ACS SOLUTIONS

Artificial Intelligence

Using AI-Driven Analytics to Augment Sales for Retail Stores

WHITE PAPER

Table of Contents

Authors

Lakshya Khanna Data Scientist
Sreeja Yalamaddi Data Scientist
Prathyusha Pervela Data Scientist

About ACS Solutions

ACS Solutions is a leading global information technology services and consulting organization that has been serving businesses globally across industries since 1998. A trusted partner to both mid-market and Fortune 500 clients, ACS Solutions has been instrumental in each of their unique digital transformation journeys. Our extensive industry-specific domain expertise and passion for innovation has helped clients envision, build, scale and run their businesses more efficiently, for over two decades.

We have a proven track record of developing large and complex software and technology solutions for Fortune 500 clients such as Microsoft, Blue Cross Blue Shield, Cox Communications, and Novartis. We deliver on our commitments and enable our customers to achieve a digital competitive advantage through flexible and global delivery models, agile methodologies, and expert frameworks. Headquartered in Duluth, GA, and with several locations across North America, Europe, and the Asia-Pacific regions, ACS Solutions specializes in 360-degree digital transformation and IT consulting services.

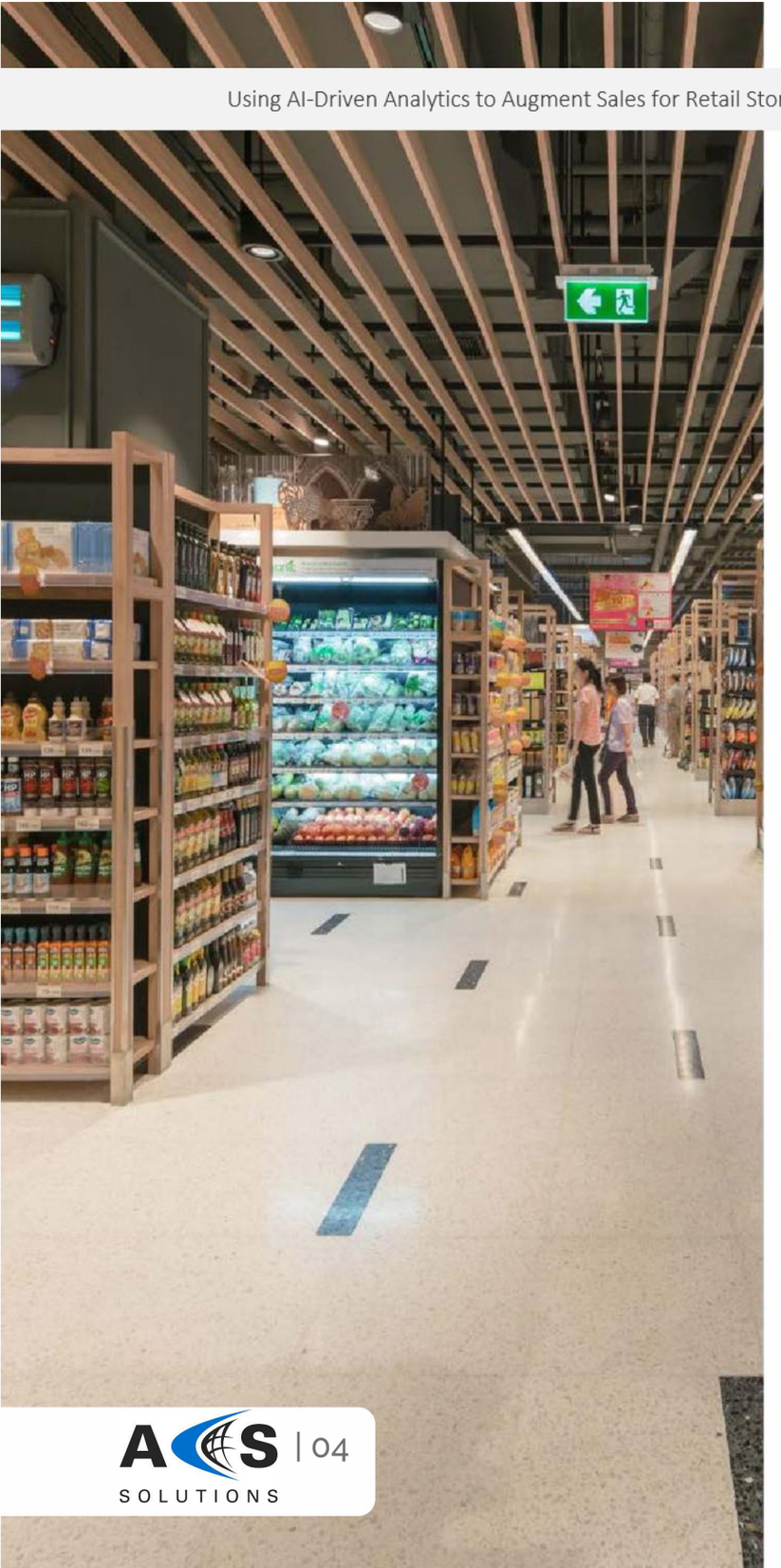
For more information, please reach us at –
acssolutions@acsicorp.com

Executive Summary

With the advent of AI, every business sector is harnessing AI's potential to improve customer satisfaction and optimize their business processes. Retail business has become highly competitive due to the competition from ecommerce platforms as well as the new players in brick and mortar. All the players are trying to maximize their market share through various avenues. Commoditization has increased which means the margins are now lower than ever before. It is imperative to study every type of customer interaction data. That helps in building an understanding of changes in customer preferences ahead, so that retailers can predict trends in buying behavior. This will eventually help in moving volumes.

Although the retail stores are capturing a lot of customer interaction data each day, it is not being utilized up to its potential for drawing any meaningful insights. One of the ways to do that is by leveraging the customers' activity captured by the store's cameras. To derive real-time customer behavioral insights, we developed a Computer Vision and AI-powered solution that consumes the video feeds captured by cameras in the stores and provides useful notes on customer-product interactions.

