University of Washington

Classical and Deep Methods for Computer Vision
EE P 596

Report: Pull your Shelf together

Authors: Sreejith S. , Rishabh Gupta

December 17, 2021

UNIVERSITY *of* WASHINGTON

Introduction

Our goal is to perform product classification in a grocery mall setup and identify misplaced products to save time and effort for the employees. It would also help customers find desired products with ease.

Problem relevance and Motivation

With a massive increase in product categories, the logistics of creating a good product placement has become taxing for the staff in supermarkets. Moving the misplaced products to their correct position adds to the time and effort of the employees. This translates to huge inefficiencies and is a precursor to monetary losses and poor customer service.

Our methodology will help the staff to automatically identify misplaced products instead of manually perusing through each item within the shelves. This will speed-up correct product placement on shelves, allowing the staff more time to focus on helping and guiding the customers.

Prior work

Traditional techniques for shelf digitization utilize local invariant features and the technique of sub-graph isomorphism between the items appearing in the given shelf and the ideal shelf layout [TD17].

Modern techniques utilize the power of deep learning to understand spatial information within the shelves. [YB21] uses YOLOv4 to monitor OSA (On Shelf Availability). They solve the annotation problem by proposing a new method that combines the two concepts of "semi-supervised learning" and "on-shelf availability" (SOSA).

Initiatives like Amazon Go have fusion sensors, proximity sensors, and LIDAR scanners to identify misplaced items on a shelf if image recognition systems fail, especially when light conditions are inadequate [Gro19].

The key contributions of the work are:

1. *Evaluating pretrained ResNet50 [He+15] network for the task of product classification in a supermarket.*
2. *Training ResNet20 V1 from scratch for the product classification task.*
3. *Proposing a strategy for identifying misplaced products on the shelves of supermarkets.*

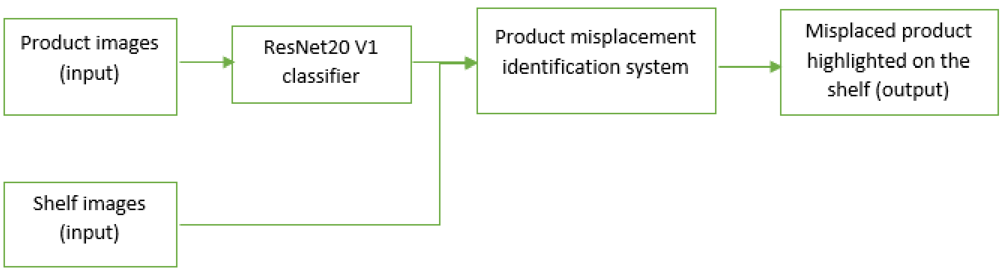


Figure 1: A high-level block diagram of the system

Method

We use [VS15] as the baseline for our product detector and classifier. The comparison of our proposed approach with the baseline approach is given in Table 1. To the best of our knowledge, there has been no previous work on misplaced product identification on the Grocery dataset [Var21].

We implemented the product position outlier detection over shelf images in the Grocery dataset [Var21] through the following steps:

1. Product detection in the densely packed shelf is performed using bounding boxes created by skilled annotators, and provided as a part of the dataset.

Table 1: Comparison of Baseline and proposed approach

Task	**Baseline**	**Proposed approach**
Detection	The Cascade object detection framework of Viola and Jones [VJ]	Bounding boxes created by skilled annotators
Classification	Support Vector Machines (SVMs)	ResNet20 V1

2. Classification of the products (which were detected in Step 1) is performed using ResNet20 V1.
   (a) First, we downsize the product images to 120x80 and normalize (division by 255) before feeding them to the classifier.
   (b) We split the data into train/test in a 70/30 ratio.
   (c) Initially, we explored the product classification task of ten classes in the Grocery dataset [Var21] using the pretrained ResNet18 [He+15] framework.
   (d) To improve the results, we experimented by training ResNet20 V1 for 12 epochs with a learning rate of 0.001 and a batch size of 50.
   (e) We get the one-hot encoded output of the product images during testing, representing the corresponding product class/label.
   (f) The accuracy is chosen as the evaluation metric for the classification task.
3. We identify misplaced products through the following procedure:
   (a) Crop out each detected product in Step 1.
   (b) Predict the class of the cropped products using the trained model from Step 2.
   (c) Now, we have a row vector (dimension: 1 x number of products in a shelf row) with location and the predictions of the class of each product on a shelf.
   (d) To get the misplaced object, we incorporate strategies to compare the neighboring elements of each of the predictions in the row vector.
   (e) We highlight the misplaced product with a red-colored bounding box and also record its location coordinates.

Base strategy for product misplacement identification

Strategy for comparing predicted classes of the neighbours:

1. In the case of a product on the extreme left and extreme right of a row in a shelf, we look at their single immediate neighbour.
2. For each other products, we look at the class of the immediate left and right neighbours.

