Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

A Hybrid Approach for Transforming Retail and Customer Experience with User Motion Analytics Using Yolov4 and Pattern Mining

Angelin Jeba P.[1], Jasmine David D.[2], Esther Alice Mathew[3], Marvin Mathew[4], Harish M.[5], Jenin Joel M.[6]

[1,3,4,5,6]Division of Artificial Intelligence and Machine Learning, Karunya Institute of Technology and Sciences, Coimbatore, India.
[2]School of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India.

E-mail: [1]angelinjeba@karunya.edu, [2]jasmine.d@presidencyuniversity.in [3]estheralice@karunya.edu.in, [4]marvinmathew@karunya.edu.in, [5]mharish@karunya.edu. in, [6]jeninjoel@karunya.edu.in

Orcid ID: [1]0009-0006-9920-3327, [2]0000-0002-4596-8133, [3]0009-0005-5316-1152, [4]0009-0000-3332-8617, [5]0009-0006-6343-3657, [6]0009-0009-4152-8577

Abstract

Advances in technology, particularly the tracking and analyzing of customer movements and behavior within retail stores, are leading to a massive revolution within the retail industry. This paper suggests a model that combines object detection, pattern mining, improved preprocessing techniques, and analysis of user motion to boost the retailing process and improve the customer experience. In the proposed method, video data preprocessing is accomplished using the Lucas-Kanade approach. The success of this technique is attributed to its ability to create a solid base for future study by successfully tracking and recording movements made by users. For object detection, the most suitable algorithm is YOLOv4, known for its high accuracy in detecting objects in real time. The motion data acquired through this approach will be utilized for pattern mining. This proposed approach enables extensive analysis of consumer behavior in retail stores. In this study, DBSCAN and K-means clustering methods are used to analyze the clusters. While K-means creates clusters from the dataset based on consumer behavior, DBSCAN is used because of its flexibility in handling different densities, including the presence of noise in the cluster. The accuracy of the proposed system is 98%, while the recall score is 99.02%, and the F1 score is 96.7%.

Keywords: User Motion Analytics, YOLOv4, Pattern Mining, Retail Transformation, Object Detection, Cluster Analysis.

1. Introduction

The retail sector has been undergoing rapid development due to technological advancements that improve operational efficiency and increase customer satisfaction. Within the ever-changing retail environment, adopting advanced analytical techniques is essential to remain competitive. According to [1], motion analytics plays a crucial role in modern retail by

DOI: https://doi.org/10.36548/jtcsst.2026.3.002
Received: 03.04.2026, received in revised form: 06.05.2026, accepted: 25.05.2026, published: 05.06.2026

enhancing traditional retailing techniques, particularly object detection and recognition. Today's customers desire an uninterrupted and personalized in-store experience. Modern consumers have specific needs and wants that must be consistently addressed. A personalized experience has been shown to increase profitability for McKinsey & Company by 10-15% and boost customer satisfaction by up to 20% [2]. By utilizing data on consumer activity in stores and applying user motion analytics, personalized marketing and advertising approaches can be implemented. Store layout and operational efficiency are critical considerations for boosting sales and minimizing operational expenses. Effective layout and operational design are key to enhancing sales performance and reducing operating costs. According to Deloitte [4], optimizing store design can lead to an additional 15-20% in sales performance by improving the consumer experience and minimizing customer search time. Employing advanced technologies such as YOLOv4 [5] allows for the analysis of consumer behavior and interaction with company products.

The process of managing queues is necessary for increasing customer satisfaction and minimizing waiting times. Managing queues has the potential to reduce waiting times by as much as 30%, thus ensuring customer satisfaction and retention [6]. Companies must address stringent privacy regulations, including the GDPR and CCPA, to ensure that all data is kept safely and anonymously [7]. The development of motion analysis techniques necessitates high costs in terms of software, hardware, and employee training [8]. For many small companies, for example, setting up an elaborate analytics system can turn out to be rather costly at first [9]. It can be rather complicated to connect motion analysis results to existing retail platforms, such as CRM and inventory management systems [10]. Poorly analyzed data might undermine the efficiency of analysis and cause various problems in organizational processes, including inventory imbalance and ineffective marketing campaigns, among others [11]. Performance consistency across locations might also affect the efficiency of the analytics [12]. To generate meaningful insights into consumer behavior, the proposed artifact will include a hybrid intelligent system for analyzing retail data that incorporates video preprocessing, object recognition through YOLOv4, multi-object tracking through Deep SORT, and pattern discovery methods like DBSCAN and K-means clustering.

2. Literature Survey

The model presented by Bochkovskiy et al. is called YOLOv4 [1], which is considered a breakthrough in real-time object detection. Indeed, YOLOv4 can provide a very high-performance rate of up to 43.5%. High accuracy makes it possible to track and identify customers in complex situations, and thus analyze their behavior patterns [8]. Such data helps merchants optimize product placement, make proper planning, and manage queues effectively. As shown by Velasquez et al. [14], businesses using motion analytics and pattern mining technology saw an average increase in sales by 15% and efficiency by 20% [15]. Additionally, thanks to specific marketing activities conducted using the pattern mining method, the retention rate could be increased by 10%, and the level of consumer engagement – by 20% [16]. Using this approach, retailers will be able to see patterns like preferred routes in a store, time spent there, busiest hours, and so forth. Stylianou et al. [17], for example, demonstrated how pattern mining could help optimize product placement and retail store design through an understanding of consumer behavior and preferences, leading to a 15-20% increase in efficiency and a sales rise of 10-15%. The findings of Vaneeta et al. [18] prove that introducing YOLOv4 technology into the queue management process decreased waiting time by 30%, significantly increasing consumer satisfaction. Moreover, personalized marketing tactics have been developed based

on pattern mining data, resulting in improved consumer engagement levels by 25% and retention rate by 12%.

A literature review is conducted using open-source benchmark datasets like COCO, KITTI, Cityscapes, and UCF-GV, as well as specialized retail datasets like Mall Dataset and Retail-100. The datasets cover a wide range of scenarios, from pedestrian detection to dense crowd tracking and retail-focused motion analysis.

The proposed research's novel contributions are as follows:

- Improved detection performance by YOLOv4 in challenging situations without any negative effect on its real-time nature.
- Tuning of the size of the anchor boxes to improve the capability of detecting smaller or closely located objects.
- Faster and more efficient management of multiple object categories for different scenarios at various retail stores.
- Reducing the necessity of training to increase YOLOv4's adaptability for different storage configurations.
- Conserving the computing resources of YOLOv4 to ensure real-time performance using multiple cameras.
- Hyperparameter tuning techniques to improve the robustness of YOLOv4.