The following white paper elaborates our solution that is implemented using advanced technologies like Convolutional Neural Networks, Image Processing, Object Detection, and Tracking Algorithms. It enriches the retailers' understanding of customers' behavior by providing analytics on purchase patterns, product placements, service times, etc.



AI for Retail Stores

Retailers often compete for customers' attention by offering a combination of selection, price, service, and convenience. But they have limited visibility into their customers' behavior in brick-and-mortar retail locations. There is little data that explains the process through which customers narrow their product choices and eventually decide to purchase. Majority of the customer analysis in these facilities is based on the observations by sales associates in stores, and it is difficult to keep track of all the customers' touch points due to human limitations.

To bridge the gap between retailers' provisions and customer requirements, Video Analytics would be the best bet, and Video Analytics in combination with Artificial Intelligence goes a long way in providing valuable inputs to retailers about their customer behavior, service time and product purchase patterns. These inputs enable the retailers to deliver great customer experience.

We chose regular eye wear store scenario, for our experiment, where customers visit the store and browse multiple eye wear frames to select one / more of their interest. The major problem with such kind of stores is that, the store associates often tend to lose track of all the models browsed during their time in the store due to the frequency of customers and volume of the models they browse.

To overcome the above-stated problem, we came up with innovative prototypes to be able to provide the retailer with better insights in the following areas:

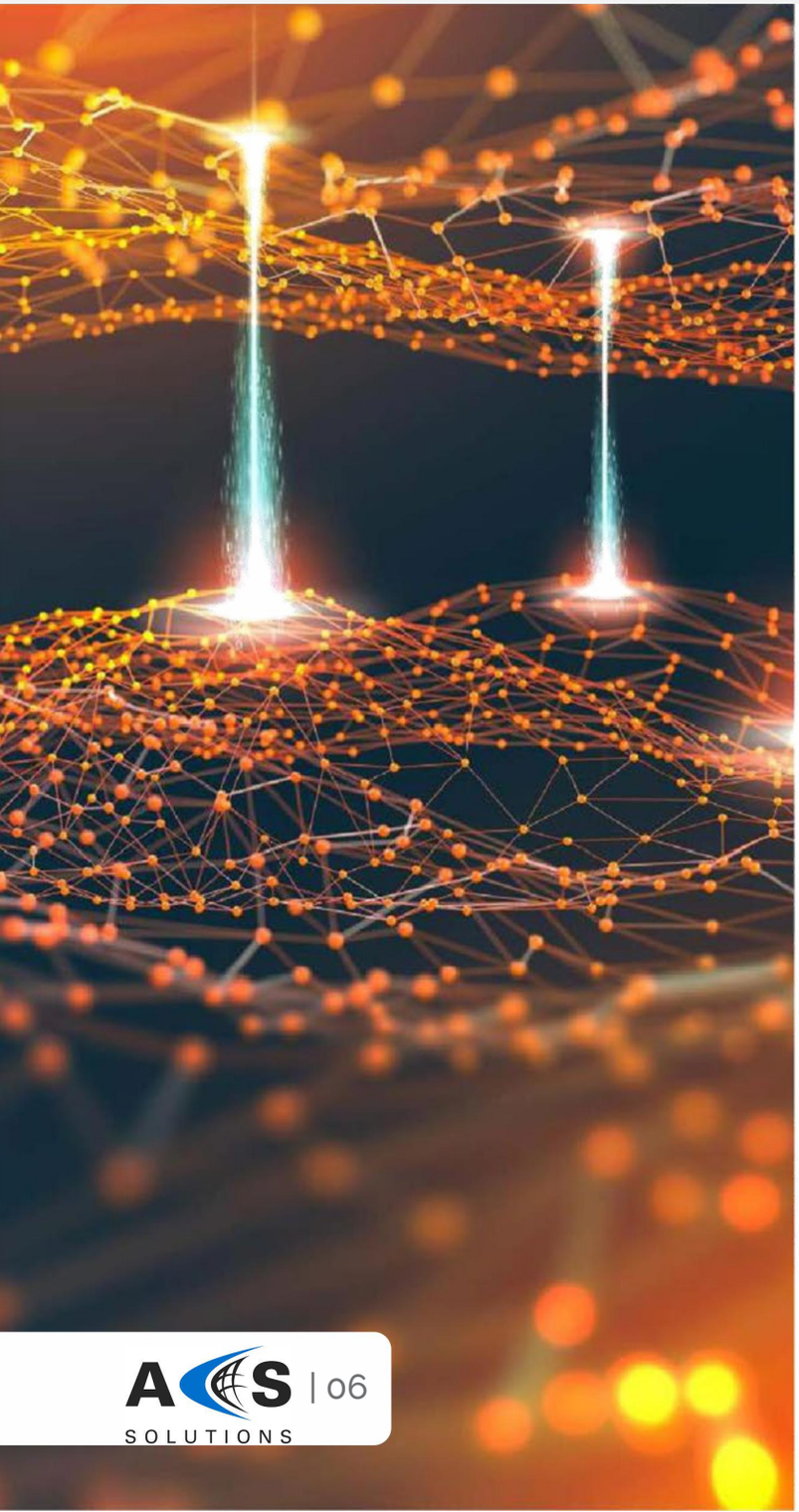
- Customer Product Interaction
- Customer's Product Preferences
- Crowd Management
- Product Placement

“A video is a sequence of images, where each image is termed as a frame. The total number of frames captured per second varies from one video to other.”

We built a model store, where we set up a rack and shelf arrangement similar to an eyewear store and placed the products (eyeglasses) on these racks. The products placed look almost identical but are actually different.

We started with a simplistic approach and developed an initial model using *Convolutional Neural Networks (CNN)* and *Image Processing* techniques to detect the presence or absence of an object. Later, we extended the solution using Object Detection and Tracking algorithms to capture and track Human-Object Interaction.

Our solutions, briefed above, use combination of Computer Vision techniques like Object Detection, Object Tracking, Image Processing to detect customer activity and generate insights based on customer-product interactions. All the *Deep Learning* and *Computer Vision* techniques we use on a video, will be applied to each frame in it.