If the predicted class of the product matches with the predicted class of its neighbour(s), we can conclude that the project is not misplaced. If the classes do not match, the object is considered to be misplaced.

However, this strategy fails in some extreme cases. Let's say we have three classes of products (say, 1, 2, 3) placed in the below pattern:

111222333<u>11</u>22333311111

In this scenario, none of the products are classified as misplaced even though the products 11 (underlined) in the middle is misplaced and belongs to one of the ends of the shelf.

The proposed method is summarised using Figure 2. The red blocks represent the preprocessing/training phase, and the green blocks denote the testing phase.

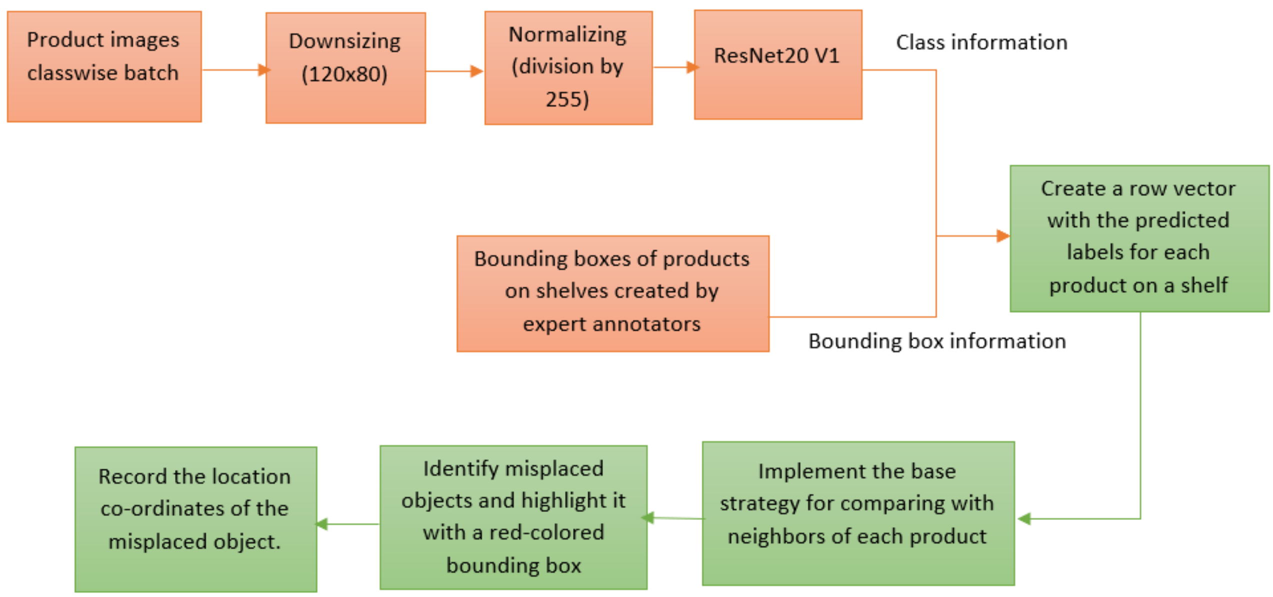


Figure 2: Block diagram of the entire system

Results

Table 2 compares the classification accuracy of the baseline classifier (Support Vector Machine) [VS15], the pretrained ResNet50 [He+15] network and our trained ResNet20 V1 (Vanilla) product classifier:

Table 2: Comparison of product classification accuracy

Method	**Accuracy**
Baseline [VS15]	85.90%
Pretrained ResNet50 [He+15]	69.91%
Proposed ResNet20 V1	83.26%

Out of the two experimented approaches, the ResNet20 V1 based classifier reports an accuracy of 83.26% which is comparable to the result obtained for the baseline classifier model.

For misplaced product identification, we compare the predicted classes of the neighbours of each product. The results are illustrated in Figure 3 (a) and Figure 3 (b).

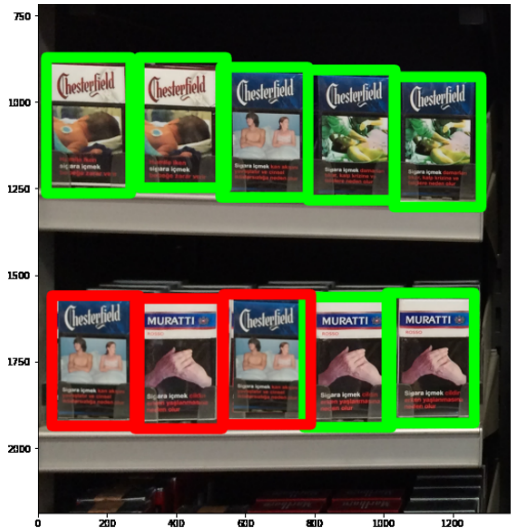
Conclusion

We proposed a strategy for identifying misplaced products on the shelves of a supermarket. We trained our classifier on a ResNet20 V1 backbone to classify products from the Grocery dataset [Var21] into ten classes. Our classification network was able to achieve an accuracy of 83.26%. We identified misplaced objects by comparing the predicted classes of their neighbours.

