AI and A-Eyes' -SMART Retail Insights and Monitoring Solution

By Abdurrehman Malekji, Darshita Rathore, Sumit Tyagi

Abstract

Image Analytics is changing the way we collect, analyze, draw insights and gather competitive intelligence from retail outlets through product layout using AI.

Abstract

Millions of dollars are spent in the retail stores for monitoring aisles, racks, and SKUs manually to check for availability, product facings, stock-out situations, etc. This process is expensive, time-consuming, manually taxing, skill indispensable and comes with a lag too which limits the customer experience and revenue potential. This solution uses artificial eyes (cameras and images), and artificial intelligence which are deployed to mine data collected and enhance customer touch-point activities by automatically regulating planogram compliance, inventory management, assortment, and generate business insights for proactive measures which otherwise would take countless human hours. The solution is developed keeping in mind the scalability across multiple industries, customer demographics, and geographies.

Background

As markets are transitioning from traditional trade to modern trade, the underlying ideology is to deliver improved customer experience.

As per recent Forbes article, for retailers of all types, there has been unprecedented uncertainty in the last few years due to the pandemic: the rapid evolution of the global pandemic and resulting policy mandates; labour push-pull demand and shortage; supply chain contraction; working from home leading to a desire for work-life balance; and 40-year-record inflation. Looking forward, retailers need to keep client needs and sentiment top of psyche. [Ref][1]

While retailers utilize an assortment of statistical surveying techniques for market research to make their image of success, for example, eye-tracking; implementing and following its execution in retail stores stays a challenge. Manual evaluation of the retail rack has ended up being tedious and erroneous with blunder rates up to 20% as portrayed in a Stanford study. [Ref][2] This is where the object detection and image recognition techniques can help the retail industries to take business decisions confidently by standardizing the store checks and getting predictable outcomes from all the sales channels.

Our real time, shelf-monitoring solution was implemented for a global Food and Beverage client. The business requirement was to provide accurate, automated shelf-level insights around display, compliance, inventory, price, competition, etc without manual checks. This was achieved by deploying a sophisticated image detection algorithm which can be used further to address similar use cases that involve examining retail audit KPIs through manual tallying of inventory.

This image-based in-store retail rack insights and compliance were proven at scale for the client, rendering astonishing results with minimal supervision. It had the ability to work with internal sales (POS) & CRM data to amplify recommendations and evaluate productivity of owned assets with ease.

The solution utilizes various open-source AI libraries and is scalable for future use cases as well. It can solve multiple objectives through employing a pre-built and custom deep learning framework with customizable integration.

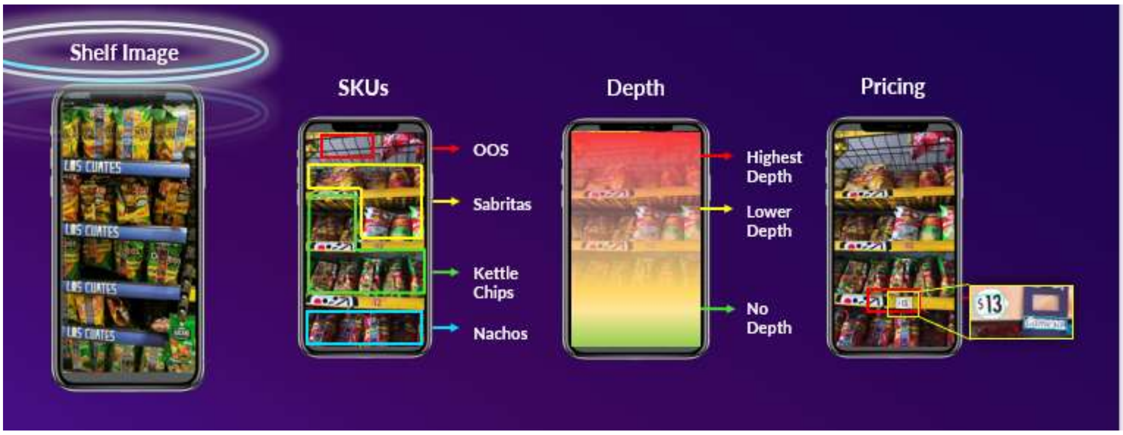


Fig. 1: Example image of real-time detection and insights

Uniqueness of the solution

Retail audits with in-person visits are an essential part of the marketing and research process. However, it is expensive, time consuming, manual and comes with a lag. We now have the capability to use artificial eyes (cameras and images) and artificial intelligence to mine data collected for deployment in several use cases. This real time solution was implemented for a global player in the Food and Beverage industry.