Simplistic Convolutional Neural Network approach

Approach

"A Similarity check engine compares the similarity between two consecutive video frames and detects presence / absence of eyeglasses at its predefined location."We analyzed the video feed and labeled each video frame as obstructed / unobstructed. A frame was considered as unobstructed, if a human does not block the view between the rack and the camera. Otherwise it is considered obstructed. Parallelly, an obstruction detection model was built using *Convolutional Neural Networks (CNN)* using the labeled data to classify video frames as obstructed / unobstructed.

> **"A Similarity check engine compares the similarity between two consecutive video frames and detects presence/ absence of eyeglasses at its predefined location."**

Video frames classified as unobstructed by CNN were passed to similarity check engine. We performed an analysis on the results from the similarity check engine which can provide retailer with information on popular products, most visited area of the store, peak times, etc.

A significant amount of pre-processing is required to convert video frames to a consumable format. We used *OpenCV* and *Skimage* frameworks for video and image pre-processing. We used *Keras* and *Tensorflow* for creating the neural network model that classifies the obstructed frames.

Implementation

Data Preparation: We created sample videos by setting up a rack and shelf arrangement similar to that of an eyewear store to perform all the experimental operations. The arrangement had a rack with 4 shelves, such that two eyeglasses are placed in each shelf at a certain distance and a camera was placed to capture the arrangement from the top.

Pre-Processing: In this step, the *region-of-interest (ROI)* was identified and selected from the original frame using *OpenCV*. The bounding boxes for each SKU in the ROI were labeled manually and details such as length, breadth coordinates, etc. were stored. An SKU is a unique identification number assigned to each of the eyeglasses. Additionally, the ROI was also labeled -'1' - for obstruction if a person blocks the ROI, otherwise '0'. This ROI labeling will be used in the pre-decision making where we build a *Convolutional Neural Network (CNN)* to detect the obstruction present in a frame.

