and low-latency real-time applications. In [1], Isikdogan et al. introduced a deep neural network design that provides an effective trade-off between flexibility and hardware efficiency. The proposed solution consists of fixed-topology hardware blocks, with partially frozen/partially trainable weights, that can be configured into a full network. Initial results in a few computer vision tasks were presented in [1]. In this paper, we further evaluate this network design by applying it to several additional computer vision use cases and comparing it to other hardware-friendly networks. The experimental results presented here show that the proposed semi-fixed semi-frozen design achieves competitive performance on a variety of benchmarks, while maintaining very high hardware efficiency.

Introduction

Rapid progress in computing technology has led to breakthroughs in numerous computer vision tasks via convolutional neural networks (CNN) in recent years. As the trend continues, deeper and heavier networks are being developed in order to pursue superior performance. At the same time, there is a need to deploy neural network inference on edge devices. However, unlike cloud servers, edge devices usually do not have sufficient computational resources to run massive networks, or would suffer from high latency and overwhelming power consumption. To overcome those obstacles, much research has been conducted on how to efficiently utilize deep networks on edge devices, like various consumer electronics products. Such research includes methods to prune weights in neural networks to reduce the size, as well as innovative topologies or data flows of convolution layers to make the network more efficient. In addition, hardware accelerators are being developed in industry, improving the power-performance trade-off significantly.

The authors in [1] proposed a fixed-topology neural network with partially frozen weights, named SemifreddoNets, that is optimized for hardware with respect to silicon area, memory requirements and power consumption. The basic SemifreddoNets topology is illustrated in Figure 1. While the proposed semi-fixed design is very different from generic neural net accelerators, it is intended to be applicable to a variety of computer vision tasks and applications. In particular, the hardware-efficient design is intended to be a good fit for the automated driving application (including front, rear and in-cabin camera use cases), where multiple vision tasks need to be executed simultaneously and in real-time.

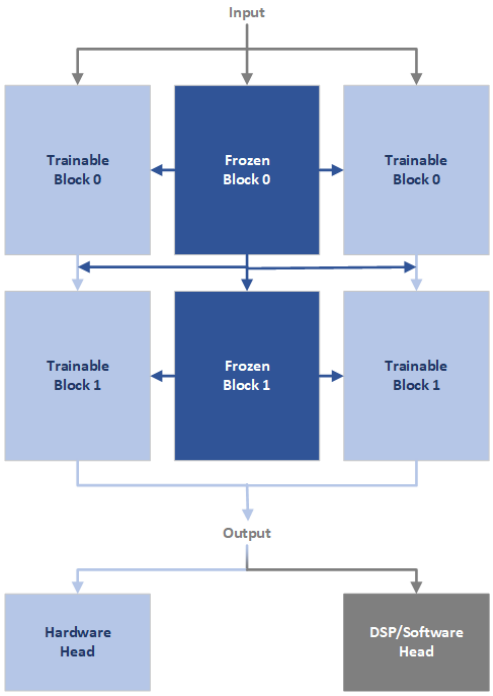


Figure 1: The illustration of basic SemifreddoNets

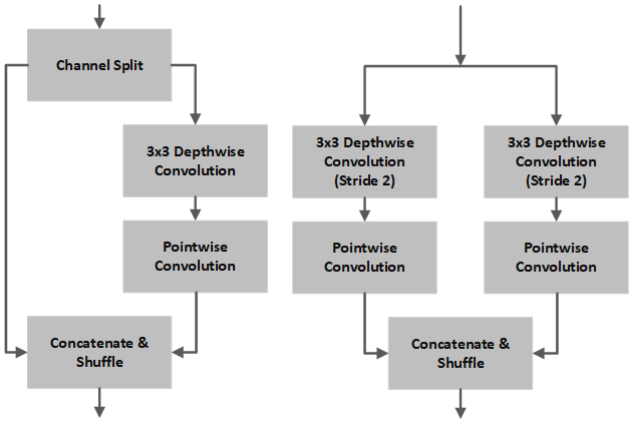
Therefore, in this work, we intend to evaluate the accuracy performance of this network and to demonstrate whether it indeed can reach satisfying performance in multiple important tasks, including pedestrian detection, pedestrian action recognition, semantic segmentation, drivable road area segmentation, and facial landmark detection. In the next section, we provide an overview of the SemifreddoNets design, as well as its configurations for specific use cases.

