A Deep Learning Framework for Grocery Product Detection and Recognition

Prabu Selvam (prabu@cse.sastra.ac.in)
SASTRA Deemed to be University
Joseph Abraham Sundar Koilraj
SASTRA Deemed to be University

Research Article

Keywords: Deep learning, Object detection, Text detection, Object recognition, Text recognition, Retail product

Posted Date: May 24th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1431986/v1

Abstract

Object detection and recognition are the most important and challenging problems in computer vision. The remarkable advancements in deep learning techniques have significantly accelerated the momentum of object detection/recognition in recent years. Meanwhile, scene text detection/recognition is also a critical task in computer vision and has gotten more attention from many researchers due to its wide range of applications. This work focuses on detecting and recognizing multiple retail products stacked on the shelves and off the shelves in the grocery stores by identifying the label texts. In this paper, we proposed a new framework is composed of three modules: (a) Retail product detection, (b) Product-text detection (c) Product-text recognition. In the first module, on-the-shelf and off-shelf retail products are detected using the YOLOv5 object detection algorithm. In the second module, we improve the performance of the state-of-the-art text detection algorithm named, “TextSnake”, by replacing the backbone network (ResNet50 + FPN) and a post-processing technique, WHBBR (Width Height based Bounding Box Reconstruction), is proposed to detect regular and irregular text. In the final module, we used a text recognition network named “SCATTER” to recognize the retail product's text information. The YOLOv5 algorithm accurately detects both on-the-shelf and off-the-shelf grocery products from the video frames and the static images. The experimental results show that the proposed text reconstruction approach WHBBR improves the performance of the state-of-the-art techniques on both regular and irregular text. The enhanced text detection and incorporated text recognition methods greatly support our proposed framework to recognize the on-the-shelf retail products by extracting product information such as product name, brand name, price, expiring date, etc. The recognized text contexts around the retail products can be used as the identifier to distinguish the product.

1. Introduction

Detection and recognition of objects in video streams are basic and challenging tasks in computer vision. Object recognition and detection have been the subject of much research in the last two decades [1, 2]. Object detection is the process of determining the existence of different individual objects in an image. The challenge of object detection and recognition has been addressed in controlled environments. Still, it remains unsolved in uncontrolled environments, particularly when items are placed in arbitrary poses in a cluttered and occluded environment [3]. The recent growth of mobile devices with high-resolution cameras has enabled applications to support daily tasks in various contexts. In this work, we focus on detecting and recognizing grocery products on shelves around the user in a grocery store. Product recognition is more similar to a complex instance recognition problem than a classification problem. It includes many identical objects yet varies in minor aspects, for example, different flavours of the same brand of lays chips [4]. Common challenges of automatic grocery product recognition are shown in Fig. 1. Automatic product detection and recognition in a video frame have many applications, ranging from recognizing specific products to providing review and price information to assisting navigation inside the grocery store.

Furthermore, automatic grocery product detection and recognition can assist the visually impaired during shopping [5]. Because product appearance varies significantly due to the substantial changes in pose, perspective, size variations, occlusion, and lighting conditions, product detection/recognition in grocery shops are complicated. Additional peculiar issues are the product's packaging can change over time, and different products look remarkably identical. Only small packaging information allows them to differentiate, such as slight differences in the text describing the product or the background of the package's colour. Detecting/recognizing specific products is complex, unlike classifying products in macro-categories such as shampoo, chips, detergent, and so on. Another notable issue in this scenario is the availability of new products. The number of new products is increasing every day, and whenever a new product is introduced, the product recognition system also needs to be scalable with no or minimal retraining.

Acquisition and manual annotation of the training images is a time-consuming task. It is not feasible because the products frequently change over time; collecting and annotating new in-store images and retraining the system is not viable. The system must endure cross-domain scenarios where testing images are obtained from different stores and with varying imaging conditions. Since the training and testing images are from varying imaging conditions, it's vital to establish an ideal system that only needs to be trained once and used in various stores and scenarios. López et al. [6] developed an automatic product recognition system using RFID (Radio Frequency Identification), sensors, or barcodes. The majority of sensor-based systems require manufacturing fabrication, increasing the product's cost, and require massive investment. The sensor-based methods cannot resolve the planogram compliance problems. Comparing to sensor-based approaches the computer vision posses cost-effectiveness and efficiency in terms of real time implementation.

This work proposes a novel framework to detect and recognize multiple on-the-shelf and off-shelf grocery products from shelf images and video frames. We divided our proposed framework into three steps, as shown in Fig. 2: 1. For *grocery product detection*, we incorporated the YOLO (You Look Only Once) algorithm [7] to perform the grocery product detection task; we trained and tested the YOLO algorithm using benchmark grocery products datasets to detect multiple objects from the shelves images and video frames. 2. *Object text detection*, in this step, the output image from the product detection step is given as input to perform product text detection to obtain the corresponding product information such as product name, brand name, price, expiry date and so on. 3. *Object text recognition*, detected texts are recognized using a scene text recognition algorithm named SCATTER [30]. The recognized texts contain complete information about the corresponding product.