One such use case *in retail is using AI-based Image Analytics for Product Insights and Layout Plans. It is a fully automated and scalable solution that generates business insights for reducing countless man-hours spent repeatedly towards

1) Designing an ideal layout

2) Compliance of positioning and placement of retail goods.

3) Understanding competitive positioning and intelligence

4) Product Assortment

Retailers spend millions of dollars manually monitoring the aisles, racks, and equipment to check availability, stock out and facings. However, due to sub-optimized monitoring and limitations of human inputs there is significant revenue loss due to stock outs.

Our real-time shelf-monitoring and analytics platform identifies products and activities happening in the aisles to optimize product stocking and recommend products. The solution using AI based image analytics can identify in real-time.

1) Layouts
2) Product placement
3) Product Pricing
4) Facings
5) Recommended assortment
6) Competitive pricing
7) Competitor placement

The solution is developed keeping in mind scalability across multiple industries such as manufacturing and supply chain. Further, it is a fully automated solution with customized dashboard to monitor key KPIs.

Introduction to Questions/Hypotheses

The idea of how much enhancement AI and artificial eyes can bring in CPG Retail runs through this solution as a hypothesis which is also demonstrated.

This solution is a holistic answer to the following:

Business questions:

- How can we remove manual intervention in a solution?
- How can our solution enhance customer experience?
- How can retail use AI and artificial eyes for an increased revenue and net profit?
- How can an AI solution be independent of industries and customer demographics?
- How can AI be used to take proactive business decisions?

Operational questions:

- How can we reduce manual efforts of sale representatives?
- How to automate manual tasks like planogram compliance regulation and inventory management?
- How can we increase freshness of KPIs by reducing refresh rate?
- How can we adapt a solution to different demographics and geographics?

Technical questions:

- How can a technology company implement XOps in their solutions?

- How to create self-scalable solution which can cater to an audience as big as a country?
- How to create an adaptive solution that can be used for a variety of data and applications?
- How to deploy distributed load-balancing GPU enabled architecture?
- How to implement continuous monitoring and re-training in MLOps pipeline?

Approach and Solution Highlights

The AI-led automated real-time Retail Shelf Monitoring solution uses Deep Learning (AI) techniques to provide a scalable, self-learning, and self-service solution for the auditors or sales representatives, that utilizes multiple modules including Pre-processing, Exploratory Data Analysis, Deep Learning modeling, Post-model processing, and Production Pipelining; each powered by cutting-edge computer vision based algorithms and part of high-end Azure Compute Pipeline which uses GPU compute engines at the backend. The complete solution can run a docker container and is completely plug and play. As the components of the model are modular, they are usable and can be deployed in the client environment with a plug-and-play approach. Besides, it helps to gather more frequent, and accurate data with limited human guardrails.

First, an atomic datapoint (image) is passed into the Detection model, which figures out the location of products, racks, rack edges, and rack rows. This spatial information is further processed and passed into a Classification model, that determines the type of product in the rack. At the same time, the image is also passed through a depth estimation model and price detection OCR to calculate missing number of packets and price mentioned on each rack row. The interim output goes through the post-model processing module and different other checks (for image quality, missing entities, and edge cases). After that, it is further passed into the Compliance module, which ascertains the degree of planogram compliance, by combining the spatial & product information, and comparing the location & type of the products with the ground truth (actual image) generating an overall compliance and stock out report.