3. Dataset

The dataset used for the proposed research is illustrated in Table 1.

Table 1. The Dataset Description for User Motion Analytics in a Retail Setting

Dataset	Training Images	Testing Images	Total Images	Split Ratio
Surveillance for Retail Stores [23]	12,468 images	3,584 unlabeled images	Varies	80:20
COCO [24]	118,000 images	5,000 images (val)	2,00,000 images	Standard (118k train / 5k val)
UCF101 [25]	2,500 videos (~70%)	1,000 videos (~30%)	Varies	70:30
KITTI [26]	7,481 images	7,518 images (test)	15,000 images	Standard split
Cityscapes [27]	2,975 images	500 images (test)	5,000 images	Standard split
Mall Dataset [28]	1400 video frames (~70%)	600 frames (~30%)	2,000 frames	70:30

The datasets used in this study include both publicly available benchmark datasets and domain-specific retail datasets. Standard datasets such as COCO, KITTI, and Cityscapes follow predefined training and validation splits. For datasets without predefined splits, including Surveillance for Retail Stores, UCF101, and the Mall Dataset, appropriate train–test splitting strategies (80:20 or 70:30) are applied to ensure balanced and unbiased evaluation. This combination enables robust performance analysis across diverse real-world retail scenarios.

4. Methodology

The images and videos from retail locations are first gathered and preprocessed using data augmentation and annotation, as shown in Figure 1. Next, the YOLOv4 model is set up to handle input photos that have been scaled to 416x416 pixels and initialized with pre-trained weights from the COCO dataset. The 416×416 resolution is selected as it provides an optimal trade-off between detection accuracy and real-time inference speed, consistent with YOLOv4 design. Features are extracted via the backbone network, CSPDarknet53, and then refined by the Feature Pyramid Network (Neck). Bounding boxes, class labels, and confidence scores are predicted by the detection layer (Head), and redundant boxes are removed using non-max suppression. Based on confidence scores, the detected results are filtered. Algorithms such as ByteTrack or Deep SORT are used to extract consumer trajectories and track valid detections over time. These trajectories are subjected to pattern mining techniques, such as clustering and sequential pattern mining, to identify common movement patterns.

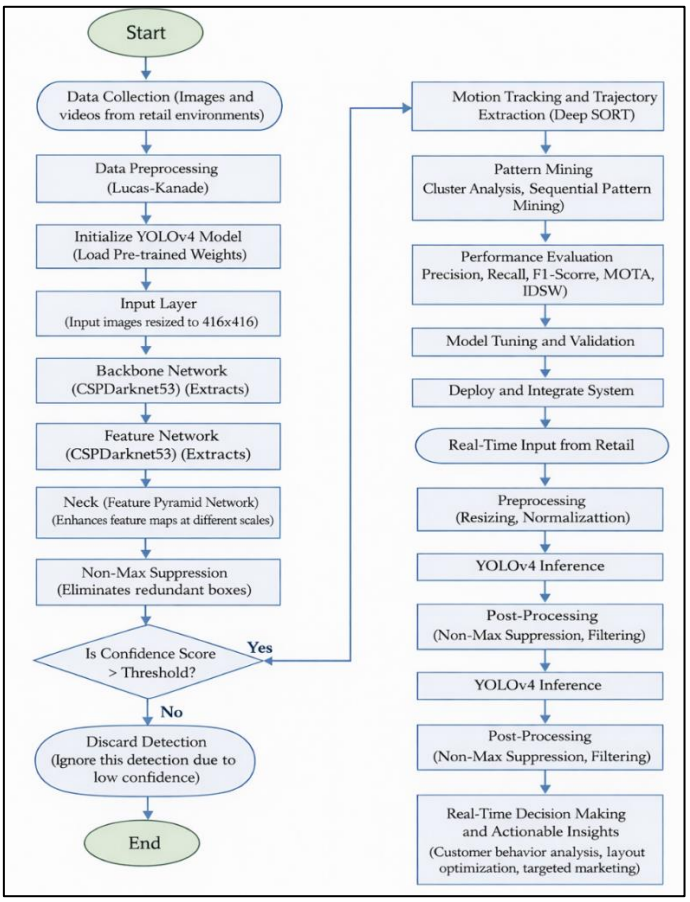

Figure 1. Architecture Diagram of the Proposed Method for User Motion Analytics

4.1 Preprocessing

The Lucas-Kanade method improves video-generated frames by calculating pixel motion between consecutive frames, thereby stabilizing and correcting frame-to-frame jitter. The optical flow constraint equation ties pixel intensity changes to motion vectors in the x and y directions. The optical flow vectors u and v are calculated by applying this constraint in a local neighborhood and solving the resulting matrix equation. Applying these modifications to the frames produces a stabilized video with reduced jitter and motion, resulting in a more

enjoyable watching experience. Several steps are required to stabilize video using the Lucas-Kanade approach for optical flow. To begin, the video is loaded, and the first frame is read and converted to grayscale for easier processing. In this initial frame, feature points such as corners are discovered and tracked. Each following frame is also converted to grayscale, and the optical flow is estimated using the Lucas-Kanade method, which predicts the movement of pixels between frames. To make processing the first frame easier, convert it to grayscale. Compared to color images, grayscale images are simpler to process since they just include intensity information. Equation (1) is utilized to do this conversion by the following action.

$$I_{prev}(x, y) = Grayscale(I_{prev}(x, y)) \quad (1)$$

where:

- $I_{\text{prev}}(x, y)$ represents the intensity value of the previous frame at pixel location (x, y).
- (x, y) denotes the spatial coordinates of a pixel in the image.
- Grayscale$(\cdot)$ is the function that converts a color image into a grayscale image by reducing RGB channels into a single intensity value.

To locate corners or other locations of interest in the first frame, the Shi-Tomasi feature detection technique is employed. These sites are selected because they offer dependable tracking and are less susceptible to noise. To achieve this, apply equation (2).

$$Features = Shi - Tomasi(I_{prev}) \quad (2)$$

where:

- Features represent the set of detected corner points (feature points) in the image. These points are typically strong, trackable locations used for motion tracking.
- I_{prev} denotes the previous frame (image) in the video sequence, usually in grayscale format.
- Shi-Tomasi $(\cdot)$ refers to the Shi–Tomasi corner detection algorithm, which identifies good features to track based on eigenvalues of the image gradient matrix.

