Analyzing computer vision models for detecting customers: practical experience in a mexican retail

Alvaro Fernández Del Carpio [a,1,*]

[a] Department of Software Engineering Universidad La Salle, Arequipa, Peru
[1] alfernandez@ulasalle.edu.pe
* corresponding author

ARTICLE INFO

Article history
Received May 12, 2023
Revised July 18, 2023
Accepted August 30, 2023
Available online February 29, 2024

Keywords
Computer vision
Object detection
Physical retail
YOLO
SSD Mobilenet

ABSTRACT

Computer vision has become an important technology for obtaining meaningful data from visual content and providing valuable information for enhancing security controls, marketing, and logistic strategies in diverse industrial and business sectors. The retail sector constitutes an important part of the worldwide economy. Analyzing customer data and shopping behaviors has become essential to deliver the right products to customers, maximize profits, and increase competitiveness. In-person shopping is still a predominant form of retail despite the appearance of online retail outlets. As such, in-person retail is adopting computer vision models to monitor store products and customers. This research paper presents the development of a computer vision solution by Lytica Company to detect customers in Steren's physical retail stores in Mexico. Current computer vision models such as SSD Mobilenet V2, YOLO-FastestV2, YOLOv5, and YOLOXn were analyzed to find the most accurate system according to the conditions and characteristics of the available devices. Some of the challenges addressed during the analysis of videos were obstruction and proximity of the customers, lighting conditions, position and distance of the camera concerning the customer when entering the store, image quality, and scalability of the process. Models were evaluated with the F1-score metric: 0.64 with YOLO FastestV2, 0.74 with SSD Mobilenetv2, 0.86 with YOLOv5n, 0.86 with YOLOv5xs, and 0.74 with YOLOXn. Although YOLOv5 achieved the best performance, YOLOXn presented the best balance between performance and FPS (frames per second) rate, considering the limited hardware and computing power conditions.

1. Introduction

Computer vision is a powerful modern technology that allows users to analyze and interpret visual input such as images and videos. This technology is used mostly in sectors such as industry, robotics [1], medicine [2], and vehicle transit [3]. Retail is one of the most important contributors to the global economy, boosted mainly by the expansion of retail businesses in various regions, resulting in substantial and significant gains [4]. The market's annual sales volume reaches approximately 27 trillion dollars [5]. While North America remains the largest sector in terms of revenue, the retail industry in Latin America and Asia are experiencing steady growth [6]. In recent years, physical retail stores have faced a disadvantage when compared to online retailers like Alibaba, Amazon, and Mercado Libre. The latter leverage massive user databanks to monitor and analyze shopping behavior.

In response, physical retailers have adopted technologies to monitor different aspects of in-person customer behavior. Surveillance cameras have been installed in many retail locations for security purposes

https://doi.org/10.26555/ijain.v10i1.1112

and to capture business intelligence information. Additionally, an increasing number of computer vision solutions are being developed for the retail industry. These solutions aim to automate store shelf monitoring [7], report out-of-stock products [8], identify misplaced products [9], evaluate retail service encounters with the analysis of consumers' facial expression [10], grocery products recognition [11], and detect and track customers [12]. Researches have also explored the use of image-based localization methods for detecting objects [13], enabling retailers to receive real-time data. This data, in turn, helps them to establish better security controls, devise effective marketing and logistics strategies, and deliver tailored products to specific customers.

This research focuses on the development of a computer vision-based solution for detecting customers in physical retail stores. The solution needs operate efficiently with limited hardware and computing power, considering the conditions and available devices at Steren's retail stores, that was considered as a unique opportunity to develop a computer vision-based solution. The project was managed by Lytica [14], a company specialized in developing software solutions using artificial intelligence (AI), being one of the authors member of the software team for this project. The project consisted of analyzing various computer vision-based models to find out the most effective solution according to the focus previously mentioned. This project is part of a larger initiative to identify and track customers, age and gender classification, and generate heat maps to identify zones of high customer concentration. The research's development is important for retail stores as it ensures accurate person detection to further identify individual features such as age, gender, and movement patterns.

This research considered several significant challenges for real-time scenarios, including: 1) obstruction and proximity of the customer image, 2) lighting conditions, 3) different points of view of the camera, 4) camera angle, 5) camera distance from a customer, 6) image quality, and 7) scalability of processes. Additionally, each store had unique elements such as shelves, showcases, sales items, etc., making it more challenging for the system to handle the complexity of detecting people. The development of the solution involved the following steps: building and preparing the dataset, configuring the models, conducting experiments with the models, and evaluating the results.

The rest of the paper is structured as follows: In Section 2 we elaborate the background and the related works, in Section 3 we describe the method and resources used in this research. In Section 4, we present the experiment results, and the corresponding discussions. Finally, our conclusions are presented in Section 5.

2. Related Works

In this section, we present the key theoretical aspects of the research work, including relevant concepts and theoretical models in the field, and related works.

2.1. Retail Business

The retail sector represents a significant portion of the world's developed economies. Understanding customer attitudes and behavior is well established as an important factor in maximizing profits and enhancing the competitiveness of retail stores [15]. Effective sales staff scheduling is also essential for profitable operations, since labor costs typically comprise one of a retailer's largest expenses [16]. In order to maintain customer satisfaction, retailers must expand their strategies to include initiatives that enhance the customer experience. Knowing customer attitudes and behavior would allow retailers to offer more specific customer preferences and needs, thereby delivering personalized and targeted experiences. By staying attuned to consumer preferences and technological trends, retailers take advantage of competitors, identifying patterns and leading to more efficient resources allocation. For instance, if customers prefer mobile assistants, retailers managers can invest in these technologies to enhance and foster a more positive shopping experience as well as loyalty. These also bring benefits to staff to focus on other essential tasks [15]. With information gathered from customers, retailers can make better predictions, and build tools to help to make decisions in favor of their products [17]. According to [18], 46% of retailers are considering boosting investments in digital channels, and 43%

of brands are planning to enhance real-time visibility of products in nearby stores. With the rise in e-commerce orders, numerous retailers are now fulfilling a significant portion of these transactions through their physical stores [19].

Using cutting-edge information technologies enables retailers to identify potential customers and gather relevant information to make decisions [20]. The retail industry has undergone significant changes in recent years, with the evolution towards Retail 4.0 involving the inclusion of various technologies such as AI, cloud computing, internet of things (IoT), augmented reality (AR), and big data analytical (BDA) [21]. Although the adoption of these technologies was previously low in many developing countries, the COVID-19 pandemic has accelerated their adoption [22].

2.2. Computer Vision

Computer vision is an AI field that allows computers and systems to capture valuable information from videos, images, and other visual components to take actions or provide recommendations [23]. These methods can identify and classify objects, resulting in recommendations or insights based on data from different systems [24]. Computer vision techniques aim to identify desired images from the surrounding background by discriminating among various image attributes, such as edges, colors, textures, corners, and other properties [25]. Through deep learning techniques that guide image processing and analysis, computer vision allows computers to understand and interpret visual information. Most computer vision workflows consist of input, feature extraction, and feature analysis. This technology has numerous applications, including object detection [26], motion tracking [27], scene reconstruction [28], semantic image segmentation [29], image enhancement [30], action recognition [31], and human pose estimation [32]. In this research work, we will refer exclusively to computer vision and its application to image processing in the retail sector.