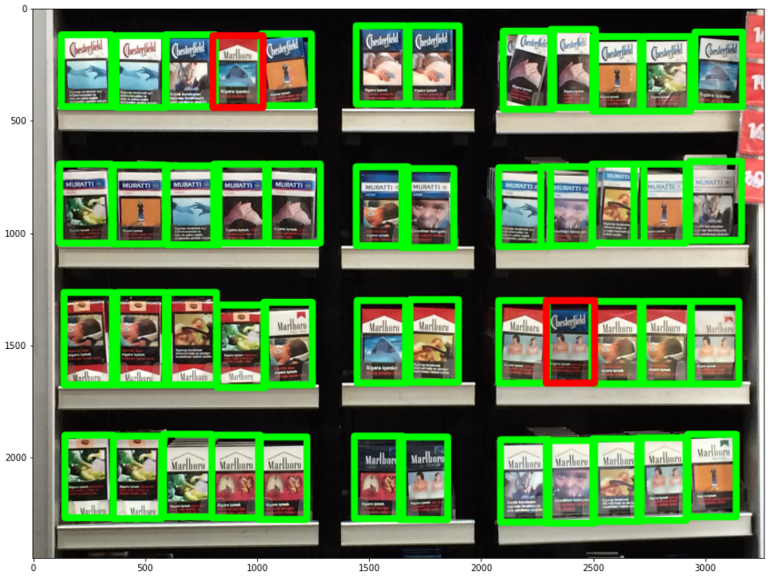
Learnings from the project:

1. Though we had less exposure initially, we explored the problem landscape and how a lot of SOTA models do not perform well for densely packet arrangements. We researched different methods to eliminate overlap between bboxes and detect the product boundaries to effectively find product misplacement.

2. We explored different models (Detectron2, RetinaNet, DenseNet, and SSD MobileNet) to solve the problem of product detection and classification. There were a lot of complicated strategies, but we focused on developing a simpler solution.



(a)



(b)

Figure 3: Identifying misplaced objects on the shelves (Red bounding box indicates misplaced objects.)

3. We implemented end-to-end pipeline for finding misplaced products.

The future extensions and scope of the project are as follows:

1. Product detection can be made over a live video taken from different perspectives.
2. Detecting misplaced objects can be done using a deep neural network based anomaly detection approach instead of using the bounding boxes created by skilled annotators.
3. An improved strategy which is capable of predicting misplaced products in all arrangement scenarios can be developed.
4. Additional functionalities like suggesting the correct placement location of the product, keeping a count of the inventory, etc., can be added to the system.
5. Different kinds of gaps in the shelves (a. Separation gaps between racks and b. gaps between products on the shelves) can be identified. The information about the product gaps can be used to detect out-of-stock items.

Appendix A

The code we use to train and evaluate our approach is available at https://github.com/sree1999/Pull-your-shelf-together.

Appendix B

Dataset

We use the Grocery dataset [Var21].

The Grocery Dataset [Var21] is an image dataset with:

1. 354 grocery shelf images were collected from approximately 40 stores, with four cameras.
2. Products from 10 different categories numbered from 1-10. Each category has a separate directory. There are 2744 product images in total.

Figure B.1: Shelf images in the Grocery dataset [VS15])

Figure B.2: Product classes in the Grocery dataset [VS15])

Bibliography

[He+15] Kaiming He et al. *Deep Residual Learning for Image Recognition*. arXiv.org, 2015. URL: https://arxiv.org/abs/1512.03385.

[VS15] Gül Varol and Rıdvan Salih. *(PDF) Toward retail product recognition on grocery shelves*. ResearchGate, Mar. 2015. URL: https://www.researchgate.net/publication/301439956_Toward_retail_product_recognition_on_grocery_shelves.

[TD17] Alessio Tonioni and Luigi Di Stefano. "Product recognition in store shelves as a sub-graph isomorphism problem". In: *arXiv:1707.08378 [cs]* (Sept. 2017). URL: https://arxiv.org/abs/1707.08378 (visited on 12/17/2021).

[Gro19] Ryan Gross. *How the Amazon Go Store's AI Works*. Towards Data Science, June 2019. URL: https://towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9.

[Var21] Gul Varol. *Grocery Dataset*. GitHub, Nov. 2021. URL: https://github.com/gulvarol/grocerydataset.

[YB21] Ramiz Yilmazer and Derya Birant. "Shelf Auditing Based on Image Classification Using Semi-Supervised Deep Learning to Increase On-Shelf Availability in Grocery Stores". In: *Sensors* 21 (Jan. 2021), p. 327. DOI: 10.3390/s21020327. URL: https://www.mdpi.com/1424-8220/21/2/327.

[VJ] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (). DOI: 10.1109/cvpr.2001.990517.