1. Environment Setup

Any deep learning solution requires a powerful GPU backend architecture and framework to work. Modules here. Every module is a part of high-end Azure Compute Pipeline which uses GPU compute engines at the backend. Azure DevOps is used for Continuous Integration, Training, Monitoring, and Deployment. [Ref][3]

Any cloud agnostic solution requires the following components as infrastructure:

- Compute Engines – VM, Kubernetes cluster

- Storage – Blob Storage, RDBMS
- Network – VPN, VPC
- Orchestrator – Kubernetes, Azure DevOps

Configuration	Preferred Parameter	Reason
Region and availability	Your preferred region	The region of business interest
Node Size	NC6v3	We need GPU-enabled machines
API server availability	99.5% - Optimize for cost	It's a batch pipeline
Scaling Method	Auto scale	Kubernetes will take care of the load
Node Count Range	1 – X	X is the max node count the business allows
Encryption	platform-managed-key	Platform managed security
Network Configuration	Kubenet	Inter-node interaction and workload balancing
Load Balancer	Standard	
Traffic Routing options	HTTP application routing	We need integration with REST API
Integrations	Add container Registry	Store & version deployable Docker containers
Monitoring	Add Cloud Monitoring	Monitor cluster health and add guard rails

Table 1: Configuration along with justification of compute configuration

To store unstructured and structured data Azure Blob Storage and Apache Hive are used respectively. Virtual Private Cloud network is set up in cloud development environments. Workload and CX are orchestrated by Kubernetes and Azure DevOps respectively.

2. Data Preparation

Over 4,000 images were pre-processed and annotated in PascalVOC format to be readily consumed by the models. The primary step for the development of an AI-based image analytics solution is perfect image annotation. A trained workforce is required to annotate a large amount of data with complex images including several products, with accuracy and level of detail. The images were annotated to capture the sub-brand's positional information, the overall rack, rack, price markings, and each row. To train the Depth Estimation Model depth masks were created. The information captured in annotations were the image dimensions (i.e., width, height, and channels), coordinates of the bounding boxes of all the sub-brands, rack rows, and the overall rack. For computer vision-based object detection not only is the location of exact position important but the class of each SKU is also needed. These images have variations in terms of the way they were taken, the variations in terms of multiple aspect ratios, spatial sizes and geometric distortion to images, the environment they were taken i.e., light, or dark, type of products as huge number of SKUs have different size and shapes, less intra class variations for some products, the way the products are placed as some might overlap with multiple products.

3. Object Detection Model

The object detection modules were designed for packet, rackrow, rack and rack edges detection. The model was made to learn the intricacies of the shapes & sizes from the coordinates stored in the translated PascalVOC files. An advanced versions of image detection architecture, YOLOv3 and YOLOv5, pre-trained on the coco dataset were tried out to predict the coordinates of chips' packets and the coordinates of each rack rows on the images. The models were developed using TensorFlow, PyTorch and monitored in real-time using MLFlow.

It was found that YOLOv5 outperforms YOLOv3 in terms of accuracy and inference speed. [Ref][4]

So, for detection two versions of YOLOv5 [Ref][5] were trained to predict the coordinates on the sales representative images:

- YOLOv5s: Slightly faster version but less accurate than medium version
- YOLOv5m: Higher mAP but slightly slower as compared to the small version

Parameters	**YOLOv5s***	**YOLO5m***
Size	14 MB	41 MB
mAP(COCO)	37.2	44.5
params(M)	7.2	21.2
speed(v100)	0.9	1.6

Table 2: Comparison of YoloV5m* - YoloV5 medium and YoloV5s* - YoloV5 small

The flag to choose the model version i.e., small, or medium, is present which can be used as per the trade-off between accuracy and inference time.

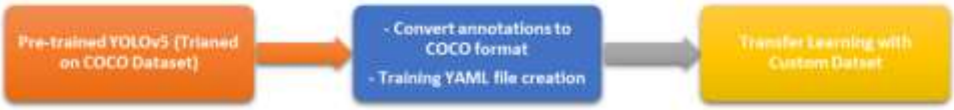

Pre-trained YOLOv5 (Trianed on COCO Dataset)
- Convert annotations to COCO format
- Training YAML file creation
Transfer Learning with Custom Dataset

Fig 2: Transfer Learning with YoLo

4. Sub-brand Classification

Each packet obtained from YOLO-v5 model was programmatically cropped and classified to a specific sub-brand using the InceptionResNet-V2 model. [Ref][5] The data pre-processing steps involved image cropping on the output of the object detection modules and for the preparation of the input for the classification module.