In summary, the significant contributions of our work are as follows:

- Incorporated the YOLOv5 algorithm to perform grocery product detection tasks, and we increased the object class size from 80 to 120. YOLOv5 algorithm works well for general object detection, whereas we trained the YOLOv5 algorithm for grocery product detection tasks, and the detection results are compared with existing methods.

- We improved the performance of the state-of-the-art text detection algorithm, TextSnake, by altering the backbone network from VGG16 to ResNet50. We proposed an algorithm to select the centering point instead of picking a random point. We proposed an accurate post-processing step for text reconstruction by combining Graham Scan algorithm and the rotating calipers technique. The modified backbone network, striding algorithm, and post-processing technique greatly enhanced the performance of the state-of-the-art algorithm. The robustness of the text detection method is evaluated using standard benchmark text detection datasets.
- We converted the videos of the complex Grozi-120 public dataset into frames, and then we performed a grocery product detection task. For the complex Grozi-120 public dataset alone, we used both videos and static images, whereas the remaining datasets contain only on shelf product and individual product images. So, we performed a recognition task with only static images.

The organization of the paper is as follows: Section 2 describes the literature review on object detection, text detection and retail product detection and recognition. Section 3 explains the proposed framework and WHBBR technique with the help of schematic diagrams. Section 4 presents a description of datasets and implementation details. Section 5 presents experimental results and a brief discussion of the research outcomes. Finally, Section 6 draws a conclusion and future work.

2. Related Work

This section describes different work carried out by various authors on general object detection algorithms, text detection, and recognition algorithms. Then a short literature review of grocery product detection and recognition methods.

2.1. General Object Detection

Object detection has been a trending area approached by the researchers in recent years. The primary aim of object detection is to identify and locate the instances of semantic objects of a specific class such as (building, human, dog, bicycle, or cats) in an image or video and, if present, spatial location information and bounding box drawn around the extent of each object instances. Generally, object detection methods fall into one of two categories: neural network-based or non-neural network methods. In the non-neural approach, object detectors can extract the features of the objects from either grayscale or colour images matched to detect the object, such as (Viola-Jones detectors, HOG (Histogram of Oriented Gradients), PCA(Principle Component Analysis), and Haar-like wavelet transform). The HOG method partitions the video frame or static image into several blocks and then looks for the object based on extracted features. The PCA approach extracts the object features using eigenvectors. Haar-like wavelet transform is the first real-time face detector that detects the objects in an image by deriving edge features, line features, and center-surrounded features from an image. SIFT (Scale Invariant Feature Transforms) was used for object detection early stage of research works. It provides some unique properties such as invariant to rotation, scale, viewpoint, and illumination. SIFT is used for human detection; it performs a significant

computation process to obtain features from the images such as scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor.

Over the past two decades, the emergence of deep learning has accelerated the development of a rich set of object detection methods. Object detection approaches based on deep learning have yielded significant advancements and outstanding results. Object detection methods are classified into two types: one-stage and two-stage methods. One stage method performs the detection in one step. The following are examples of typical one-stage algorithms: YOLO [7], SSD (Single Shot MultiBox Detector), DetectNet, and SqueezeDet. One-stage methods only localize the object without computing region proposals directly by performing bounding box regression and classification tasks. Two-stage method follow two steps for object detection. The original image is used to generate region proposals in the first step. The region proposals are classified, and their locations are fine-tuned in the second step, which involves classification and regression tasks. R-CNN (Regional Convolutional Neural Network) series are popularly known for detecting region proposals. R-CNN performs an external selective search over the image to generate region proposals and feeds the computed region proposals into CNN (Convolutional Neural Network) to perform classification and bounding box regression tasks. The pace of training and detection quite sluggish with R-CNN since it involves forward computation of different object regions that may overlap. Instead of extracting region proposals from each image multiple times, Fast R-CNN uses a feature extractor to extract all the features of the entire image to perform object detection. The processing time decreases since Fast R-CNN extracts all the features at an instance.

Faster R-CNN is based on the same architecture as Fast R-CNN. RPN (Region Proposal Network) substitutes the selective search approach in Faster R-CNN, which overcomes the issue of significant time overhead in producing ROI (Region Of Interest), SSD provides a considerable performance over Faster R-CNN in detecting the more prominent objects. The network creates a variety of feature maps of various sizes. On multi-scale feature maps, classification and bounding box regression tasks are performed concurrently. YOLO is a prevalent object detection technique based on the one-stage method. YOLO detects multiple objects simultaneously by predicting class probability values and bounding boxes. For object detection, YOLO does not employ multi-scale feature maps. Compared to SSD, generalization capabilities are inferior in YOLO for large-scale changes in an object. YOLO has the problem of poor recognition accuracy and a high missed detection rate. YOLOv2 uses an anchor mechanism to predict bounding boxes, so the feature map's spatial information is substantially maintained. YOLO, which employs a fully connected layer to predict bounding boxes, YOLOv2 uses convolutional layers. When a fully connected layer is used to predict bounding boxes, the feature map may lose; the YOLOv3 algorithm adapts multi-scale feature maps and uses FPN (Feature Pyramid Networks) to predict bounding boxes. FPN technique helps to merge the middle layers' output with the latter layer's, and the smaller objects present in the low-level feature can be spotted by passing high-level features to the bottom layers.

