A deep learning computer vision iPad application for Sales Rep optimization in the field

Edward R. Sykes¹

Accepted: 12 December 2020 / Published online: 4 February 2021
© The Author(s) 2021

Abstract

Computer vision is becoming an increasingly critical area of research, and its applications to real-world problems are gaining significance. In this paper, we describe the design, development and evaluation of our computer vision Faster R-CNN iPad App for Sales Representatives in grocery store environments. Our system aims to assist Sales Reps to be more productive, reduce errors, and provide increased efficiencies. We report on the creation of the iPad app, the data capturing guidelines we created for the creation of good classifiers and the results of professional Sales Reps evaluating our system. Our system was tested in a variety of conditions in grocery store environments and has an accuracy of 99%, a System Usability Score usability score of 85 (high). It supports up to 40 classifiers running concurrently to perform product identification in less than 3.8 s. We also created a set of data capturing guidelines that will enable other researchers to create their own classifiers for these types of products in complex environments (e.g., products with very similar packaging located on shelves).

Keywords Applications of computer vision · Deep learning · Mobile computing · Object detection in real-world environments

1 Introduction

The way in which consumer packaged goods companies are keeping track of product availability and positioning at retail locations is not efficient [2]. Currently, the most common approach involves having company representatives manually checking shelf layouts, stocks, position, and orientation of products [2,17,22]. Sales Reps assess stores to ensure they meet their client's standards and are compliant with contracts and protocols. At this time, virtually all of these processes are done manually with paper and pen using planograms [22]. Of the relatively few apps available to support Sales Reps, they are limited in functionality and typically only address a small number of business-oriented questions using tablets [17,22]. The motivation behind this work is to enhance the use of mobile technologies in combination with computer vision to increase efficiencies of Sales Reps in the field. To this end, we set out to design and implement a system to accomplish this goal. Table 1 presents a list of the desirable

characteristics for the proposed system that was specified at the onset of the work.

To our knowledge, there are no systems that provide these features; this provides motivation and relevance for this research study. We validated our system against these desirable characteristics and with Sales Reps in real-world scenarios and report on their evaluation of the system.

This paper is structured as follows: Sect. 2 provides a literature review of work in this field, Sect. 3 presents our methodology on the design, development, and evaluation of our computer vision system, Sect. 4 presents the findings, Sect. 5 provides a discussion of our research, and Sect. 6 provides a conclusion.

2 Background

This section provides a literature review of relevant studies in this problem domain. We present a review of current workflows for Sales Reps in the field; the current state-of-the-art in product recognition and identify the main problems in this area; and provide a summary of key research studies that have used image recognition for grocery product identification.

✉ Edward R. Sykes
ed.sykes@sheridancollege.ca
http://cmi.sheridancollege.ca

¹ Centre for Mobile Innovation, Sheridan College, 1430
Trafalgar Road, Oakville, ON L6H 2L1, Canada

Table 1 List of desirable characteristics

	Desirable characteristic
1	An advanced tool for Sales Reps to be more productive, reduce human errors, and increase their efficiencies
2	The classifiers need to be accurate at recognizing objects of interest while discerning objects that are very similar (e.g., must distinguish between Minute Maid Original and Minute Maid Calcium)
3	Given an image of a grocery shelf containing potentially 50 different products, a successful classifier must recognize at least one of several objects of interest (e.g., one Minute Maid Original from several on the shelf) and not identify any other products (no False Positives)
4	The classifiers need to be robust and operate successfully in a variety of conditions in grocery stores (e.g., variances in the lighting, orientation of products, obstructions, etc.)
5	The implementation needs to be computationally efficient and run in real-time on the Sales Rep's iPad without any network connectivity (the memory limitation is 1.5 GB, and the system must be able to recognize up to 40 unique products within 20 s)
6	Create a repeatable design process that is generalizable to address similar types of problems (e.g., products that are very similar in terms of appearance and packaging)

Fig. 1 Planogram—visual aid for layout and placement of products on shelves

2.1 Current workflows for Sales Reps in the field

Planograms have long been the de facto standard in the majority of supply chain processes and retail environments [13,26]. Planograms show the placement of products on shelves based on guidelines created by retail distribution managers (see Fig. 1). The advantages of using planograms include assigned selling potential to all available shelf space, better visual appeal, inventory control, easier product replenishment for staff, and better related product positioning [13,24,26]. The disadvantages of planograms are that they are complicated to implement, shelves may not accommodate all products, and new staff may not comply due to lack of training [24]. Another disadvantage of planograms is the fact that they are paper-based and therefore error-prone leading to human mistakes [24]. This may lead to other problems with the management and workflow associated with the products [24]. Another flaw with planograms is that they are not real-time. Inconsistencies between what the planogram depicts versus what is actually on the shelves could be quite different. These discrepancies could lead to delays within the entire workflow (retailer's priorities, supply chain, and in-store operations). To reconcile these differences may take days or weeks at a significant cost to productivity [24].

2.2 Beyond planograms for sales reps in the field

Recent studies have shown there have been very few advancements in the evolution of Planograms and the processes surrounding their use [10,11,33]. Leading research points to computer vision technology as a way that may substantially increase sales force productivity, improve shelf condition insights, and help drive sales [10,28,34]. Gartner researchers state that 'CIOs of consumer goods manufacturers should understand current uses and limitations, as well as potential retailer benefits and interest' [28].

This view is also reinforced by a recent study that was conducted on the state-of-the-art of product recognition in shelf images [10]. In that study, researchers stated that activities such as monitoring the number of products on the shelves, completing the missing products and matching the planogram continuously have become important and that an autonomous system is needed for operations such as product or brand recognition, stock tracking, and planogram matching. [10,11] state the current problems about product recognition below.

- The lack of visual difference among the different products of the same brand creates problems in classification [10, 11];
- The images taken at different angles and at different distances, image quality, and light reflections create problems in classification [10,11]; and
- The methods used to increase classification success can lead to incorrect product classification or the inability to classify the product [10,11].

To our knowledge, these problems have not been solved. They also serve to frame our research efforts in this project to create a solution that addresses these problems.

Fig. 2 Color version of minute maid original and calcium

Fig. 3 Grayscale version of minute maid original and calcium

2.3 Image recognition: tools and technologies

One of our goals is to create accurate classifiers that recognize objects of interest while discerning objects that are very similar. For example, the classifier needs to distinguish between a Minute Maid Original product and a Minute Maid Calcium product. Figures 2 and 3 present these two images in color and grayscale. The difference between these images is very slight and subtle—especially at a distance, amongst many similar products, and in real-world environments.

At the onset of this research, we surveyed computer vision systems that shared the same objectives as proposed in this work. Three relevant studies were found. Advani et al. [1] aimed to classify products belonging to different categories [1]. In this work, a graphical model was created where objects were represented as nodes and connected to each other by edges when viewed together on a visual scene. The weight of a node is a function of the number of times the object

is observed. The test was conducted on 11 different classes with 73% success rate.