Random adjustments were made to the images, so that the model was robust at handling distorted inputs such as random height and width adjustments, altered aspect ratios and adjusting pixel dimensions.

5. Depth Estimation

For each pixel (RBG), a depth value is predicted by depth estimation model. Using positional information generated by packet detection model, depth is estimated for each packet by averaging the per pixel depth value generated by depth estimation model. Depth value for each packet section is then translated into missing number of packets (depth-wise) by normalizing the value using maximum and minimum depth provided by business for each type of rack and the average depth of a packet.

6. Pricing OCR

For each rack row edge is passed through OCR (price detection) model to get price information for each rack-row. Packets prices are calculated by mapping the packet with its nearest price marking.

7. Post-Processing

The post-processing module comprises many steps starting from checking the sanity of the images for passing them to the compliance formulation. The image sanity is done based on the formulation of confidence scores generated from the object detection models; average rack row and packets area; and total rows and packets count. Since the model quality is paramount, the low sanity score indicates that the image is of low resolution or poor quality and is not passed for the further compliance calculations.

Once the image sanity is done the output from the packet detection model, rack-row detection model, and sub-brand classification model is integrated into a single JSON file treated for any unforeseen inaccuracy in the predicted positional information. Accuracy is improved by identifying the occurrences where racks or rows have been missed to improve the object detection model results or where there were empty spaces on the rack. The post-processing step also leads to identifying the location, i.e., the row and column of the detected packets, and handling cases where a greater number of racks have been detected. The threshold based on the exploratory data analysis on the image data is set in each of these steps.

Since the order in which the rows and packets are detected is not in the actual order in which they are present in the rack, the post-processing begins by arranging the identified rack rows and packets in the order in which they are present the actual image. This arrangement and column-wise matrix creation for packets is done based on the coordinates and area intersection of packets with the row.

The undetected row if missed by the object detection model is identified based on the area left between the two subsequent rows. A separate logic to find out the missing top row is applied which detects it by checking the coordinates of the complete rack and the coordinates of the first row detected. Packets/Strips are only considered within a row if their overlapping area is greater than a defined threshold. To remove any extra Rows detected, row overlapping percentage with the complete rack is calculated and a threshold is set to remove those rows. In a similar manner any packet bounding boxes that are not present inside a rack row are discarded.

To identify empty spaces inside the racks, the distance between two subsequent packets in a row is calculated. If the distance is more than the average size of a packet, the required number of packets are inputted inside the row. Given that there are many different possible cases of planogram, the average size of the packets is calculated based on packets being detected on the same row and the adjacent rows. The outputs from post processing are passed to further calculations as a dictionary.

8. Shelf Analytics – Compliance, Stock-out Situations, Pricing & Depth

The final step is calculating the compliance, leveraging all the information gathered by the object detection and classification models. Rack insights compliance was calculated by comparing the previous images and the after images (same rack captured at start and end of day).

Two methodologies are put in for the comprehensive compliance formulation:

- Position-Sensitive Compliance Calculation: Position Sensitive compliance calculation is an element wise comparison between individual cells of the two matrices. If packets in two corresponding cells match, the position level compliance for that cell would be 100%, on the other hand if packets in the two corresponding cells do not match, the position level compliance for that cell would be 0%.

- Position Insensitive Compliance Calculation: Position Insensitive Compliance is calculated by comparing the frequency count of each individual packet in both the matrices and it is irrespective of the position of individual packets. Over here the frequency of each individual packet is 1 in both the matrices and hence the compliance calculated here would be 100%.

The compliance calculations are done on three different levels:

(i) Overall Rack Level: Differences relative to the entire rack. Measures the compliance by comparing the count of each sub-brand in previous and after images. This comparison is independent of the position of packets.

