Empty Shelf Detection

Group No: 13

Dhruvisha Patel
M.Sc.
ComputerScienceLakeheadUniversity,
Thunder Bay,Canada
(E) dpate138@lakeheadu.ca

Krishna Gandhi
M.Sc.
ComputerScienceLakeheadUniversity,
Thunder Bay,Canada
(E) kgandhi1@lakeheadu.ca

Niharika Sojitra
M.Sc.
ComputerScienceLakeheadUniversity,
Thunder Bay,Canada
(E) nsojitra@lakeheadu.ca

Palak Patel
M.Sc. Computer Science
Lakehead University,
Thunder Bay, Canada
(E)ppate109@lakeheadu.ca

Abstract— Out of Stock detection (OOS) has the key role to improve on-shelf availability of products in the supermarket. This project proposes a system that actively detects empty spaces from densely packed racks in retail stores. We approached a currently used method of Object detection in which the features of an image is outlined by bounding box. In contrast, our proposed method of image segmentation that robustly detects the voids as the Region of Interest (RoI) which is the input to the machine learning model. The outcome of this model is the mask that indicates whether the void is present or not.

Index Terms—image segmentation, bounding box augmentation, RCNN,annotations,RoI.

I. INTRODUCTION

In the past few years, the popularity of research in computer vision has been increasing rapidly. To teach computers how to see the world through the human eyes is the most interesting challenge to accomplish because the visual world is vast and complex. There has been development of new models and efficient algorithms towards computer vision problems. One application of this computer vision that has highly gained attention is object detection. Object detection model trains a computer to find all the objects in the given image and draw a bounding box on them. One of the major roles of object detection is in the supermarket and convenience store, to check on-shelf availability of products. When customers see a desired product is not on shelf, they purchase a similar product from the competent retailer. Therefore, on-shelf availability is the key factor for improving profit in a business for retailers. In order to improve profit, products need to be restocked frequently in a manner that should be at the front of the shelves. In the present work, we define a system that finds empty shelves and acknowledges retailers or employees about no on-shelf availability of products that should be restocked on an emptyshelf.

On shelf availability maintenance is a very tedious task for clerks and employees as they have to look around a store frequently and need to replenish the products in such a way that they should be at the front. However, this is completely labor intensive work. Therefore, to address the challenge of Out of Stock (OOS), a method is needed to automatically detect the OOS by extracting image features focusing on empty regions without product template matching or feature matching. Several other factors such as multiple aisle geometry, artificial light intensity and reflections will affect the automatic recognition of emptyracks.

II. RELATED WORK

Audits performed by store employees for manual inspection of shelves is a method that is labor-intensive and lacks reliability because of factors, such as employee availability and the time of audit. The very first approach for automatic OOS detection is to monitor inventory data to calculate stock-outs. However, inaccurate inventory information may lead to cases where the product might be in the store but not on the shelf. Therefore, only inventory data alone is not enough for OOS detection.

RFID (Radio-Frequency Identification) is one of the approaches used for tracking products in the current supply chain. Even though it achieves better accuracy for OOS detection and improves on-shelf availability, its use is hindered by its cost [4]. Since it is not cost effective, only a set of well-established retailers have implemented thismethod.

Weight sensing is another approach used out-of-stock detection. However, due to the same reasons as RFID, it is not used widely. This approach requires wires and sensors connected all over the store along with constant monitoring. Costs for such an infrastructure is very high, even after which the model's accuracy is sensitive to human errors such as misplaced products leading to incorrect readings due to change in weight.

Most of the retail stores have now started using computer systems for inventory management and CCTV cameras for surveillance. This allows us to implement computer-based detection approaches for OOS estimations instead of physical store audits.

Several image processing techniques have been proposed to monitor on-shelf availability in retail shelves. Moorthy et al. [3]have applied feature matching using SURF algorithm for reference images to achieve the goal. Using on-shelf objects as the points of interest, voids were recognized as the regionsthat lacked a key point. This made the matching process more difficult.

