https://doi.org/10.2352/ISSN.2470-1173.2020.12.FAIS-172

Deep Learning based Fruit Freshness Classification and Detection with CMOS Image sensors and Edge processors

Tejaswini Ananthanarayana[1,2]***, Raymond Ptucha***[1]***, Sean C. Kelly***[2]***;***
[1] ***Rochester Institute of Technology, Rochester, New York, USA;***
[2] ***ON Semiconductor, Rochester, New York, USA***

Abstract

CMOS Image sensors play a vital role in the exponentially growing field of Artificial Intelligence (AI). Applications like image classification, object detection and tracking are just some of the many problems now solved with the help of AI, and specifically deep learning. In this work, we target image classification to discern between six categories of fruits – fresh/ rotten apples, fresh/ rotten oranges, fresh/ rotten bananas. Using images captured from high speed CMOS sensors along with lightweight CNN architectures, we show the results on various edge platforms. Specifically, we show results using ON Semiconductor's global-shutter based, 12MP, 90 frame per second image sensor (XGS-12), and ON Semiconductor's 13 MP AR1335 image sensor feeding into MobileNetV2, implemented on NVIDIA Jetson platforms. In addition to using the data captured with these sensors, we utilize an open-source fruits dataset to increase the number of training images. For image classification, we train our model on approximately 30,000 RGB images from the six categories of fruits. The model achieves an accuracy of 97% on edge platforms using ON Semiconductor's 13 MP camera with AR1335 sensor. In addition to the image classification model, work is currently in progress to improve the accuracy of object detection using SSD and SSDLite with MobileNetV2 as the feature extractor. In this paper, we show preliminary results on the object detection model for the same six categories of fruits.

Introduction

Image classification and object detection are two of the many applications of AI widely used in most of all the industries today. The choice of CMOS (Complementary Metal Oxide Semiconductor) sensor, CNN (Convolutional Neural Network) architecture for training the deep learning model and edge platforms play a vital role in improving the accuracy and performance of the deep learning model. High resolution/speed CMOS image sensors provide the requisite image quality for exacting tasks such as CNN based image classification and object detection. The CNN model being trained is highly dependent on the training data. If the training data is comprised of low-quality images, then the CNN will not be able to identify the object under consideration correctly and is most likely to provide poor performance for image classification and object detection tasks. Agricultural imaging classification and detection use cases like freshness of fruits, vegetables and meat particularly benefit when statistically relevant training images are used in conjunction with optimized CNNs and inference routines on edge platforms. In order to inference on the edge platforms and obtain high performance it is necessary to choose CNN architectures that are light-weight and platform-aware. Architecture like SqueezeNet [10] introduces a *Fire module* wherein the 3×3 filters are replaced with 1×1 filters thus decreasing the number of input channels to 3×3 . The squeeze and expand layer in SqueezeNet reduces the number of parameters required in half when compared to a traditional AlexNet architecture while maintaining similar accuracy. MnasNet [15] and NasNet [16] focus on the accuracy vs latency tradeoff while maintaining less number of MAdd (Multiply-Adds) parameters. MobileNetV1 [8], MobileNetV2 [13] and MobileNetV3 [7] introduce depthwise convolution with different variations. MobileNetV2 introduces bottleneck residual block and offers approximately 47% less number of MAdds and 19% less training parameters as compared to MobileNetV1 with better Top 1 accuracy and faster inference results on ImageNet [13]. MobileNetV3 [7] on the other hand combines MobileNetV1 and MnasNet to provide better Top-1* accuracy than MobileNetV1 and MobileNetV2. MobileNetV3 provides around 4% improvement in accuracy on ImageNet vs 37% increase in the number of training parameters. We choose MobileNetV2 in order to maintain an acceptable balance between accuracy, latency and number of training parameters. Some of the other applications which can take advantage of such an image classification and object detection model are defect inspection and detection, surface inspection, textile inspection [14].

Our work mainly focuses on agricultural image classification and detection to determine the type and freshness of fruits (fresh/ rotten apples, fresh/ rotten oranges, fresh/ rotten bananas) where the images are captured using ON Semiconductor's high speed 13 MP CMOS sensor AR 1335, XGS-12 and inferenced on NVIDIA Jetson Xavier [4], an edge-inference platform.