SemifreddoNets Overview

SemifreddoNets are fixed-topology networks composed with several building blocks. Each building block is a sub-net constructed from two kinds of subblocks as shown in Figure 2. The *basic subblock* is used to extract more abstract features but maintain the same feature dimensions while the *downscaling subblock* doubles the number of channels and downscales the activations by two in both dimensions. The detailed design of these building blocks are listed in Table 1.

Most weights in frozen blocks, except the parameters for batch normalization, are hard-wired in the silicon in order to save die size and computational complexity. The weights in frozen blocks are pre-trained for image classification using the ImageNet

dataset [2], such that the frozen blocks can provide general elementary features to other use cases. On the other hand, the weights of trainable blocks are fully flexible and can be loaded dynamically, based on offline training results for different computer vision tasks. As indicated by the arrows in Figure 1, features from frozen blocks can be passed on to trainable blocks, but not the other way around. In other words, trainable blocks benefit from the computation performed in frozen blocks, while features computed by trainable blocks will not impact the computation in frozen blocks. With the objective of controlling the contribution of the features from frozen blocks, there is a set of trainable per-channel parameters that determine the blending factor between activations from trainable blocks and activations from frozen blocks. Hence, if the frozen blocks are not beneficial to a specific use case, the network can solely use the features from trainable blocks. The property gives SemifreddoNets more flexibility in adapting to manifold computer vision tasks.



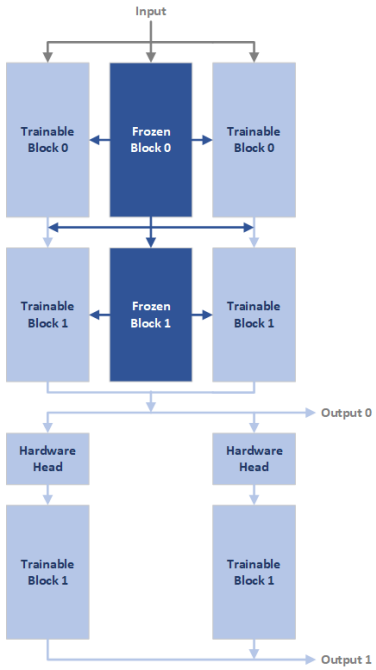
(a) Basic subblock (b) Downscaling subblock

Figure 2: Layer components in SemifreddoNets building subblocks

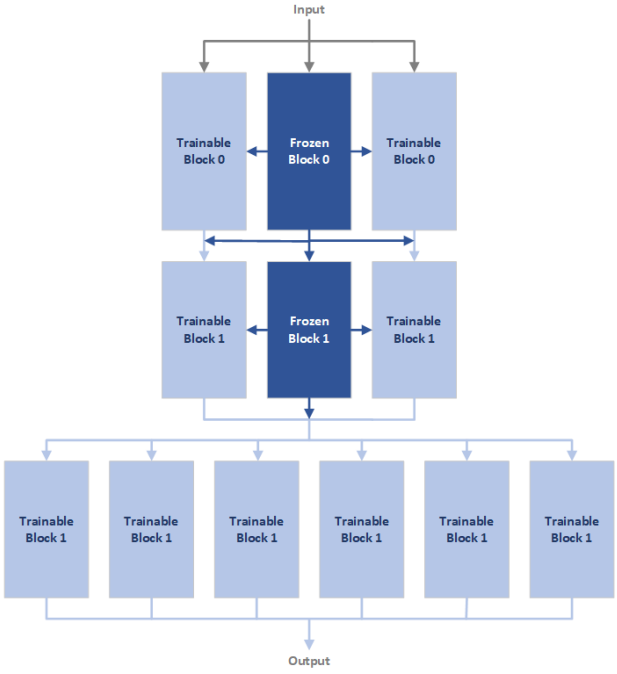
In general, SemifreddoNets itself is mainly used as the feature extraction backbone, and it is concatenated with a task-specific head to tailor it to specific classification or regression tasks. Even though the topology uses fixed-function sub-nets, the system can still be configured to create a deeper or wider network, depending on the requirements and the characteristics of the application. This adaptation is achieved by conceptually repeating trainable block 1 from the basic network multiple times and defining an optimal flow over the repeated blocks. Two examples of this concept are shown in Figure 3. It should be noted that additional iterations over such blocks may increase the overall latency and reduce the hardware efficiency. In hardware, the repetition is done by storing the output activations and re-running the same fixed-function block loaded with different weights.