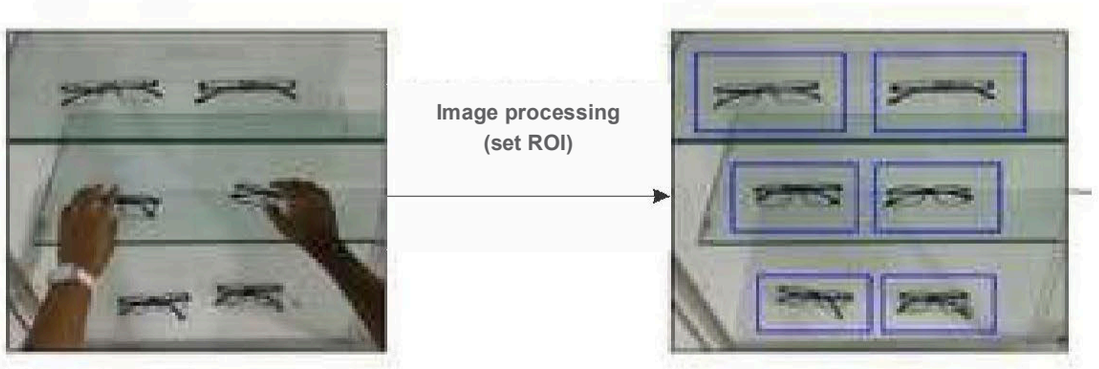
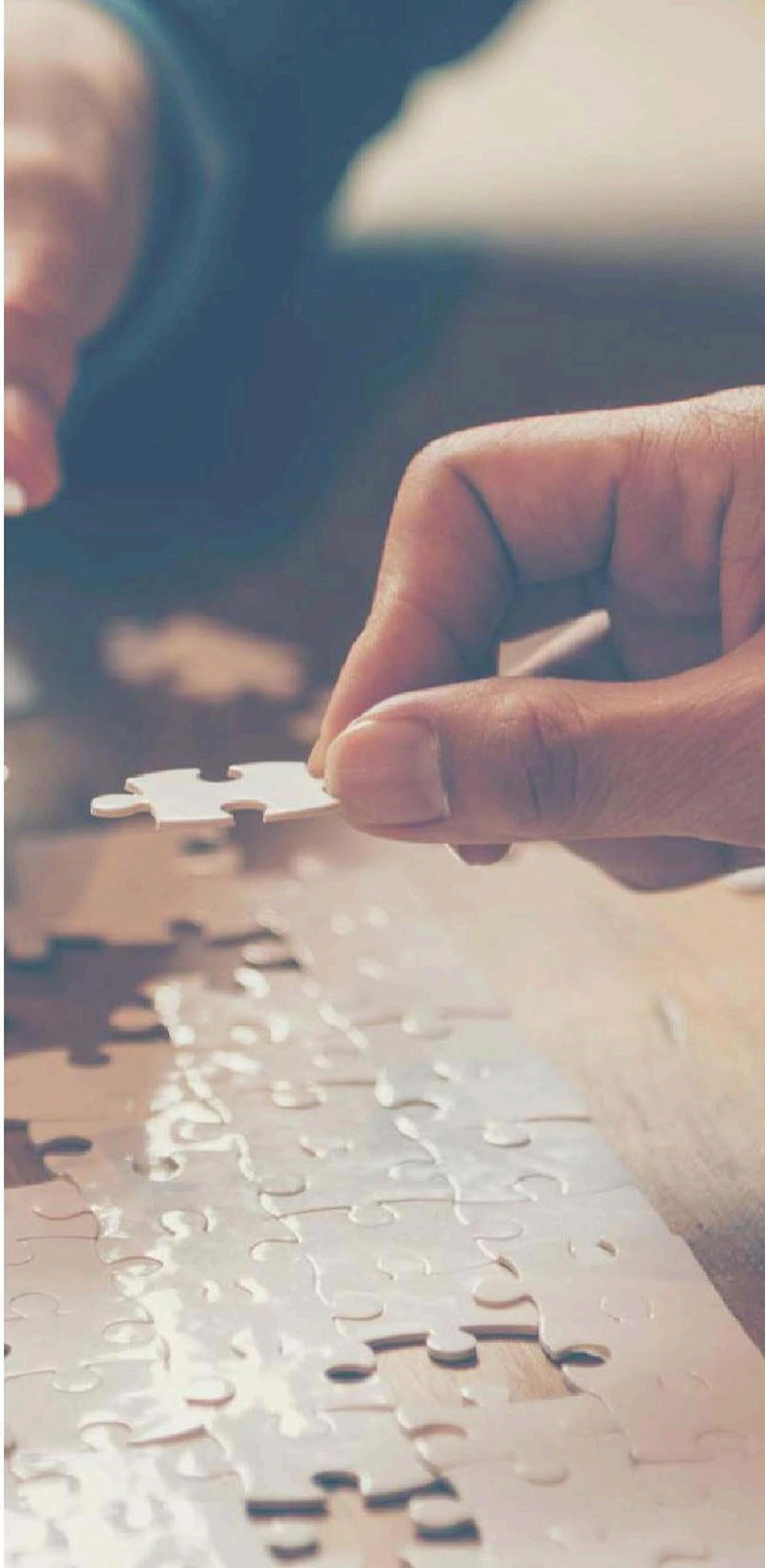
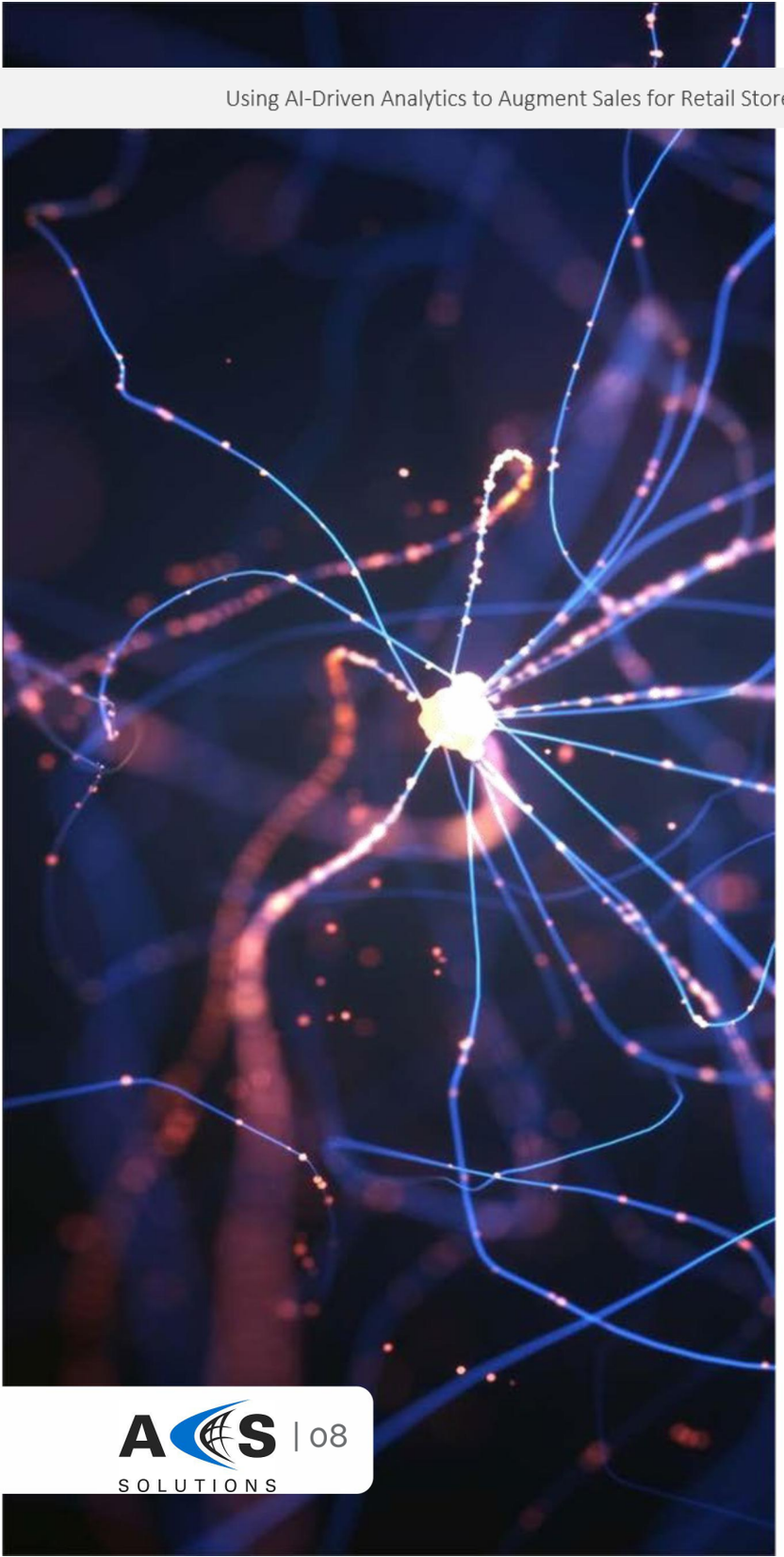


Figure 1 : Pre-processing- ROI Selection





Convolutional Neural Networks are a class of Deep Neural Networks which are especially useful for analyzing images. They ingest and process images as tensors. A CNN consists of several convolutional and pooling layers optionally followed by fully connected layers. Each layer of a CNN modifies the dimension of the processed input image as per its operation and extracts information.

A convolutional layer has a set of multiple filters (kernels). Each filter extracts a feature. Pooling layer is used between convolutional layers to reduce the spatial dimensions, avoid overfitting and gain computational performance. A fully connected layer performs classification based on the features extracted by the preceding layers.

Pre-Decision Making: Decision making is the task of determining if an SKU is present in its defined bounding box or not, given a sequence of frames. The pre-decision making step is the one where we get to filter the obstructed frames. We used the labeled data prepared in the previous step to build a CNN model using *Keras* and *Tensorflow* in Python. We used 3 pairs of convolutional and max-pooling layers with 32, 64, 128 filters, followed by a flatten layer, a fully connected layer, and *reLu* activation function to classify obstructed and unobstructed frames. We fine-tuned the model using regularization techniques like *Dropout and Batch Normalization*. The tuned model was able to estimate the presence or absence of an obstruction in an ROI with 96% accuracy.