The second study by Hafiz et al. [18] used object recognition on images of six different beverage types (Coca-Cola, coffee, Fanta, Pepsi, mineral water and coconut water) [18]. In their approach, the object was first separated from the image background with the saliency map followed by the mean shift segmentation and then passed through an HSV (hue, saturation, value) color-based threshold. The feature vectors obtained by color and SURF descriptors from training datasets with 195 examples were classified by an SVM (support vector machine). A success rate of 89% was obtained on test data with 194 examples.

The third relevant study involved a hybrid classification system consisting of 2 phases that tackled the issue of distinguishing similar products [4]. In the first phase, an SVM was used to classify the information obtained from the retail product image. In the second phase, the classifications obtained from the first phase were combined with the learned statistical product sequence model and the product classes were extracted. This approach was tested on datasets consisting of 108,090 soft drinks images with 794 different classes of 11,557 horizontal shelf sequences with no overlap. Their system had a success rate of 68%.

We also conducted a survey of candidate computer vision algorithms based on the review conducted by Pouyanfar et al. [29] and the following criteria: (1) accuracy, (2) computational speed, (3) memory efficiency, (4) ease of use (i.e., implementation with supporting documentation), and (5) portability (for mobile device use) [29]. We evaluated the following object detection algorithms using this criteria: cascade object detector, mean shift segmentation, support vector machines, You Look Only Once V3, single shot multibox detector, and Faster R-CNN using similar work in the literature as a base [29]. The results are shown in Table 2. During the review and evaluation process, it became clear that the Faster R-CNN was the most feasible since it provided the greatest potential to satisfy the requirements of the proposed research.

2.3.1 Faster R-CNN object detector

Faster R-CNN uses region proposal networks (RPN) that shares full-image convolutional features with the detection network and enables nearly cost-free region proposals [30]. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. RPNs are trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection [30]. Figure 4 presents the high-level architecture of the Faster R-CNN network that was used in the creation of our classifiers.

Table 2 Evaluation of various computer vision algorithms

Criteria	Cascade object detector	Mean shift segmentation	Support vector machines	YOLO V3	Single shot multibox detector	Faster R-CNN
Accuracy	✓✓	✓✓	✓	✓✓	✓✓	✓✓
Computational speed	✓	✓	✓✓	✓✓	✓✓	✓✓
Memory efficiency	✓	✓	✓✓	✓✓	✓✓	✓✓
Ease of use (implementation with documentation)	✓✓✓	✓✓	✓✓	✓✓	✓✓	✓✓
Portability (for mobile device use)	✓✓✓	✗	✓✓	✓✓	✓✓	✓✓

✓: acceptable, ✓✓: good, ✓✓✓: very good, ✗: poor

Fig. 4 High-level architecture of the Faster R-CNN network

3 Methodology

This section discusses the methodology used to design, develop, and evaluate our computer vision system. Our system was initially developed in MathWork's MATLAB. MATLAB's Computer Vision System Toolbox and the Deep Learning Toolbox enabled research and development using the Faster R-CNN Object Detector and ultimate deployment to the iPad. We chose MATLAB for the following reasons:

1. it is a popular and widely used software package used by millions of Engineers and Scientists [25]. Furthermore, our research group has years of experience using MATLAB;
2. it is taught in our undergrad courses in Computer Science and Engineering at our institution; Consequently, there are many students that have acquired MATLAB knowledge and skills;
3. the academic institution has a campus-wide license for MATLAB and Simulink. All members (faculty, students and staff) have access to all 92 MATLAB toolboxes;
4. MATLAB offers exceptional technical support which we have used on numerous occasions to help with our technical issues and to provide advice. This level of support is unlike other packages or tools we have encountered;

Fig. 5 Steps involved in creating the system

5. the research that we conduct in our centre is largely applied in nature. All our projects are with industry partners—many of whom have MATLAB;
6. MATLAB is very good for rapid prototyping and testing various machine learning models [25].

Figure 5 depicts the steps that were used in designing the system to satisfy the desirable characteristics (see Table 1). These steps are elaborated later in this section.

3.1 Transfer learning, custom image set design, ROIs and data augmentation

3.1.1 Transfer Learning

We recognized the limitation of being able to acquire a vast number of images in this problem and thus implemented a transfer learning approach. We experimented and evaluated the performance of our detectors using the five most common convolutional neural networks that have been pre-trained on the ImageNet dataset: VGG16, VGG19, ResNet50, ResNet101, and Inception V3 [20]. The ImageNet

Table 3 List of juice products of interest in retail stores

	Product
1	Simply OJ Pulp Free 1.75L
2	Simply OJ Pulp Free 2.63L
3	Simply Lemonade 1.75L
4	Simply Apple 1.75L
5	Minute Maid Original Orange Juice 1.75L
6	Five Alive Passionate Peach Citrus 1.75L
7	Nestea Lemon 1.75L
8	Minute Maid Calcium Juice 1.75L

has a total of 14 million images and 22 thousand visual categories [20]. The advantage of using this approach is that the pretrained network has already learned a rich set of image features that are applicable to a wide range of images and this learning is transferable to the new task by fine-tuning the network. We fine-tuned the network so that the feature representations learned from the original task (general image feature extraction) would be customized to support our specific classification problem. The advantage of this transfer learning approach is that the number of images required for training and the training time is significantly reduced [29].

3.1.2 Custom image set design

We created a custom image set that incorporated healthy variations of our products. A healthy variation is referred to as uniquely variant pictures of the products of interest containing different backgrounds as well as variations including rotation, noise, contrast, obfuscation, etc. [11]. Our industry partner provided us a list of their top products which was used as the products of interest for training (Table 3).

Our preliminary research showed that glare and shadows often occur on products in retail environments. This is usually due to different lighting conditions. In generating our image set, we aimed to replicate grocery store shelf environments. Products were used as props in different positions, orientations, lighting conditions and environments. For example, we placed products under a shelf to imitate the shadow from a shelf or a lamp shining at a product to imitate glare. Figures 6 and 7 show products with artificially created shadows and glare, respectively, using different shelving and lighting conditions. This figure illustrates that lighting can significantly alter the appearance of a product. Each image set for each product was created using at least 60 front facing pictures of products in these types of configurations ranging from close (0.5–2 m) to far (2–3 m) distances from the product. From our preliminary research, we recognized that overfitting was a possibility. We attempted to partially mitigate this by ensur-

Fig. 6 Artificially created shadow on a product

Fig. 7 Artificially created glare casted onto products of interest for use in creating our positive image set for training

ing that every image was taken with a completely different background.