Some solutions are oriented towards indoor navigation assistants for visually impaired people, utilizing techniques such as YOLO (You Only Look Once), monocular depth estimation, linear interpolation, and HRT (Head Related Transfer Function) to generate binaural sounds for the detected objects and specify the position of objects in 2D space [33]. Also, combining dead reckoning through visual-inertial odometry with the user's location enabled them to determine turn-by-turn directions [34]. On the other hand, retail store solutions involve shelf control, utilizing low power microcontroller with embedded cameras [35]. An important aspect is the product counting using YOLOv5 and DeepSORT [36] or Mobilnet V3 [37] algorithms, which automate fast product checkout. Similarly, counting shoppers can be done using semantic segmentation with convolutional neural networks and depth-sensors [38].

2.3. Object Detection

Object detection involves identifying the object's class and estimating its location by producing a bounding box around it. The identifications of instances of a class from an image are known as single-class object detection and multi-class object detection [39]. Object detection algorithms typically involve two main components: a classifier, which predicts the class or category of an object, and a localizer, which determines the object's position and size within the image. Object detection and localization are typically employed to identify and locate an object in a given image while determining its corresponding category under object classification. The conventional process of detecting objects is categorized into three main stages: a) region detection, b) feature extraction, and c) classification. The region detection process identifies the regions of interest where the object detection scheme should be applied. The feature extraction stage obtains features to get semantic and visual information of the object within the identified region [40]. Object detection is a crucial component of various applications, including autonomous vehicles [41], robotics [42], and image recognition systems [43].

The detection of objects in images is achieved through machine learning algorithms and involves a convolutional neural network that detects a limited (or specific) number of objects within the image. To be considered as such, a detection machine learning algorithm must: a) Detect multiple objects with their classes, and b) Give the X and Y position of the bounding box object in the image (or its center) and draw a rectangle around it.

The development team at Lytica Company experimented with several computer vision models from the literature, prioritizing those that were cost-effective and provided accurate results. To determine which of these approaches would be a suitable solution for the requirements and conditions of the test retail store in Mexico, a detailed analysis was performed for each one. The conceptual description of each type of network model is presented below:

- SSD MobileNetV2: Single-Shot MultiBox Detector (SSD) MobileNetV2 is a fast one-stage object detection model commonly deployed on mobile devices. This model relies on a feed-forward convolutional network that generates a set of fixed-size bounding boxes and scores for object class instances in those bounding boxes. The early network layers are based on a conventional design for high-quality image categorization called the base network. The model includes convolutional feature layers that decrease in size towards the end of the base network to predict detection at multiple scales. Additionally, each feature cell is associated with a set of default bounding boxes [44]. SSD uses hierarchical structure to extract typical and representative features from input images. It gradually acquires the advanced features needed from low-level features [45].
- Yolo-FastestV2: It is simple, fast, small, easily portable, resource-efficient, and high-performing model that consumes less energy than other models [46]. It is a popular lightweight object recognition model that replaces the backbone network of YOLOv5 with ShuffleNet V2 and simplifies the feature pyramid network structure. With a parameter size of only 237.55 kB, it can achieve real-time detection on embedded devices with limited computing resources. However, the model's feature fusion lacks richness, resulting in inadequate extraction of location and semantic features of small objects. This situation results in inaccurate positioning of bounding boxes and a low recognition rate [47]. Some advantages of this model include: a) The backbone is replaced with a lighter shufflenetV2 structure; b) It features various loss weights for various scale output layers; c) YOLOv5 is substituted for its anchor matching mechanism and loss; d) It decouples the detection head, the foreground background classification, the category classification, and the detection frame regression; and e) It considers the softmax cross-entropy from sigmoid.
- YOLOv5: It comprises a set of compound-scaled object detection architectures and models that are pre-trained on the COCO dataset. It has minimal capabilities for Test Time Augmentation (TTA), model ensemble, hyperparameter evolution, and can convert to ONNX, CoreML, and TFLite formats. Developed by Ultralytics, YOLOv5 is an open-source research project that aims to advance future vision AI technologies [48]. The YOLO (You Only Look Once) object identification method splits images into a grid system, where each cell is responsible for detecting items inside it. Due to its speed and accuracy, YOLO is one of the most well-known object identification algorithms. The YOLOv5 model begins by adaptively scaling the input image and utilizing a genetic algorithm to automatically learn and adjust the depth and width of the network to meet the needs of different scenes [49]. It consists of four modules: input, backbone, neck, and prediction, which work together to achieve its objective [50]. YOLOv5 uses CSPNet as the backbone to extract features, achieving a better balance of inference speed and accuracy.
- YOLOX: This high-performance, advanced object detection system integrates anchor-free mechanisms, which significantly reduces the number of parameters. It supports ONNX, TensorRT, NCNN, and Openvino. YOLOX is an improved version of YOLO that includes a decoupled head for higher performance and an advanced label-assigning strategy. These improvements were essential for achieving an end-to-end property. YOLOX achieves a superior balance between speed and accuracy compared to its competitors [51].

2.4. Related computer vision works

In this section, we examined previous research and efforts in the field of computer vision. We explored various solutions developed for detecting people, including categorizing, counting, and tracking them; as well as recognizing products. Identifying customers entering retails stores provides vital information to retail managers, which can be used to make strategic and marketing decisions. For example, this data can reveal how many customers visit the store, individual or group visits, and the routes they follow indoors. Analyzing customer attitudes and behavior in a retail store can help maximize profits and enhance a retailer's competitiveness.

Thus, some interesting architectures were adopted to ease the customer detection [12], [52], [53], [54]. In [12], the deep learning-based framework enabled to identify faces (using the Haar Cascade object detection model), gender and age (using Wide ResNet 16-8 with a 64x64 RGB image as input for the network), and facial expressions of customers (using the mini Xception model). Similarly, [55] implemented a system to detect faces, estimate age and gender of customers, and track them into the shopping area. The researchers adopted a framework for customer recognition, which consisted of a Siamese network with a 99.95% value of accuracy. The facial images captured through a CNN allowed the system to estimate the customer's age and gender, achieving a classification accuracy of 71.8% for age and 85.2% for gender estimation. On the other hand, the novel architecture proposed by [52] was not only to detect people, but also to count them through four programs: a) publisher, b) object detection, c) video combiner, and d) dashboard program. The publisher program sends CCTV videos to the object detection and video combiner programs. The object detection program implements a YOLOv3 algorithm using a Darknet-53 network for detecting objects in CCTV videos. It sends the number of detected objects and a video with detections to the dashboard program. Finally, the video combiner mix the data received from the publisher program into a single video. The two-stage scheme developed by [53] for detecting people included motion detection and human detection. Various methods were used for detecting motion, such as background subtraction, frame difference, and optical flow. Feature extraction is performed using Histogram of Oriented Gradients (HOG), which is invariant to changes in shadows, contrast, and illumination. This technique can be deployed on embedded devices such as Raspberry Pi, which have the necessary capabilities. Special environmental conditions for operating the computer vision solutions were described in [12] through its framework to operate under poor illumination, varying angles of view, and over-exposure of the background. In many cases conventional cameras for CCTV were used, but with fisheye cameras were also achieved notable results. These cameras offer a larger field of view and reduced occlusion, making them ideal for applications like video surveillance and customer flow statistics. However, the distortion inherent in fisheye images presents challenges for existing object detection algorithms. To tackle these issues, [54] presented a modification based on YOLOv5 that incorporated a radius-aware loss function to adapt to the distortion effects in fisheye images.