There is no initial transformation indicated by the transformation matrix initialization as an identity matrix. Equation (3) is used to calculate the transformations for every frame, updating this matrix in the process.

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The matrix T represents the initial transformation applied to each frame. Since it is an identity matrix, it indicates that no transformation has been applied yet. During processing, this matrix is updated frame-by-frame to capture motion such as translation, rotation, and scaling between consecutive frames.

To proceed with the processing, read the subsequent frame from the video file. To estimate motion, this frame will be compared to the prior frame. To make optical flow

calculations easier, similarly to the first frame, turn the current frame to grayscale, as indicated by equation (4).

$$I_{curr}(x, y) = Grayscale(I_{curr}(x, y)) \tag{4}$$

where:

- $I_{\text{curr}}(x, y)$ represents the pixel intensity value of the current frame at spatial location (x, y).
- (x, y) denotes the coordinates of a pixel in the image.
- Grayscale$(\cdot)$ is the function that converts a color image (RGB) into a single-channel grayscale image.

Optical flow measures the intensity patterns of an image's movement between frames. The Lucas-Kanade approach uses equation (5) to solve for the motion vectors (u, v), assuming that the flow is almost constant in a limited area of pixels.

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{5}$$

where:

- $I(x, y, t)$ represents the image intensity function at spatial location $(x, y)$and time t.
- $\frac{\partial I}{\partial x}$ is the intensity gradient in the horizontal (x) direction.
- $\frac{\partial I}{\partial y}$ is the intensity gradient in the vertical (y) direction.
- $\frac{\partial I}{\partial t}$ is the temporal intensity change between consecutive frames.
- u is the velocity component in the x-direction (horizontal motion).
- v is the velocity component in the y-direction (vertical motion).

This is expressed as follows in matrix form.

$$Av = b \tag{6}$$

where:

- A represents the matrix of spatial image gradients, typically formed using intensity derivatives: $A = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}$ or stacked over multiple pixels in a local window.
- v denotes the velocity vector of a pixel (optical flow), given by: $v = \begin{bmatrix} u \\ v \end{bmatrix}$, where, u is the horizontal motion and v is the vertical motion
- b represents the temporal intensity change vector, defined as, $b = -\frac{\partial I}{\partial t}$.

The optical flow calculation's status array may be used to find and save consistently tracked sites. Points that have a "successful tracking" are chosen. Calculate how to translate the feature points from the previous frame to the equivalent points in the current frame using an estimated transformation matrix.

$$A=\begin{bmatrix} \frac{\partial I}{\partial x_1} & \frac{\partial I}{\partial y_1} \\ \frac{\partial I}{\partial x_2} & \frac{\partial I}{\partial y_2} \\ \vdots & \vdots \\ \frac{\partial I}{\partial x_N} & \frac{\partial I}{\partial y_N} \end{bmatrix}, v = \begin{bmatrix} u \\ v \end{bmatrix}, b = -\begin{bmatrix} \frac{\partial I}{\partial t_1} \\ \frac{\partial I}{\partial t_2} \\ \vdots \\ \frac{\partial I}{\partial t_N} \end{bmatrix} \quad (7)$$

where, A is the matrix of spatial gradients for Npixels, v is the optical flow vector representing horizontal and vertical motion components, and b is the temporal gradient vector. This matrix formulation enables robust estimation of motion using multiple pixel constraints.

Solve for v:

$$v = (A^T A)^{-1} A^T b \quad (8)$$

where, A is the spatial gradient matrix, A^T is its transpose, b is the temporal gradient vector, and $v$represents the estimated motion vector. This formulation provides a least-squares solution for computing optical flow in the Lucas–Kanade method.

Translation, rotation, and scaling are included in this transformation matrix by equation (9).

$$T = EstimateAffine(prev - pts, cur - pts) \quad (9)$$

where, T denotes the affine transformation matrix, $prev - pts$ and $cur - pts$ $r$epresent corresponding feature points in consecutive frames, and $EstimateAffine(\cdot)$computes the transformation that best aligns these points.

To update the overall stabilizing impact, multiply the current transformation matrix by the cumulative transformation matrix, which maintains a cumulative record of the modifications performed to the video. It is shown in equation (10).

$$T_{cumulative} = T_{cumulative}\ x\ T \quad (10)$$

where:

- $T_{\text{cumulative}}$ $r$epresents the cumulative transformation matrix, which stores the overall transformation from the initial frame up to the current frame.
- T denotes the current frame-to-frame transformation matrix, typically obtained using affine estimation between consecutive frames.

Equation (11) shows how to apply a smoothing function, such as a moving average, to the cumulative transformation matrix to eliminate jitter and provide a smoother transition and trajectory.

$$T_{smooth} = Smoothing(T_{cumulative}) \quad (11)$$

where:

- T_{smooth} represents the smoothed transformation matrix, which reduces noise and sudden fluctuations in motion estimation.
- $T_{\text{cumulative}}$ denotes the cumulative transformation matrix, obtained by accumulating frame-to-frame transformations over time.
- Smoothing(·) is a function that applies temporal filtering (such as moving average, Gaussian filter, or low-pass filter) to stabilize the transformation values across frames.

To stabilize the current frame, apply the smoothed transformation to it. Therefore, warp the current frame using the smoothed transformation matrix to account for motion and get a stabilized output by equation (12).

$$I_{stabilized} = WarpAffine(I_{curr}, T_{smooth}) \tag{12}$$

where, I_{curr} is the current frame, T_{smooth} is the smoothed transformation matrix, and WarpAffine(·)applies the transformation to generate the stabilized frame $I_{\text{stabilized}}$.

In the following iteration, update the feature points for tracking by setting the current frame as the previous frame.

$$I_{prev} = I_{curr} \tag{13}$$

where, I_{prev} and I_{curr} represent the previous and current frames, respectively. This update step ensures continuity in the frame processing pipeline by shifting the current frame to be used as the reference in the next iteration.

$$prev - pts = curr - pts \tag{14}$$

where *prev_pts* and *curr_pts* represent feature points in consecutive frames. This update step ensures that the tracked feature points from the current frame are used as reference points for the next frame, enabling continuous motion estimation.

The Lucas–Kanade method is applied for camera motion stabilization and does not suppress meaningful customer motion, as tracking is performed on residual motion vectors post stabilization.