Gonc¸alves et al. [1] have used panorama stitching to produce high resolution shelf images from multiple viewpoints. They have further applied image segmentation along with binary masking to the cropped RoI, to obtain a clear grayscale image with minimized noise. Void spaces are therefore identified from these processed images, using OOS segmentation. In order to train the label classifier, 185 label images are manually annotated from 297 high resolution fisheye images, not included in the set used for panoramic generation [1]. To build the classification module for OOS detection, 272 OOS situations from the 23 panoramic images were manually annotated by two experienced professionals in the retail area [1]. The performance evaluated by this approach showed quite good results.

It is quite clear that many of the existing approaches use the on-

shelf products as the region of interest (RoI) and then use different techniques to find voids. Adopting the understanding from[2], this project proposes a method using image segmentation technique for object detection on the SKU110K dataset. However, the void spaces present in the image will be used as RoI during the model training. Training images will be manually annotated to identify the void spaces in them. These images will then be fed to the Masked R-CNN model for training, which will be further tested with the plain test images, to get experimental results.

III. PROPOSED METHOD

A. ModelFitting

Once the data has been processed, it has to be fed into a Convolutional Neural Network for Model Training. The network should successfully be trained to identify voids using the annotated data for further use. Various networks have been proposed till today.

R-CNN:

Region-based CNN (R-CNN) is a simple approach for bounding-box detection to retrieve maximum object detection accuracy. It takes into accountRoI to evaluate the convolutional networks independently. However, RCNN takes a lot of time for extracting the region proposals (2000 per image), which in turn consumes a lot of memory. Considering the size of the SKU110K dataset, RCNN approach does not seem feasible for model training.

Faster RCNN:

Faster R-CNN improved the performance by learning the attention mechanism with a Region Proposal Network (RPN). A separate network is built to predict the region proposal instead of the selective search algorithm. Shaoqing Ren et al
[5] proposed such a model that could let the network learn the region proposals. Class label and a bounding box offset are the two outputs that are generated by this network.

Mask RCNN:

Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs [6]. Mask R-CNN extends Faster R-CNN by adding a third branch to the output, that is the objectmask.
It performs object detection to classify individual objects and localize each using a bounding box and semantic segmentation to further classify each pixel into a fixed set of categories without differentiating the object instances.
The two stage procedure of Mask R-CNN has been taken from Faster R-CNN. Region Proposal Network produces candidate object bounding boxes.
However, in the second stage, Mask R-CNN differs from Faster R-CNN as it produces a binary mask for each RoI along with the class label and box offset.

IV. EXPERIMENTAL SETUP

A. Dataset Description

We have used SKU-110K dataset [7] which contains a total 11,762 images including 8233 images for training, 2941 images for testing and 588 images for validation. All the images are taken from different super markets under various constraints such as various scales, lighting conditions, viewing angles and noise levels. This dataset considers different scenarios that maybe encountered in the real-time CCTV recordings of a superstore, for example a person standing in front of the aisle. Some example images are as shown in Fig [1].

Fig. 1

B. Data Preprocessing

To detect objects from the images, annotation files for train and test data are provided with this dataset. However, in our proposed method, RoI is different. Therefore manual annotation for void spaces is needed.

First and foremost, the images were resized to a size of 256 x 256 pixels, since the SKU-110K dataset had all the images in different height, width and size. This preprocessing step introduced uniformity in the dataset and had a huge impact on the computation time.
The next step was to annotate the images using VGG Image Annotator (VIA) [8] version 2.0.11 for identifying the coordinates of the void spaces in training images. Polygons are used instead of rectangles for a better and precise annotation.
Only one region attribute “name” with default value “void” was used, resulting in two classes: void and background. An image could have single or multiple instances of the *void* class. Fig [2] displays details of the VGG Image Annotator tool used in this project.