In addition to utilizing the last output activation as the feature map, all intermediate output activations from each iteration can optionally be concatenated and used as multiscale feature maps. That is an important capability when multiscale feature maps are required in several use cases to boost the performance, such as the object detection network in the Single Shot MultiBox Detector (SSD) [3].

The extracted features should be decoded into meaningful task-specific information, and this is implemented by the head network. In some applications, this can be done using a simple 1×1 convolution layer and implemented by the hardware head block. For others, a more complicated decoder head network is essential to reach optimal performance. For instance, a Pyramid



(a) Deeper configuration with multiscale features



(b) Wider configuration

Figure 3: SemifreddoNets topology configuration options

Table 1: Components of SemifreddoNets building blocks

Block	Layer / Subblock	Input Size	# in Frozen Block	# in Trainable Block
Block 0	3×3 Convolution	$640 \times 480 \times 3$	1	1
	Downscaling subblock	$320 \times 240 \times 32$	1	1
	Basic subblock	$160 \times 120 \times 64$	3	0
	Downscaling subblock	$160 \times 120 \times 64$	1	1
	Basic subblock	$80 \times 60 \times 128$	3	0
Block 1	Basic subblock	$80 \times 60 \times 128$	4	1
	Downscaling subblock	$80 \times 60 \times 128$	1	1
	Basic subblock	$40 \times 30 \times 256$	3	0
Head	1×1 Convolution (+ Optional pooling)	$40 \times 30 \times 256$	0	1

Scene Parsing (PSP) head [4] can be used with SemifreddoNets for semantic segmentation. In this case, the hardware-based network would communicate with a CPU or a DSP to complete the full flow.

Use Case Evaluations

Using the SemifreddoNets semi-frozen semi-fixed neural network, promising accuracy performance for image classification and face identification was shown in [1]. In this work, the same topology concept is applied to additional computer vision tasks. We intend to demonstrate the potential of SemifreddoNets used in a wide range of applications in terms of both accuracy performance metrics and hardware efficiency.

Pedestrian detection

We first report results on the pedestrian detection application. In this experiment, the JAAD dataset [5] is used for both training and evaluation. Note that pedestrians can be relatively small in the field of view and become challenging to detect. To overcome this difficulty, a multiscale approach was taken, where feature maps from finer resolution layers can be used to detect smaller pedestrians in the scene, using additional detection heads. The detection head at each resolution level is a simple 1×1 convolutional layer, as described in [3].

We found that SemifreddoNets with only 2 iterations can outperform a much larger VGG16-like backbone network [6] in terms of log average miss rate. The performance can be further enhanced using more iterations as suggested by the results in Table 2. Detection examples are shown in Figure 4, where blue boxes are the ground truth and red boxes are the detection results with detecting confidence scores.