Regarding tracking people, it was an important aspect considered by retailers and motivated mainly by security reasons, retail surveillance, and marketing strategies. Tracking has two steps: to measure the distance between detections to determine the likelihood of correspondence to the same object; and to derive the assignment of pairs of detections from two frames, that can be resolved using algorithms like Kuhn-Munkres [56]. Additionally, the analysis of groups of customers to be identified and tracked implied the implementation of a monocular real-time computer vision system [57]. This solution required a temporal silhouette map that integrated foreground silhouette detection and motion detection information. The group of individuals was analyzed by calculating relative distances using the trajectories of each individual. Continue people tracking can result in high computational costs. To overcome this issue, [58] suggested that models track over short periods to associate the detected faces across time, resulting in computational efficiency.

The identification of retail products implied to not only recognize categories of products, but also analyze store shelves out of stock (SOOS). [59] built a system to collect data and survey store shelves through autonomous navigation and monitoring. A deep convolutional neural network (CNN) was used to analyze images and map SOOS with an accuracy of up to 87%. As to classifying retail products, this carries important challenges such as the complexity of the scene, variability of products, uneven lighting conditions, and the viewing angle of the cameras. An interesting approach to classify retail products was based on two-level analysis. General features of the product were captured utilizing a reconstruction-classification network (RC-Net) to then be discriminated by finer characteristics utilizing a convolutional LSTM (conv-LSTM) [60]. Results showed an accuracy around 90% for the training dataset. With the purpose of optimizing the training of models when new products are added to a dataset, [11] applied an end-to-end architecture with a generative adversarial network (GAN) in order to avoid retraining the whole data. The samples generated by a GAN served to train a deep CNN to learn embedding products images in the dataset.

Furthermore, there are several methods for detecting merchandising in retail stores using computer vision techniques [60]. As important challenges identified were the complexity of the scene, variability of products, uneven lighting conditions, and the viewing angle of the cameras. The features for detecting retail products were based on key points, gradients, patterns, color, and deep learning. The problem of product detection was categorized into single product, multiple products, recognition of products, and retrieval of rack images. An extensive literature review of current studies about deep learning-based retail product recognition included analysis of the challenges, techniques, and datasets [58]. The product recognition is a complex task and is more challenging than common object detection. The associated were categorized as: large-scale classification, data limitation, intraclass variation, and model flexibility when a new product is introduced.

The numerous studies reviewed in the literature focused on developing solutions related to predicting customer characteristics and behavior in stores. The models used in these studies included for example: CNN with Siamese network [58], Wide ResNet, Adaboost, Xception [12], Fast YOLO [56], and YOLOv3 [52]. For our research, we selected models (YOLOv5, YOLO FastestV2, SSD MobileNet V2 and YOLOX) after conducting a comprehensive review of models used in the literature and multiple open-source repositories on GitHub. The primary factors influencing our choice were their reputation in the research community, the level of support and updates offered by the developers, as well as their successful applications in previous studies, as evidenced by the literature. Each chosen architecture exhibits unique characteristics in terms of precision, inference speed, and memory usage, allowing us to provide a broad comparison and evaluation, especially in the context of operating on limited hardware like the Jetson Nano.

Many of the models reviewed from the literature have benefited from access to robust hardware infrastructures, enabling the implementation of computationally intensive algorithms and models. In contrast, our work was performed under limited hardware conditions. Despite these limitations, we undertook the challenging task of applying sophisticated object detection models such as the mentioned above. The contribution of our work lies in demonstrating the feasibility and efficiency of performing complex object detection tasks in constrained hardware environments, such as the Jetson Nano.

3. Method

This section provides a detailed description of the steps followed employed for building the dataset and the models. Hence, the implemented steps included: building an own dataset with images captured from the retail stores, making some data adaptations to be processed in different models, testing the models on the Jetson Nano, and getting metrics of the models in order to obtain the most representative model as solution of the problem (See Fig. 1).

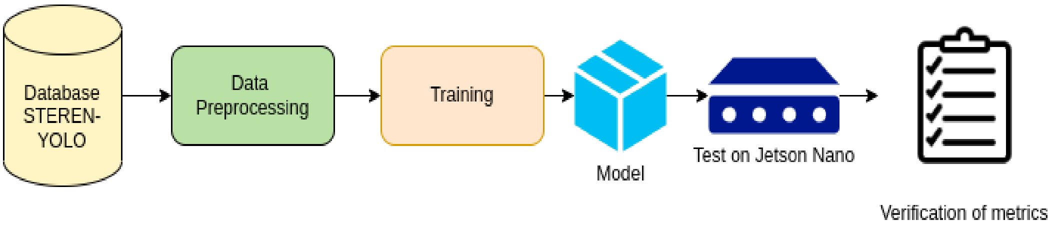


Fig. 1. Steps of the research project

3.1. Building the Dataset

This section presents the steps followed to create the dataset for the study, as described in Fig. 2.

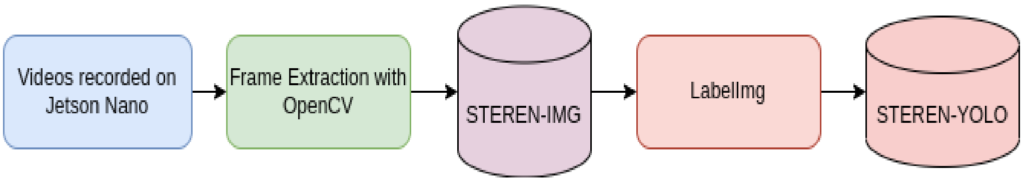
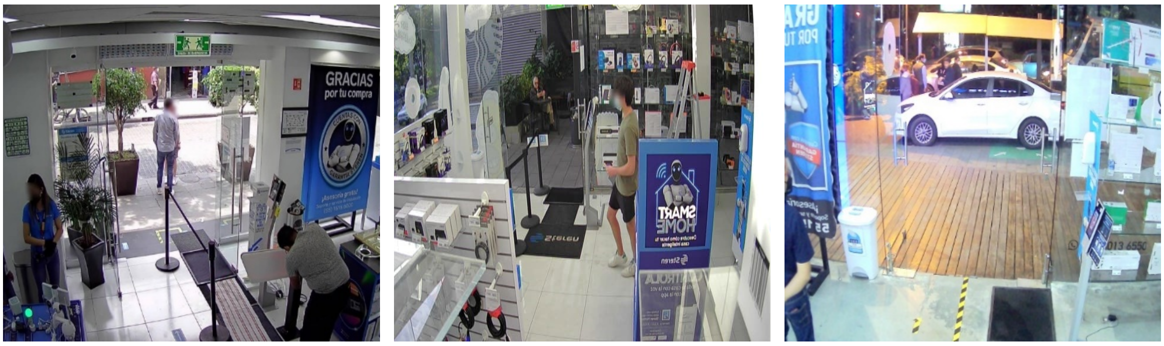


Fig. 2. Steps to create the dataset for training

For the dataset, one-minute long videos were captured randomly throughout the day from five Steren supermarket stores. Steren is a Mexican multinational company dedicated to marketing electronic products and technology. In each store, a camera was located considering factors such as vision angle, height, and lens aperture, to get the whole image of a person entering the store. The camera used to capture the videos was a Raspberry Pi Camera Module v2. The videos were saved in 1080p format at 30 FPS, and they were taken using OpenCV version 4.5.

Initially, due to budget constraints, at this point only one camera was installed at each store entrance. Subsequently, the stores will have more cameras to cover a wider retail surface. Each camera was connected to a Jetson Nano with 4 GB of RAM, and each captured video was sent to an S3 Bucket on Amazon through an OpenCV script. The videos were captured at different periods of time, and under diverse illumination conditions to make the data more useful when generating a more suitable and reliable model. These conditions are shown in Fig. 3.