Sensors and camera

Our AI demonstration system utilizes the ON Semiconductor 13 megapixel AR1335 color image sensor. The sensor is integrated into the e-con Systems color camera system. This sensor is a 1.1µm, rolling shutter running at 30fps. The e-con ISP (Image Signal Processor) processes 10bit linear CFA (Color Filter Array) data to YUV for viewing and injecting into the network. This camera system operates in 1080p mode at 30fps and delivers streaming data to the NVIDIA Jetson Xavier processor [4]. It images with a 4.3mm FL (Focal Length), 1/2.3", f/2.8, 67deg FoV (Field of View), $M12 \times 0.5$ lens. Images from this camera system are streamed into NVIDIA Jetson Xavier where they are processed through our MobileNetV2 network topology.

*Top-N accuracy indicates that the target label must be predicted in any of the top N highest probabilities. Top-1, thus indicates that the predicted label must match the target label in the highest probability.

In addition to the Kaggle [1] dataset we have captured and augmented images using ON Semiconductor's XGS-12 based camera system. The image sensor is a 3.2µm, global shutter, 90 fps image sensor designed with high speed imaging tasks in mind. XGS-12 targets the taxing high resolution and high speed capture conditions common to industrial imaging. This sensor outputs HiSpi (High-Speed Serial Interface) data and will then be converted to 30 fps MIPI (Mobile Industry Processor Interface) with a specialized FPGA (Field Programmable Gate Array). In this system, image signals are processed from 12bit linear CFA sensor response to YUV with ON Semiconductor's AP 1302 Image Signal Processor. This data is then delivered to the Xavier processor via MIPI interface and subsequently used to conduct inference of classification and detection tasks.

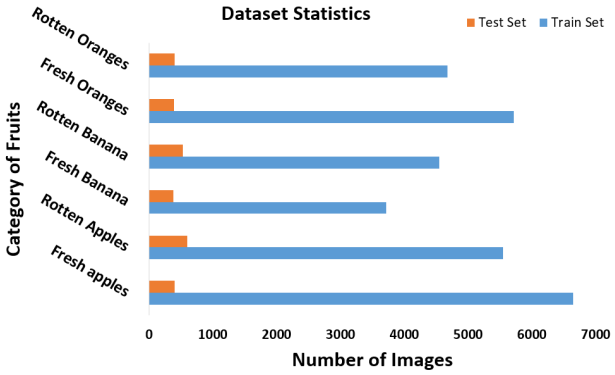


Figure 1: Training and testing dataset statistics.

Dataset Information

For freshness of fruits classification and detection we target six category of fruits, fresh/ rotten apples, fresh/ rotten bananas, fresh/ rotten oranges. Images have been captured using ON Semiconductor's XGS-12 camera. Augmentations like scaling, translation, rotation, Gaussian noise addition, brightness variation, have been performed on these images. In addition to the 12 MP based images, images are also added from a publicly available fruits dataset from Kaggle [1]. The dataset is not used as-is, instead custom augmentations are performed on a selected number of images from the Kaggle dataset. Our testing set is directly taken from the Kaggle fruits data test set [1] so that the assessment is done on a diverse set of data. Our main goal is to perform accurate real-time testing using live video/ images on an edge platform. Figure 1 shows the dataset statistics of our training and testing data. Our dataset consists of 30,846 training images and 2698 testing images.

In order to perform the task of object detection, ground truth annotations are required which provide the bounding box information. To obtain bounding boxes, we use the LabelImg tool [3]. LabelImg tool allows the bounding boxes to be captured in PASCAL VOC or YOLO format. For our work, we use the PASCAL VOC format which provides the upper left and bottom right coordinates of the box $(x_{TopLeft}, y_{TopLeft}, x_{BottomRight}, y_{BottomRight})$.