Table 2: Pedestrian detection experimental results on JAAD validation dataset

Topology	Output scales	Log Average Miss Rate
SemifreddoNets with 2 iterations and 2-scaled outputs	x16, x32	0.432
SemifreddoNets with 3 iterations and 3-scaled outputs	x8, x16, x32	0.391

Figure 4: Pedestrian detection experimental examples on JAAD dataset

Pedestrian action recognition

In addition to the pedestrian bounding box annotations, the JAAD dataset also provides the label of pedestrian actions. In this experiment, we trained the network to classify the pedestrian in a video into two action classes: walking/running or standing/background. Unlike normal recognition tasks, temporal information is required to identify the motion of pedestrians. To extract the time-domain features, the gated recurrent unit (GRU) [7] is used and connected to the end of network. The time-domain processing can be done using a DSP.

For both training and evaluation, it is assumed that action recognition occurs after pedestrian detection and tracking, i.e. the series of bounding box crops corresponding to the same pedestrian over time are provided to the network, and the cropped images are resized to 112×80 and fed to the network. The table below shows that, compared with the reference network LT-RCNN [8], SemifreddoNets can achieve competitive performance using the basic topology. When switching to a wider configuration, SemifreddoNets can further exceed the reference network in both metrics.

Table 3: Pedestrian action recognition experimental results on JAAD validation dataset

Topology	Sample Accuracy	Mean Accuracy
LT-RCNN	0.867	0.641
SemifreddoNets basic	0.855	0.675
SemifreddoNets wider configuration	0.874	0.696

Semantic segmentation

We utilized the Cityscapes dataset [9] to evaluate the accuracy performance of SemifreddoNets on the semantic segmentation. In this use case, backbone extractor SemifreddoNets outputs the downscaled feature maps, and the subsequent head network digests those features and classifies each pixel as belonging to which object class. Therefore, unlike other use cases, the requirement for the segmentation application is to provide per-pixel results, instead of application metadata. In practice, the network outputs a segmentation map at a lower resolution than the input image (due to downscaling in the backbone), and the full resolution output is achieved by bilinear interpolation.

Based on the experimental results shown in Table 4, the semantic segmentation performance can benefit from the additional wider layers and a dedicated decoder head network, such as PSP [4]. Compared with other compact neural networks, such as SkipNet [10], SemifreddoNets can provide competitive accuracy with a much smaller number of weights. Figure 5 shows some segmentation results using SemifreddoNets.

Table 4: Semantic segmentation experimental results on Cityscapes validation dataset

Topology	# Parameters in Backbone (Million)	Mean IOU (%)
SkipNet with MobileNet	4.3	61.5
SkipNet with ShuffleNet	1.8	55.5
SemifreddoNets with simple head	0.91	50.4
SemifreddoNets with PSP head	0.91	59.1
SemifreddoNets (5-iteration) with MSP* head	1.2	61.1

*MSP: multiscale pooling (3 scales)

Drivable road segmentation

The purpose of drivable road segmentation is to identify the area of the road where automated vehicles can proceed. Different from normal semantic segmentation of roads, it additionally needs to detect lanes and to distinguish between the primary lane, i.e. the lane that the vehicle is currently driving on, and alternative lanes, i.e. the lane that the vehicle can change into. Also, the lanes in the opposite direction should be classified as non-drivable. For this task, we used BDD100K dataset [11] for both training and evaluation.

Figure 5: Semantic segmentation experimental examples on Cityscapes dataset

In this use case, the multiple-scale configuration with 5 iterations and 3 scales is chosen, and the multiscale pooling module with global averaging pooling is used in the segmentation head, followed by the bilinear upscaling. As shown in Table 5, when comparing to the reference network, DLA-34 [12], SemifreddoNets can achieve better accuracy in all metrics with much less parameters, which translates to higher frame-rate and lower power consumption. Road segmentation example images are provided in Figure 6, where the primary drivable lanes are labeled in red and the alternative drivable lanes are labeled in blue.