(a) Natural daylight (b) Natural afternoon light (c) Natural light at night

Fig. 3. Natural light conditions in Steren Chain Store

For labeling images from the database for detection, the open source tool labelImg has been selected because it is straightforward to install and use. For this project we followed the next steps: 1) To install LabelImg from GitHub; 2) To choose the directory where the images are stored; 3) When creating bounding boxes, we had to ensure that the visible part of the people was labeled, avoiding estimated sizes or labeling assumptions; and 4) After image annotation was completed, it was saved in YOLO format. These annotations included the coordinates of the bounding box and the class label. Each image has been labeled as shown in Fig. 4. A total of 5579 images, with each image having its own label in the “txt” format. Each text file has a class ID of ”0” because this project only considers the “person” class. In YOLO format, each “txt” file includes annotations for the related image file, such as:

$$<object-class><x><y><width><height>$$

During this process, some challenges emerged that required careful attention and explanation. First, the database creation involved two persons, one for labeling and the other for reviewing the labeling. It became apparent that both individuals may have different interpretations of where exactly to draw the bounding boxes. To address this issue, clear and detailed guidelines were established to ensure agreement on the database. Second, manual annotation was a time-consuming process, especially for images containing up to 20 people. Sometimes, there were doubts about whether to label certain individuals, particularly when they were far away or obstructed. Despite these challenges, every effort was made to accurately label the data. Third, recognizing the significance of annotation quality on model performance, a separate person was assigned to review and correct the annotations, ensuring the best possible dataset.

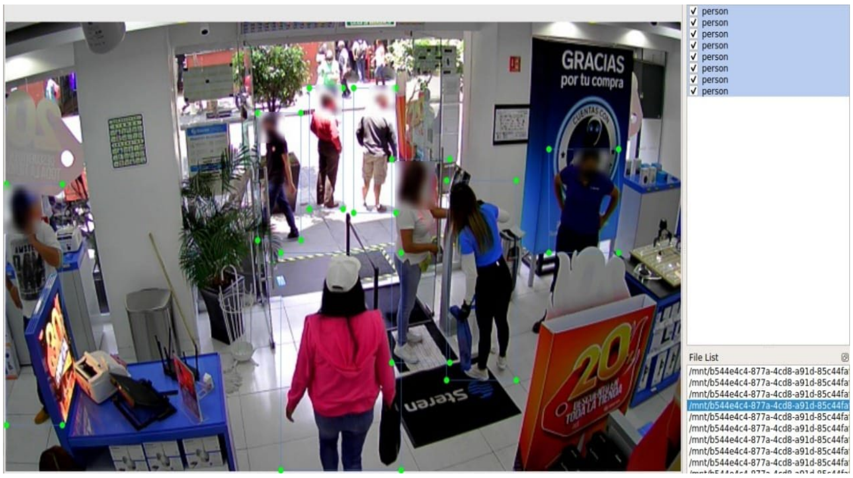

☑ person
☑ person
☑ person
☑ person
☑ person
☑ person
☑ person
☑ person
File List
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
/mnt/b544e4c4-877a-4cd8-a91d-85c44fa
GRACIAS
por tu compra
Steren

Fig. 4. Tool for labeling images (Labelimg)

Finally, we got a database which was stored in an AWS S3 bucket and divided into training (70%), validation (20%), and testing (10%), such as is shown in Table 1.

Table 1. Partition of the dataset for training, validation and testing

Database	Total data	Amount of training data	Amount of validation data	Amount of testing data
Steren Detection	5579	3905	1115	557

3.2. Identifying the Headings

This process required several changes to the input data in order to adapt it to the different detection models. The detection models included in this research were: SSD Mobilenet v2, YOLO FastestV2, YOLOv5n, YOLOv5s, and YOLOXn.

The input data was the same for YOLOv5n and YOLOv5s detection models, but some adaptations were necessary for YOLOXn and SSD Mobilenet v2 models. In the case of YOLOXn, the input data was converted from YOLO format to COCO format. For the SSD Mobilenet v2 model, the data input was changed to the PASCAL VOC format using a tool named Yolo2Pascal-Annotation-Conversion. After preparing the dataset, the next step consisted of configuring the object detection models, which are described in section 4.3. The values related to epochs, batch size, input size, and learning rate were standardized across all object detection architectures, in this way: epochs equal to 50, batch size equals to 64, input image size equals to 640x640, and the learning rate equals to 0.001.

3.3. Configuring the Models for Training

This section presents the configuration of each model and describes the training process.

- SSD Mobilenetv2: As explained in the previous section, the data was transformed into the PASCAL VOC format for this model. Once data was obtained, we used the Jetson Inference repository. After cloning this repository to the project's local files, we executed the train file "ssd.py" in the command line. After training, we obtained the weights of the architecture and transformed them into the ONNX open format to enable their execution on the Jetson Nano.
- YOLO FastestV2: This model was a suitable option to be included in this research due to its interesting architecture. The steps taken to train this network are shown below:
    - Get anchor bias. To obtain these parameters, a command was executed on the entire training database to obtain the anchors. This command is located in the official repository of the networking model.

  - Build a custom coco.data. In this file, both the number of epochs (50) and the learning rate (0.001) was standardized for all the experiments. The anchors obtained from the training dataset were configured and the dataset address and names are also specified in the 'coco.names' file.
  - Transform weights to ONNX format: After training, the weights are transformed into the ONNX format. This format is used to enable execution on the Jetson Nano.

- YOLOv5xs and YOLOv5n models: It is important to emphasize that both networks are being discussed together here because the training procedure was identical, with the only difference being the size of the backbone. For both YOLO models (v5n and v5xs), the repository is located at https://github.com/ultralytics/yolov5. Additionally, YOLOv5xs is a development based on YOLOv5s (located in the previous link). The number of convolutional layers was reduced from 5 to 3 (depth multiple) and the filters were reduced from 4 to 2 (width multiple). The depth multiple corresponds to the 'depth' of the model, meaning that it adds more layers to the neural net. In contrast, 'width' adds more filters into the layers, adding more channels to the layer outputs. These multipliers are a common way of building scalable models. These parameters can be found in the "yolov5s.yaml" file in the models' folder, which comes from the original repository. Table 2 illustrates the differences between these two models.

Table 2. Comparison between YOLOv5xs and YOLOv5n

Model	Size (pixels)	Params (Milions)	FLOPs
YOLOv5n	640	1.9	4.5
YOLOv5xs	640	1.6	3.8

For both networks, the labels were required to be in YOLO format. To achieve this, 640x640 images were labeled using the LabelImg tool. To facilitate the training process, the file called "custom data.yaml" was used, which was configured with the paths of the training, validation, and test files. The number of classes was set to 1 since only the "person" class was defined.

- YOLOXn: This network stored in the repository is available at https://github.com/Megvii-BaseDetection/YOLOX. This repository contains all the instructions to train the network. Table 3 illustrates the characteristics of this model.

Table 3. YOLOXn network features

Model	Size	mAPval 0.5:0.95	Params (Milions)	FLOPs (G)
YOLOX-Nano	640	25.8	0.91	1.08

After cloning the YOLOX network, we combined the model configuration, training, and test configuration into a single experiment file (Exp file). This allowed us to easily configure and run experiments with the YOLOX network. For this project, a copy of the Exp file named "yolox_base.py" was made and then configured with specific data, as follows: depth = 0.33; width = 0.25; input size = (640, 640); mosaic scale = (0.5, 1.5); random size = (10, 20); test size = (640, 640); exp name = os.path.split(os.path.realpath(file)).split("."); enable mixup = False; data dir = "datasets/STEREN COCO format 9k"; train ann = "instances train2017.json"; val ann = "instances val2017.json"; num classes =1.