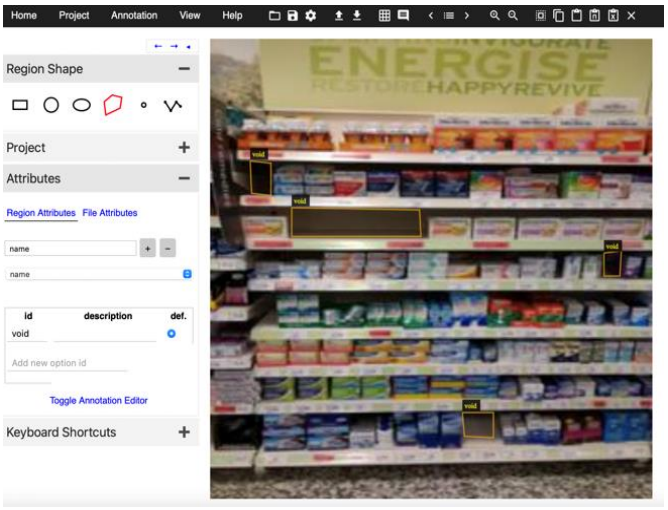


Fig. 2

Annotation files for the training and testing dataset are saved in all the available formats including CSV, JSON and COCO format, for flexibility in future use. Annotation file in JSON format was used ahead in this project because of its compatibility with the MRCNN module [Link to Matterport MRCNN reference] used.

Now, we have created a subset of this large SKU-110K dataset containing 600 manually annotated images. Among them 300 images are used for training and 50 images are used for testing the model due to the hardware constraints.

V. MODEL ARCHITECTURE AND IMPLEMENTATION

A. Requirements and Set-Ups

The Mask-RCNN base architecture model generates bounding boxes and segmentation masks for each instance of an object in the image[9]. Resnet101 is the backbone for this model. Pre-trained weights from MSCOCO have been used to implement a transfer learning approach during the training of this model. The main Mask RCNN implementation is contained in three files: *model.py* that includes the entire model architecture, *utils.py* containing the intermediate functions for the model and *config.py* that defines different configuration parameters for the model architecture.

MRCNN Architecture requires TensorFlow version 2.6.0 tightly integrated with Keras that provides a high level API used to train the Mask RCNN model. TensorFlow object recognition algorithms can robustly classify and identify objects from larger images.

We have used Kaggle Environment to execute our code because Kaggle provides free access to NVIDIA TESLA P100 GPUs to train deep learning models. Also, Kaggle provides a 16GB GPU for 9 hours straight in a single commit and 40 hours per week. Another benefit of Kaggle is that it has a large number of datasets that can be simply used without downloading them.

B. Implementation

To implement the Matterport MRCNN[6] model with a custom dataset, the configurations and dataset have to be customised to match the compatibility with the base architecture.

MRCNN model was trained using the pre-trained weights from MS-COCO [6] . A transfer learning approach was used for faster and better training of the model. Transfer Learning approach is a machine learning technique that uses knowledge, in this case, weights from an already trained model to a different model that solves a similar or related problem. COCO dataset is the most accurate dataset for object detection. It already identifies many objects that can be found on a superstore shelf.

A separate class has been used that defines the custom configurations for the model to be trained. Firstly, the learning rate for the model is set to a value of 0.01. A lower value than this resulted in a very slow training. With the LEARNING_RATE = 0.001 , training of a dataset as small as 15 images for 30 epochs resulted in a Wall time of 3 hours. Thus LEARNING_RATE=0.01 proved to be an optimal value that allowed each epoch to complete in 92s for 300 images.

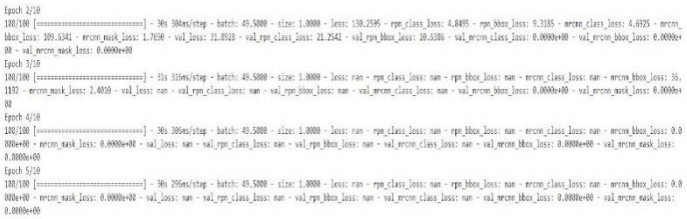
IMAGE_MIN_SIZE and IMAGE_MAX_SIZE are the two other variables that require customer configuration for each dataset. Different datasets have images of different sizes. Since, all the images were set to a standard size in the preprocessing step, the same values are used for these two variables. Model was trained in Kaggle environment that had an upper cap on the memory used as input or output. Hence, an image size of 256 x 256 is well suited for model training and also with the resource availability.