The detection speed and accuracy of YOLOv3 have been considerably improved than the earlier versions. YOLOv4 algorithm adapts the architecture of YOLOv3 with modifications in the backbone and neck. The major difference in the YOLOv4 is only the backbone. YOLOv4 uses CSPDarknet53, whereas YOLOv3 uses

Darknet53 as their respective backbone network. YOLOv4 backbone architecture comprises mainly three parts: CSPDarknet53, Bag of special, and Bag of freebies. A bag of special methods is used to increase inference cost, but object detection accuracy were greatly improved. YOLOv5 is a one-stage object detector composed of three crucial components: Backbone, Feature Pyramid, And Final Detection CSPNet (Cross-Stage Partial Network) is used as the backbone to extract rich features from an input image. CSPNet greatly improves processing time. PANet (Path Aggregation Network) is used as a feature pyramid, and the final detection part generates an output vector with bounding boxes, class probabilities objectness score and applies anchor boxes on features.

Google Translate's NMT (Neural Machine Translation) performs language translation in the form of text inscriptions which doesnot intended to identify or classify the object class. For example, simple process of translation of the text from a language English to French. Whereas YOLOv5 performs object detection tasks with the intension to identify and stratify the object class. Also, it finds the exact location of an object and draws a bounding box around the object. In our proposed framework, we restrict the text detection model to perform detection only within the bounding box coordinates. Whereas NMT performs text detection and translation for the entire image. The WHBBR technique is proposed to enhance the accuracy of state-of-the-art text detection methods and this technique works well with irregular text detection models.

Many researchers have widely explored the major problems of object detection in videos and scene images. In this research work, many solutions have been suggested: [1–3]. Most of the video object detection algorithms includes two networks. Firstly, CNN were used in the backbone network whereas the last layer was taken up for the feature extraction by replacing the Fully connected layer. Secondly, the detection network classifies the objects and predicts the bound boxes. The methods for detecting video objects using deep learning can be classified into LSTM (Long Short Term Memory)-based, tracking-based, flow-based, attention-based, and other methods. The flow-based method uses optical flow. A deep feature flow framework propagates deep feature maps from sparse keyframes to different frames via an optical flow. A 3D CNN model for video sequence object detection developed a network to learn the spatiotemporal properties of a video series using several input video frames and combine video data and optical flow for video classification. It improved detection speed and accuracy. Missing Recovery Recurrent Neural Network (MR-RNN) is an object detection algorithm to capture the missing objects and many objects missed by basic object detectors earlier by capturing the temporal information from the video frames. To retrieve complete details about the moving objects. Geometric properties of moving objects are described by taking the pixels of moving objects and object-background pixels pairs and finally, establish**es** the relationship between the moving object's geometric properties and the model parameters. A dual-stream detection mechanism was utilized to boost tiny object detection by combining appearance flow with motion flow. Wang et al. [8] proposed MANet to find the optical flow information between adjacent frames, to deliberate the optical flow information in which it extract the global image features together. This method was efficiently used to extract features by acquiring the instance-level calibration across frames with optical flows technique, and then the pixel-level feature calibration was to improve the performance of video object detection.

Flow-oriented temporal coherence module and ABTC (Attention-Based Temporal Context Module) methods are used to extract and integrate high-level features from keyframes, so these integrated features were given as an input to the detection network for object detection which results higher accurate frame alignment. D&T proposed a ConvNet architecture [9] to improve object detection and object tracking performance by introducing a multi-task objective frame-based object tracking by adopted techniques as follows, frame track, regression, correlation features, and frame-level detection based on tracklets. Temporal contextual information was extracted using the STMN (Spatio-temporal memory module). Seq-NMS [10] proposed a heuristic method composed of sequence selection, re-scoring, and suppression for re-ranking bounding boxes [11] in a video sequence. TSSD (Temporal Single-Shot Detector) method integrates ConvLSTM-based attention used for background and scale suppression and SSD.

2.2. Text Detection

A novel method proposed by Shivakumara et al. [12] to detect text from video frames based on neighbour component grouping and GVF (Gradient Vector Flow) use dominant edge pixels to extract TC (Text Candidates). They presented two grouping schemes: the first scheme finds nearest neighbours to produce CTC (Candidate Text Components). The second scheme extracts neighbouring and restores missing CTC to detect arbitrary text in video frames [44–47]. Hybrid text detection and text tracking work proposed based on MSER (Maximally Stable Extremal Region). Delaunay Triangulation is used to identify the text candidates and multi-scale integration to solve multi-font and multi-sized texts by occurring spatial and temporal information. It also utilized convolving Laplacian with wavelet sub-bands to enhance low-resolution text pixels and combined MSERs and SWT (Stroke Width Transform) to obtain text candidate regions so which they improved the performance of arbitrary shaped text in video frames.