3.1.3 Identify the region of interests (ROI) on custom training images

We used MATLAB's Image Labeler app to specify the precise product Region of Interest (ROI) on our custom set of images for ground truth training for the classifier. The Image Labeler app provides an easy way to mark rectangular regions of interest (ROI) labels, polyline ROI labels, pixel ROI labels, and scene labels in a video or image sequence [25]. With this tool, we manually labeled the ROIs around the products of interest from the image collection. We then exported this labeled ground truth data for training and testing purposes. Figure 8 presents the Image Labeler app with one image from the collection highlighted with a rectangular ROI demarking the Five Alive product. Experimentation on different ROIs was explored and evaluated based on the performance of the classifiers.

3.1.4 Data Augmentation

Data augmentation enabled the addition of more variety to the training data without actually having to increase the

Fig. 8 MATLAB Image Labeler app. Establishing ground truth datasets for training networks

number of labeled training samples [35]. This method is used to improve network accuracy by randomly transforming the original data during training. Data augmentation for deep learning has shown to yield consistent improvements over strong baselines in image classification, object detection and person re-identification [29,35]. We experimented extensively using the following data augmentation techniques: rotation, reflection, translation, shearing, scaling, random cropping, and erasing on our custom image set. Our data augmentation approach was in the context of a grocery store product.

From the 60 original images per product we created, data augmentation generated a total of 4200 images per product which were used for training (10 generated images from each of these seven augmentation techniques $\times$ 60 original images).

3.2 Train the network and determine optimal parameters

We systematically explored all significant training parameters to produce the most accurate results for the detectors based on the desirable characteristics. We used the Bayesian optimization algorithm for optimizing the hyperparameters of classification model [32].

3.2.1 Bayesian optimization

Bayesian optimization attempts to minimize a scalar objective function $f(x)$ for x in a bounded domain. The function can be stochastic or deterministic. The components of x can be integers, floats, or categorical. The key aspects of this minimization process are:

Algorithm 1: Bayesian Optimization

```

1 initialization
2   Network Section Depth  $\leftarrow d \in \mathbb{Z} : [1, 3]$ ;
3   Initial learning rate  $\leftarrow r \in \mathbb{R} : [0.001, 1]$ ;
4   Stochastic gradient descent  $\leftarrow g \in \mathbb{R} : [0.8, 0.98]$ ;
5   L2 regularization strength  $\leftarrow s \in \mathbb{R} : [1e-10, 1e-2]$ ;

6 begin
7   Evaluate  $y_i = f(x_i)$  for NumSeedPoints points  $x_i$ , taken at
   random within the variable bounds;
8   if there are evaluation errors then
9     take more random points until there are NumSeedPoints
     successful evaluations. The probability distribution of
     each component is either uniform or logarithmic-scaled,
     depending on the Transform value set during
     initialization;
10  repeat
11    Update the Gaussian process model of  $f(x)$  to obtain a
    posterior distribution over functions  $Q(f|x_i, y_i)$  for
     $i = 1, \dots, t$ ;
12    Find the new point  $x$  that maximizes the acquisition
    function  $a(x)$ ;
13  until ((max_iterations not reached) and (time_limit not
    reached) and (stopping_criterion not met));

```

- A Gaussian process model of $f(x)$.
- A Bayesian update function that modifies the Gaussian process model at each new evaluation of $f(x)$.
- An acquisition function, $a(x)$ (based on the Gaussian process model of f) that is maximized to determine the next point x for evaluation.

Algorithm 1 presents the Bayesian optimization algorithm that was used in the refinement of our classification models [32]. In our research, the following parameters were optimized:

- *Network section depth.* This parameter controls the depth of the network. The total number of layers in the network is $9 * SectionDepth + 7$. The network has three sections, each with *SectionDepth* identical convolutional layers. The total number of convolutional layers is $3 \times SectionDepth$. The objective function takes the number of convolutional filters in each layer proportional to $\frac{1}{\sqrt{SectionDepth}}$. Consequently, the number of parameters and the required amount of computation for each iteration are roughly the same for different section depths. Range: [1 3], Type: integer.
- *Initial learning rate.* Range: [0.001 1], ‘logarithmic.’
- *Stochastic gradient descent momentum.* Momentum adds inertia to the parameter updates by having the current update contain a contribution proportional to the update in the previous iteration. This results in more smooth parameter updates and a reduction of the noise inherent to stochastic gradient descent. Range: [0.8 0.98].

Fig. 9 Accurate identification of all Simply Lemonade products

- *L2 regularization strength.* L2 regularization is used to prevent overfitting. This algorithm searches the regularization strength space to find a good value. Data augmentation and batch normalization also help regularize the network. Range: [1e-10 1e-2], ‘logarithmic.’

Training progress plots were used to analyze the progression of training. The plots showed the ‘TrainingAccuracy,’ ‘ValidationAccuracy,’ and ‘TrainingLoss.’ These plots were helpful in analyzing how quickly the network accuracy was improving and to determine if the network started to overfit the training data [8,31]. After training, we systematically analyzed the results of the Faster R-CNN networks, namely ‘TrainingLoss,’ ‘TrainingAccuracy,’ ‘TrainingRMSE,’ and ‘BaseLearnRate.’ Once detection was found on one or more the objects of interest, our system placed bounding boxes around the object.

Figure 9 shows bounding boxes around objects of interest illustrating accurate identification (true positives), while inaccurate identification (false positives) is shown in Fig. 10. Figure 10 also illustrates a classifier that is unacceptable based on our requirements even though it correctly identified one object of interest.

3.3 Create test image set

Test images representing real-world scenarios were created using the iPad. To recreate real-world scenarios accurately, staging of test images was done to provide the most authentic representation possible. Photographs were taken at specific distances based on how far Sales Reps would typically stand relative to product shelves in a grocery store. For example, please see Figs. 9 and 10. Additional factors were taken into consideration such as lighting conditions, orientation of products, as well as the relative position of the product on the shelf. A total of 40 different grocery store shelves were col-

Fig. 10 Unacceptable classification of Simply Lemonade products. This classifier does not meet the requirements because of the falsely identified products even though one was found

lected each with potentially 30 or more different products. For example, in Fig. 9, there are over 30 different products represented in this single photograph. This approach facilitated well over a thousand unique test images (i.e., products) that were used for testing purposes in our research.

3.4 Implementation on iPad

We implemented our classification system as a native iPad app. The specifications of the iPad used in our research was: iPad (Pro), A10X Fusion, 2GB memory (total), 12-megapixel camera.

The open-source computer vision library (OpenCV) was used for the object recognition on the iPad [5, 15]. OpenCV is also used in MATLAB's Computer Vision System Toolbox, allowing our classifiers to be ported from MATLAB to iOS. The main area of focus for this part of the research involved: (1) evaluating the performance and accuracy of the classifiers on the iPad and (2) exploring human factor research questions such as, how is the Sales Rep holding the iPad and what is the impact on how the classifiers' function as a result? Are there ways to guide the user to take good pictures to ensure that the classifiers will perform well?