4.2 Object Detection

The backbone network (CSPDarknet53), with its convolutional layers, batch normalization, activation functions, and CSP connections, is an efficient network architecture used for feature extraction in YOLOv4. The network neck (SPP, PANet) improves context and localization by aggregating multi-scale information. To forecast bounding boxes, scores, and class probabilities, the network's head uses equations for bounding box transformation, objectness scoring, and class probability calculation. YOLOv4 employs Convolutional Neural Networks (CNNs) for feature extraction of each frame. Some of the layers used in this network are activation function layers, batch normalization layers, and convolutional layers. Feature maps obtained from the CNN are used to predict object position and classification. YOLOv4 has many important components, such as the Backbone Network (CSPDarknet53), Neck (SPP,

PANet), and Head. The architecture of YOLOv4 for user recognition, as shown in Figure 2, comprises a series of deep learning techniques to achieve real-time person detection.

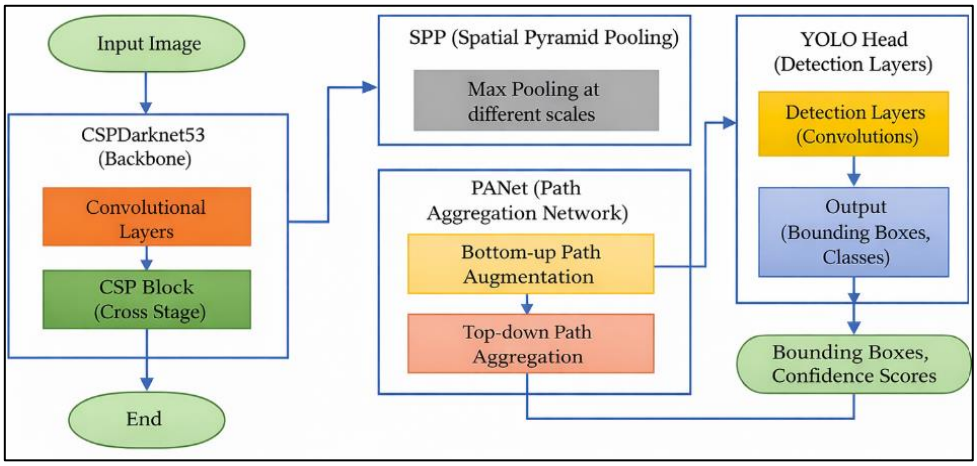

Figure 2. YOLOv4 Architecture for Identifying Users

4.2.1 Backbone Network (CSPDarknet53)

Feature extraction is an important aspect of YOLOv4 that is done through CSPDarknet53. The function of this layer is to efficiently extract features from images to enable object detection. The CSPDarknet53 architecture is a modification of the Darknet53 architecture that allows for efficient gradient flow and reduced computations. The key features of the model include:

4.2.2 Convolutional Layer

CSPDarknet53 is the backbone network, and through convolutional operations, this network helps to extract information from the images fed as inputs. This network consists of various convolutional layers that work to detect various features such as edges, texture, and complex patterns among others. Below is the equation representing the process involved in convolutional operation within a convolutional layer.

$$Output(i,j,k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{c=0}^{C-1} Input(i+m, j+n, c) \cdot Filter(m,n,c,k) \quad (15)$$

where:

- $Output(i,j,k)$ is the output feature map at position (i,j) for the kth filter.
- $Input(i+m, j+n, c)$ is the input image value at position $(i+m, j+n)$ for the cth channel.
- $Filter(m,n,c,k)$ is the filter value at position (m,n) for the cth channel and kth filter.
- M and N are the dimensions of the filter.
- C is the number of input channels.

4.2.3 Batch Normalization

Batch Normalization helps stabilize and speed up the training process, which is crucial for real-time applications, by normalizing the activations within each mini-batch. Equation (16) provides a mathematical expression for this normalization, which stabilizes training.

$$BN(x) = \frac{x-\mu}{\sqrt{\sigma^2+\in}} \tag{16}$$

where x represents the activation, μ and σ^2 are the mean and variance of the activations, and $\in$ is a small constant for numerical stability.

4.2.4 Activation Function (Leaky ReLU)

An activation function called Leaky ReLU (Rectified Linear Unit) adds nonlinearity to the model to aid in learning intricate representations and patterns. The function is defined as follows.

$$Leaky\ ReLU(x) = \begin{cases} x & if\ x > 0 \\ \alpha x & if\ x \leq 0 \end{cases} \tag{17}$$

where α is a small constant, often set to 0.01 to define the slope for negative inputs.

4.2.5 Residual Block

Overcoming a few levels and adding the input straight to the block's output is made possible using a residual connection. This is expressed mathematically as follows.

$$Output = Input + ConvBlock(Input) \tag{18}$$

where $ConvBlock$ represents one or more convolutional layers and other operations such as Batch Normalization and activation functions. This block is responsible for transforming the input features. ConvBlock uses Batch Normalization, Leaky ReLU activations, and a sequence of convolutional processes.

$$ConvBlock(Input) = Leaky\ ReLU(BatchNorm(Conv(Input))) \tag{19}$$

4.2.6 CSP Connection

To extract fundamental characteristics, the input frame is first processed through a first convolutional layer. Part 1 and Part 2 of the first layer's output are separated, and the outcomes are then concatenated. The first part is processed via several layers (convolutional, Batch Normalization, Leaky ReLU).

$$Processed\ Part\ 1 = Leaky\ ReLU(BatchNorm(Conv(Part\ 1))) \tag{20}$$

$$Output = Concat(Part1, Part2) \tag{21}$$

4.2.7 Neck (SPP and PANet)

The layers that link the head detection layers to the backbone feature extractor are commonly referred to as the neck of a neural network. Spatial Pyramid Pooling (SPP) and Path

Aggregation Network (PANet) are two often utilized neck components for advanced feature aggregation and improvement.

4.2.8 Spatial Pyramid Pooling (SPP)

SPP uses pooling operations on feature maps, produced by the backbone network, at various sizes. Different-sized feature maps are produced as a consequence, and are then concatenated.

$$SPP\ Output = Concat(MaxPool_1, MaxPool_2, MaxPool_3, Input) \quad (22)$$

where $MaxPool_n$ is the max-pooling operation with window size $n \times n$ times.

4.2.9 Path Aggregation Network (PANet)

A more robust gradient for the detection head and improved feature representation is made possible by PANet's ability to aggregate information from various levels of the feature pyramid. A bottom-up channel propagates strong low-level characteristics to higher levels, which enhances the conventional top-down feature pyramid network.

$$PANet = Concat(HighLevelFeatures, LowLevelFeatures) \quad (23)$$

4.2.10 Head

Based on the finely tuned information supplied by the neck, the head of a neural network is in charge of generating the final predictions. This includes estimating the bounding box's height, breadth, and center coordinates around discovered users.