Architecture for Image Classification and Object Detection

The goal of identifying the fruits based on its freshness is met by modeling an image classification network that classifies the fruits in one of the six categories and an object detection network that classifies and localizes the fruits based on its freshness. In the subsections below we discuss in brief, the image classification and object detection networks used in our experiments.

Image Classification

For image classification we use MobileNetV2 [13]. MobileNetV2 is built upon the concept of Depthwise-separable convolutions [8]. MobileNetV2 consists of 19 inverted residual bottleneck layers [13]. Each bottleneck block consists of a 1×1 *Expansion Layer*, 3×3 *Depthwise Convolution Layer* and 1×1 *projection Layer*. MobileNetV2 is based on an inverted residual structure. A typical residual network starts with an input that has high number of channels and then follows with a few squeeze layers and an expand layer. The residual block here is constructed by connecting the two expanded layers with the help of a skip connection which skips the squeezed layers, thus following a wide-narrow-wide network. The inverted residual network is the inverse of this method. It follows a narrow-wide-narrow structure. The residual connection is between the two narrow layers. The 1×1 *Expansion Layer* widens the network followed by a 3×3 *Depthwise Convolution Layer* which reduces the number of parameters and finally squeezes the network with a 1×1 *projection Layer* [13]. The residulal between the two narrow layers is learnt in this process. We choose MobileNetV2 because it uses less number of Multiply-Add operations and less number of training parameters while maintaining/ improving the accuracy of an image classification model.

For our image classification experiment we apply transfer learning by initializing MobileNetV2 with ImageNet weights and fine-tuning by freezing the first 130 layers of MobileNetV2 and training on the remaining 25 layers.

Object detection

For Object detection based localization and classification we experiment using SSD (Single Shot Detector) [11] as the detection network using MobileNetV2 as the backbone for feature extraction. Contrary to region proposal network (RPN) based architectures like R-CNN [6] fast R-CNN [5], faster R-CNN [12], SSD needs only one shot to identify mutilple objects within a particular frame. Originally, SSD architecture is built upon VGG-16 [11]. For our experiments, we use SSD that is built upon MobileNetV2. The final fully connected layer is discarded and a few convolutional layers are introduced. These convolutional layers provide predictions of detections at different scales [11]. For each location, k bounding boxes of different sizes and aspect ratios are obtained. For each of these bounding boxes, c class scores are computed with 4 offsets with respect to the original ground truth bounding box shape. Thus, at each feature layer of size $m \times n$, *(c+4) kmn* outputs are obtained [11].

Results

In this section we show results on the image classification and object detection model. Results are evaluated on the test set obtained from the Kaggle dataset [1] and random images from

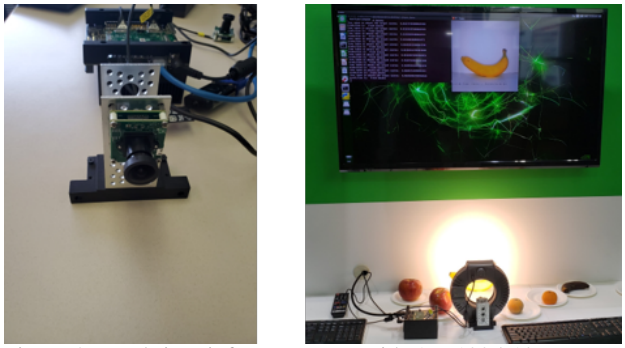


Figure 2: Real-time inference setup with AR 1335 13 MP connected to NVIDIA Jetson Xavier.

the internet. In addition, we also show real time inference on NVIDIA Jetson AGX Xavier [4] edge platform. NVIDIA Jetson AGX Xavier uses TensorRT inference engine that accelerates the inferencing process. The real-time set up with ON Semiconductor's AR1335 13 MP camera connected to Xavier is shown in the Figure 2. The following two subsections discuss training and inference details for image classification and object detection.

Image Classification

During training, the MobileNetV2 image classification model is initialized with pretrained ImageNet weights. The first 130 layers are frozen during training and fine-tuned on the remaining layers. The model is trained on six category of fruits to determine the freshness of the fruit under consideration. The MobileNetV2 model is implemented using Keras framework with TensorFlow backend [2]. The model is trained for 1000 epochs using Adam's optimizer, batch size of 32, and a learning rate of $1e^{-05}$. The model is trained to reduce the categorical loss.