Figure 6: Drivable road segmentation experimental examples on BDD100K dataset

Facial landmark detection

For facial landmark detection, we trained SemifreddoNets to detect the locations of 5 landmarks, i.e. the centers of the two eyes, the center of the nose, and the left and right corners of the mouth, as the red marks depicted in Figure 7. The network was trained and evaluated on the AFLW dataset [13].

According to the experimental results in Table 6, the basic single-iteration SemifreddoNets, followed by the detection head composed of a flatten layer and a dense layer, can provide reasonable accuracy when compared with the state-of-the-art MTCNN

Table 5: Drivable road segmentation experimental results on BDD100K validation dataset

Topology	# Parameters in Backbone (Million)	Class mIOU (%)			Mean IoU Drivable (%)	Mean IoU All (%)
		Non-drivable	Primary drivable	Alternative drivable		
DLA-34	15	N/A	73.1	55.4	64.2	N/A
SemifreddoNets	1.2	95.7	76.0	60.5	68.2	77.4

[14] in terms of normalized error percentage.

Table 6: Facial landmark detection experimental results on AFLW validation dataset

Topology	Normalized Error (%)
MTCNN	6.9
SemifreddoNets	8.6

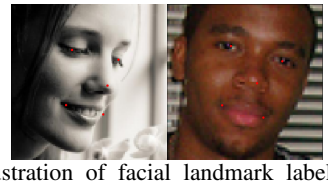


Figure 7: Illustration of facial landmark labels from AFLW dataset

Conclusions

In this work, we evaluated SemifreddoNets, proposed in [1] as a fixed-function neural net for hardware-efficient inference, in multiple applications related to automated/autonomous driving: pedestrian detection, pedestrian action recognition, semantic segmentation, drivable road area segmentation, and facial landmark detection. Serving as a feature extractor, it can be configured for all aforementioned applications, and achieve acceptable accuracy, and approach or even exceed the accuracy performance of reference networks of much larger size. Considering the significant benefits of small network size, low power and low latency, SemifreddoNets is a highly competitive approach toward achieving efficient real-time inference.

References

[1] Leo Isikdogan, Bhavin Nayak, Chyuan-Tyng Wu, Joao P. Moreira, Sushma Rao and Gilad Michael, "SemifreddoNets: Partially Frozen Neural Networks for Efficient Computer Vision Systems", Proceedings of the European Conference on Computer Vision (2020)

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, "ImageNet: a Large-Scale Hierarchical Image Database", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C. Berg, "SSD: Single Shot MultiBox Detector", Proceedings of the European Conference on Coputer Vision (2016)

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang and Jiaya Jia, "Pyramid Scene Parsing Network", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

[5] Amir Rasouli, Iuliia Kotseruba and John K. Tsotsos, "Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior", Proceedings of the IEEE International Conference on Computer Vision Workshops (2017)

[6] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv preprint (2014)

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", Proceedings of the Conference on Empirical Methods in Natural Language Processing (2014)

[8] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko and Trevor Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth and Bernt Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)

[10] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani and Martin Jagersand, "RTSeg: Real-time Semantic Segmentation Comparative Study", Proceedings of the IEEE Conference on image processing (2018)

[11] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan and Trevor Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2020)

[12] Fisher Yu, Dequan Wang, Evan Shelhamer and Trevor Darrell, "Deep Layer Aggregation", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2018)

[13] Martin Koestinger, Paul Wohlhart, Peter M. Roth and Horst Bischof, "Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization", Proceedings of First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)

[14] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li and Yu Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks", IEEE Signal Processing Letters, vol. 23, no. 10 (2016)

Author Biography

Chyuan-Tyng Wu received the Ph.D. in Electrical and Computer Engineering from Purdue University. His research was focused on computer vision and depth information capture systems. He is now working as an Algorithm Engineer in the Imaging and Camera Technologies Group at

Intel. His work includes multiple fixed function developments for imaging signal processors and emerging IP incubations for computational photography and the machine learning applications.