We also had the following constraints: (1) the iPad's memory limitation (a maximum of 1.5 GB is available at any given time), (2) 40 classifiers must complete execution within 20 s, and (3) the entire process must be done on the device without any network connectivity.

3.5 Test and evaluate the iPad app in real-world environments

Despite many accurate classifiers, there were also many inaccurate ones. The goal of this phase of the methodology was

to determine appropriate configurations for object detection that improved accuracy and performance. The areas explored were:

1. Experimenting with a collection of different product layout configurations for creating the custom image sets. This work led to the development of data capturing guidelines (heuristics) for the curation of positive images with the intent that such guidelines may be generalized and applied to other related problems.
2. Experimenting with size and style of the ROIs for the custom image set. Several variations of ROIs were investigated, namely: having no overlap with the background, variations between 1–10% overlap with the background, etc. Figure 12 presents four scenarios: (a) full ROI around product with 1% overlap with the background, (b) tight ROI completely containing the front of the object of interest, (c) restricted ROI on the front face including any detail unique to the product (potentially reducing the impact of shadows from the shelf above), and (d) tight ROI, capturing only the front face that contains the major details of the object of interest.
3. Evaluating the impact on accuracy by changing the parameters for detection, namely 'NumStrongestRegions,' and 'MiniBatchSize.' *NumStrongestRegions* is the maximum number of strongest region proposals that the detector has identified. *MiniBatchSize* is used for each training iteration and represents a subset of the training set that is used to evaluate the gradient of the loss function and update the weights.
4. False Positive suppression. The *NumStrongestRegions* detector parameter was particularly important in our approach to suppress the number of false positives. Recall, the desirable characteristics that were established at the onset of this research, specifically, desirable characteristic #3: 'given an image of a grocery shelf containing potentially 50 different products, a successful classifier must recognize at least one of several objects of interest (e.g., one Minute Maid Original from several on the shelf) and not identify any other products (no False Positives).' To achieve this goal, we designed our system to reduce the number of false positives. This was accomplished by using the *NumStrongestRegions* detector parameter to essentially create a threshold of whether or not the object detected should be classified as a true positive or discarded. Figure 11 presents an example of multiple bounding boxes around a product. If the *NumStrongestRegions* is above the threshold, then it would be classified as a true positive.

Fig. 11 False positive suppression using the *NumStrongestRegions* detector parameter. The region (represented by the multiple bounding boxes) must be sufficiently strong for the object to be classified as a true positive

Fig. 12 From left to right: region of interest with 1–2% overlap with background, tight ROI around product, all details specific to product on front face, and subset of details on front face

3.6 Participants

This study involved eliciting user experience feedback to determine the effectiveness of our system. Twenty-one volunteers were involved as participants. Two groups of participants provided their perspectives and evaluations of our iPad app. The first group consisted of 13 participants from our research office. The second group consisted of 8 professional sales representatives working in the field.

3.7 Analysis

Both quantitative and qualitative analyses were performed: evaluating our classifiers using standard machine learning evaluation techniques, standard descriptive statistics, classifier performance and a usability study with sales reps in real-world environments. The results drove the refinement process which involved experimenting with the machine learning training and run-time parameters. It also resulted in a set of data capturing guidelines and structure for how to create similar detectors.

1. Computing confusion matrices: Compare the classifier's accuracy at recognizing at least one of potentially several objects of interest to ground truth (object present or not) with the additional requirement that there can be no False Positives. Confusion matrices and derived measures (e.g., *accuracy*, *sensitivity* (recall, or true positive rate), *specificity* (true negative rate), *miss rate* (false negative rate), *fall-out* (false positive rate), *precision*, and the *F1 score*) are commonly used in the evaluation of machine learning algorithms, please see: (Elkan [14]; Forman and Scholz

Table 4 Confusion matrix derivations

True positives (TP):	Classifier correctly identifies the correct product
True negatives (TN):	Classifier correctly identifies that it is not the product of interest
False negatives (FN):	'Didn't identify the product when it should have'
False positives (FP):	'False Alarm'—classifier said it was a product of interest when it shouldn't have
Accuracy:	$\frac{TP + TN}{TP + TN + FP + FN}$
Sensitivity:	$\frac{TP}{TP + FN}$
Specificity:	$\frac{TN}{TN + TP}$
Miss rate:	$\frac{FN}{FN + TP}$
Fall-out:	$\frac{FP}{FP + TN}$
Precision:	$\frac{TP}{TP + FP}$
F1 score:	$\frac{2TP}{2TP + FP + FN}$

[16]; Hamilton [19]; Kohavi and Provost [21]; Lu et al. [23]). Table 4 presents the statistics that were collected for each classifier.

- Performance was measured by execution time and memory required to run the detectors. In the context of running the detectors on the iPad, higher performance was recognized by faster execution requiring the least amount of memory.
- The following methodology and analysis were used for the qualitative analysis: as the Sales Reps were already comfortable with using iPads for tasks during their work routine, they were asked to proceed with their usual work routine and concurrently use our app. Four user testing sessions were conducted each separated by at least one week to ensure the feedback was analyzed and incorporated into the refined app. Information was collected from participants from System Usability Scale (SUS) surveys, and researcher observation as participants tested our app [6] (see “Appendix” for the survey). An analysis of 5000 SUS results in 500 studies across a variety of applications found that the average score is 68 [7]. A SUS score above 68 would be considered above average [7].

4 Findings (analysis and evaluation)

This section presents the findings of the performance evaluation of the detectors; the impact of changing the detector parameters; data capturing guidelines; and user experience findings.

4.1 Performance evaluation of the detectors

Numerous confusion matrix computations were performed for each detector created across the test image set. Table 5 presents these results for our best detector (Five Alive Passionate Peach) with an accuracy of 99.4%, a sensitivity (recall or true positive rate) of 87.5%, specificity (true negative rate) of 100%, miss rate (false negative rate) of 12.5%, fall-out (false positive rate) of 0%, precision of 100%, and an F1 score of 93.3%. Other successful detectors had similar results as shown in Table 6.