Figure 3 shows the confusion matrix on the test set of the dataset. Using this confusion matrix table we evaluate performance on the test set by estimating precision and recall values as shown in Table 1.

Table 1 represents results on unseen data providing an average precision of 96.92%, average recall of 96.90% and overall average accuracy of 96.88%. In order to mimic real time testing we gathered a few images randomly from the internet falling into the six categories of fruits. The confusion matrix for the same is shown in the Figure 5. The model performs fairy well to unseen data, but does take a hit at the average precision and recall values. The average accuracy on the images randomly chosen from the internet is 79.16%. Detailed values are shown in the Table 2.

Ground Truth Class	**Precision**(%)	**Recall**(%)
Fresh Apples	91.04	97.72
Fresh Bananas	99.21	98.50
Fresh Oranges	98.65	94.58
Rotten Apples	97.61	95.17
Rotten Bananas	97.05	99.43
Rotten Oranges	97.96	95.53

Table 1: Image classification performance on the test set.

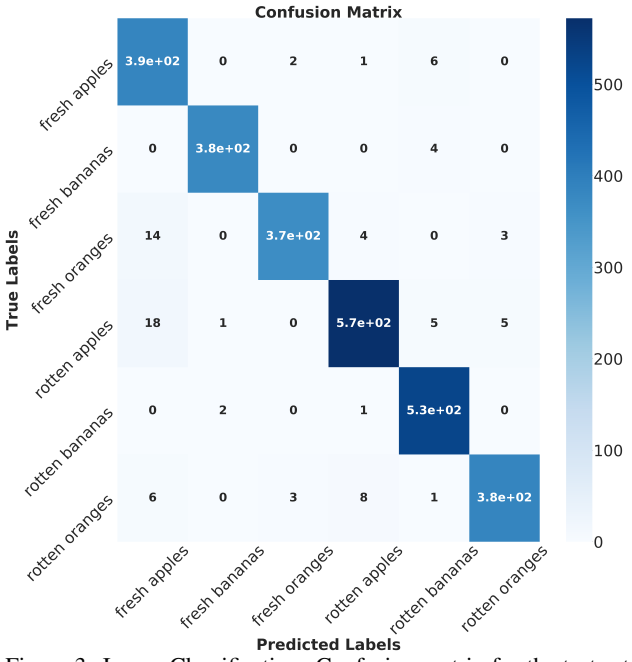


Figure 3: Image Classification: Confusion matrix for the test set images.

Ground Truth Class	**Precision**(%)	**Recall**(%)
Fresh Apples	100	50
Fresh Bananas	100	87.5
Fresh Oranges	70.58	75
Rotten Apples	86.67	81.25
Rotten Bananas	61.53	100
Rotten Oranges	81.25	81.25

Table 2: Image classification performance on random images from the internet.

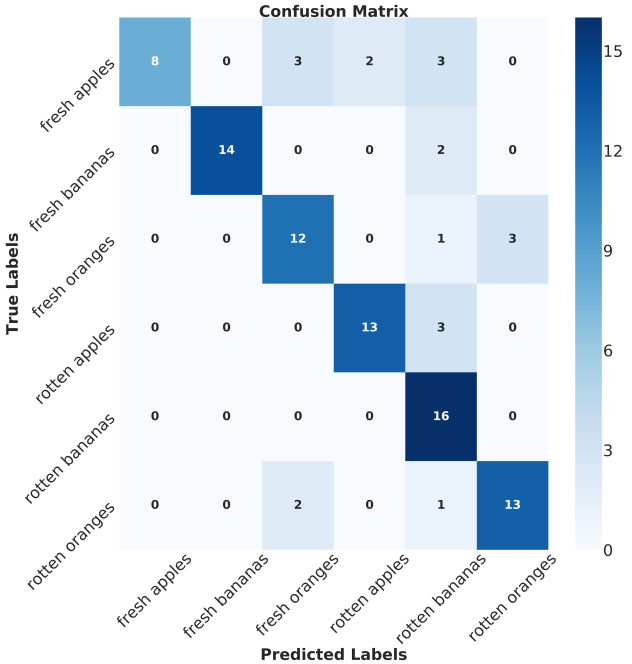


Figure 5: Image Classification: Confusion matrix for random images from the internet.