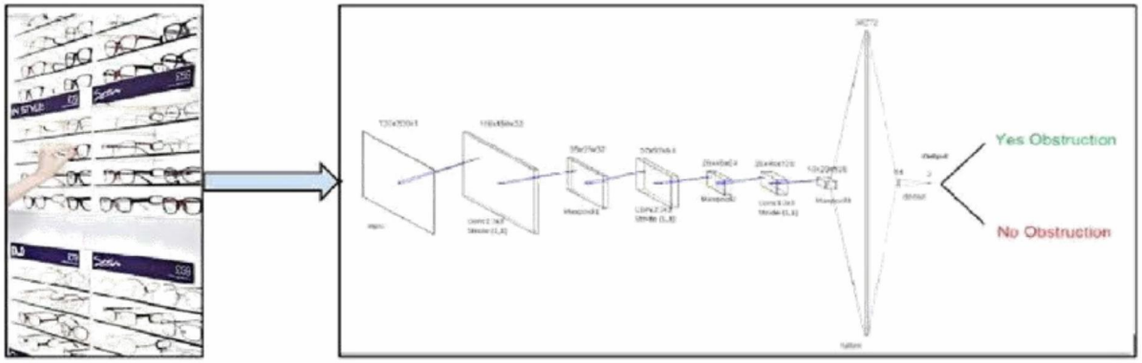


Figure 2 :Obstruction Detection Model

Decision Making: Once all the obstructed frames are filtered out, a similarity check engine was developed using the *Structural Similarity Index Metric (SSIM).* This engine was used to check the presence/absence of the SKUs in their respective bounding boxes. For each SKU, two flags - *isPicked* and *isPutBack* were maintained using this similarity check engine. Try-on counter of an SKU is incremented by one each time these flags were raised sequentially i.e., isPutBack after isPicked.

Evaluation & Results

The performance of this approach was evaluated on a test video which we had labeled for Try-on counts. We performed an error analysis between the model output and actual counts by Comparing Tryon counts obtained using each of Structural similarity engine and CNN model separately and with a combination of both. Average error between the actual and the estimated try-on counts from combination turned out to be ***0.178.***

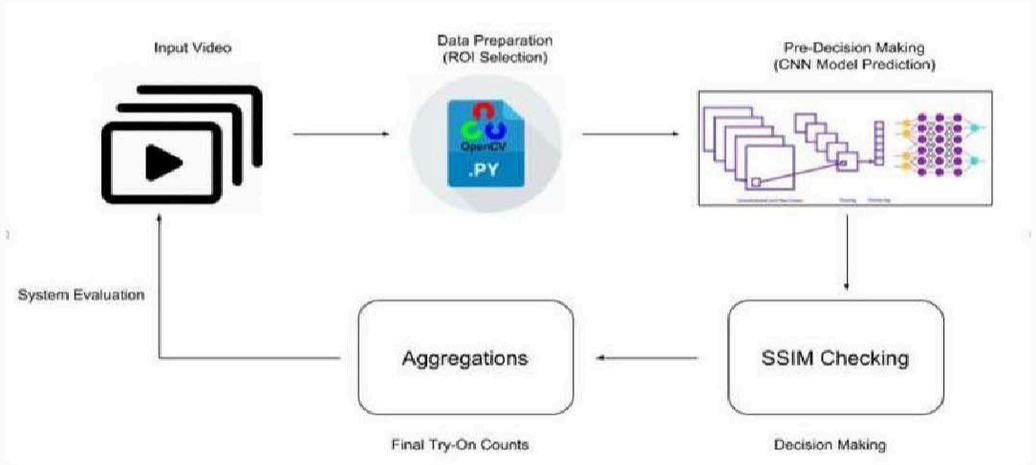
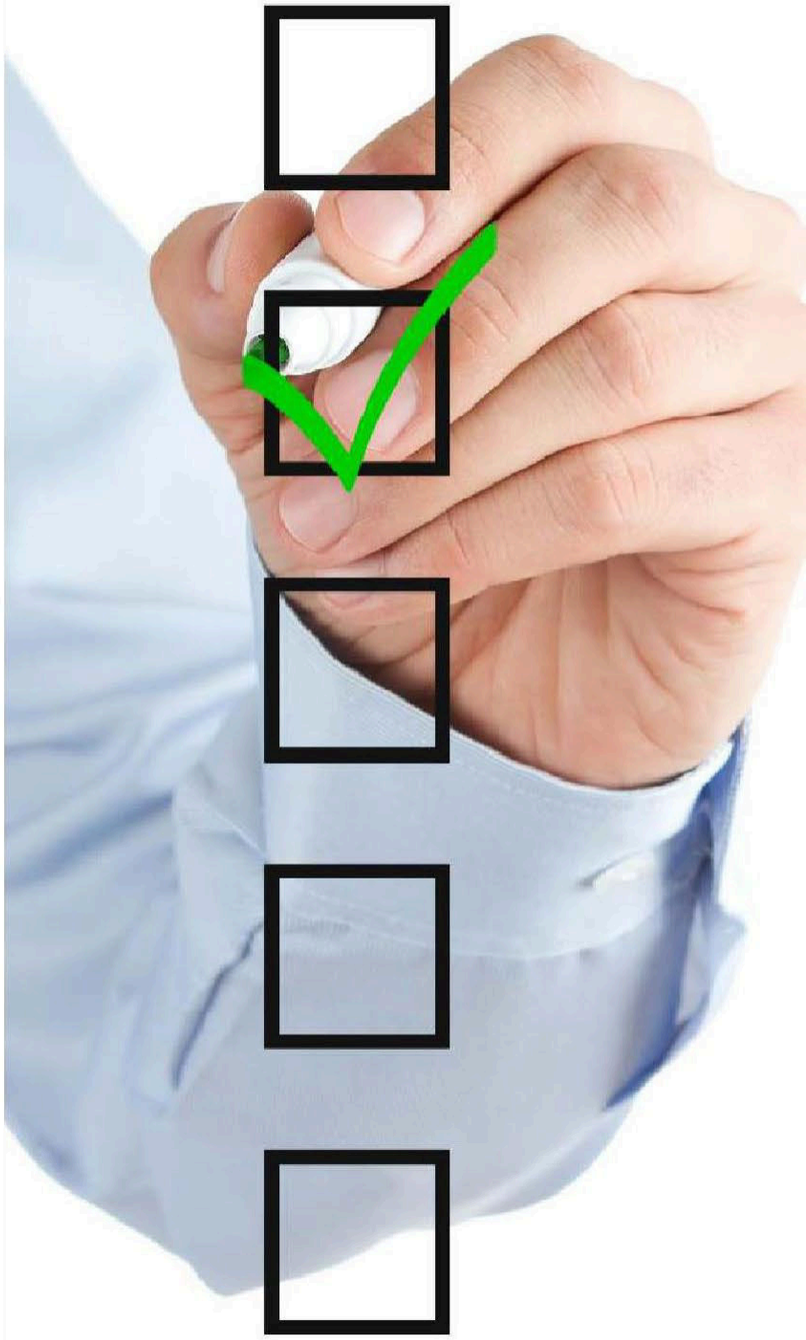


