MSc in Business Analytics

Machine Learning & Content Analytics

Empty Selves Stock Detection on Supermarkets

Instructor: Mr. Haris Papageorgiou

Kalavasis Pavlos - (AM: p2822116)

Papazisis Zisis - (AM: p2822127)

Tsonos Grigorios - (AM: p2822135)

Tzimas Ferdinandos - (AM: p2822132)

Athens, 14/09/2023

Content of tables

EMPTY SHELVES DETECTION

Introduction

With the coming of age of Artificial Intelligence, everyone wants to see how they can use their data to solve their business problems. For example, some retail stores are silently working on how they can stay competitive by predicting empty supermarket shelves. This leads to the idea that the store can automatically track which items are being pulled off the shelves and take the guesswork out stocking. With the advancements and neural networks and artificial intelligence, we wanted to see if we could predict the empty spaces on supermarket shelves, given a limited amount of data and no initial bounding boxes to train on. This problem is already solved when we have bounding boxes to train on.

Object detection has rapidly evolved to revolutionize industries through its ability to recognize and localize specific objects within images or videos. In the context of supermarket management, object detection technology is poised to empower businesses with the capability to pinpoint vacant shelf spaces in real time. By harnessing the prowess of deep learning algorithms, pretrained models, and data-driven insights, AI-driven object detection systems can discern not only the presence of products on shelves but also the absence thereof, offering a panoramic view of shelf occupancy and stock availability.

This technological endeavor not only enhances inventory control but also shapes customer satisfaction. The ability to preemptively restock depleted shelves improves supply chain efficiency, reduces customer frustration arising from out-of-stock products, and provides valuable insights for optimizing product placement strategies. The amalgamation of AI and object detection heralds an era of responsive retail environments, where data-driven decisions streamline operations, elevate customer experiences, and amplify business success.

In this exploration, we delve into the multifaceted realm of object detection, unveiling the mechanisms that underpin the identification of empty shelves in supermarkets. We navigate the intricacies of dataset collection, model selection, fine-tuning strategies, and deployment considerations. Through this journey, we lay the foundation for a solution that not only mitigates a common retail challenge but also illuminates the transformative potential of artificial intelligence in modern commerce.

Our Project

The success of deep learning to solve complex problems is not hidden from anyone these days. Deep learning plays an important role to automate problems in all walks of life. In this document, we have used the different deep learning-based

algorithm, to detect empty inventory in grocery stores. Usually, when we go to a grocery store, and we see a shelf that doesn't have the product we need, then many customers will leave without asking the store workers if they have that item. Even if the store had that item in their warehouse. This can cause the store to lose out on potential sales for as long as the inventory remains empty. We have used machine learning models to help stores replenish inventory quickly so that they don't lose customers and sales. Currently supermarkets are more popular, and the local stores are leaving the competition. when people go to supermarkets, they find various items stocked on seemingly unlimited shelves.

Supermarket shelves needed to be filled with the items accordingly. The most common problems in the supermarkets are identifying the empty shelves, on-shelf availability, and future sales. The labors cannot always track the empty shelves and on shelf availability levels due to their workloads. Moreover, it is a time-consuming method for the labors which can affect the customer satisfaction and business profit. Every month, supermarkets buy the required number of products from related manufacturing companies by analyzing the previously purchased products and their sales. This is usually done manually by managing excel sheets which is also time consuming and not reliable. Especially during the seasonal times or pandemic situations they cannot use the manual method which must also be done as fast as possible. Therefore, this system can be used to assist in empty shelf detection, percentage of on-shelf availability and in the prediction of future sales. The implementation of on-shelves percentage detection service is done using machine learning. Machine learning processes are carried out for implementing the necessary functionalities and algorithms. Initially, the camera captures clear and real time images regularly. Then the system processes and detects the image like the threshold percentage or detect the empty shelves. When the system detects the threshold percentage or empty shelves, the system will provide an alert to the labors. The Implementation of the predicting the future supply and demands is done using time series analysis using several existing machine learning algorithms by utilizing historical data. In this research the prediction of future sales and demand in the supermarkets is done by considering the customers' behavior, the variety of product groups they buy and seasonal changes. These predictions are made on the assumption of a constant per capital supply of products and demand in our system.