We used the Bayesian optimization technique to optimize hyperparameters of classification models for all our detectors [32]. For the Five Alive Passionate Peach detector, the ResNet101 pre-trained network produced the best results amongst all the pre-trained networks (see Table 7). The following values were discovered from the Bayesian optimization process (Please refer to Sect. 3.2 ‘Train the network and determine optimal parameters’):

- Network section depth: 3 (the range was: [1 3])

Table 5 Confusion matrix results and derivations for our best detector (Five Alive Passionate Peach)

Predicted class	Actual class		Total
	Five Alive Peach	Non Five Alive Peach	
Five Alive Peach	TP = 35	FP = 0	TP + FP = 35
Non Five Alive Peach	FN = 5	TN = 918	FN + TN = 923
Total	TP + FN = 40	FP + TN = 918	958
Accuracy:			99.48%
Sensitivity (recall or true positive rate)			87.59%
Specificity (true negative rate)			100.00%
Miss rate (false negative rate)			12.50%
Fall-out (false positive rate)			0.00%
Precision			100.00%
F1 score			93.33%

Table 6 Partial confusion matrix results and derivations for our best detectors

	Product	Accuracy (%)	Sensitivity (%)	Specificity (%)	Miss rate (%)	Fall-out (%)	Precision (%)	F1 Score (%)
1	Simply OJ Pulp Free 1.75L	95.10	84.38	100	15.63	0	100	91.53
2	Simply OJ Pulp Free 2.63L	93.18	79.07	100	20.93	0	100	88.31
3	Simply Lemonade 1.75L	91.50	76.79	100	23.21	0	100	86.87
4	Simply Apple 1.75L	95.88	83.58	100	16.42	0	100	91.06
5	Minute Maid Original Orange Juice 1.75L	96.92	75.44	100	24.56	0	100	86.00
6	Five Alive Passionate Peach 1.75L	99.48	87.50	100	12.50	0	100	93.33
7	Nestea Lemon 1.75L	93.63	74.68	100	25.32	0	100	85.51
8	Minute Maid Calcium Juice 1.75L	95.73	82.61	100	17.39	0	100	90.48

- Initial learning rate: 0.005 (the range was: [0.001 1])
- Stochastic gradient descent momentum: 0.91 (the range was: [0.8 0.98])
- L2 regularization strength: $1.52e-6$ (the range was: [1e-10 1e-2]).

Other parameters we discovered that led to improved training were:

- *MaxEpochs*: 30
- *MiniBatchSize*: 128 observations at each iteration
- *PositiveOverlapRange*: [0 0.3] and *NegativeOverlapRange*: [0.6 1]

The progress plot presented in Fig. 13 shows a representative training progress plot from our experiments; this one showing the ResNet50 for the Five Alive Passion Peach Product for ‘TrainingAccuracy,’ ‘ValidationAccuracy’ and ‘TrainingLoss.’ Progress plots depict how we monitored the training of our deep learning networks. The plots show various metrics during training (e.g., how quickly the network accuracy is improving, and whether the network is starting to overfit the training data, etc.). The plot displays training metrics every iteration. Each iteration is an estimation of the gradient and an update of the network parameters. The figure also presents each training epoch using shaded backgrounds in the columns. An epoch is a full pass through the entire data set. The figure presents the following:

- *Training accuracy* Classification accuracy on each individual mini-batch.
- *Smoothed training accuracy* Smoothed training accuracy computed by applying a smoothing algorithm to the training accuracy which makes it easier to spot trends.
- *Validation accuracy* Classification accuracy on the entire validation set.
- *Training loss, smoothed training loss, and validation loss* The loss on each mini-batch, its smoothed version, and the loss on the validation set, respectively.

4.2 Findings from detector parameter investigation while running on iPad

Extensive testing was conducted to evaluate desirable characteristic #6 (computational and time efficiency), desirable characteristic #3 (at least one TP, no FP), the memory and time requirements to run the detectors on the device (please refer to Table 1). The following sections present the findings from a set of experiments that aimed to minimize the app’s memory usage and running time by changing the detector parameters (*NumStrongestRegions*, and *MiniBatchSize*) without compromising accuracy.

Table 7 Accuracy comparison of pretrained networks for Five Alive Passionate Peach product detectors

Pretrained network	Accuracy (%)	Sensitivity (%)	Miss rate (%)	Precision (%)
VGG16	93.15	78.50	21.84	100.00
VGG19	92.74	81.75	18.42	100.00
ResNet50	96.57	82.33	14.86	100.00
ResNet101	99.48	87.50	12.50	100.00
Inception V3	95.43	83.25	12.94	100.00

Fig. 13 Training progress plot for Five Alive Passionate Peach Product depicting ‘TrainingAccuracy,’ ‘ValidationAccuracy’ and ‘TrainingLoss.’ Progress plots were used to monitor the training of our deep learning networks

4.2.1 The impact on running time and accuracy with respect to NumStrongestRegions

The *NumStrongestRegions* is the maximum number of strongest region proposals. We discovered that reducing the *NumStrongestRegions* speeds up processing time however at the cost of detection accuracy. Figure 14 shows a representative plot from our experiments showing *NumStrongestRegions* vs. detection accuracy.

We also discovered that by increasing the MiniBatchSize the detection processing is sped up; however, it takes up more memory. Figure 15 shows a representative plot from our experiments showing MiniBatchSize vs. iPad memory utilization.

4.3 The impact on running time in relation to MinSize and MaxSize

A series of experiments were conducted to find the optimum values for *MinSize* and *MaxSize* such that the running time

Fig. 14 NumStrongestRegions vs. detection accuracy (all other variables were held constant)

Fig. 15 MiniBatchSize vs. iPad memory utilization (all other variables held constant)

Fig. 16 Surface plot of maxX, minY in reference to running time (minX=100, maxY=500).

would be minimized. *MinSize* is the minimum region size that contains a detected object, a [height width] vector measured in pixels. *MaxSize* is the maximum region size that contains a detected object, a [height width] vector also measured in pixels. It was discovered that adjusting the min and max sizes proved to have a significant impact on the detectors by significantly reducing the runtime detection time. In our experiments, we ran 40 classifiers (5 copies of our 8 classifiers) running concurrently with the following parameters held constant: minX = 100, maxY = 500. This was based on the previous findings and minX and maxY represent the smallest x and largest y values for the images in our positive image set based on the rectangular shaped juice boxes. Figure 16 depicts a surface plot of maxX, minY in reference to running time (s). The final analysis revealed the minimum

running time was 3.82 s for 40 classifiers executing in parallel with MinSize = 100,300 and MaxSize = 200,500. This is a speed increase of 162% over the previous parameter values.

5 Data capturing guidelines for the creation of good training image sets

Throughout this project, extensive research was conducted to create classifiers capable of differentiating between products with only slight variations to their packaging (see Fig. 2). We experimented with the training and classifier parameters which had an impact on the accuracy and performance of the classifier. However, we discovered that the most significant impact on the quality of the classifier lies with the creation and quality of the positive image set. We carefully constructed all of the positive image sets with healthy variation images with unique backgrounds. Table 8 presents a design pattern representing the culmination of the knowledge and heuristics that we gained in conducting this research in this context.