Figure 3: Simplistic CNN Approach



Limitations

- The accuracy of this approach depends on the video quality. Poor video quality can affect the accuracy of CNN model. Thus leading to an increased error in the Try-on count.
- The labeling process and training is a one-time activity but is time-consuming.

To overcome these limitations, we extended the above approach by using advanced techniques like Object Detection and Tracking algorithms in place of the CNN model.

Object Recognition Approach

Approach

We developed an advanced solution using hand-detection model and clustering algorithm to detect hands present in a frame. A tracker was assigned to each detected hand and the path traversed by each hand was tracked. We used this tracked path to determine the pick-up or put-back of the respective SKU. The presence/absence of the SKU in its expected location was detected using similarity check engine (developed in the previous approach). We extensively used *Tensorflow* and *Caffe* framework for hand detection model along with *SKimage* and *OpenCV* for hand tracking.

Implementation

Hand Detection We used *OpenPose* hand detection model to detect hands in the video frame. OpenPose is a key-point detection model developed by Carnegie Mellon University (CMU) that identifies each hand as a combination of 22 key-points. Key-points returned from OpenPose were fed to a clustering algorithm that groups the key-points corresponding to each hand present in the video frame.

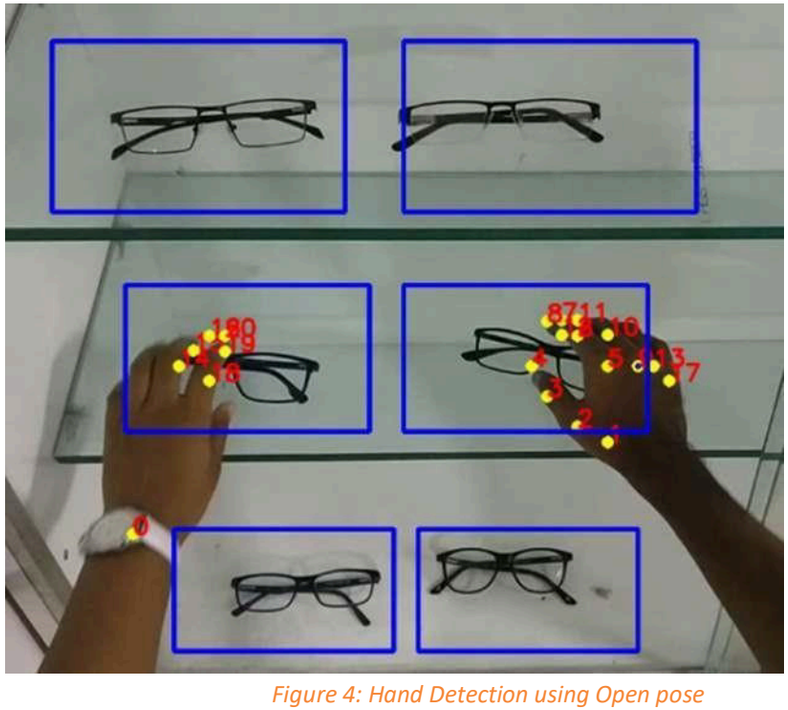


Figure 4: Hand Detection using Open pose

Hand Tracking :

We developed the tracking algorithm in a way that a tracker is initiated whenever a hand is detected in the frame. Next, it creates a bounding box around the detected hand. A criterion is set using *Intersection over Union (IOU)* metric on the bounding boxes between consecutive frames, such that if the IOU is greater than the threshold set, the tracker continues to track. Otherwise, the tracker stops, and algorithm looks for the peak in the path traversed by the tracker. The peak signifies the coordinates of the extreme point in the path. The detected coordinates fall into one of the bounding boxes labeled around the SKUs.

The algorithm checks all the bounding boxes for peak coordinates to return SKU and then deletes the tracker. The identified SKU is used in the next step to check whether it's picked or put back.

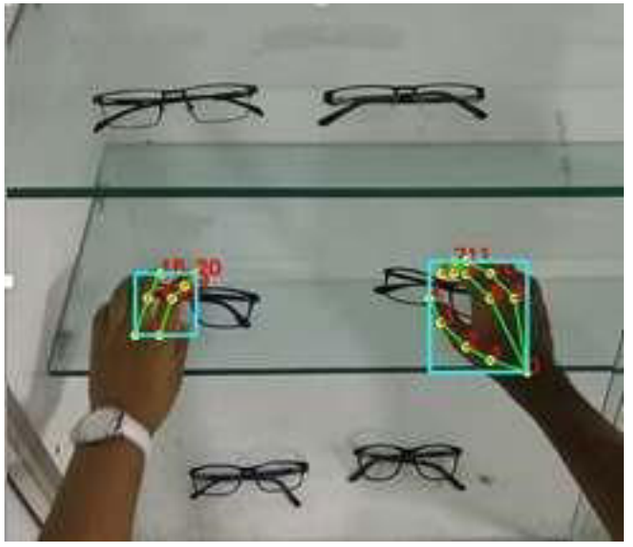


Figure 5 : Clustering and Hand Tracking

Similarity Check: We reused the Similarity Check engine that we developed as part of the previous approach. But instead of unobstructed frames, SSIM was implemented on the bounding boxes drawn around the SKUs returned from the Hand Tracking. Pickup, Put back, and Try-on count details from the Similarity Check engine could be used to help retailers get better insights into their customers' interests.