4. Results and Discussion

4.1. Experimental results

This section shows all the results of the experiments carried out with the models. An EC2 instance on Amazon (AWS Deep Learning AMI, Ubuntu 18.04 - g4dn.2xlarge) was used to measure and compare the mean accuracy (AP) and the IoU of the trained models. This instance consisted of an Ubuntu Server

18.04 LTS, 32 GB of RAM, 8 virtual cores CPU, Nvidia T4 GPU (16 GB), 225 GB SSD storage, and a 25 Gbps network. Table 4 illustrates the results of the precision, recall, mAP 0.5, and mAP 0.5:0.95 metrics of the trained models.

Table 4. Results of metrics of the proposed detection models

Model	Precision	Recall	F1	mAP 0.5	mAP 0.5:0.95
YOLO FastestV2	0.6000	0.6774	0.6363	0.6363	0.5915
SSD Mobilenetv2	0.7485	0.7295	0.7388	0.8372	0.6742
YOLOv5n	**0.8771**	0.8458	**0.8611**	0.9265	**0.6881**
YOLOv5xs	0.8710	**0.8509**	0.8609	**0.9296**	0.6498
YOLOXn	0.717	0.759	0.7374	0.894	0.581

Fig. 5 illustrates the graphical results of the metrics in the training process.

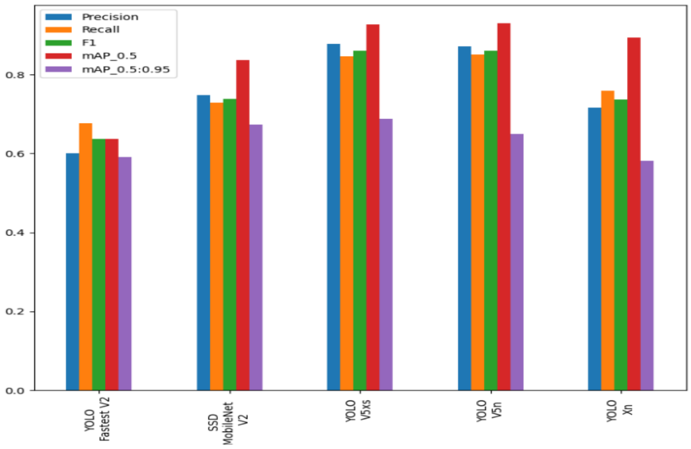


Fig. 5. Metric results from trained models

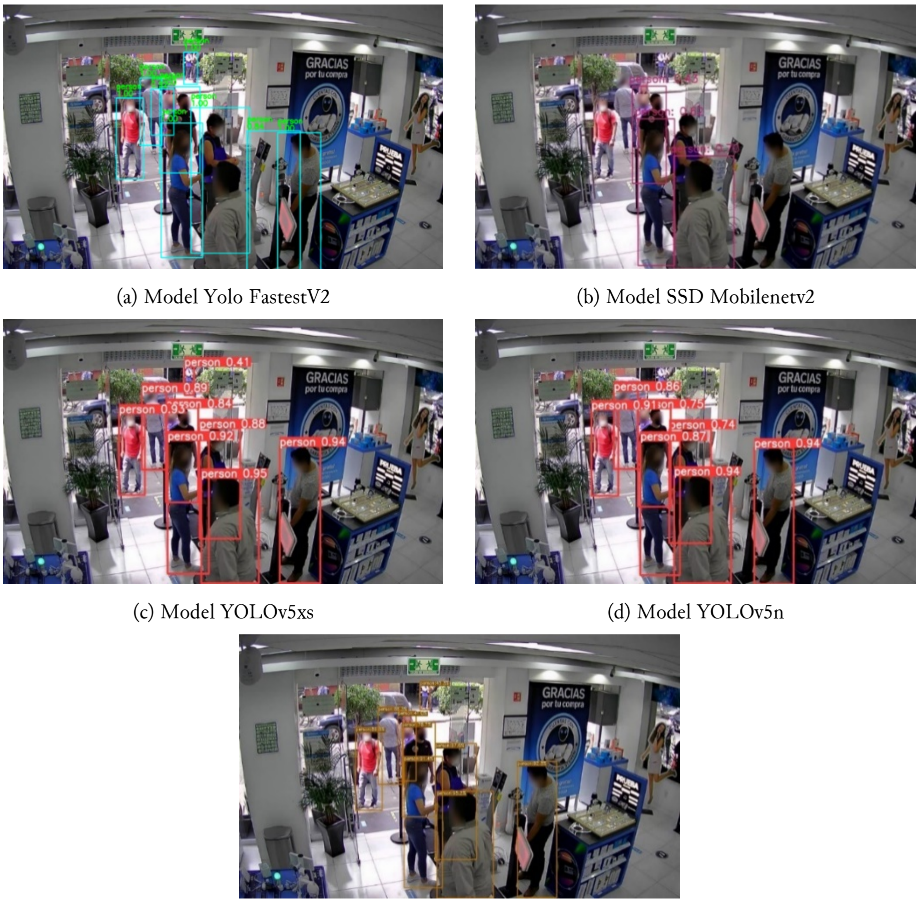
These results were achieved by manually adjusting the parameter values to ensure their reliability and integrity. The images without the presence of people have been removed from the dataset during the training process. Finally, a random sampling process was carried out on complete images. The final dataset consisted of 11,158 images, divided into a training set (4,016 images), a validation set (1,004 images), and a test set (558 images).

All the experiments were executed in a Jetson Nano with the following characteristics: 1) GPU: 128core NVIDIA Maxwell architecture based on GPU; 2) CPU: Quadcore ARM A57; 3) Video: 4K @ 30fps (H.264/H.265) / 4K @ 60fps (H.264/H.265) encode and decode; 4) Camera: MIPI CSI-2 DPHY lanes, 12x (Module) and 1x (Developer Kit); and 5) Memory: 4GB 64-bit LPDDR4; 25.6 gigabytes/second. In addition, the test took a minute and a half for each of the tested detectors to have a reliable result. These results are illustrated in Table 5.

Table 5. Network performance results on the Jetson Nano

Model	CPU Latency (%)	GPU Latency (%)	Memory Usage (GB)	Time (m)	CPU Temperature (C)	GPU Temperature (C)	FPS
YOLO FastestV2	98.3	**85**	51.8	**5:06**	**63.0**	**60.0**	**5.89**
SSD Mobilenetv2	**69.9**	96	**39.9**	7:38	67.5	63.5	3.70
YOLOv5n	106.7	99	58.8	12:31	70.0	68.5	2.27
YOLOv5xs	93.5	99	56.9	10:08	72.5	64.5	2.80
YOLOXn	88.3	98	61.2	5:20	63.5	62.5	5.42

To better understand how the Jetson Nano performs in production, the last column in Table 5 shows the FPS (Frames Per Second) performance for each model. The project places a greater emphasis on FPS performance because higher values are better for real-time execution. The results in inference of the different models are shown in Fig. 6.



(a) Model Yolo FastestV2

(b) Model SSD Mobilenetv2

(c) Model YOLOv5xs

(d) Model YOLOv5n

(e) Model YOLOXn

Fig. 6. Results in inference of the models

In Fig. 6a, a crowd of people is shown, and although the model detects people correctly in some cases, there also errors For example, it sometimes produces false positives with a higher confidence level (0.84), and it fails to detect obvious people. Although this model is quite fast, it lacks accuracy, and for our specific case, we require high precision. In Fig. 6b, the model works well, but there are instances where it fails to detect even very obvious people. In Fig. 6c, while this model generally performs quite well, it falls short in our crucial case due to its slow inference speed. In Fig. 6d, similar to the previous model, this one achieves high accuracy with increased confidence. However, the issue lies in its inference time, which is crucial for us, as we intend to use it on an edge device. In Fig. 6e this model works quite well and is efficient in terms of inference speed. It accurately fits all bounding boxes and consistently delivers good results with high confidence.