4.2.11 Bounding Box Prediction

This entails projecting the bounding boxes' sizes and locations around identified users. The final bounding box coordinates are calculated by applying the anticipated offsets to the anchor boxes.

$$x_{bbox} = \sigma(x_{center}) + c_x \quad (24)$$

$$y_{bbox} = \sigma(y_{center}) + c_y \quad (25)$$

$$w_{bbox} = p_w \cdot e^w \quad (26)$$

$$h_{bbox} = p_h \cdot e^h \quad (27)$$

where σ is the sigmoid function, c_x and c_y are the cell coordinates, and p_w and p_h are the anchor box dimensions.

Anchor boxes are optimized using K-means clustering on retail datasets to better capture human-scale object distributions in crowded environments.

4.2.12 Objectness Score

The likelihood that a specific bounding box includes a pertinent item rather than background noise is determined by the objectness score. The network uses this score to

distinguish between important areas and those that are not. The objectness score, denoted as $\sigma(S_{obj})$, is computed using the sigmoid activation function applied to a raw score $\sigma(S_{obj})$ as follows.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{28}$$

4.2.13 Class Prediction

Different user categories, such as adults, children, and staff, may be included in the courses in a shopping mall setting. The raw class scores are converted into probabilities using the softmax method. For a given bounding box, let s_i be the raw score for class *i.* The softmax function calculates the probability $P(class_i)$ for class *i* as follows:

$$P(class_i) = \frac{e^{s_i}}{\sum_{j=1}^{C} e^{s_j}} \tag{29}$$

where s_i is the score for class *i*, and the denominator is the sum of exponentiated scores for all classes.

4.2.14 Detection Equation

The final detection output is calculated for each grid cell and anchor box.

$$Detection_{i,j,k} = \sigma(o_{i,j,k}) \cdot \sigma(class_{i,j,k}) \cdot BoundingBox_{i,j,k} \tag{30}$$

where:

- $\sigma(o_{i,j,k})$ is the objectness score.
- $\sigma(class_{i,j,k})$ is the softmax probability for the class.
- $BoundingBox_{i,j,k}$ includes$x_{bbox}, y_{bbox}, w_{bbox}, and\ h_{bbox}$.

The detection confidence score in equation (30) is formulated as the joint probability of object presence and class prediction.

$$P(class, object) = P(object) \cdot P(class \mid object) \tag{31}$$

where:

- P(object)represents the objectness score (probability that an object exists in the bounding box),
- P(class | object)represents the conditional class probability given that an object is present.

4.2.15 Post-processing

YOLOv4 uses Non-Maximum Suppression (NMS) to remove superfluous boxes and retain only the most reliable detections after creating the bounding boxes. NMS is carried out by the IoU, for two bounding boxes B_1 and B_2 calculated as:

$$IoU(B_1, B_2) = \frac{Area(B_1 \cap B_2)}{Area(B_1 \cup B_2)} \quad (32)$$

where $Area(B_1 \cap B_2)$ is the area of the intersection and $Area(B_1 \cup B_2)$ is the area of the union of the two bounding boxes.

4.2.16 Pattern Mining for User Motion Analysis

Using clustering and sequential pattern mining approaches, similar movement patterns and behaviors are found by first recognizing and tracking users to extract trajectories for user motion analysis.

4.2.17 Trajectory Extraction

Compute trajectories by utilizing the Kalman filter to correlate detected locations over time to forecast and modify the status of every item being monitored. Drift accumulation is mitigated using Kalman filtering with periodic reinitialization and appearance-based re-identification to maintain trajectory consistency over long video sequences.

State Vector(x):

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}] \quad (33)$$

where, u,v: Bounding box center coordinates; s: Scale (area) of the bounding box; r: Aspect ratio; $\dot{u}, \dot{v}, \dot{s}$: Velocities of u, v, and s.

Prediction Step:

$$x_{k|k-1} = Fx_{k-1} + Bu_{k-1} \quad (34)$$

$$P_{k|k-1} = FP_{k-1}F^T + Q \quad (35)$$

where, F: State transition matrix, B: Control input model, omitted if there's no control input, Q: Process noise covariance.

Updating Step:

$$y_k = z_k - Hx_{k|k-1} \quad (36)$$

$$S_k = HP_{k|k-1}H^T + R \quad (37)$$

$$K_k = P_{k|k-1}H^T S_k^{-1} \quad (38)$$

$$x_k = X_{k|k-1} + K_k y_k P_k = (I - K_k H)P_{k|k-1} \quad (39)$$

where, H: Observation model, R: Measurement noise covariance, y_k: Innovation or measurement residual, S_k: Innovation covariance, K_k: Kalman gain, z_k: Measurement vector.

Motion Cost (Mahalanobis Distance):

$$d^2(z_i, x_j) = (z_i - Hx_j)^T S_j^{-1}(z_i - Hx_j) \quad (40)$$

where, z_i: Detection, x_j: Predicted state, S_j: Covariance of the predicted state.

Appearance Cost (Cosine Distance):

$$d_{cos}(a_i,\ a_j) = 1 - \frac{a_i \cdot a_j}{\|a_i\|\|a_j\|} \tag{41}$$

where, a_i: Appearance feature of detection, a_j: Appearance feature of the track.

Combined Cost:

$$C_{i,j} = \lambda d^2(z_i, x_j) + (1-\lambda) d_{cos}\,(a_i, a_j) \tag{42}$$

where, λ: Weighting factor balancing motion and appearance costs.

Loss Function for LLM Fine-Tuning:

For the chatbot module, the Large Language Model (LLM) is fine-tuned using the cross-entropy loss function, which measures the divergence between predicted token probabilities and ground-truth labels.

$$\mathcal{L} = -\sum y \log\,(\hat{y}) \tag{43}$$

where, y_i: ground-truth token distribution, $\hat{y}_i$: predicted probability, and N: number of tokens

4.2.18 Trajectory Formation

Update the tracks according to the assignment for every frame. The method to create trajectories by appending the detected positions to the appropriate tracks is as follows.

$$Trajectory_i = \{(x_{i,1}, y_{i,1}, t_1), (x_{i,2}, y_{i,2}, t_2), \ldots, (x_{i,n}, y_{i,n}, t_n) \tag{44}$$

where, Trajectory_i denotes the movement path of the i^{th}object across frames, $(x_{i,k'}\, y_{i,k})$ are spatial coordinates at time t_k, and n is the total number of frames. This representation is used to analyze motion patterns and customer behavior in the scene.