Data Collection

SKU-110K images were collected from thousands of supermarket stores around the world, including locations in the United States, Europe, and East Asia. Dozens of paid associates acquired our images, using their personal cellphone cameras. Images were originally taken at no less than five mega-pixel resolution but were then JPEG compressed at one megapixel.

Otherwise, phone and camera models were not regulated or documented. Image quality and view settings were also unregulated and so our images represent different

scales, viewing angles, lighting conditions, noise levels, and other sources of variability. Bounding box annotations were provided by skilled annotators. We chose experienced annotators over unskilled, as we found the boxes obtained this way were more accurate and did not require voting schemes to verify correct annotations. We did, however, visually inspect each image along with its detection labels, to filter obvious localization errors.

Dataset Overview

The Sku110k dataset provides 11,762 images with more than 1.7 million annotated bounding boxes captured in densely packed scenarios, including 8,233 images for training, 588 images for validation, and 2,941 images for testing. There are around 1,733,678 instances in total. The images are collected from thousands of supermarket stores and are of various scales, viewing angles, lighting conditions, and noise levels. All the images are resized into a resolution of one megapixel. Most of the instances in the dataset are tightly packed and typically of a certain orientation.

Process

First, we are going to use the SKU dataset. The SKU dataset is specifically for detecting items in grocery store shelves and refrigerators. SKU has all images and the annotations for each item on the shelf. But for our application, we don't need the annotations of items on shelves. We want the opposite of that. Because we are only going to use those images in which the inventory was empty for any product(s) on the shelf. For annotation for label absence, we use LabelImg tool.

Experiments

We have investigated the performance of current state-of-the-art object detection algorithms on the SKU-110k dataset. The idea is to draw an analysis that explains how well object detection algorithms can perform under harsh conditions. We employed SSD_Mobilenet_V2_Fpnlite, SSD ResNet101 v1 FPN, SSD_Mobilenet_v1_FPN to benchmark their performance on the SKU- 110K dataset. We have leveraged the capabilities of transfer learning in our experiments. All the object detection networks are incorporated with a backbone of ResNet50 pre-trained on COCO dataset. We fine-tuned all the models for 5000 epochs and used Adam as an optimizer. We resized images to 640 × 640 during the training and testing phases.

Setup, Configuration

In the table below, we list each such pre-trained model including:

- a model name that corresponds to a config file that was used to train this model in the samples/config's directory,
- a download link to a tar.gz file containing the pre-trained model,
- model speed - we report running time in ms per 600x600 image (including all pre and post-processing), but please be aware that these timings depend highly on one's specific hardware configuration (these timings were performed using an Nvidia GeForce GTX TITAN X card) and should be treated more as relative timings in many cases. Also note that desktop GPU timing does not always reflect mobile run time. For example: MobileNet V2 is faster on mobile devices than MobileNet V1 but is slightly slower on desktop GPU.
- detector performance on subset of the COCO validation set, Open Images test split, iNaturalist test split, or Snapshot Serengeti LILA.science test split. as measured by the dataset-specific mAP measure. Here, higher is better, and we only report bounding box mAP rounded to the nearest integer.
- Output types (Boxes, and Masks if applicable) You can un-tar each tar.gz file via, e.g.,:

  tar -xzvf ssd_mobilenet_v1_coco.tar.gz

Inside the un-tar' ed directory, you will find:

- a graph proto (graph.pbtxt)
- a checkpoint (model.ckpt.data-00000-of-00001, model.ckpt.index, model.ckpt.meta)
- a frozen graph proto with weights baked into the graph as constants (frozen_inference_graph.pb) to be used for out of the box inference
- a config file (pipeline.config) which was used to generate the graph. These directly correspond to a config file in the samples/configs) directory but often with a modified score threshold. In the case of the heavier Faster R-CNN models, we also provide a version of the model that uses a highly reduced number of proposals for speed.
- Mobile model only: a TfLite file (model.tflite) that can be deployed on mobile devices.