After applying these guidelines to all of the positive image sets, the majority of classifiers resulted in zero false positives (i.e., the classifier did not mistakenly identify a product). These classifiers performed extremely well on test images with products of interest straight on, or near center field of view. They did not, however, perform to expectations on extreme cases where the products of interest were in the far corners. Figure 18 shows one case where the classifier identified only one product of interest (*Simply Apple*) because it was located in the top left corner of the image.

In order to create classifiers that were capable of detecting products located in the extremities within a given image, a quadrant system was constructed. The goal of the quadrant system approach was to improve training of classifiers by training them to recognize more products of interest positioned in a variety of different natural angles. Figure 19 shows the seven-quadrant system that we created. We populated each quadrant with at least twenty images from each of those perspectives. Figure 20 displays a test result of the same extreme case image with a classifier built with the seven-quadrant system.

Another significant area that we explored was the problem of how to best support a user to take good pictures while looking at the products on the shelf. This is essential so that the classifier is presented with the best possible image for analysis. We discovered that if test images had rotation within them, then the classifiers did not identify as many products as it possibly could. Although difficult for the human to eye to see, slight rotations (i.e., Fig. 21), demonstrate the results of a *Simply Lemonade* classifier only identifying two out of four products in this image with a rotation of 1° CCW. Upon

Table 8 Data capturing guidelines—heuristics for the curation of positive image sets with healthy variations

✓	Aim to curate at least 60 unique images for the image set
✓	Pictures should be taken straight on
✓	Have an equal representation of images that are closeup (within 1m) and medium distance away (1m-2m range). Products in the images should never be so far away that text becomes unreadable
✓	Have an equal mix of cluttered and non-cluttered backgrounds
✓	Do not take extremely dark pictures, as text and other details become lost
✓	Take pictures with different lighting (i.e., both natural and artificial light)
✓	Imitate glare and shadow (please see Figs. 6 and 7)
✓	Avoid taking pictures that are angled toward the product. Instead, use natural angles from a person's point of view. For example, the field of view is in the center (straight), but the product may be located in the top-right corner of the view
✓	If multiple pictures are taken with the same or very similar backgrounds, select the one that looks most appropriate to avoid overfitting; and
✓	Specifying the ROIs: Once the images are created, the next critical step is to specify the ROI for the products in each of the images (see Fig. 12). We determined that the best ROI for our products was a tight ROI wholly containing the front region of the object of interest. Another important finding from the ROI process was when the side of a product is visible, the ROIs should only denote the front facing portion of the product (see Fig. 17)

Fig. 17 Tight ROI surrounding the front face of the product when the side of a product is visible **Fig. 19** Seven-quadrant system for taking positive images **Fig. 18** Extreme case with products of interest (Simply Apple) in far top left corner

rotating the image 1° CW, the classifier was able to identify another product (see Fig. 22).

Based on these findings, we created visual guides in our app to support users to take good pictures. We created a cus-

Fig. 20 100% accuracy—results of the extreme case after incorporating the seven-quadrant system

Fig. 21 Simply Lemonade classifier identifying two products

Fig. 22 Simply Lemonade classifier identifying three products after 1° rotation CW

tom camera module to guide the user to take straight, level pictures with no tilt. This custom camera provided the following features: (a) a grid layout (to support straight-on picture taking), (b) a level (to support left-right vertical orientation), and (c) a rectangle (to support forward-backward tilt of the iPad). Figure 23 shows these features. We designed the app so that user can only take pictures when both the level and tilt indicators are green (the button is disable otherwise). This design principle ensured that the app is given a suitable image to work with for product recognition. Figure 24 presents the final version of the app.

6 User experience findings

The section presents the qualitative findings elicited from user experience feedback. Two groups of users were involved in this study corresponding to the first deployment of the app and a series of versions leading to the final version. Both groups of users provided their evaluations of the app, completed a System Usability Scale survey and were observed by

researchers for refinement purposes to the classifiers and the app itself. We report on the Group 2 (Sales Reps) findings as they were the most significant and reflect the usability of the final version of the app.

6.1 Professional sales reps usability findings

This group consisted of professional Sales Reps working in the field. The following presents the summarized results:

- All of the participants showed enthusiasm and eagerness while using the iPad app;
- All of the participants felt comfortable using the app while checking call reports, taking pictures, and reviewing them (i.e., integration with their typical workflow activities);
- Many participants were excited when they saw the app detecting the products of interest that they manage and expressed they want and need this to be incorporated into their own work routines to increase their day-to-day efficiencies.

Regarding the findings from the questionnaires, the System Usability Scale odd-numbered items (I1, I3, I5, I7, and I9) express positive statements on the application¹. All of these scored 4 or 5 ('strongly agree' or 'agree' with the statement), except for I5, which scored mostly 4 ('agree'). In total, 100% of the respondents gave scores of 4 or 5 to I1; 100% to I3; 75% to I5; 100% to I7; and 87.5% to I9. Figure 25 presents positively rated items showing user satisfaction.

The mean SUS score for this group was 85 (min = 72.5, max = 92.5, $\sigma = 5.86$). The average SUS score from 500 studies is 68 [7]. A one-way ANOVA was performed to determine whether there was a difference in the user satisfaction in our app compared to the SUS average. There was a statistically significant difference between the groups at the 0.05 level, $F(1, 8) = 23.37$, $p = 0.013$, indicating that the usability of our app is well above average.

The even-numbered items in the SUS questionnaire (I2, I4, I6, I8, and I10) express negative statements in using the app.² All of the respondents gave scores of 1 or 2 ('strongly disagree' or 'disagree') for all items except for I6 where 12.5%

¹ SUS positive rated items: Item I1: 'I think that I would like to use this system frequently,' Item I3: 'I thought the system was easy to use,' Item I5: 'I found the various functions in this system were well integrated,' Item I7: 'I would imagine that most people would learn to use this system very quickly,' Item I9: 'I felt very confident using the system.'

² SUS negative rated items: Item I2: 'I found the system unnecessarily complex,' Item I4: 'I think that I would need the support of a technical person to be able to use this system,' Item I6: 'I thought there was too much inconsistency in this system,' Item I8: 'I found the system very cumbersome to use,' Item I10: 'I needed to learn a lot of things before I could get going with this system.'

(a) tilted and not level

(b) level but tilted

(c) not level but not tilted

(d) level and no tilt

Fig. 23 Custom camera on the iPad app with gridlines, level and tilt indicators to help Sales Reps take proper pictures: **a** not acceptable (tilted and not level), **b** not acceptable (level but tilted), **c** not acceptable (not level but not tilted), and **d** acceptable (level and no tilt)

responded with ‘agree,’ collectively indicating a high user satisfaction (see Fig. 26).

Throughout this entire phase of user testing, comments and constructive feedback were recorded. The following are comments from Sales Reps during their use of the app.