We highlight that a common pattern observed in all the networks is significant improvement under good light conditions and fewer people. Additionally, these networks also perform better when people do not have backpacks, caps, purses, glasses, etc. These characteristics are well-known in the field of computer vision. Additionally, YOLOv5 architectures produce very similar results, which can be attributed to their almost identical architectures. On the other hand, YOLOXn shows some similarity in inference time with YOLO FastestV2, but the key difference lies in YOLOXn's higher confidence in its results, as shown in Fig. 6a and Fig. 6e.

4.2. Discussion

As can be seen in the training results in Table 4, the YOLOv5 networks showed the best performance. The reason is that the YOLOv5 network architectures are much more robust than the others, we can observe this in the number of parameters and the FLOPs of each one. However, YOLO Fastest V2 showed the lowest accuracy among the tested models, possibly due to its small size. We recommend using a slightly more robust network considering the amount of data and the quality of the images in real conditions. Finally, the YOLOXn and SSD MobileNet V2 networks demonstrated moderate performance compared to the other models.

We can observe in Table 5 that the models utilized both CPU and GPU technologies, although GPU is the primary technology. The FPS metric is particularly important, as it directly affects real-time performance. In this regard, YOLO FastestV2 achieved the highest performance with 5.89 FPS, followed closely by YOLOXn with 5.42 FPS. Among the models, SSD MobileNet v2 had the best memory optimization and did not strain the resources of the Jetson Nano. On the other hand, the models based on YOLOv5 exhibited strong performance, with YOLOv5n. However, our research found that YOLOXn performed optimally, with strong results in terms of accuracy rate under real conditions and on Jetson Nano.

While it is true that this work has a limitation in terms of hardware, there are other options that could be more optimal for handling complex models. For instance, using another device like Jetson AGX Xavier or exploring different configurations for the models could be beneficial. In our case, we worked with the ONNX format, but there are also other formats such as OpenVino or TensorRT, which can help reduce inference time, although some confidence might be compromised. For this project, we chose to work with ONNX due to its generalizability. This means it can be used and tested on various types of devices, not just edge devices but also on smartphones, servers, and more.

We also believe that this solution could be generalized, taking the following points into consideration:

- The models like the ones discussed in this project are widely applicable for object detection tasks in various environments, not limited to retail.
- To ensure models generalize well across different retail environments, it is crucial to train them on diverse data. This includes data from different types of retail stores, various lighting conditions, and different individuals (customers, employees, etc.). More varied training data leads to better model generalization to new and unfamiliar environments.
- In the training data, include diverse scenarios with customers carrying different items like backpacks, caps, purses, glasses, etc., as these factors can impact the model's detection ability.
- Fine-tuning the models may be necessary to adapt them to specific conditions in other retail environments. This could involve adjusting parameters, retraining the models with new data, or applying optimization techniques to improve performance.
- The ONNX format was chosen for its ability to generalize across multiple platforms. It can be used and tested on various types of devices, not only edge devices but also on devices like smartphones and servers.
- Keeping the models up to date with the latest data is essential to maintain their effectiveness and generalizability over time, as retail environments can change.

However, it is important to remember that each retail environment is unique, and what works well in one may not perform as effectively in another. Therefore, thorough testing and validation of the models in new environments are crucial to ensure their successful generalization.

On the other hand, deploying these models in different settings can present a variety of challenges:

- Depending on the device, there may be restrictions related to memory, processing power, and other hardware limitations that could affect model performance and inference speed. For example, some models may not function as effectively on smartphones as they do on more powerful servers or specialized devices like the Jetson AGX Xavier.
- Different environments may present unique conditions for which a model has not been trained. For example, lighting conditions, crowd density, or the presence of multiple objects can greatly affect the performance of person detection models.
- To maintain optimal performance over time, regular maintenance and updates may be necessary. Changes in the environment or the behavior of individuals within that environment may require model updates or retraining.
- In our specific case (retail environment), the models must provide real-time or near-real-time inference, which can be challenging, particularly with hardware limitations or larger, more complex models.

To overcome these challenges, careful planning, regular updates, and robust hardware are essential. Comparing the results of our study with the existing literature reveals significant contributions and some novel findings. Previous research has typically focused on demonstrating the individual performance of YOLO and other models, often in isolation or in comparison to one or two alternatives. In contrast, this research directly compares the performance of five different object detection models: SSD MobilenetV2, YOLO FastestV2, YOLOv5n, YOLOv5xs, and YOLOXn.

The results indicated that the YOLOv5 models (v5n and v5xs) demonstrated the highest accuracy, recall, and F1 scores. This finding aligns with related works, where YOLOv5 is recognized for its high accuracy and robustness. However, despite the fact that these models presented a good accuracy, they exhibited lower inference speed. This aspect affects real-time person detection, especially on devices like the Jetson Nano. Although YOLO FastestV2 was not as accurate as other models, it offered the fastest inference times. We find results that establish a balance between accuracy and speed, offering a different way to evaluate the performance of customer detection models.

In contrast, the YOLOXn model, which was not mentioned in the reviewed literature, has demonstrated balanced performance in our study, offering fairly good accuracy and speed. Unlike YOLO FastestV2, YOLOXn showed higher confidence in results by maintaining adequate inference times, positioning it as a strong model for applications that require a balance between detection accuracy and speed. In addition, this study shows environmental and situational factors that can affect model performance, such as lighting conditions and the number of people present in the frame.

Overall, this research broadens the perspective on people detection models by offering a comprehensive comparison of different models, considering both their accuracy and speed, and introducing environmental factors into performance evaluation. It adds to existing knowledge by demonstrating the trade-offs between different models and suggesting that the choice of model should be tailored to the specific requirements of an application, in our case, the need for real-time sensing with good accuracy.

5. Conclusion

This research paper presents the results of analyzing highlighted small and light architectures for people detection using a real database. The models were evaluated with the F1-score metric, obtaining values of 0.64 with YOLO FastestV2, 0.74 with SSD Mobilenetv2, 0.86 with YOLOv5n, 0.86 with YOLOv5xs, and 0.74 with YOLOXn. Although YOLOv5 models demonstrated the highest accuracy, recall, and F1 scores, they exhibited lower inference speed, which impacts to the detriment of real-time person detection, when using resources like Jetson Nano. Taking into account the importance of the FPS metric on the real-time performance, YOLO FastestV2 achieved the highest score with 5.89 FPS, followed by YOLOXn with 5.42 FPS. Considering that the solution should show a strong performance and a satisfactory FPS due to limited conditions of hardware and computing power, YOLOXn was determined to be currently in production. Although YOLOXn architecture does not present the best detection results in people, nor is it the fastest, it was found to be the system with the best overall performance. The YOLOXn has good accuracy and does not sacrifice much in FPS, making it ideal for this project. To ensure the application of the models analyzed in this research in different retail environments, it is necessary to train them on diverse data considering various scenarios. Additionally, there are alternative resources that can be taken into consideration to handle models that are more complex. For instance, using a Jetson AGX Xavier or employing formats like OpenVino or TensorRT to reduce inference time. We think that this project is highly relevant as it addresses the most current techniques in retail sectors, and will be an important starting point for future research in this area.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, “Exploring impact and features of machine vision for progressive industry 4.0 culture,” *Sensors Int.*, vol. 3, p. 100132, Jan. 2022, doi: 10.1016/j.sintl.2021.100132.