To detect text from complex video frames, Ye et al. [13] proposed a texture-based method, LBP (Local Binary Pattern), to extract features of text candidates. PNN (Polynomial Neural Network) was developed to classify text and non-text regions. A three-stage text detection method was proposed First it extracts the features from a video frame then, text candidates are detected by optimizing RBFNN (Radial Basis Function Neural Network) model. Finally, a post-processing was done on the false detected text candidates [50]. To classify textual and non-textual components, He et al. [14] presented a framework for text detection called Text-CNN (Text-Attentional Convolutional Neural Network). In this reseachers introduced a new training mechanism to increase the robustness against a complex background. CE-MSERs (Contrast Enhancement Maximally Stable Extremal Regions) detector was developed to improve the video frame's intensity. This method can detect complex text patterns with a high accuracy rate. Predicting arbitrary orientated and quadrilateral shaped text line or word incorporates proper loss function. The false detection rate reduced drastically, NMS (Non-maximum Suppression) [51, 53, 57] produces final bounding boxed text regions [15]. To detect text region from low-quality images with complex background from video frames the script identification was performed by extracting low and high-level featues using CNN-LSTM framework and Attention-based patch and their respective weights were calculated.

The video frames were converted into patches and fed into CNN-LSTM. Local features are extracted by performing patch-wise product patch weights, and global features are extracted from the final LSTM cell. Weights of local and global features fused dynamically to perform script identification. Coarse candidate regions detection and fine text line detection are effective in detecting multi-scale candidate text areas. Candidate text regions are segmented and fed into CNN, which generates a confidence map for each frame's text regions. Finally, projection analysis refines text candidates and divides them into text lines. The performance of the video text detection technique was enhanced using a novel refined block structure constructed efficiently using a fully convolutional network. High-resolution semantic feature maps are generated by extracting multi-resolution features from video frames to capture highly varied video text appearances and feature fusion, an efficient correlation filter [54] used to improve the overall detection performance.

An eMSER (Edge-enhanced Maximally Stable Extremal Regions) was proposed to reduce the text detection duration in video frames, also retain the character's shapes and converge the detection. A hierarchical convolutional neural network is used to extract descriptive features. A multi-scale deformable convolution structure extracts additional features and spatiotemporal information from the video frames [16]. It incorporates a bipartite graph model and the random walk algorithm. Firstly, text candidates and background regions are extracted from the video frame. Shape, motion, and spatial relation between text and background are exploited to refine text candidates. Obtains the correlations between text and background regions that was greatly improvise accuracy in video text detection. Fusion-based detection method [17] proposed extracting text regions and locating characters; tracking trajectories are linked to refine detection results. A polygon-based curve text detector combined the R-CNN and TLOC (Transverse And Longitudinal Offset Connection) [18] for the precise detection of irregularity texts. Post-processing methods named NPS (Non-Polygon Suppress) and PNMS (Polygonal Non-Maximum Suppression) produces efficient and accurate text detection result.

2.3. Grocery Product Detection and Recognition

In 1999, the first significant effort was made to recognize retail products in isolation. Naturally, the problem of localization is not addressed. Merler [19] introduced a retail product detection problem with a dataset consists of with rack and product images in 2007; it took almost eight years to develop a more comprehensive method to detect and recognize multiple retail products. Marder [20] proposed a two successive layers multi-product detection scheme. In the first layer, they followed three different techniques to detect retail products in the rack (i) Vote map, (ii) HOG and (iii) BoW (Bag of Words) based on a sliding-window approach. A saliency map was used for product recognition and to address the second layer's planogram compliance problem. Beis et al. introduced a k-d tree representation [21] of all product images SURF (Speeded Up Robust Features) descriptors for retail product recognition. The products in the rack images are recognized using a previously constructed k-d tree and the Best-Bin-First search algorithm. In addition, a pose-class histogram in high dimensional space were used to perform fine-grained recognition.

George et al. proposed a three-phase detection and recognition method [22]. In the first phase, They developed a non-parametric probabilistic model based on SIFT features. Fine-grained product categorization is performed in the second phase. The first and second phases are coupled with the KLT (Karhunen-Loeve Transform) in the final phase it tracks the boxes detected in a video. Geng et al. developed a product detection system [23]; they created a saliency map for the shelf images to identify the locations of the products in the frames. The saliency map is constructed by SURF key points using the rack image's AIM (Attention and Information Maximization). Finally, a CNN is used to recognize the products. Ray et al. present a Conditional Random Field (CRF) [24] based method for classifying structured objects. A CNN extracts the visual features and linearly fed to a CRF model. Viterbi and forward-backward algorithms were used to generate the labels of the product sequence.