Figure 4: Image classification inference on the test set.

In order to obtain real-time performance, we capture video using AR1335 attached to NVIDIA Jetson Xavier. The frames are then passed through the trained model to obtain appropriate classification. Figure 4 shows results on the held out test set. Measures are being taken to improve the dataset by increasing the number of images captured from 12 MP camera in order to obtain better real-time performance.

Object Detection

We train a SSD (Single Shot Detector) model with MobileNetV2 as the backbone using TensorFlow Object detection API [9]. The model has been initialized with pre-trained weights from COCO-dataset. Bounding boxes have been generated using LabelImg tool [3]. We train many models with different training parameters to obtain the best model. Since bounding boxes have to be annotated for all the images from training and test set, we only use approximately 10,000 images for training and approximately 2600 images during testing. We start with an exponential decay learning rate of $4e^{-04}$. Different data augmentation options like *random_horizontal_flip*, *random_vertical_flip*, *ssd_random_crop* are used. We also explore different aspect ratios like $1.0, 2.0, 0.5, 3.0, 0.3333, 4.0, 0.25, 5.0, 0.2$.

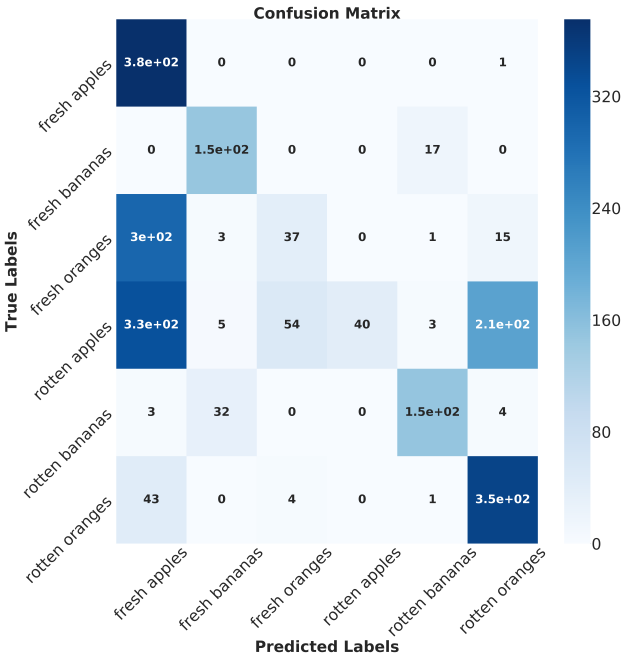


Figure 6: Object detection: Confusion matrix for the test set images.

Table 3 shows the average precision calculated at 11 recall values from 0.0 to 1.0 and the average IOU for each category on the held out test set. The IOU is calculated by considering a threshold of 0.5. The object detection model needs a lot of improvement as it performs with a mean average precision of 62%. The confusion matrix for this test set of object detection model is shown in the Figure 6. As seen in the confusion matrix, the model gets very confused between rotten apples, fresh apples and fresh oranges.

Ground Truth Class	Average Precision (%)	Average IOU's (%)
Fresh Apples	94.71	74.75
Fresh Bananas	26.74	21.39
Fresh Oranges	57.71	47.31
Rotten Apples	90.34	78.07
Rotten Bananas	25.82	17.36
Rotten Oranges	76.89	65.60

Table 3: Object detection performance on the test set.

Figure 7 shows the inference images for object detection on the held out test set. The top figure in Figure 7 shows the objects that were correctly detected and classified while the bottom figure shows the objects that were detected but misclassified. There is a lot of scope for improvement on the object detection model, as the amount of data is insufficient to provide good performance on real-time videos. Currently, annotations are being collected for all the 30,846 images and work is in progress to expand the dataset itself which will potentially lead to improvements in the object detection model by improving the accuracies of the detection and classification tasks.