Evaluation & Results

Implementation

We checked the performance of our approach on a sample video which was labeled for Try-On counts. We performed an error analysis on output from the model to determine the actual Try-On counts. The model gave exact Try-On counts for all the SKU units except for ones in the bottom shelf which can be visualized from the plot below. Results went hand in hand for the first 4 data points but differed for last 2 data points (bottom shelf). Bottom shelf is at a farther distance from the camera compared to other shelves and it had poor lighting on it. Therefore, the model couldn't detect hands accurately.

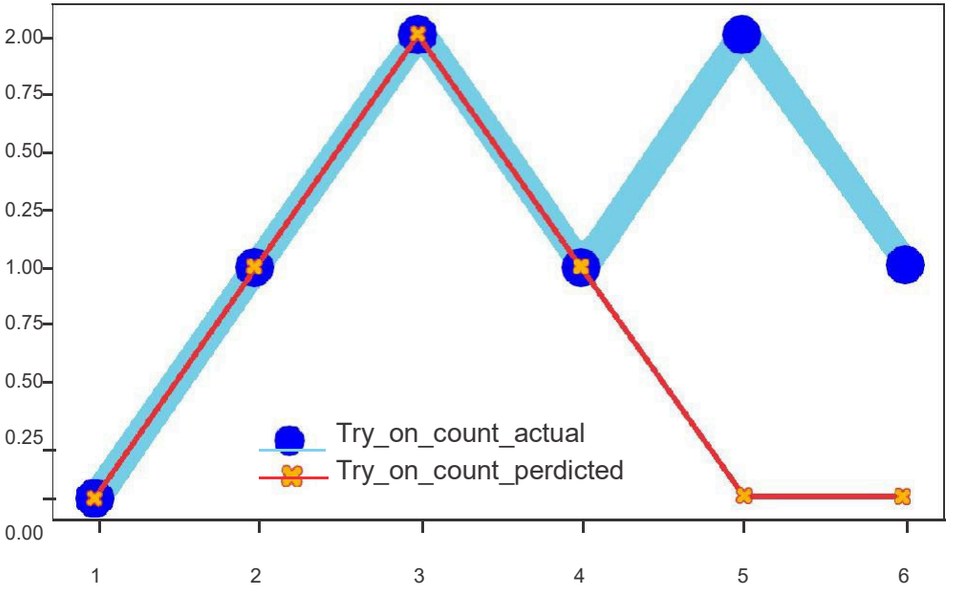
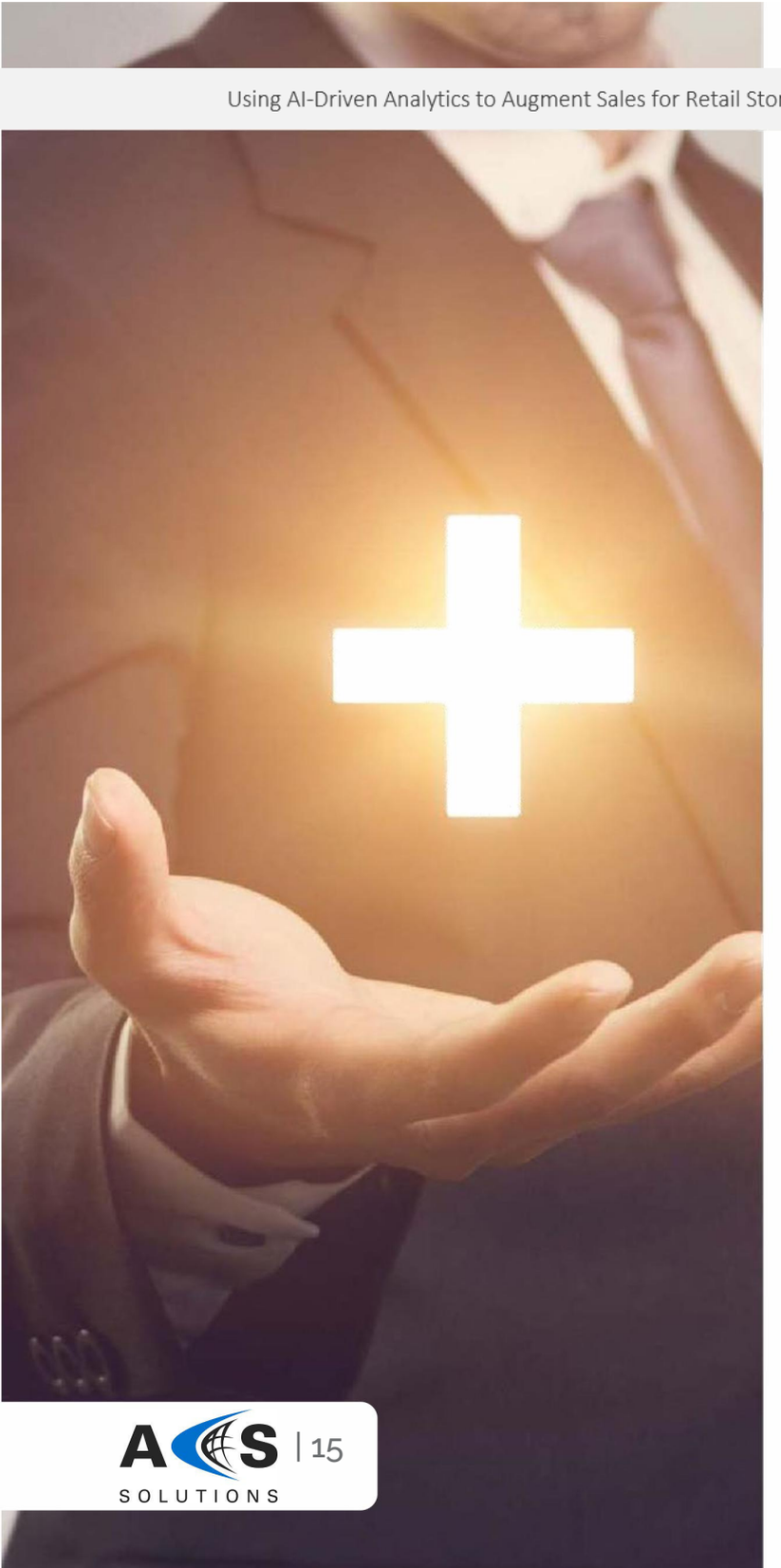
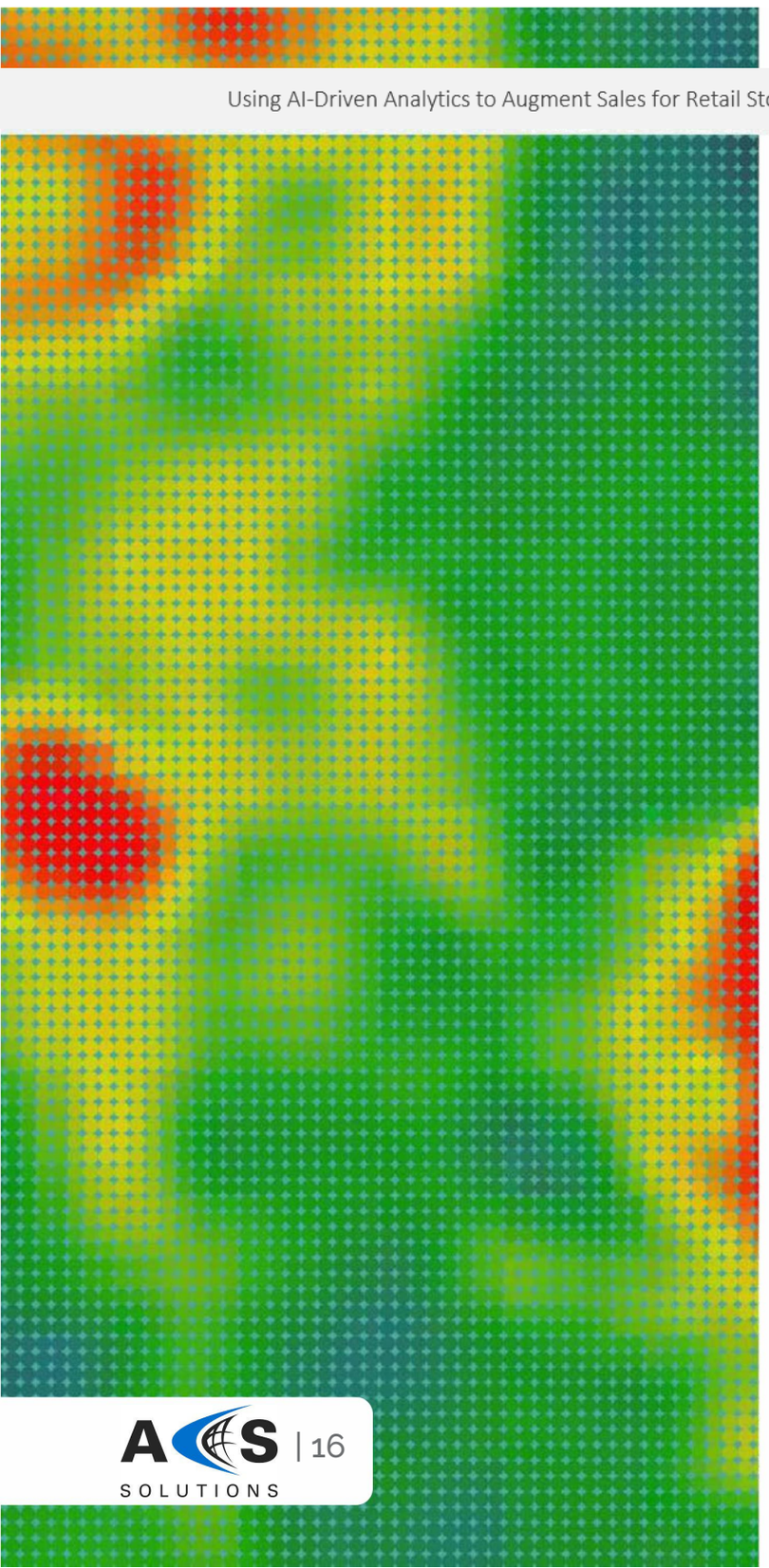