Compliance = (Match of the count of each sub-brands in both images)/ (Rack Capacity)

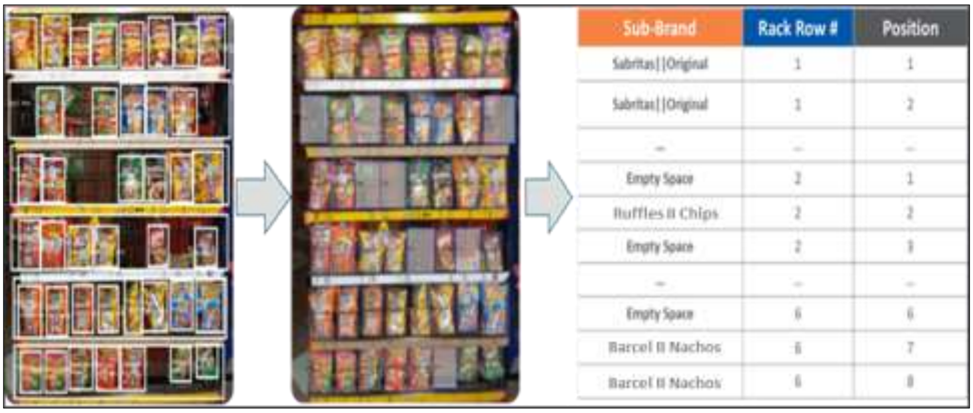
(ii) Rack Row Level: Differences relative to a row in the rack. Measures the compliance by comparing the count of each sub-brand in a particular rack row. The comparison depends only on the position of the rack row & is independent of the position of the packet in that rack row.

Compliance = (Match of the count of each sub-brands in both rack rows)/ (Rack Row Capacity)

(iii) Positional Level: Differences relative to a particular position on the rack. Measures the compliance by comparing the packets at each position of the rack.

Compliance = (Count of the positional match of sub-brands in both images)/ (Rack Capacity)

The outputs received after post-processing are converted into matrices or data frames which will then be compared with hypothetical best rack images and racks filled with all the packets in them.



Sub-Brand	Rack Row #	Position
Sabritas||Original	1	1
Sabritas||Original	1	2
–	–	–
Empty Space	2	1
Ruffles II Chips	2	2
Empty Space	2	3
–	–	–
Empty Space	6	6
Barcel II Nachos	6	7
Barcel II Nachos	6	8

Fig. 3: Detected product on the shelf along with the sample output

The missing packets for each packet row (depth-wise) are calculated below:

(i) *missing_depth = (estimated_depth_for_packet/ max_depth_detected) X total_depth_of_rack*

(ii) *missing packets = missing_depth/ depth_of_single_packet*

How is the solution consumed?

As per Retail Drive, Retail companies see practically 70% deviation in the system that was arranged Versus what is being executed in stores [Ref][6]. But Retail is in numerous ways mathematics yet not just.

This AI-led automated solution helps the stakeholders in retail to acquire constant visibility into the stores by catching only a couple of pictures of retail racks. Each SKU can be tracked on a shelf efficiently across many stores in any geo-area, time region, and at any given moment.

In-store automation is accomplished with two techniques:

- By enabling the sales representative to capture in-store pictures utilizing their mobile devices
- By introducing IoT-based cameras on the retail shelves to capture pictures repeatedly throughout the day

Once the images are captured the application AI algorithm checks the quality of the image and calculates all the KPIs. This robust, fully automated, and scalable solution generates business insights by reducing countless man-hours spent repeatedly towards regulatory compliance of positioning and placement of retail goods, managing out of the stock issue and checking overall store analysis. And the data collected, business insights generated, and compliance intelligence are shared with the decision makers immediately. It can work with internal sales (POS) & CRM data to amplify recommendations and evaluate productivity of owned assets with ease.