Some remarks on frozen inference graphs:

- If you try to evaluate the frozen graph, you may find performance numbers for some of the models to be slightly lower than what we report in the below tables. This is because we discard detections with scores below a threshold (typically 0.3) when creating the frozen graph. This corresponds effectively to picking a point on the precision recall curve of a detector (and discarding the part past that point), which negatively impacts standard mAP metrics.

- Our frozen inference graphs are generated using the v1.12.0 release version of TensorFlow; this being said, each frozen inference graph can be regenerated using your current version of TensorFlow by re-running the exporter, pointing it at the model directory as well as the corresponding config file in samples/configs.

Quantitative and Qualitative Analysis Results

Evaluation Matrix

The quantitative analysis demonstrates that the proposed approach achieved a minimum loss on the provided dataset SKU-110k, whereas the qualitative evaluation indicates increase in sales and customers' satisfaction level. This section discusses the well-known evaluation criteria essential to standardize state-of-the-art results for object detection in difficult situations. Finally, we will present the outcome of our experiments on the SKU-110k dataset. Evaluation Criteria The standardization of how to assess the performance of approaches on unified datasets is imperative. Since object detection in a challenging environment is identical to generic object detection, the approaches appraise similar evaluation metrics.

Precision tells in what ratio the object detection model found the correct objects in the image. Or in other words, how many of the positive results are positive.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{total\ positive\ results\ found\ by\ the\ model}$$

Recall tells in what ratio the model managed to identify those cases that are positive.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{total\ number\ of\ dog\ images}$$

The image explains the visual difference between precise and imprecise prediction in object detection. The green color represents the ground truth, and the red color depicts the predicted boundary. Considering the IOU threshold value equals 0.5, the left prediction is not precise because the IOU between the ground truth and the inferred bounding box is less than 0.5. The bounding box prediction on the right side is precise because it covers almost the complete ground truth area.

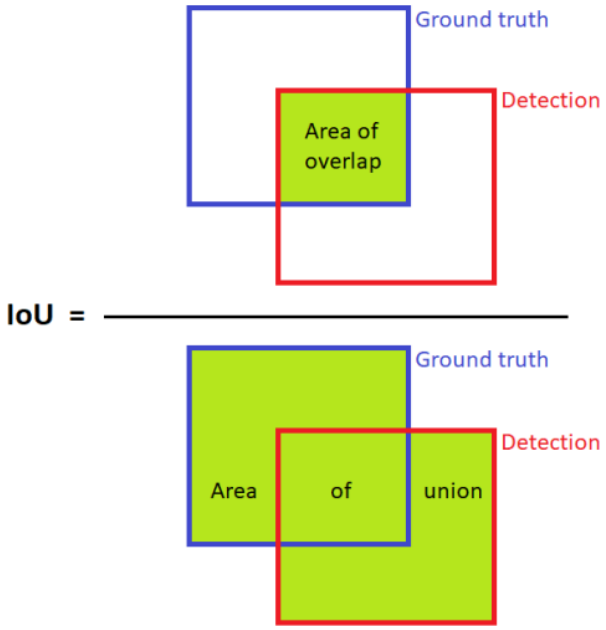
Where TP is True Positives and FN represents False Negatives.

Intersection Over Union

Intersection Over Union (IOU) is one of the most important evaluation metrics that is regularly employed to determine the performance of object detection algorithms. It is the measure of how much the predicted region is overlapping with the actual ground truth region. IOU is defined as follows:

$$\frac{\text{Area of Overlap region}}{\text{Area of Union region}}$$

Intersection over Union (IoU) is the name of the calculation which gives “the overlap divided by the union” of 2 bounding boxes: ground truth bounding box and detection (prediction) bounding box. A simple visual example of IoU is shown in figure. For most evaluation cases like competitions, an IoU threshold of 0.5 is sufficient. This number means that there is most likely an object inside the ground truth box. IoU is used to determine whether a prediction is positive or negative. For example, if mAP is being calculated for IoU value of 0.5.