Conclusion

Achieving good performance on tasks like image classification and object detection for freshness of fruits not only depends on the deep learning architecture being used but also on the quality of data that the model is trained on. Leveraging the XGS-12 camera and 13MP AR1335 camera we obtain high quality images and use it for training. Using MobileNetV2 for image classification we reduce the number of MAdd parameters thus improving the performance on edge platforms. SSD with MobileNetV2 as backbone forms a good basis for the object detection model. For the freshness of fruits application we consider fresh/ rotten apples, fresh/ rotten bananas, fresh/ rotten oranges. We obtain real-time performance with AR1335 connected to NVIDIA Jetson Xavier and achieve 97% accuracy on the image classification model and 62% accuracy on the object detection model. Further efforts are in progress to expand the current dataset by including more number

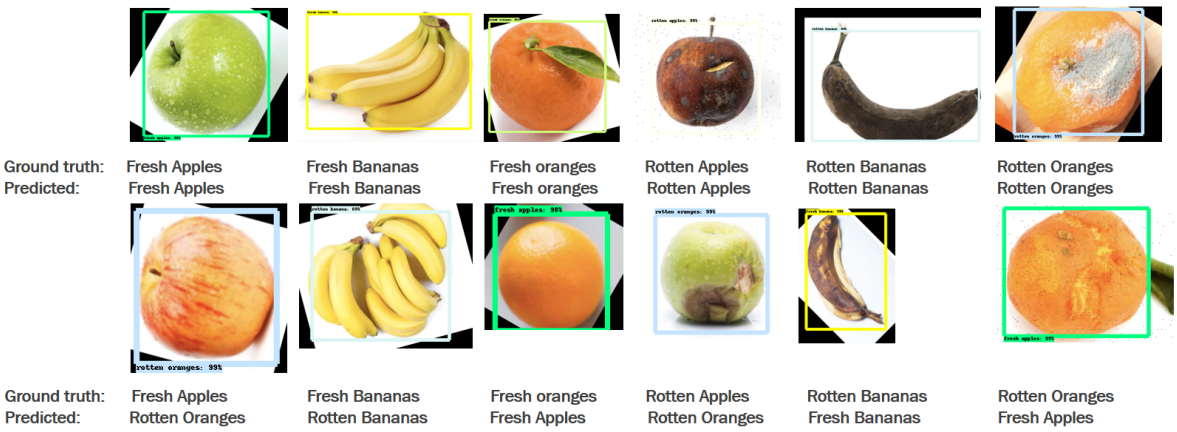


Figure 7: Object detection inference on the test set.
Top row are correct predictions and bottom row are incorrect predictions.

of images captured from 12 MP camera.

Acknowledgments

We would like to thank Sudershan Vuruputoor, Umesh Holalu, Ashutosh Gupta, Sujit Kumar, Siva R, Joydeep Adhikari and Mayur Verma from ON Semiconductor, Bengaluru, India, for helping us in collecting annotations for the object detection task using LabelImg. We would also like to thank Christophe Piron from ON Semiconductor, Belgium, for capturing images using the XGS-12 camera system.

References

[1] Fruits dataset kaggle. `https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification`.

[2] Keras framework. `https://keras.io/backend/`.

[3] Labelimg tool. Tzutalin - Git code (2015): `https://github.com/tzutalin/labelImg`.

[4] Nvidia jetson xavier platform. `https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit`.

[5] Ross Girshick. Fast r-cnn, 2015.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

[7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[8] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[9] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors, 2016.

[10] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size, 2016.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

[13] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018.

[14] Dr. Stephan Se. Deep learning for manufacturing inspection applications.

[15] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile, 2018.

[16] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition, 2017.

Author Biography

Tejaswini Ananthanarayana received her BE in Electronics Engineering from Mumbai University, India in 2010 and her MS in Electrical Engineering with a digital design focus from Rochester Institute of Technology, Rochester, NY, USA in 2015. She is currently pursuing PhD in Engineering with Deep Learning as the focus area at Rochester Institute of Technology, Rochester, NY, USA. In addition to being a full-time PhD student, Tejaswini is also an intern at ON Semiconductor, Rochester, NY, USA, working as a Deep Learning Development Engineer. Her research focus includes, but is not limited to, sign language translation, image classification, object detection and self-supervised learning.