Positive comments

1. ‘The application is awesome! I would love to see this in use as it is very fast and will definitely be more efficient than current work routines!’
2. ‘This is a fantastic idea! Would love to see this integrated into our current application as well as direct integration with Yammer to upload pictures since it’s a total time saver!’
3. ‘It’s amazing to see detection still works on products with price tags overhanging in front of them.’
4. ‘This application will for sure make calls more efficient and accurate.’
5. ‘I’m extremely pleased to see improvements to the application and detection since last time!’

Negative comments/constructive feedback

1. ‘It’s unfortunate that some products were not detected behind the glass door shelving.’
2. ‘Sometimes obstructions like large promotional tags would interfere with detection.’

7 Discussion

This section presents a summary of the significant factors in designing good classifiers for grocery store environments, limitations and future work.

7.1 Significant factors in designing good classifiers

We discovered that there are several factors that contribute to the design of good classifiers for the problem explored in this research: classifier training process and parameters, classifier runtime parameters, and the heuristics for effective image data capturing and curation.

1. *Training process and parameters* The training process implemented was effective due to sourcing representative images, using data augmentation, and transfer learning. The most impactful area that influenced the quality of the classifiers during the training phase was in the specification of the Regions of Interest on the training images. Data augmentation techniques were also important. We encountered several circumstances in which the images taken naturally capture products that have some tilt or

Fig. 24 Final version of our computer vision App

Fig. 25 Positively rated items showing user satisfaction

Fig. 26 Negatively rated items showing the lack of user satisfaction

curvature (see the products located in the top right section of Fig. 9). We discovered there are several factors that contribute to this phenomenon such as, how far back the

Sales Rep is while taking pictures; the size of the products; and the shape and number of products on the shelf. We observed that some natural rotation and/or tilt was particularly noticeable in photographs taken where products are positioned at the extremes (i.e., top left, top right, bottom left, and bottom right in the image). We discovered this is due to the physical properties of the iPad's camera. The curvature of the camera's lens can cause 'barrel distortion' and 'pincushion distortion' [9]. This is also a function of the physical properties of lines of perspective [3,12]. We employed the use of data augmentation to address these issues and to improve the accuracy of the detectors. For example, evidence of the impact of rotation may be seen in Figs. 21 and 22. Other data augmentation techniques, namely reflection, translation, shearing, scaling, random cropping, and erasing, were important to capture the scenarios that a Sales Rep may encounter in a grocery store (e.g., reflections from glass doors and shelves; scaling to capture different distances the Sales Reps may be from the shelf when taking a picture; and random cropping and erasing for various obfuscation scenarios such as price tags, special product coupons or shelf structures that block portions of the product). The training phase was also improved by Bayesian optimization that facilitated the selection of the best parameter settings aimed for the highest degree of accuracy while minimizing the time required and the amount of memory to run on an iPad. The most significant parameters involved during the training phase were 'learning rate,' 'momentum,' 'MaxEpochs' and 'miniBatchSize.'

2. *Classifier runtime parameters* The significant parameters involved when the classifier executes are: ‘Num-StrongestRegions,’ ‘miniBatchSize,’ ‘minSize’ and ‘max-Size.’ We found that these parameters play important roles in increasing the accuracy of the classifier, reducing memory consumption, and reducing the time for product identification.
3. *Data capturing guidelines* One of the main contributions of this work is the data capturing guidelines that we created for the construction of good image training sets. The heuristics supporting these data capturing guidelines have been carefully prepared with the objective of assisting other researchers exploring similar types of problems (e.g., detecting and discriminating objects with very similar features).

7.1.1 Limitations

1. *Good training sets* When good training sets are used, the classifier’s accuracy is increased significantly. However, to create the training sets takes a considerable amount time and effort. Acquiring images to create good training sets in complex real-world settings is difficult.
2. *Product packaging changes* Companies invest a considerable amount time and money to keep their products up to date and to meet the needs of consumer’s expectations [27]. The materials, designs and manufacturing processes of products are frequently refined to respond to new trends and customer feedback. Packaging redesigns occur due to one or more of the following reasons: (1) Brand changes/logo changes, (2) Regulation changes, (3) Environmental changes (e.g., green initiatives), and (4) trying something different [27]. In the context of this work, if product packaging changes, new images will be needed to train and create new classifiers.
3. *Grocery store environments change* We have noticed that grocery stores are starting to enclose chilled products behind glass doors. Furthermore, the lighting in shelving units varies from store to store. For every configuration and type of lights used, there is an impact on the way in which light reflects on the products. As discussed in the data capturing guidelines section, these lighting and configurations need to be considered in the image sets for training. This makes research in this area a challenge.

7.2 Limitations and future work

7.2.1 Future work

There are several natural extensions of this work. Some of the most feasible ones are presented below. Future research should:

1. *explore ways to accelerate the process of creating product images for training* A future enhancement of this work would be an automated process involving an on-premise, high-speed, high-resolution video recording system that capture images and automatically demarks ROI boundaries under a variety of different conditions (lighting, glare, etc.);
2. *test and evaluate the data capturing guidelines* In this work, we created a seven-quadrant data capturing methodology for the curation of good image training sets. Future research should explore the degree of generalizability of our data capturing methodology, its application, feasibility and evaluation to other settings and products.
3. *explore the use of increased computational resources* To train classifiers in our current system took at least 45 min on a MacBook Pro (2.9 GHz processor, 32 GB DDR4, and Radeon Pro 4096MB GPU). A high-performance computing system with substantial GPU processing capability would be beneficial for future exploration.

8 Conclusion

This paper described a computer vision system that was created and evaluated that specializes in grocery shelf product identification for Sales Reps in the field using iPads. The main contributions of this work are:

1. the computer vision app-enabled Sales Reps to be more productive, reduce human errors, and increase their efficiencies;
2. the mean SUS usability score was 85 (high) across the Sales Reps who tested our app in real-world grocery store environments;
3. the classifiers are robust, operate successfully in a variety of conditions in grocery stores and have demonstrated an accuracy up to 99% (with no false positives) which is beyond that of competitor systems;
4. the app is computationally efficient, runs in real time on the Sales Rep’s iPad without any network connectivity and can run 40 classifiers concurrently to identify objects of interest in less than 3.8 s from amongst 50 or more competing products; and
5. a set of data capturing guidelines that provides the methodology for creating an accurate classifier for identifying products in grocery stores. This last contribution is intended to aid other computer vision researchers.

In the spirit of furthering science and this work, all of the source code (MATLAB, Swift iOS code) for this project including the classifiers, models, and data sets will be openly available on the author’s and/or journal’s website. For a video presentation of our app, please see <https://www.youtube.com/watch?v=...>

[com/watch?v=NirKSSFrjE](https://www.youtube.com/watch?v=NirKSSFrjE). We hope this will encourage other researchers to explore and extend our work.