Franco et al. divided the product detection and recognition task into three steps [25]: (i) Candidate pre-selection; in this step, they segmented the foreground from the background using fixed-threshold binarization. (ii) Fine-selection, They utilized a customized DNN (Deep Neural Network) and a BoWto select the most robust features (iii) Post-processing reduces the false positives by eliminating the multiple overlapped detections of the same products [42, 43]. Karlinsky et al. and Zientara et al. calculated a homography matrix [26, 27] to identify the grocery products in shelf images by matching SURF key points of product images with corresponding rack images. Goldman et al. [28] used a Hough voting scheme based on matched SURF key points to determine the pose of products, and then they determined the location of products by estimating their pose.

Bukhari et al. [67] developed a vision-based ARC (Automatic Retail Checkout) system, which uses CNNN for object detection, Canny edge detector and hysteresis thresholding to perform non-maximal suppression and generates a binary image containing the edges, respectively. Morphological operations are performed to fill out holes and gaps. This method highly depends on a motor-powered conveyor-belt mechanism. Ciocca et al. [60] introduced a multi-task learning network to extract features from the images. Meanwhile, the Authors performed the classification in both supervised and unsupervised learning methods. Yilmazer and Birant [61] combined two concepts SOSA (Semi-Supervised Learning and On-Shelf Availability) to identify the empty shelves. Similarly, Santra et al. [62] use GCN (Graph Convolutional Network) for feature extraction, Siamese network architecture (SNA) was used to capture the similarity of the neighbouring superpixels and Finally, the features extracted from GCN and SNA are fed to SSVM for the identity gaps on the rack. Leo et al. [63] assessed the performance of different classification models. Olóndriz et al. [66] introduced FooDI-ML and Glovo a dataset and an application respectively were used to recognize the retail product information. Machado et al. [64] developed a product recognition system for visually impaired people. The authors infered that the ResNet-50-based approach achieves better results than other deep learning-based models. Domingo et al. [65] use Cross-Validation-Voting (CVV) scheme to classify the retail products.

3. Proposed Framework

The overall architecture of our proposed framework is shown in Fig. 2; it consists of three important modules for grocery product recognition. The first module is to detect grocery products based on the product class using a single shot object detection algorithm, YOLOv5. The second module uses a text detection algorithm to detect the appropriate text on grocery product packing (brand name, product name, quantity, and other information). Finally, the third module recognizes the text using the text recognition algorithm called WHBBR. The recognized text has unique information about the corresponding product named as grocery product recognition model.

3.1. Pre-Processing

In a pre-processing step, the input videos in Grozi 120 datasets are converted into video frames to perform object text detection and recognition. Here, the video frames are captured for every 0.5 seconds, i.e., two frames are extracted per second. We use cv2.VideoCapture() and vidcap.read() predefined function to capture the video frames.

3.2. Grocery Product Detection using YOLOv5

YOLOv5 incorporates CSPDarknet and CSP (Cross Stage Partial Network), which makes it easier to train the object detection model and reduces the computation cost, respectively. When compared to other YOLOv5 models seems much better at detecting smaller objects or far away objects, inference speed is good when compared to Faster-RCNN, Fast-RCNN and SSD. Unlike R-CNN and SPP-net, No overlapping boxes around the objects. YOLOv5 is more efficient than other popular object detection models.

We incorporated the YOLOv5 algorithm for our grocery product detection. Figure 3 shows the overall pipeline of the YOLOv5 object detection algorithm. YOLOv5 algorithm was chosen as our product detector for three main reasons. Firstly, YOLOv5 uses CSPDarknet as its backbone. CSPNet is incorporated into the darknet created as CSPDarknet. CSPNet successfully addresses the issue of repeating gradient information, which often occurs in large-scale backbones. The gradient changes are included in the feature map, significantly improving CNN's learning ability in cases where accuracy was reduced due to light-weight and reduce needless energy usage by spreading the entire computation across each layer in CNN. CSPDarknet reduces the model's size by compressing the feature maps during the feature pyramid generation step via cross-channel pooling, significantly minimizes memory consumption costs.

CSPNet improves the inference speed and accuracy. Detection speed, model size, and accuracy are imperative in our grocery product detection task. The inference efficiency of product detection on low-resource edge devices is determined based on model size. CSPNet can reduce the model size efficiently. Secondly, the YOLOv5 algorithm incorporates a PANet as its neck to increase the flow of information. PANet adopts bottom-up path augmentation and a new FPN to enhance the localization capability of the entire feature hierarchically. Adaptive Feature Pooling allows high-level features to access fine details and high localization of low-level feature similarly large receptive fields and capture richer context information of high-level feature access by low-level feature this pooling feature help to produce accurate prediction.

A fully connected Fusion used for mask prediction that differentiate instances and recognize the various portions of the same object.