4.2.19 Pattern Mining

When examining recorded trajectories to find significant patterns and behaviors, the process is known as pattern mining in the context of user video tracking. In this case, cluster analysis and sequential pattern mining are the two main methods for pattern mining.

4.2.20 Cluster Analysis

It groups users with comparable movement patterns. While K-means divides the data into K clusters by minimizing the variation among clusters, DBSCAN uses density to identify clusters. The number of clusters (k) is selected using the elbow method, minimizing the Within-Cluster Sum of Squares (WCSS). DBSCAN parameters are set as ε = 0.5 and minPts = 5, empirically tuned for dense retail scenarios.

4.3 IoT-Chatbot Integration

IoT devices such as smart cameras, RFID sensors, and BLE beacons capture real-time customer movement data. This data is transmitted to an edge server where inference is performed. The chatbot interface queries analytics results via REST APIs. For example: (i) a store manager queries footfall density via chatbot, (ii) system retrieves clustered trajectory insights, (iii) chatbot returns recommendations such as optimal product placement.

4.4 Hyperparameter Settings

Table 2. Hyperparameter Settings of the Proposed Model

Component	Hyperparameter	Value / Description
YOLOv4 Training	Learning Rate	0.001
	Batch Size	16
	Number of Epochs	50
	Input Image Size	416 × 416
DBSCAN Clustering	Epsilon (ε)	0.5
	Minimum Points (minPts)	5
K-means Clustering	Number of Clusters (k)	Determined using Elbow Method

The hyperparameter values, chosen through trial-and-error and references from past research to attain optimum performance from the detection and clustering modules, are presented in Table 2. YOLOv4 parameters have been adjusted for optimum accuracy and processing speed, while clustering parameters have been set for accurate identification of customer movements.

5. Results and Discussion

Integrating YOLOv4 and pattern mining techniques offers an effective way to analyze consumer behavior in retail stores. This section explores the implications of the findings, provides the main conclusions, and evaluates the system. The results in the table below suggest that the model generates trustworthy detection results, which are necessary for further analyzing user motions. High accuracy implies that the model rarely confuses non-customer objects with customers, while high recall means that the majority of customer instances can be detected.

The following table presents the algorithm's accuracy, precision, recall, and F1 score results, with the model demonstrating its superiority with rates of 98%, 99.02%, and 96.7%, respectively.

The definition of accuracy within the scope of object detection refers to the variation between the number of objects detected and the number of objects that should have been detected as shown in Figure 3. YOLOv4 refers to an architecture used for detecting objects in images. There exist methods by which users in videos can be marked using blue lights through the use of YOLOv4 to detect and highlight them. This involves extracting all frames from the video and scaling them into 416×416 pixels for input into the YOLOv4 framework.

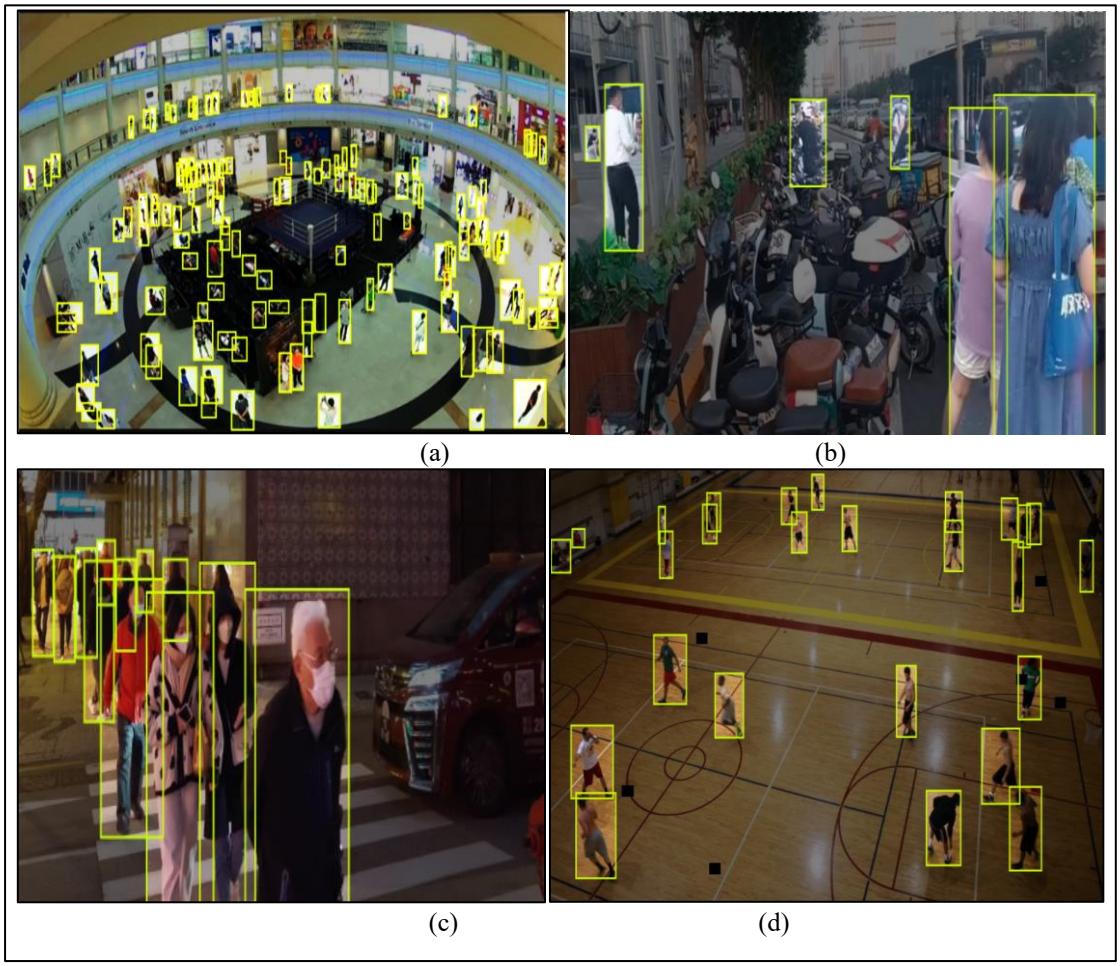

Figure 3. Object Identification Using the Proposed YOLOv4 Architecture in the Image (a)-(d) Identified Users in a Video Sequence Frame

The model is evaluated using detection, tracking, and clustering metrics to ensure comprehensive validation.