Acknowledgements We gratefully acknowledge the Natural Science and Engineering Research Council (NSERC: nserc-crsng.gc.ca) for the funding for this research (Grant Number NSERC CARD2 469614) and Tuan Mai's contributions.

Compliance with ethical standards

Conflict of interest The author declares that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material

is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

9 Appendix

This section presents a graphical abstract which provides an overview of the work completed and illustrates the overall findings as Fig. 27.

Fig. 27 Graphical Abstract

References

- Advani, G., Smith, B., Tanabe, Y., Irick, K., Cotter, M., Sampson, J., Narayanan, V.: Visual co-occurrence network: using context for large-scale object recognition in retail. In: 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia), pp. 1–10 (2015)
- Akkas, A.: The impact of shelf space on product expiration. <https://ssrn.com/abstract=3095346> (2018)
- Andersen, K.: The Geometry of an Art: The History of the Mathematical Theory of Perspective from Alberti to Monge. Springer, New York (2007)
- Baz, Y.E., Çetin, M.: Retail product recognition with a graphical shelf model. In: 25th Signal Processing and Communications Applications Conference (2017)
- Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Newton (2017)
- Brooke, J.: SUS: a “quick and dirty” usability scale. Usability Evaluation in Industry, Taylor and Francis, London (1996)
- Brooke, J.: Sus: a retrospective. *J. Usabil. Stud.* **8**(2), 29–40 (2013)
- Brownlee, J.: Better deep learning: train faster, reduce overfitting, and make better predictions. *Mach. Learn. Mastery* (2018)
- Bukhari, F., Dailey, M.N.: Automatic radial distortion estimation from a single image. *J. Math. Imaging Vis.* **45**, 31–45 (2013)
- Castelo-Branco, F., Reis, J.L., Vieira, J.C., Cayolla, R.: Business intelligence and data mining to support sales in retail. In: Marketing and Smart Technologies, Springer Singapore, pp. 406–419 (2020)
- Ceren, M.G., Elena, S.B., Albayrak, S.: A survey of product recognition in shelf images. In: 2017 International Conference on Computer Science and Engineering, IEEE, p.p 146–150 (2017)
- Damisch, H.: The Origin of Perspective. MIT Press, Translated by John Goodman (1994)
- Drèze, X., Hoch, S.J., Purk, M.E.: Shelf management and space elasticity. *J. Retail.* **70**(4), 301–326 (1994)
- Elkan, C.: Evaluating classifiers. <http://cseweb.ucsd.edu/~elkan/250B/classifiereval.pdf> (2011)
- Faruqui, N.: Open Source Computer Vision for Beginners: Learn OpenCV Using C++ in Fastest Possible Way, 2nd edn. Amazon Publishing, Seattle (2017)
- Forman, G., Scholz, M.: Apples to apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explor.* **12**(1), 49–57 (2010)
- Franco, A., Maltoni, D., Papic, S.: Grocery product detection and recognition. *Expert Syst. Appl.* **81**(15), 163–176 (2017)
- Hafiz, R., Islam, S., Khanom, R., Uddin, M.S.: Image based drinks identification for dietary assessment. In: Computational Intelligence (IWCI) International Workshop, pp. 192–197 (2016)
- Howard, H.J.: Confusion matrix. http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html (2011)
- ImageNet: Imagenet. <http://image-net.org> (2019)
- Kohavi, R., Provost, F.: Glossary of terms: special issue on applications of machine learning and the knowledge discovery process. *Mach. Learn.* **30**, 271–274 (1998)
- Liu, S., Li, W., Davis, S., Ritz, C., Tian, H.: Planogram compliance checking based on detection of recurring patterns. *Comput. Vis. Pattern Recogn.* (2016)
- Lu, Z., Szafron, D., Greiner, R., Lu, R., Wishart, D.S., Poulin, B., Anvik, J., Macdonell, C., Eisner, R.: Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics* **20**(4), 547–556 (2004)
- Mankodiya, K., Gandhi, R., Narasimhan, P.: Challenges and opportunities for embedded computing in retail environments. In: Martins, F., Lopes L., Paulino H. (eds.) *Sensor Systems and Software. S-CUBE 2012.*, Springer, Berlin, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 102, pp. 121–136 (2012)
- MATLAB: Matlab–mathworks–matlab & simulink. <https://www.mathworks.com/products/matlab.html> (2020)
- Mentzer, J.T., Min, S., Zacharia, Z.G.: The nature of interfirm partnering in supply chain management. *J. Retail.* **76**(4), 549–568 (2000)
- Moreau, P.C.: Brand building on the doorstep: the importance of the first (physical) impression. *J. Retail.* (2020). <https://doi.org/10.1016/j.jretai.2019.12.003>
- Porter, E., Tuong, N.H.: Image recognition can help consumer goods manufacturers win at the retail shelf. Gartner Research (2017)
- Pouyanfar, S., Sadiq, S., Yan, B., Tian, H., Tao, Y., Reyes, M.P., Shyu, M.L., Chen, S.C., Iyengar, S.S.: A survey on deep learning: algorithms, techniques, and applications. *ACM Comput. Surv.* **51**(5), 1–36 (2019). <https://doi.org/10.1145/3234150>
- Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern J. Mag.* **39**, 1137–1149 (2015)
- Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., Schmidt, L.: A meta-analysis of overfitting in machine learning. In: *NeurIPS* (2019)
- Snoek, J., Rippel, O., Swersky, K., Ryan, K., Satish, N., Sundaram, N., Prabhat, M.M.A.P., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: *Proceedings of the 32nd International Conference on Machine Learning, W&CP*, vol. 37 (2015)
- Soltia, A., Raffela, M., Romagnoli, G., Mendling, J.: Misplaced product detection using sensor data without planograms. *Decis. Support Syst.* **112**, 76–87 (2018)
- Song, Y., Xue, Y., Li, C., Zhao, X., Liu, S., Zhuo, X., Zhang, K., Yan, B., Ning, X., Wang, Y., Feng, X.: Online cost efficient customer recognition system for retail analytics. In: 2017 IEEE Winter Applications of Computer Vision Workshops (WACVW), IEEE (2017)
- Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. *arXiv preprint arXiv:1708.04896* (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Edward R. Sykes has an Honours B.Sc. (1992) in Mathematics and Computer Science (Summa Cum Laude) from McMaster University, an M.Sc. (2005) in Computer Science from McMaster University and a Ph.D. in Computer Science from Guelph University. He also has a B.Ed. (1993) from Western University, an M.Ed. (1998) and a Ph.D. (2006) in Education (Cognition and Learning) from Brock University. Ed is the Director for Sheridan College's Centre for Mobile Innovation (CMI). The

CMI's focus is on mobile health, creating innovative solutions to industry focuses problems by using technologies, such as, IoT, Wearable Computing, AR/VR, and Machine Learning.