PANet helps to identify smaller products in our grocery product detection using the shared pooling feature and ensures that products are not missed. Thirdly, to achieve multi-scale prediction, the head of the YOLOv5 algorithm adapts the YOLO layer and produces feature maps of different sizes such as 19×19, 38×38, and 76×76 helping the model that can handle and detect small, medium, and oversized objects. It also predicts anchor boxes for feature maps. Grocery products can be of different sizes, such as small, medium, and large. The multi-scale detection mechanism in the YOLO layer ensures that the model able to detect the grocery product even if the size changes during the detection process.

3.3. Text Detection

We propose a method for detecting text information in grocery items, such as product name, brand name, amount, and so on, using an efficient post-processing technique. As shown in Fig. 4, it describes the overall architecture of the text detection model. Conventional text detection algorithms generally assume the text instances in linear form. This linear form could not hold the representation and geometric properties of curve text instances. To address this problem, we use the curve-shaped text detection method. Inspired by TextSnake [29], the text instances are represented as a sequence of overlapping disks, each centered on the text center line and associated with an orientation and radius. The various transformations of text instances such as rotation, bending, and scaling are captured.

The text instance (t_i) represents an ordered list $O(t)=\{W_0, W_1,...W_i,.., W_n\}$ consisting of multiple characters. Where 'W_i and 'n' in the ordered list represent ith disk and the total number of disks, respectively, each disk (W) in the ordered list O(t) is correlated with a set of geometrical characteristics, i.e., W=(c,r,θ). In the center, the radius equals half of the text instance t's local width. The disk orientation is determined by the tangential direction of the text center line around the center (c). W is represented by the numerals c, r, and θ. The geometrical characteristics in O(t) are mainly used to amend irregular shape text instances and change them into rectangular image regions. The text area (t) can be readily reconstructed by calculating the union of the disks in O(t). The proposed FCN model predicts text regions (TR), text center line (TCL), and its geometric attributes such as radius (r), sin θ, and cos θ. Further, masked TCL is computed from TR; TCL is a component of TR. Each other instance segmentation can avoid TCL overlapping, and disjoint sets are calculated and utilized. The central axis point lists are obtained using a striding method, and a proposed post-processing technique is used for text instances reconstruction and eliminates false detection.

3.3.1. Backbone Network

According to recent studies, ResNet50 captures well-defined feature representations. It is used very frequently in many computer vision tasks. ResNet allows us to train extremely deep neural networks with more than 150 layers. ResNet has the technique called skip connections, which addresses the problem of vanishing gradient by providing an alternate path for the gradient to flow through and allows the model to

learn an identity function that ensures that the higher layer will perform at least as good as the lower layer, and not worse. So, we adopted ResNet50 with batch normalization as our backbone network to extract features from an image. The block diagram of our backbone network is illustrated in Fig. 5. Similar to U-Net, we use ResNet's skip connections in the decoding stage to aggregate low-level features. This network has divided into five stages of convolution, and the fully-connected (FC) layers replace the feature merging network, which is made up of grouping feature maps of each step. In a merging network, several stages are piled one on top of the other, and each stage has its merging unit that extracts feature maps from its previous stage. The following Eq. (1–4) interprets the merging branch.

$$e_5 = d_1 \quad (1)$$

$$d_i = Conv_{3x3}(Conv_{1x1}[e_i\text{-}1; UpSample_{x2}(d_i\text{-}1)]), \text{ for } i \geq 2 \text{ and } i \leq 4 \quad (2)$$

$$d_5 = Conv_{3x3}(Conv_{3x3}(Conv_{3x3}(Conv_{1x1}[e_i\text{-}1; UpSample_{x2}(d_i\text{-}1)]))), \text{ for } i = 5 \quad (3)$$

where e_i and d_i represent feature maps of i^{th} stage and the corresponding upsampling and merging units, respectively. After the merging, the final detection output size is the same as the size of the input image. The final output has four channels, TR/TCL, and the last three are geometric attributes of text instances such as r, sinθ, and cosθ, respectively. The network generates the TCL, TR, and geometry maps after feed-forwarding. Text Region (TR) is a binary mask, with 1 for foreground pixels (those inside the polygon annotation) and 0 for background pixels, and Text Center Line (TCL) is computed using the sequencing process.

3.3.2. TCL and TR Generation

Masked TCL is extracted by performing the intersection of TR and TCL. Disjoint-set accurately divides the TCL pixels into discrete text instances. The enhanced striding algorithm predicts the shape and course of the text instances. It consists of three essential tasks: centralizing, striding, and sliding. Firstly, we chose a pixel by centralizing; we made it the starting point. Striding and centralizing are recursively performed in both opposite directions from the starting point until it reaches the end. The searching operation produces two ordered point lists combined to construct a final central axis list. The final axis list precisely describes the text flow and the text shape.

Table 1
Algorithm to compute an initial center point in the TCL.