Table 3. Comparison of Results with Competitive Methods

Method	Precision %	Recall %	F1-Score %
Proposed YOLOv4	98	99	96
YOLOv3 [19]	87	84	85
Faster R-CNN [20]	85	82	83
SSD [21]	80	78	79
RetinaNet [22]	88	85	86

As shown in Table 3, YOLOv4 outshines YOLOv3, Faster R-CNN, SSD, and RetinaNet regarding accuracy, precision, and recall compared to other object recognition models used for analysis of user motion. As per the results obtained, the accuracy, recall, and precision of YOLOv4 outperformed YOLOv3 [19], Faster R-CNN [20], SSD [21], and RetinaNet [22]. Due to its high efficiency, YOLOv4 is the best choice for applications that need accurate and thorough monitoring of user movement.

Table 4. Customer Flow Statistics for Store Classifications

Class Name	Thursday			Saturday		
	Customer Flow	Ratio of Customer Flow	Class Flow Density * (customers/m^2)	Customer Flow	Ratio of Customer Flow	Class Flow Density * (customers/m^2)
Clothing	138,397	0.6812	6.1947	246,824	0.6123	11.7997
Food	39,770	0.1980	2.7241	66,791	0.1676	4.6695
General	20,870	0.0987	3.4820	38,382	0.1924	6.9855
Shoes and Accessories	8157	0.0836	5.2478	24,878	0.1617	14.8118

Table 4 above shows an analysis of customer migration trends in various retail categories. This table shows the average client population, busiest hours of flow, and the total flows of various store classes like electronic stores, clothing stores, and food stores. It can show how customer flows vary in distribution and intensity within these stores, hence giving insight into the stores that have either high or low traffic rates. Implementing proper strategies, enhancing retail design, and understanding customer needs through analyzing such statistics would help improve customer participation. The flow density ratio in a class of stores is determined by dividing the whole area of a store class by its number of customers.

Table 5. Stage-Wise Computational Latency

Stage	Latency (ms/frame)
Preprocessing (Lucas–Kanade)	8–12 ms
YOLOv4 Inference	15–25 ms
Tracking (Deep SORT)	5–10 ms
Pattern Mining	10–18 ms
Total Pipeline	~40–60 ms

From Table 5, it can be seen that the proposed pipeline can provide close to real-time results with an overall latency of 40-60 ms per frame. The Lucas-Kanade preprocessing algorithm consumes 8-12 ms, YOLOv4 object detection consumes 15-25 ms, DeepSORT tracking consumes 5-10 ms, and finally, pattern mining takes 10-18 ms.

In addition to object detection accuracy, the performance of the proposed solution on motion analytics was assessed via the use of multi-object tracking and clustering metrics.

As illustrated in Table 6, the very high MOT accuracy (MOTA) of 92.4% means that our solution performs robust and efficient tracking with very few tracking errors. In addition, with only 12 ID switches, we have very stable identification of individuals across all frames. Finally, with cluster purity of 89.6%, it is evident that the pattern mining model successfully clusters together customers with similar behavioral patterns.

Table 6. Motion Analytics Performance Evaluation

Metric	Value
MOTA (%)	92.4
ID Switches	12
Cluster Purity (%)	89.6

YOLOv4 is capable of processing images at a speed of 110 frames per second on an Intel Atom X7-E3950 HD505/GPU, thus assuring instant recognition of clients. The high accuracy rate, with mAP greater than 98% of the dataset, enables precise detection of client movement paths. The combination of this technique with pattern mining makes it possible to

analyze the behavioral aspects of customers by identifying their preferred ways through shops and other behavioral characteristics.

Table 7 proves the consistency of the proposed model on different datasets since it has the highest efficiency when working with the COCO dataset due to the high number of annotated data samples in the set. At the same time, there is a slight decrease in efficiency with the Mall Dataset due to complex motion patterns and crowded scenes.

Table 7. Dataset-wise Performance Evaluation

Dataset	Precision (%)	Recall (%)	F1-Score (%)	mAP (%)	Remarks
Surveillance for Retail Stores [23]	97.5	98.2	97.8	97.2	Strong performance in controlled retail environment
COCO [24]	98.0	99.0	96.7	97.8	Robust general object detection
UCF101 [25]	96.2	99.8	96.0	95.5	Effective for motion-based activity analysis
KITTI [26]	98.1	98.5	96.8	96.2	Good performance in outdoor scenes
Cityscapes [27]	98.8	98.2	95.5	96.0	Accurate urban scene detection
Mall Dataset [28]	98.9	98.8	95.3	94.6	Slight drop due to crowd density

Table 8. Cross-Dataset Generalization Analysis

Dataset Type	Avg Precision (%)	Avg Recall (%)	Avg F1-Score (%)
Retail-specific	96.7	96.5	96.6
General datasets	97.3	97.0	97.1

According to Table 8, the cross-dataset generalization test results show that the proposed model sustains excellent performance irrespective of whether retail or general benchmark datasets are used. For retail datasets, the model is characterized by high precision of 96.7%, recall of 96.5%, and an F1-score of 96.6%. As a result, it can be concluded that the model has the ability to perform efficiently under real-life circumstances involving changing customer activities and crowd density. The same applies to the performance recorded for general benchmark datasets, with even better figures such as average precision of 97.3%, recall of 97.0%, and an F1-score of 97.1%.

Other techniques, such as heatmaps, can show areas with heavy foot traffic, and anomaly detection techniques will find any abnormal behavior patterns. Examples include cluster detection techniques, such as DBSCAN, that help detect natural clusters of customers behaviour, and sequence mining to identify patterns and predict behavior. This combination makes it possible to design store layouts effectively, provide excellent customer service, and create a unique experience. Generally, combining the use of YOLOv4 and pattern mining with other techniques provides insights into customer behaviors.

Table 9. Failure Case Analysis

Scenario	Issue Observed	Impact on Model
Dense crowd	Overlapping objects	Missed detections
Occlusion	Partial visibility	Identity switches
Low illumination	Poor feature extraction	False negatives
Motion blur	Distorted frames	Reduced accuracy
Background clutter	Similar patterns	False positives

Despite achieving high overall performance, the proposed model exhibits certain limitations under challenging conditions. Failure cases are primarily observed in scenarios involving dense crowd regions, where multiple individuals overlap, leading to missed

detections or incorrect bounding box assignments. Additionally, low-light environments and motion blur can reduce detection confidence, resulting in false negatives. Tracking errors such as identity switches may occur when individuals with similar appearances intersect or occlude each other. Furthermore, complex background patterns and reflections in retail environments may introduce false positives. These observations highlight the need for further improvements in handling occlusion, illumination variability, and fine-grained feature discrimination.