- IoU >= 0.5, then true positive (TP): ground truth object is detected with the correct class.
- IoU < 0.5, then false positive (FP): ground truth object is detected with a wrong class.
- False negative (FN): ground truth object is not detected.

The losses for the Final Classifier

Loss/classification_loss: Loss for the classification of detected objects into various classes: object, empty shelves etc.

Loss/ Localization_loss: Localization Loss or the Loss of the Bounding Box regressor.

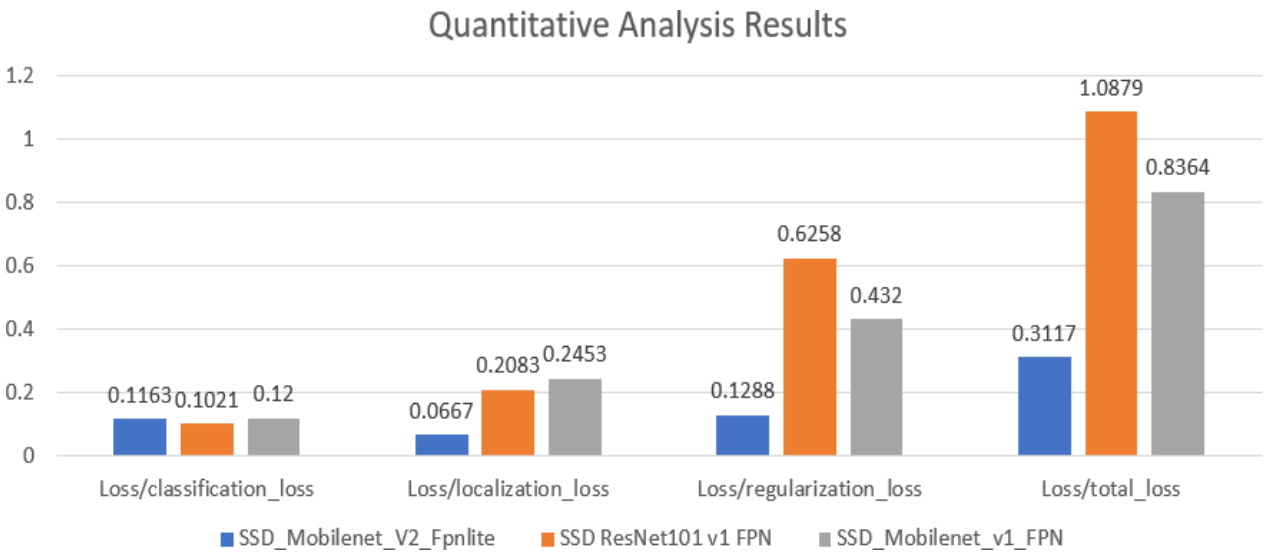
Loss/ Regularlization_loss: Regularization refers to the act of modifying a learning algorithm to favor "simpler" prediction rules to avoid overfitting. Most commonly, regularization refers to modifying the loss function to penalize certain values of the weights you are learning.

Loss/ total_loss: Sum of classification_loss, localization_loss and regularlization_loss and total_loss.

Learning Rate: The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

Quantitative analysis of different Deep Learning Models on SKU-110k dataset

Losses & Learning-rate	***SSD_Mobilenet_V2_Fpnlite***	***SSD ResNet101 v1 FPN***	***SSD_Mobilenet_v1_FPN***
Loss / Classification-loss	0.1163	0.1021	0.12
Loss / Localization-loss	0.667	0.2083	0.2453
Loss / Regularization-loss	0.1288	0.6258	0.432
Loss/ Total-loss	0.3117	1.0879	0.8364
Learning-rate	0.0787	0.03991	0.038



SSD ResNet101 v1 FPN

SSD_Mobilenet_V2_Fpnlite

SSD ResNet101 v1 FPN

SSD MobileNet V2 FPNlite:

Quantitative: Demonstrated strong real-time performance with competitive mAP, particularly suitable for scenarios demanding low-latency responses.

Qualitative: Impressive adaptability to varying lighting and shelf configurations, although occasional false negatives were observed in densely packed shelves.