Figure 6 : Object Recognition Approach



Benefits

- Better customer understanding: Provide better insights to retailers about customer's product preferences.
- Enhance customer experience: Provide insights into Crowd Management, Product Placement based on peak time and most populated region of stores which can be derived from results obtained.
- Improve customer conversion: Retailers will be able to understand the role of product interactions in the customer conversion funnel.
- Understand product performance: Retailers will be able to analyze the performance of specific product locations.



Recommendations

Motion Heatmaps

To bolster the customer preference analytics for retail stores we can generate the motion heatmaps to get custmer movement patterns. The movement patterns can be used to identify the most popular locations and racks in the store. The insights derived from these patterns can be combined with our solution to provide an in-depth analysis of the role of product interaction in the customer conversion funnel.

Annexures

Annexure 1: List of Technologies Used

- Caffe is one of the early deep learning frameworks that was originally written in C++ and comes with Python and Matlab bindings.
- **Keras** is an open source neural network library written in Python.
- **OpenCV** is a library of programming functions mainly aimed at real-time computer vision.
- **Skimage** stands for scikit-image, which is a python package dedicated to image processing and using natively NumPy arrays as image objects.
- Tensorflow is a computational framework for building Deep Learning models. It provides a variety of toolkits that allow constructing models at a preferred level of abstraction.

Annexure 2: List of Figures