6. Conclusion

The proposed solution for boosting the efficiency of retailers' activities by using YOLOv4 along with user motion analytics via a pattern mining method presents a holistic way to enhance retailer’s performance and customers' experience. Proper and real-time analysis of client movements is provided with the help of combining advanced object detection using YOLOv4 equipped with CSPDarknet53 architecture and a Feature Pyramid Network. Client motion paths can be followed by ByteTrack or Deep SORT to conduct further detailed research. By implementing clustering and sequential pattern mining, along with many other methods, it becomes possible to derive meaningful insights from clients' movements and interactions. Real-time implementation enables constant video feed analysis and production of insights. Implementation in a real-time environment implies a constant analysis of video streams, allowing the derivation of useful information for adjusting store layout and conducting targeted marketing strategies. This method enables merchants to make well-informed decisions, increasing clients' satisfaction, efficiency, and making a push toward expansion.

References

[1] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934 (2020).

[2] Celestin, Mbonigaba, M. Vasuki, S. Sujatha, and A. Dinesh Kumar. "How Businesses Create Personalized Experiences to Boost Customer Retention: The Role of Technology and Human Interactions in Customer Satisfaction." International Journal of Applied and Advanced Scientific Research 9, no. 2 (2024): 75-80.

[3] Dhanushkodi, Kavitha, Akila Bala, Nithin Kodipyaka, and V. Shreyas. "Customer Behavior Analysis and Predictive Modeling in Supermarket Retail: A Comprehensive Data Mining Approach." Ieee Access 13 (2024): 2945-2957.

[4] Sharma, Dheeraj. "A Study of Generative AI-Driven Innovation in Retail: Enhancing Customer Experience and Market Competitiveness." European Economics Letters 15, no. 3 (2025).

[5] Haque, Ahasanul, Naznin Akther, Irfanuzzaman Khan, Khushbu Agarwal, and Nazim Uddin. "Artificial Intelligence in Retail Marketing: Research Agenda Based on Bibliometric Reflection and Content Analysis (2000–2023)." In Informatics, vol. 11, no. 4, MDPI, 2024, 74.

[6] C. Zhu, J. Jia and T. Arslan, "FVOR-YOLO: A Real-Time Model for Fruits and Vegetables Detection in Complex Supermarket Self-Checkout Environments," in IEEE Internet of Things Journal, vol. 13, no. 7, 1 April1, 2026, 13872-13887.

[7] Kimwaki, Benedict Mutnda. "The Efficacy of the Office of the Data Protection Commissioner in Safeguarding the Right to Privacy in Kenya: An Empirical Review." The Easta Journal Law and Human Rights 4, no. 01 (2025): 85-94.

[8] Sharma, Alka, Vibhu Johar, and Ketan Bhatt. "Technology in Fashion Retail: Exploring the Nexus of In-Store Technology, Customer Experience, and Store Image." The International Review of Retail, Distribution and Consumer Research 35, no. 3 (2025): 313-333.

[9] Mattegunta, Venkata Krishna Pradeep. "Retail-Time Inventory Visibility: Transforming Retail Operations Through Real-Time Data Integration." Journal of Computer Science and Technology Studies 7, no. 3 (2025): 58-64.

[10] F. Li and J. Xu, "Revolutionizing AI-Enabled Information Systems Using Integrated Big Data Analytics and Multi-Modal Data Fusion," in IEEE Access, vol. 13, 2025, 212316-212340.

[11] Kusumawati, Ratna. "Integrating Big Data Analytics into Supply Chain Management: Overcoming Data Silos to Improve Real-Time Decision-Making." International Journal of Advanced Computational Methodologies and Emerging Technologies 15, no. 2 (2025): 17-26.

[12] Patel, Saumil, and Rohith Naini. "Advanced Computer Vision-Based Retail Analytics: A Novel Approach to Customer Behavior and Product Movement Tracking." In IISE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2025, 1-6.

[13] Sapkota, Ranjan, Rahul Harsha Cheppally, Ajay Sharda, and Manoj Karkee. "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection." arXiv preprint arXiv:2509.25164 (2025).

[14] Velasquez, Juan D. "An Analysis of Trends, Challenges, and Opportunities in Retail Analytics." International Journal of Market Research 67, no. 4 (2025): 394-422.

[15] Muniasamy, Anandhavalli, Arshi Naim, and Sayeda Meeraj. "Leveraging Queuing Theory for Efficient Electronic Business Management in Supermarkets: A Case Study of Giant Stores in the Gulf Region." Trends in Business Process Modeling and Digital Marketing: Case Studies and Emerging Technologies (2024): 136-147.

[16] Ibitoye, Ayodeji OJ, Oluwaseun Kolade, and Olufade FW Onifade. "Customer Retention Model Using Machine Learning for Improved User-Centric Quality of Experience Through Personalised Quality of Service." Journal of Business Analytics (2025): 1-19.

[17] Stylianou, Tasos, and Aikaterina Pantelidou. "A Machine Learning Approach to Consumer Behavior in Supermarket Analytics." Decision Analytics Journal 16 (2025): 100600.

[18] Vaneeta, M., Swathi Mugada, Prachi Bhausaheb Patil, V. J. Preethi, and K. Deeksha. "Real-Time Object Detection for an Intelligent Retail Checkout System." In 2025 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), IEEE, 2025, 1-6.

[19] Redmon, Joseph, and Ali Farhadi. "Yolov3: An Incremental Improvement." arXiv preprint arXiv:1804.02767 (2018).

[20] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in neural information processing systems 28 (2015). 1137-1149.

[21] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single Shot Multibox Detector." In European conference on computer vision, Cham: Springer International Publishing, 2016, 21-37.

[22] Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection." In Proceedings of the IEEE international conference on computer vision, 2017, 2980-2988.

[23] Surveillance for Retail Stores, https://www.kaggle.com/competitions/surveillance-for-retail-stores/data, last accessed on 12th April 2026.

[24] COCO Dataset 2017, https://www.kaggle.com/datasets/sabahesaraki/2017-2017, last accessed on 12th April 2026.

[25] UCF101 Videos, https://www.kaggle.com/datasets/pevogam/ucf101/data, last accessed on 12th April 2026.

[26] KITTI Object Detection, https://datasetninja.com/kitti-object-detection, Last accessed on 12th April 2026.

[27] Cityscapes Image Pairs, https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs/data, last accessed on 12th April 2026.

[28] Mall Dataset, https://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html, last accessed on 12th April 2026.