[2] T. Habuza *et al.*, “AI applications in robotics, diagnostic image analysis and precision medicine: Current limitations, future trends, guidelines on CAD systems for medicine,” *Informatics Med. Unlocked*, vol. 24, p. 100596, Jan. 2021, doi: 10.1016/j.imu.2021.100596.

[3] V. Kakani, V. H. Nguyen, B. P. Kumar, H. Kim, and V. R. Pasupuleti, “A critical review on computer vision and artificial intelligence in food industry,” *J. Agric. Food Res.*, vol. 2, p. 100033, Dec. 2020, doi: 10.1016/j.jafr.2020.100033.

[4] D. Atkin, B. Faber, and M. Gonzalez-Navarro, “Retail Globalization and Household Welfare: Evidence from Mexico,” *J. Polit. Econ.*, vol. 126, no. 1, pp. 1–73, Feb. 2018, doi: 10.1086/695476.

[5] M. Naidoo and A. Gasparatos, “Corporate environmental sustainability in the retail sector: Drivers, strategies and performance measurement,” *J. Clean. Prod.*, vol. 203, pp. 125–142, Dec. 2018, doi: 10.1016/j.jclepro.2018.08.253.

[6] A. Parfenov, L. Shamina, J. Niu, and V. Yadykin, “Transformation of Distribution Logistics Management in the Digitalization of the Economy,” *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 1, p. 58, Mar. 2021, doi: 10.3390/joitmc7010058.

[7] C.-C. Lin, K. N. Ramamurthy, and S. U. Pankanti, “Moving Camera Analytics: Computer Vision Applications,” in *Embedded, Cyber-Physical, and IoT Systems*, Cham: Springer International Publishing, pp. 89–113, 2020, doi: 10.1007/978-3-030-16949-7_5.

[8] X. Ding, C. Chen, C. Li, and A. Lim, “Product demand estimation for vending machines using video surveillance data: A group-lasso method,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 150, p. 102335, Jun. 2021, doi: 10.1016/j.tre.2021.102335.

[9] A. Milella, A. Petitti, R. Marani, G. Cicirelli, and T. D’orazio, “Towards Intelligent Retail: Automated on-Shelf Availability Estimation Using a Depth Camera,” *IEEE Access*, vol. 8, pp. 19353–19363, 2020, doi: 10.1109/ACCESS.2020.2968175.

[10] E. Pantano, “Non-verbal evaluation of retail service encounters through consumers’ facial expressions,” *Comput. Human Behav.*, vol. 111, p. 106448, Oct. 2020, doi: 10.1016/j.chb.2020.106448.

[11] A. Tonioni and L. Di Stefano, “Domain invariant hierarchical embedding for grocery products recognition,” *Comput. Vis. Image Underst.*, vol. 182, pp. 81–92, May 2019, doi: 10.1016/j.cviu.2019.03.005.

[12] E. P. Ijjina, G. Kanahasabai, and A. S. Joshi, "Deep Learning based approach to detect Customer Age, Gender and Expression in Surveillance Video," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225459.

[13] Y. Jiang, D. Pang, and C. Li, "A deep learning approach for fast detection and classification of concrete damage," *Autom. Constr.*, vol. 128, p. 103785, Aug. 2021, doi: 10.1016/j.autcon.2021.103785.

[14] Lytica, "About.". [Online]. Available at: https://lytica.com/about/.

[15] N. Eriksson, C.-J. Rosenbröijer, and A. Fagerstrøm, "Smartphones as decision support in retail stores – The role of product category and gender," *Procedia Comput. Sci.*, vol. 138, pp. 508–515, Jan. 2018, doi: 10.1016/j.procs.2018.10.070.

[16] P. Pandey, H. Gajjar, and B. J. Shah, "Determining optimal workforce size and schedule at the retail store considering overstaffing and understaffing costs," *Comput. Ind. Eng.*, vol. 161, p. 107656, Nov. 2021, doi: 10.1016/j.cie.2021.107656.

[17] D. Grewal, A. L. Roggeveen, and J. Nordfält, "The Future of Retailing," *J. Retail.*, vol. 93, no. 1, pp. 1–6, Mar. 2017, doi: 10.1016/j.jretai.2016.12.008.

[18] A. Damen, "53 Data-Backed Retail Statistics Shaping Retail and Beyond," 2022. [Online]. Available at: https://www.shopify.com/retail/retail-statistics.

[19] McKinsey, "A new playbook for retail leaders," 2023. [Online]. Available at: https://www.mckinsey.com/industries/retail/our-insights/retail-reset-a-new-playbook-for-retail-leaders.

[20] M. R. Pinto, P. K. Salume, M. W. Barbosa, and P. R. de Sousa, "The path to digital maturity: A cluster analysis of the retail industry in an emerging economy," *Technol. Soc.*, vol. 72, p. 102191, Feb. 2023, doi: 10.1016/j.techsoc.2022.102191.

[21] P. Sakrabani, A. P. Teoh, and A. Amran, "Strategic impact of retail 4.0 on retailers' performance in Malaysia," *Strateg. Dir.*, vol. 35, no. 11, pp. 1–3, Nov. 2019, doi: 10.1108/SD-05-2019-0099.

[22] L. L. Har, U. K. Rashid, L. Te Chuan, S. C. Sen, and L. Y. Xia, "Revolution of Retail Industry: From Perspective of Retail 1.0 to 4.0," *Procedia Comput. Sci.*, vol. 200, pp. 1615–1625, Jan. 2022, doi: 10.1016/j.procs.2022.01.362.

[23] J. S. Raj, "A Comprehensive Survey On The Computational Intelligence Techniques And Its Applications," *J. ISMAC*, vol. 01, no. 03, pp. 147–159, Dec. 2019, doi: 10.36548/jismac.2019.3.002.

[24] F. Alsakka, I. El-Chami, H. Yu, and M. Al-Hussein, "Computer vision-based process time data acquisition for offsite construction," *Autom. Constr.*, vol. 149, p. 104803, May 2023, doi: 10.1016/j.autcon.2023.104803.

[25] M. Helmy, T. T. Truong, E. Jul, and P. Ferreira, "Deep learning and computer vision techniques for microcirculation analysis: A review," *Patterns*, vol. 4, no. 1, p. 100641, Jan. 2023, doi: 10.1016/j.patter.2022.100641.

[26] K. Das and A. K. Baruah, "Object Detection on Scene Images: A Novel Approach," *Procedia Comput. Sci.*, vol. 218, pp. 153–163, Jan. 2023, doi: 10.1016/j.procs.2022.12.411.

[27] W. Mrabti, K. Baibai, B. Bellach, R. O. Haj Thami, and H. Tairi, "Human motion tracking: A comparative study," *Procedia Comput. Sci.*, vol. 148, pp. 145–153, Jan. 2019, doi: 10.1016/j.procs.2019.01.018.

[28] P. Wu, W. Li, and M. Yan, "3D scene reconstruction based on improved ICP algorithm," *Microprocess. Microsyst.*, vol. 75, p. 103064, Jun. 2020, doi: 10.1016/j.micpro.2020.103064.

[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[30] P. Kaur, B. S. Khehra, and A. P. S. Pharwaha, "Color Image Enhancement based on Gamma Encoding and Histogram Equalization," *Mater. Today Proc.*, vol. 46, pp. 4025–4030, Jan. 2021, doi: 10.1016/j.matpr.2021.02.543.

[31] J. Xiao, X. Cui, and F. Li, "Human action recognition based on convolutional neural network and spatial pyramid representation," *J. Vis. Commun. Image Represent.*, vol. 71, p. 102722, Aug. 2020, doi: 10.1016/j.jvcir.2019.102722.