To filter the predicted regions in the output a minimum confidence level is defined. DETECTION_MIN_CONFIDENCE is set to 0.9. This means the model will only those predicted regions of the output, where it is 90% confidence. Lower the value, lower the regions are discarded. 90 becomes an optimal value because a value higher than this may increase *false negatives*, and a value lower than this tends to increase *false positives*.

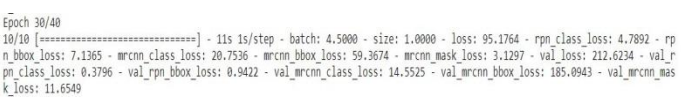
The dataset was then customized to fit it in the final model. After the preprocessing step, the dataset consists of 300 images for training and their annotation file in JSON format. Annotations available in the JSON format are easy to understand, parse and more compatible with the underlying Mask RCNN model architecture.

IMAGES	EPOCHS	STEPS_PER_EPOCHS	LEARNING_RATE
300	20	100	0.01
300	20	25	0.01
300	30	10	0.01

VI. RESULTS



IMAGES = 300. EPOCHS = 20, STEPS_PER_EPOCH = 100, LEARNING_RATE = 0.01



20 epochs 100 steps per epoch 300 images

20 epochs 25 SPE 300 images

```
Epoch 2/10
100/100 [==============================] - 30s 304ms/step - batch: 49.5000 - size: 1.0000 - loss: 130.2595 - rpn_class_loss: 4.8495 - rpn_bbox_loss: 9.3185 - mrcnn_class_loss: 4.6925 - mrcnn_
bbox_loss: 109.6341 - mrcnn_mask_loss: 1.7650 - val_loss: 31.8928 - val_rpn_class_loss: 21.2542 - val_rpn_bbox_loss: 10.6386 - val_mrcnn_class_loss: 0.0000e+00 - val_mrcnn_bbox_loss: 0.0000e+
00 - val_mrcnn_mask_loss: 0.0000e+00
Epoch 3/10
100/100 [==============================] - 31s 316ms/step - batch: 49.5000 - size: 1.0000 - loss: nan - rpn_class_loss: nan - rpn_bbox_loss: nan - mrcnn_class_loss: nan - mrcnn_bbox_loss: 36.
1192 - mrcnn_mask_loss: 2.4010 - val_loss: nan - val_rpn_class_loss: nan - val_rpn_bbox_loss: nan - val_mrcnn_class_loss: nan - val_mrcnn_bbox_loss: 0.0000e+00 - val_mrcnn_mask_loss: 0.0000e+
00
Epoch 4/10
100/100 [==============================] - 30s 306ms/step - batch: 49.5000 - size: 1.0000 - loss: nan - rpn_class_loss: nan - rpn_bbox_loss: nan - mrcnn_class_loss: nan - mrcnn_bbox_loss: 0.0
000e+00 - mrcnn_mask_loss: 0.0000e+00 - val_loss: nan - val_rpn_class_loss: nan - val_rpn_bbox_loss: nan - val_mrcnn_class_loss: nan - val_mrcnn_bbox_loss: 0.0000e+00 - val_mrcnn_mask_loss:
0.0000e+00
Epoch 5/10
100/100 [==============================] - 30s 296ms/step - batch: 49.5000 - size: 1.0000 - loss: nan - rpn_class_loss: nan - rpn_bbox_loss: nan - mrcnn_class_loss: nan - mrcnn_bbox_loss: 0.0
000e+00 - mrcnn_mask_loss: 0.0000e+00 - val_loss: nan - val_rpn_class_loss: nan - val_rpn_bbox_loss: nan - val_mrcnn_class_loss: nan - val_mrcnn_bbox_loss: 0.0000e+00 - val_mrcnn_mask_loss:
0.0000e+00
```