Raymond Ptucha is an Associate Professor in Computer Engineering and the Director of the Machine Intelligence Laboratory at the Rochester Institute of Technology. His research includes machine learning, computer vision, and robotics, with a specialization in deep learning. Ray was a research scientist with the Eastman Kodak Company where he worked on computational imaging algorithms and was awarded 31 U.S. patents. He earned a Ph.D. in computer science from RIT in 2013. Ray was awarded an NSF Graduate Research Fellowship in 2010 and his Ph.D. research earned the 2014 Best RIT Doctoral Dissertation Award. Ray is a passionate supporter of STEM education, an NVIDIA certified

Deep Learning Institute instructor, and the Chair of the Rochester area IEEE Signal Processing Society.

Sean Kelly is the Director of Industrial Imaging Ecosystems at ON Semiconductor, currently focusing on the application of AI to industrial imaging and machine vision. Sean was previously VP of Imaging at Motorola Mobility, leading imaging commercialization and technology teams. Before that he was Senior Director of Imaging at Flextronics supporting the development of cell phone camera modules. He also spent 17 years at Eastman Kodak working on a wide range of imaging technologies and products. Sean holds 6 US patents and an MS in Electro Optics from the University of Dayton. Sean's work has spanned R&D through product commercialization and he is passionate about all aspects of imaging and artificial intelligence.

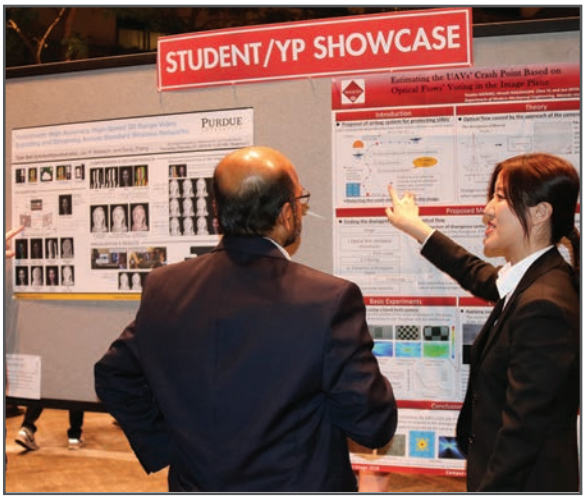
JOIN US AT THE NEXT EI!

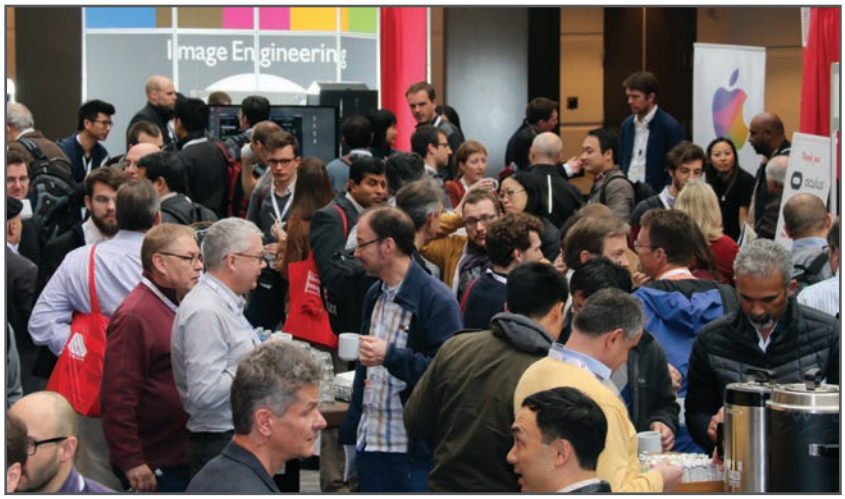
IS&T International Symposium on

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