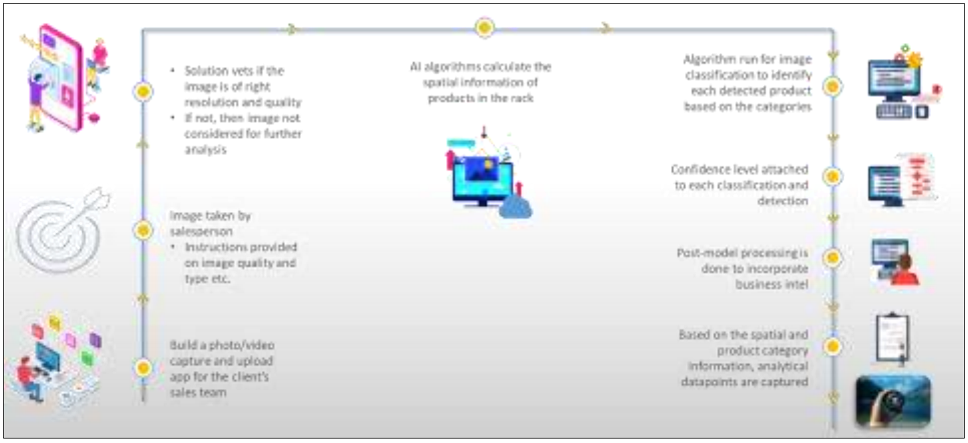

Build a photo/video capture and upload app for the client's sales team
Image taken by salesperson
• Instructions provided on image quality and type etc.
• Solution vets if the image is of right resolution and quality
• If not, then image not considered for further analysis
AI algorithms calculate the spatial information of products in the rack
Algorithm run for image classification to identify each detected product based on the categories
Confidence level attached to each classification and detection
Post-model processing is done to incorporate business intel
Based on the spatial and product category information, analytical datapoints are captured

Fig 4: Business Process Flow of the Solution

Architecture and Design

1. Data Flow:

Fig. 5 shows the end-to-end pipeline starting from the capture of the image, generating in-store retail rack insights and compliance data for the client rendering astonishing results with minimal supervision.