[32] J. Wang *et al.*, "Deep 3D human pose estimation: A review," *Comput. Vis. Image Underst.*, vol. 210, p. 103225, Sep. 2021, doi: 10.1016/j.cviu.2021.103225.

[33] S. Davanthapuram, X. Yu, and J. Saniie, "Visually Impaired Indoor Navigation using YOLO Based Object Recognition, Monocular Depth Estimation and Binaural Sounds," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, May 2021, vol. 2021-May, pp. 173–177, doi: 10.1109/EIT51626.2021.9491913.

[34] G. Fusco, S. A. Cheraghi, L. Neat, and J. M. Coughlan, "An Indoor Navigation App Using Computer Vision and Sign Recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, vol. 12376 LNCS, pp. 485–494, 2020, doi: 10.1007/978-3-030-58796-3_56.

[35] M. E. Yücel and C. Ünsalan, "Shelf control in retail stores via ultra-low and low power microcontrollers," *J. Real-Time Image Process.*, vol. 19, no. 4, pp. 751–762, Aug. 2022, doi: 10.1007/s11554-022-01222-2.

[36] M. Shoman, A. Aboah, A. Morehead, Y. Duan, A. Daud, and Y. Adu-Gyamfi, "A Region-Based Deep Learning Approach to Automated Retail Checkout," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2022, vol. 2022-June, no. 1. M. Shoman, A. Aboah, A. Morehead, Y. Duan, A. Daud, and Y. Adu-Gyamfi, IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. 2022-June, 3209 (2022), pp. 3209–3214, doi: 10.1109/CVPRW56347.2022.00362.

[37] R. Y. Lee, S. Y. Chua, Y. L. Lai, T. Y. Chai, S. Y. Wai, and S. C. Haw, "Cashierless Checkout Vision System for Smart Retail using Deep Learning," *J. Syst. Manag. Sci.*, vol. 12, no. 4, pp. 232–250, Aug. 2022, doi: 10.33168/JSMS.2022.0415.

[38] A. Abed, B. Akrout, and I. Amous, "A Novel Deep Convolutional Neural Network Architecture for Customer Counting in the Retail Environment," in *Communications in Computer and Information Science*, vol. 1589 CCIS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 327–340, doi: 10.1007/978-3-031-08277-1_27.

[39] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of Deep Learning for Object Detection," *Procedia Comput. Sci.*, vol. 132, pp. 1706–1717, Jan. 2018, doi: 10.1016/j.procs.2018.05.144.

[40] V. K. Sharma and R. N. Mir, "Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1687–1699, May 2022, doi: 10.1016/j.jksuci.2019.09.012.

[41] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, Jul. 2021, doi: 10.1016/j.array.2021.100057.

[42] Z. Zhou, L. Li, A. Fürsterling, H. J. Durocher, J. Mouridsen, and X. Zhang, "Learning-based object detection and localization for a mobile robot manipulator in SME production," *Robot. Comput. Integr. Manuf.*, vol. 73, p. 102229, Feb. 2022, doi: 10.1016/j.rcim.2021.102229.

[43] K. Ohri and M. Kumar, "Review on self-supervised image recognition using deep neural networks," *Knowledge-Based Syst.*, vol. 224, p. 107090, Jul. 2021, doi: 10.1016/j.knosys.2021.107090.

[44] H. Lu, C. Li, W. Chen, and Z. Jiang, "A single shot multibox detector based on welding operation method for biometrics recognition in smart cities," *Pattern Recognit. Lett.*, vol. 140, pp. 295–302, Dec. 2020, doi: 10.1016/j.patrec.2020.10.016.

[45] G. Ma, M. Wu, Z. Wu, and W. Yang, "Single-shot multibox detector- and building information modeling-based quality inspection model for construction projects," *J. Build. Eng.*, vol. 38, p. 102216, Jun. 2021, doi: 10.1016/j.jobe.2021.102216.

[46] GitHub, "dog-qiuqiu/Yolo-FastestV2," 2022. [Online]. Available at: https://github.com/dog-qiuqiu/Yolo-FastestV2.

[47] H. Zhang *et al.*, “An Improved Lightweight Yolo-Fastest V2 for Engineering Vehicle Recognition Fusing Location Enhancement and Adaptive Label Assignment,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 2450–2461, 2023, doi: 10.1109/JSTARS.2023.3249216.

[48] GitHub, “v7.0 - YOLOv5 SOTA Realtime Instance Segmentation,” *Glenn Jocher*, 2022. [Online]. Available at: https://github.com/ultralytics/yolov5/releases.

[49] M. Wang *et al.*, “FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection,” *J. Vis. Commun. Image Represent.*, vol. 90, p. 103752, Feb. 2023, doi: 10.1016/j.jvcir.2023.103752.

[50] T. Ming and Y. Ju, “SAR ship detection based on YOLOv5,” in *Third International Conference on Computer Vision and Data Mining (ICCVDM 2022)*, Feb. 2023, vol. 12511, p. 82, doi: 10.1117/12.2660100.

[51] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” *arXiv*, vol. 5, p. 12, Jul. 2021, Accessed: Mar. 03, 2024. [Online]. Available at: https://arxiv.org/abs/2107.08430.

[52] K. A. Winanta, T. Kirana, R. D. Hefni Al-Fahsi, A. Patar Jiwandono Pardosi, O. F. Suryani, and I. Ardiyanto, “Moving Objects Counting Dashboard Web Application Design,” in *2019 International Electronics Symposium (IES)*, Sep. 2019, pp. 45–48, doi: 10.1109/ELECSYM.2019.8901580.

[53] A. Farouk Khalifa, E. Badr, and H. N. Elmahdy, “A survey on human detection surveillance systems for Raspberry Pi,” *Image Vis. Comput.*, vol. 85, pp. 1–13, May 2019, doi: 10.1016/j.imavis.2019.02.010.

[54] S.-H. Chiang, T. Wang, and Y.-F. Chen, “Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches,” *Image Vis. Comput.*, vol. 105, p. 104069, Jan. 2021, doi: 10.1016/j.imavis.2020.104069.

[55] Y. Song *et al.*, “Online Cost Efficient Customer Recognition System for Retail Analytics,” in *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Apr. 2017, pp. 9–16, doi: 10.1109/WACVW.2017.9.

[56] D. A. Mora Hernandez, O. Nalbach, and D. Werth, “How Computer Vision Provides Physical Retail with a Better View on Customers,” in *2019 IEEE 21st Conference on Business Informatics (CBI)*, Jul. 2019, vol. 1, pp. 462–471, doi: 10.1109/CBI.2019.00060.

[57] I. Haritaoglu and M. Flickner, “Detection and tracking of shopping groups in stores,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I-431-I–438, doi: 10.1109/CVPR.2001.990507.

[58] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, “Deep Learning for Retail Product Recognition: Challenges and Techniques,” *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–23, Nov. 2020, doi: 10.1155/2020/8875910.

[59] M. Paolanti, L. Romeo, M. Martini, A. Mancini, E. Frontoni, and P. Zingaretti, “Robotic retail surveying by deep learning visual and textual data,” *Rob. Auton. Syst.*, vol. 118, pp. 179–188, Aug. 2019, doi: 10.1016/j.robot.2019.01.021.

[60] B. Santra, A. K. Shaw, and D. P. Mukherjee, “Part-based annotation-free fine-grained classification of images of retail products,” *Pattern Recognit.*, vol. 121, p. 108257, Jan. 2022, doi: 10.1016/j.patcog.2021.108257.