Peter van Beek is a senior algorithms engineer with the Intel Imaging and Camera Technologies Group. He is supporting development of deep learning inference engines as well as optimization of camera imaging pipelines. Previously, he was with the Intel Autonomous Driving Group and Mobileye. Before joining Intel, Peter was a technical lead at Sharp Laboratories of America. He is a senior member of the IEEE and received a PhD from Delft University of Technology.

Phillip Schmidt received his Dipl.-Ing. in Automation Technology from the University of Stuttgart (2013). He had worked at the German Aerospace Center (DLR), Institute of Robotics and Mechatronics in Wessling, Germany, from 2014 to 2017. Since then he has worked at Intel Corp. in Santa Clara, CA, where his work has focused on machine learning, robotics and SLAM.

Joao Moreira received his MS in Electrical Engineering from the University of Aveiro (2015) with a thesis developed at Intel Netherlands. Once graduated, he joined Intel as a full time Software engineer, working on simulation models for Intel's Imaging and Camera Group. Since 2019 Joao is a Deep Learning Software Engineer at the Advanced Solutions R&D team. His work is focused on Software Development supporting Deep Learning Vision research.

Thomas Gardos is a Principal Engineer and manager of an imaging and computer vision R&D team at Intel. Previously, he managed Intel's video compression algorithm team and represented Intel in video standards bodies. Before joining Intel, Tom was in the Imaging Research Labs at Eastman Kodak and a graduate of Kodak's 2-year Imaging Scientist training program. Tom received his MS and PhD in Digital Image and Signal Processing from Georgia Tech and his BSEE from University of Delaware.

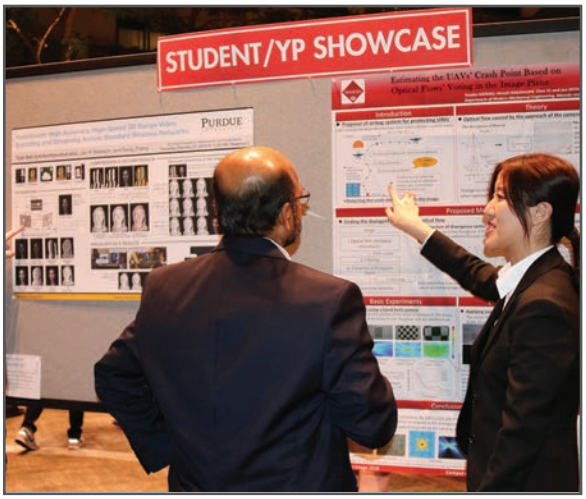
JOIN US AT THE NEXT EI!

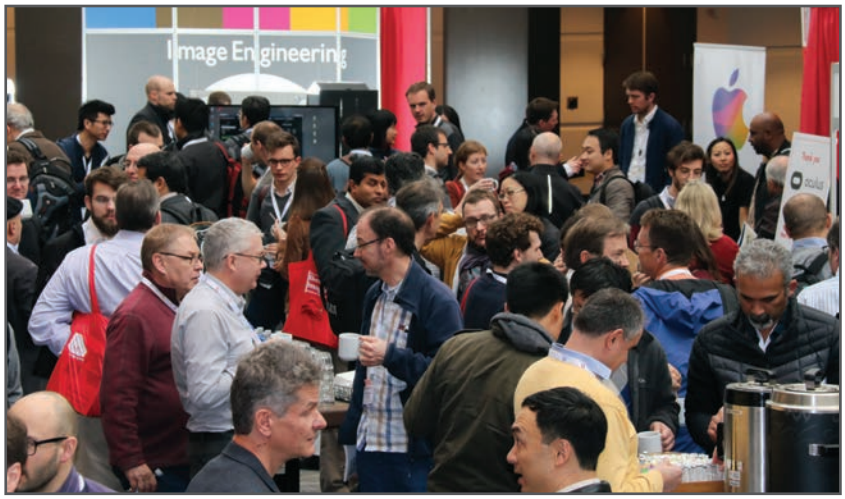
IS&T International Symposium on

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