Procedure Centralizing
Input: x- axis Leftmost point (x_l), Leftmost point (x_2), y- axis topmost point (y_l), bottommost point (y_2)
Output: x- axis (x_{cp}),y- axis (y_{cp}), text center point (t_{cp})
1. x_{cp} = median(x_1,x_2)
2. y_{cp} = median(y_1,y_2)
3. t_{cp} = (x_{cp},y_{cp})

Centralizing: As given in Table 1, we follow three steps to calculate the center point coordinate using the Instance segmented TCL as shown in Fig. 6. (i) calculate the x-axis center point (x_{cp}) by finding the leftmost point (x_1) and rightmost point (x_2) of segmented TCL. (ii) Likewise, calculate the y-axis center point (y_{cp}) by finding the topmost point (y_1) and bottommost point (y_2) of segmented TCL. (iii) Find the center point coordinates ($x_{cp,}$ y_{cp}).

Striding: once a center point is obtained, the next step is to perform a striding operation. This technique looks for points by taking a stride in two opposite directions within the TCL area.

$$Disp1 = \left(\frac{1}{4}r \times cos\theta, \frac{1}{4}r \times sin\theta\right) \quad (4)$$

$$Disp2 = \left(-\frac{1}{4}r \times cos\theta, -\frac{1}{4}r \times sin\theta\right) \quad (5)$$

Eqn (4) and Eq. (5) are the offset value for each stride in two opposite directions. If the points move out of the text area, the stride offset value is decremented gradually until the points move inside the text area or it hits the end.

Sliding: Finally, the sliding procedure iteratively moves along the central text line, drawing circles on predicted text instances with a radius r calculated from the r map. For each point on TCL: The distance between two points on the sides is used to determine the radius (r); by drawing a straight line across the TCL points in the text area, the orientation (θ) is determined. Since the TCL is a straight line, it is simple to compute it using algebraic triangles and quadrangles. But, it isn't easy to use a generic algebraic technique for polygons with more than four sides. An illustration of the TCL extraction and TCL expansion is shown in Fig. 7 and Mask to TCL conversion is given in Table 2.

Text instances (t) represented a set of vertices (v_0,v_1,v_2,.....,v_n). We assumed that text instances had two edges, one at the top and one at the bottom and that the two edges connected to the head or tail are parallel and traverse in the opposite direction. Each edge is measured as $M(e_{i,i+1}) = \cos(e_{i-1,i,i+1,i+2})$, head and tail edge measurement M is set to -1. Then, possible text control points are sampled on text sidelines. TCL is extracted by computing midpoints of corresponding text control points. The Head and tail edges of TCL get shrunk by ¼ of the radius of control points so that most of the TCL pixels remain within TR. If we take ½ of the radius of control points, we lose the heads and tail of the text areas. At last, the TCL area is expanded by 5 pixels.

3.4. Width Height based Bounding Box Reconstruction (WHBBR) Algorithm

As shown in Fig. 8, in post-processing, the final result is in the form of polygon-shaped bounding boxes that detect starting and ending characters in a text that can be solved using the Width Height based Bounding Box Reconstruction (WHBBR) algorithm. Firstly, the polygon-shaped bounding box coordinates are obtained using Graham Scan Algorithm. Secondly, two antipodal points were selected through sidelines, and two directed tangent lines of support were drawn at antipodal points (a_i) and (a_j). Thirdly, the diameter of the polygon (A) is calculated by a pair of antipodal points of (A). Where several techniques are proposed in the past to boost the performance of finding the diameter of a polygon, and some of the methods proved to be incorrect, we proposed an efficient way to calculate the diameter of a polygon using an antipodal point as given in Table 3, that they allow parallel lines of support. These two parallel lines visit all pairs of antipodal vertices by rotating clockwise. At each iteration, the angle of θ_i and θ_j are compared to compute A's diameter is determined in constant time.

The width is calculated as given in Table 3. A support line is constructed through an edge; for example, a_{i-1} is the vertex furthest from this edge. Continue this process until all the edges are visited at least once. In each step, the width is computed, and the most negligible width value can be considered. Finally, the width and height of the arbitrary shaped text are enclosed with an accurate bounding box. In post-processing, Missed characters and overlapped characters are efficiently identified. The arbitrarily shaped bounding boxes are converted into rectangular bounding boxes.

Procedure mask_to_tcl	Algorithm 1: Width Height based Bounding Box Reconstruction (WHBBR)
Input: pred_sin, pred_cos, pred_radii, tcl_contor, direction, initial_x, initial_y **Output**: text center line	**Input**: Set of points A={ a_0,a_1,a_2,..., a_n} polygon bounding box vertices.

initialize H,W, flag = 1, x_shift = initial_x, y_shift = initial_y
 	
initialize result = [], max = 200, iteration = 0
 	
while in_contour(tcl_contour,(x_shift, y_shift)) **do**
 	
iteration = iteration + 1
 	
sin_orient = pred_sin[y_shift, x_shift]
 	
cos_orient = pred_cos[y_shift, x_shift]
 	
x_center, y_center = centralizing(W,x_shift,H,y_shift)
 	
sin_center = pred_sin[y_center, x_center]
 	
cos_ center = pred_cos[y_center, x_center]
 	
radii_ center = pred_radii[y_center, x_center]
 	
#Append the x_center, y_center and radii_center into list
 	
result[]
 	
result.append(sin_center, cos_ center, radii_ center)
 	
while !contour_end **do**
 	
stride = (1/4) * radii_center
 	
x_shift_front = x_center + cos_ center * stride * flag
 	
y_shift_front = y_center + sin_ center * stride * flag
 	
x_shift_back = x_center - cos_ center * stride * flag
 	
y_shift_back = y_center - sin_ center * stride * flag
 	
if size_of_result = 1 **then**
 	
final_x = x_shift_front
	Initialize count = 0, Detected = 0.
	