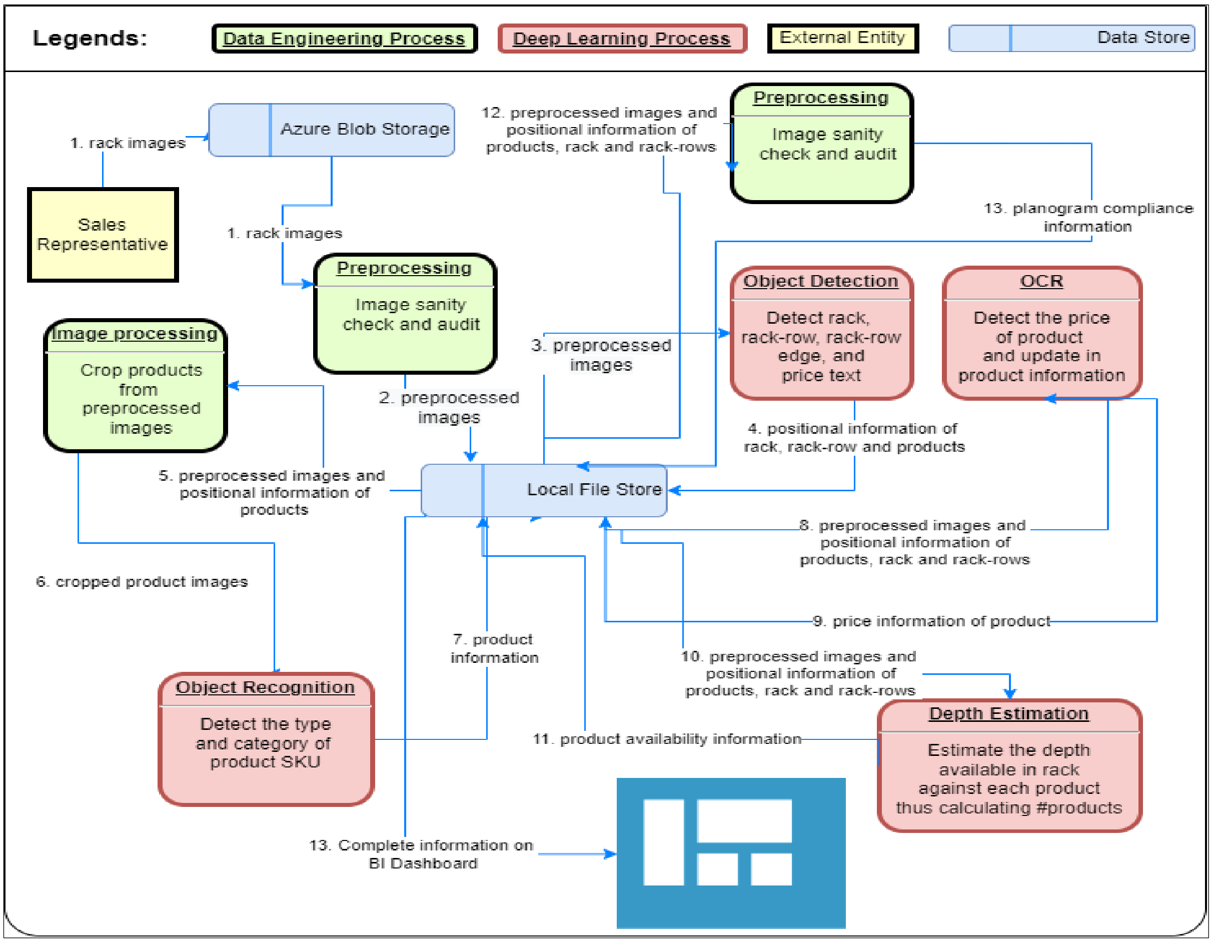

Legends:
Data Engineering Process
Deep Learning Process
External Entity
Data Store
Sales Representative
1. rack images
Azure Blob Storage
1. rack images
12. preprocessed images and positional information of products, rack and rack-rows
Preprocessing
Image sanity check and audit
13. planogram compliance information
Preprocessing
Image sanity check and audit
Image processing
Crop products from preprocessed images
3. preprocessed images
2. preprocessed images
Object Detection
Detect rack, rack-row, rack-row edge, and price text
OCR
Detect the price of product and update in product information
4. positional information of rack, rack-row and products
5. preprocessed images and positional information of products
Local File Store
8. preprocessed images and positional information of products, rack and rack-rows
6. cropped product images
9. price information of product
7. product information
10. preprocessed images and positional information of products, rack and rack-rows
Object Recognition
Detect the type and category of product SKU
11. product availability information
Depth Estimation
Estimate the depth available in rack against each product thus calculating #products
13. Complete information on BI Dashboard

Fig 5. Data Flow Diagram

2. Model Metrics:

Accuracies in terms of precision, recall, map_.5 and validation loss of the object detection models built along with the epochs on which they were trained are provided in the table below:

Model Type	**Model Name**	**Precision**	**Recall**	**Epoch**	**IoU Train**	**map_.5**	**Validation Loss**
Packets	yolov5m	0.943	0.917	74	0.6	0.950	0.042
Packets	yolov5s	0.938	0.886	63	0.6	0.938	0.046

Rackrow	yolov5s	0.947	0.953	48	0.6	0.969	0.027

Table 3: Model Metrics

Classification model results are given below (Illustrative):

Sub-Brand	Total	TP	FP	FN	Recall	Precision	F1
Product 1	1	1	0	0	100.0	100.0	100.0
Product 2	644	643	15	1	99.8	97.7	98.8

Table 4: Classification model results summary

3. Time Comparison – YoloV5 Small Vs YoloV5 Medium

Results on the processing time for the object detection models consisting of time per iteration and time per image:

Model	Model Type	Iteration	Time (sec)	Time/Iteration (sec)	Time/Image (sec)
yolov5s	Packets	10	53.934	5.394	0.034
yolov5m	Packets	10	62.352	6.235	0.040
yolov5s	Packets	1	14.227	14.227	0.091
yolov5m	Packets	1	15.195	15.195	0.097

Table 5: Time Comparison

Summary

Conclusion

By automating in-store analytics, the clients were able to identify out-of-stock situations, optimal assortment, and compliance well in advance, leading to the impact on the key performance metrics:

- Revenue optimization:
    1. The solution considers heuristics on planogram & sales; hence it has led to an increase in revenue.
    2. Real time inventory management, so no sales loss because of Out of Stock.
- Operational Efficiency:
    1. Less human resources and man-hours required (Cost reduction).

2. Automates manual vision task, that reduces TAT from minutes to milliseconds.
3. The single sales representative can target more shops, as manual labour for them is greatly reduced.
4. Frequency of compliance audit can be increased since it’s an automated solution.