SSD ResNet101 V1 FPN:

Quantitative: Exhibited a higher mAP due to its ability to capture intricate features, although inference speed was comparatively lower.

Qualitative: Robust detection of empty shelves even in complex scenarios, yet resource-intensive nature might affect deployment feasibility.

SSD MobileNet V1 FPN:

Quantitative: Achieved a balance between speed and accuracy, offering competitive mAP and favorable inference speed.

Qualitative: Proven resilience to subtle lighting changes and well-suited for medium-sized supermarkets; occasional false positives occurred.

So, the holistic analysis of quantitative and qualitative aspects offers a nuanced understanding of each model's performance in the context of identifying empty and full shelves. The SSD MobileNet V2 FPNlite excels in rapid real-time responses, while SSD ResNet101 V1 FPN showcases higher accuracy at a slightly reduced inference speed. SSD MobileNet V1 FPN emerges as a balanced contender, suitable for various supermarket scenarios. Each model's strengths and limitations provide essential insights, guiding the selection of the most fitting model based on the specific demands of supermarket shelf identification.

From the above Observation we found SSD_MobileNet_v2_FPNlite has more speed for detecting of objects, it classification loss, regularization loss and localization loss is less as compare to others model. According to our research, we can SSD MobileNet_v2_FPNlite is more effective than other pre-trained models.

Discussion

We note that the IoU threshold for calculating the mAP is low (0.1), this is due to two factors. First, the weakly supervised method is unable to fit the entire object. Second, it gives a combined bounding box in the cases where the boundary between similar objects is very hard to distinguish. The latter is unique to shelf datasets and shouldn’t be a limitation of the method in general datasets. The former can be tackled as well. Our refinement module is complementary to other methods like which helps to improve the extent of the localization and we expect our method’s performance to increase when combined with such methods. Future work can be focused on improving the method’s performance on higher thresholds. Another thing to note is that, our method is more suited to structured object instances which is not necessarily the case

with other WSOL methods. To make our method more robust to any object, one should collect training instances of the object spanning different views, orientations etc.

The application of object detection to identify empty shelves in supermarkets represents a significant leap forward in the realm of retail management and customer satisfaction. The synergy between artificial intelligence and computer vision empowers businesses with real-time insights into shelf occupancy, facilitating efficient inventory control, enhancing supply chain operations, and elevating customer experiences. As technology continues to evolve, the potential for object detection systems to seamlessly integrate with existing supermarket infrastructure offers a compelling solution to a persistent challenge.

In conclusion

Object detection and recognition are the most important and challenging problems in computer vision. The remarkable advancements in deep learning techniques have significantly accelerated the momentum of object detection/recognition in recent years. Meanwhile, scene text detection/recognition is also a critical task in computer vision and has gotten more attention from many researchers due to its wide range of applications. This work focuses on detecting and recognizing multiple retail products stacked on the shelves and off the shelves in the grocery stores by identifying the label texts. In the first module, on-the-shelf and off-shelf retail products are detected using the YOLOv5 object detection algorithm. The YOLOv5 algorithm accurately detects both on-the-shelf and off-the-shelf grocery products from the video frames and the static images. The enhanced text detection and incorporated text recognition methods greatly support our proposed framework to recognize the on-the-shelf retail products by extracting product information such as product name, brand name, price, expiring date, etc. The recognized text contexts around the retail products can be used as the identifier to distinguish the product.

Appendices

Models seen in appendix 1 were tested and competed at least in the sku-110k Challenge and got a spot in the comparison charts. Easiest way would have been to choose the one with the best rankings. But there are many factors that prevent this, like reverse ratio between accuracy and speed or unexpected results caused by image resolutions.

For example, it can be seen from the list that ‘SSD_Mobilenet_V2_Fpnlite” seems to be the most accurate model to determine the object.

Another reason for testing so many models is the difference in image resolutions which can be seen at the end of the model names. A model would transform the

resolution of the input image into its own, 640x640, 800x1333, 1024x1024 etc. So, detection with higher resolution would last longer yet results would be more accurate.