VII. DISCUSSION

In Supervised learning for Out-of-Stock detection in panoramas of retail shelves[1], On Shelf Availability (OOS) classification results are presented in 3 matrics. First is sensitivity matrix which detects labels with accuracy of 86.6%. Second matrix is a specificity matrix which gives 84.5% accuracy for OOS detection and finally 79.4% overall accuracy is achieved. Since we are not using any label detection technique, sensitivity matrix is not considered in our approach. Also, it takes average 2 minutes to generate panoramic image. However, in our system, images are annotated already. So that it reduces the computation time

VIII. LIMITATIONS

To annotate such a large dataset, limited resource availability puts a cap on the images that could be train for an optimal number of epochs. Using a much larger dataset with more number of epochs may have resulted in a better outcome but exhausts the memory available in the used environment.

IX. CHALLENGES

A. *Hardware requirements*

As we have used SKU 110K dataset containing a large number of images around 11,000 which require a powerful processor such as GPU or i7/i9 having minimum of 16 GB ram.

B. *Data Annotation*

Data annotation is the biggest challenge for us because our dataset is containing a large number of images of approximately 11k and we have to manually annotate void in each image. That task is very time consuming.

X. CONCLUSION

The proposed project demonstrates a novel approach to find voids that represents empty shelves in grocery stores using image processing techniques and a Mask-RCNN model of deep learning.

In the future,this approach can be extended by implementing a system that captures images of aisles in the grocery store by using CCTV cameras and notifies store personnel about the on shelf availability of products in real time.

XI. ACKHNOWLEDGEMENT

We would like to extend our sincere thanks toDr. Garima Bajwa, (Lakehead University) for guiding us and giving us her valuable time and advice. She always enriched us with her knowledge and gave us the necessary input to carry out this work. We are grateful to her for her extra efforts and for being patient with us.

XII. REFERENCES

[1] L. Rosado, J. Gonçalves, J. Costa, D. Ribeiro and F. Soares, "Supervised learning for Out-of-Stock detection in panoramas of retail shelves," *IEEE International Conference on Imaging Systems and Techniques (IST)*, no. 5, pp. 406 - 411, 2016.

[2] K. Higa and K. Iwamoto, "Robust Shelf Monitoring Using Supervised Learning for Improving On-Shelf Availability in Retail Stores," *Sensors*, vol. 19, no. 12, p. 2722, 2019.

[3] R. Moorthy, S. Behera, S. Verma, S. Bhargave and P. Ramanathan, "Applying Image Processing for Detecting On-Shelf Availability and Product Positioning in Retail Stores," *WCI '15: Proceedings of the Third International Symposium on Women in Computing and Informatics*, pp. 451-457, 2015.

[4] R. Moorthy, S. Behera and S. Verma, "On-Shelf Availability in Retailing," *International Journal of Computer Applications(0975 – 8887)*, vol. 115, no. 23, pp. 47-51, 2015.

[5] S. Ren, K. He, R. B. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137 - 1129, 2015.

[6] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2961-2969, 2019.

[7] E. Goldman, "SKU-110K," [Online]. Available: https://github.com/eg4000/SKU110K_CVPR19.

[8] A. Dutta, A. Gupta and A. Zisserman, "VGG Image Annotator (VIA)," Visual Geometry Group,Department of Engineering Science, University of Oxford, [Online]. Available: https://www.robots.ox.ac.uk/~vgg/software/via/.

[9] Esri, Nvidia, M.-D. County, D. Kudinov, D. Hedges and O. Maher, "Mask R-CNN for Object Detection and Segmentation," 2018. [Online]. Available: https://github.com/matterport/Mask_RCNN.

Participation Log:

Name	Student Id	FDE Section – Nov 25	FDE Section – Nov 30	FDE Section – DEC 02
Dhruvisha Patel	1159961	ATTENDED	ATTENDED	ATTENDED
Krishna Gandhi	1170559	ATTENDED	ATTENDED	ATTENDED
Niharika Sojitra	1170232	ATTENDED	ATTENDED	ATTENDED
Palak Patel	1166610	ATTENDED	ATTENDED	ATTENDED