- Customer Experience:
    1. Reduces chances of the store going Out of Stock.
    2. Planogram enforcement creates awareness of product spatial information to customers so it’s easy to find their favourite product as it’s at the same location everywhere.
- Technical scalability:

1. The product is widely scalable and can be implemented to any of the FMCG products.

Results

Instead of a singular use case of shelf compliance, the solution was employed to cater to multiple use cases in real time, that are predominantly manual in nature. Now, the solution provides product stocking, forecasting, store insights and customer DNA as well. The client is in turn able to improve customer satisfaction and overall NPS.

Here are some highlights of the impact the solution ensured:

- No. businesses supported: 5,000 outlets recorded every week.
- No. of hours saved: ~200,000 work hours saved annually
- % Revenue Increase (YOY): 1.5x increase (as per initial assessment) in product orders for specific SKUs. To keep the confidentiality intact, the exact numbers for the KPIs have not been shared.
- Automation of the manual vision task, that reduces TAT from minutes to milliseconds.
- A single sales representative can target more shops, as manual labour for them is greatly reduced.
- Frequency of compliance audit can be increased since it’s an automated solution.
- Planogram enforcement creates awareness of product spatial information among customers. So, it’s easier for them to find their favourite product, as it’s at the same location in every outlet.
- Competitive intelligence (verified) with actual stock positions, pricing information etc.

Limitations of the approach/method/technique

Experiments and data that require high manual intervention can be easily replaced by AI. These solutions ensure more frequent and accurate collection of data with limited human bias.

Key drawbacks:

1. The initial solution needs to be trained with labelled data
2. High-end computational sophisticated is required

Challenges:

There were certain challenges faced during the implementation of the solution, which are in the process of remediation:

- Data availability and quality issues:
    - Poor image quality (low resolution, parallax in images etc.)
    - Unknown/new products in the Test data
    - Wide variety of shapes of the same product
    - Aberrations caused by new data was handled by implementing sanity checks on data.
- Processing and modelling:
    - Batch processing was done due to the size of images.
    - The pipeline generates a lot of intermediate data outputs, out of which some are inputs to other models, for example, Detection outputs are input to classification models.
    - Though the ML model solved most of the use cases, the business intel also needed to be incorporated over and above the ML model. Hence, post-model processing module was required.
    - As the processes are computationally heavy, Azure GPU Engines are used to run the state-of-the-art models.

References:

1. Francois Chaubard, 2022, The Current Business Climate Demands Grocers and Retailers Become Fast Adopters, Forbes Business Council
2. Timothy Chong, Idawati Bustan and Mervyn Wee, 2016, Deep Learning Approach to Planogram Compliance in Retail Stores, Stanford University
3. Contributors, 2022, What is Azure DevOps, Microsoft Learn
4. Joseph Redmon and Ali Farhadi, 2018, YOLOv3: An Incremental Improvement, arXiv, Cornell University
5. Glenn Jocher, 2022, ClearML Dockerfile fix, Ultralytics
6. 2021, Your Complete Guide to Image Recognition for In-Store Retail Execution, Infilect Blog

Technical appendices:

1. Planogram: Planogram is a diagram or model that indicates the placement of retail products on shelves to maximize sales.
2. YOLO: YOLO an acronym for 'You only look once', is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself.
3. InsceptionResNetv2: Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database
4. Model Metrics [Ref]:
   a. Precision: Precision is a measure of how many of the positive predictions made are correct (true positives). The formula for it is TP/TP+FP, where TP = True Positives and FP = False Positives
   b. Recall: Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data. The formula for it is TP/TP+FN, where TP = True Positives and FN = False Negatives
   c. F1 –Score: Harmonic mean of Precision and Recall
   d. IoU: A number that quantifies the degree of overlap between two boxes, used while training the models.
   e. mAP: mAP is calculated by finding Average Precision (AP) for each class and then average over a number of classes. mAP_.5 is the mean average precision over 0.5 IoU.
5. Epoch: Epoch is once all images are processed one time individually of forward and backward to the network.

Authors

Abdurrehman Malekji, Data Science Head and SME at Absolutdata

Abhimanyu Saraf, AI Solution Architect at Absolutdata

Darshita Rathore, Lead ML Engineer at Absolutdata

Sumit Tyagi, Lead Data Scientist at Absolutdata