	Compute x_{min} , x_{max} $y_{min,}$ and y_{max}
	
	Draw two vertical parallel lines of support on A through y_{min} and y_{max}
	
	While visited edges[] !=NULL **do**
	
	if one || both parallel lines tangents with an edge, **then**
	
	Detected = an antipodal-edge or edge-edge pair
	
	Add edge pair in the visited edges list
	
	Max_distance = length of width edge pair
	
	Min_distance = length of height pair
	
	flag = flag + 1
	
	end if
	
	if Detected edge pair ⊥ to the x-axis, **then**
	
	W_Deteced = Detected
	
	else
	
	H_Deteced = Detected
	
	end if
	
	Rotate the parallel lines until one is connected to the next
	
	polygon edge is detected.
	
	if new antipodal pair is detected, **then**
	
	Add edge pair in the visited edges list
	
	Temp_distance = new edge length
	
	flag = flag + 1
	
	if Temp_distance > Max_distance **then**
	
	Max_distance = Temp_distance

Procedure mask_to_tcl	Algorithm 1: Width Height based Bounding Box Reconstruction (WHBBR)

final_y = y_shift_front
else
compute the distance_front and distance_back
if distance_front > distance_back **then**
final_x = x_shift_front
final_y = y_shift_front
else
final_x = x_shift_front
final_y = y_shift_front
endif
end while
if final_x ≥ W || final_x < 0 || final_y ≥ H || y_shift < 0 **then**
break
endif
end while
return result #contains the coordinates of text center line
	W_Detected = an antipodal-edge or edge-edge pair
	else
	Min_distance = Temp_distance
	H_Detected = an antipodal-edge or edge-edge pair
	end if
	end if
	if visited edges[] contains all the edges & parallel lines reach
	their original position, **then**
	break
	end if
	end while
	return W_Detected, H_Detected, Max_distance, Min_distance

Table. 2 Algorithm to convert the mask to TCL.	**Table. 3** Algorithm to compute width and Height for Bounding Box reconstruction

3.5. Text recognition

We adapt the context attentional network [30] as our text recognizer. Cropped text images are fed into a text recognition model. A four-step mechanism was incorporated to process each image; firstly, the cropped text image has been transformed into the normalized image using Thin-Plate Spline (TPS) transformation to reduce the burden for the subsequent feature extraction stage. Secondly, a 29-layer ResNet is used as the convolutional neural network's backbone to extract essential features from the input image. The final feature map is 512 channels. Thirdly, CTC-based decoding is used to embed characters with each column and the output of the embedded sequence to the CTC decoder to generate output. Finally, the selective contextual refinement block is employed to mitigate the lack of contextual information. To overcome the problem of long-term dependency, a two-layer Bi-LSTM is used over the feature map. The output from the Bi-LSTM network is combined with a visual feature map to generate a

new feature map. In the selective decoder, a two-step attention mechanism is employed; in the first step, 1D attention operates on the output feature map generated from Bi-LSTM; further, an attention map is generated as a fully connected layer from these features. Next, the element-wise product is calculated between yielded attentional features and the attention map. In the second step, a separate encoder-decoder decodes the attention map, and LSTM generates the text characters.

4. Experiment

This paper uses a synthetic dataset (SynthText) proposed by Gupta et al. [31] to train our model. We evaluate our model on seven standard benchmarks that contain four ‘regular’ datasets (IC03, IC13) and three ‘irregular’ datasets (IC15, Total-Text, SCUT-CTW1500).

Regular Text Datasets

the performance of our proposed framework have been evaluated using standard benchmark datasets such as ICDAR 2011 [32] and ICDAR 2013 [33]. The majority of the text images in these datasets are almost horizontal text images.

- **ICDAR2011 (IC11)** [32] ICDAR 2011 dataset is inherited from previous ICDAR contests benchmarks. Some of the prior dataset’s flaws, such as inconsistent definitions and inaccurate bounding boxes, have been resolved. This dataset contains 484 images, 299 images for training, and 255 images for testing.
- **ICDAR2013 (IC13)** [33] contains 462 images. Most of the text images are inherited from IC03, 229 images for training, and 233 for testing. There are 849 text instances in the training set, whereas the testing set contains 1095 text instances.

Irregular text Datasets

ICDAR 2015 Incidental Text [34], Total-Text [35] and SCUT-CTW1500 [36] are the benchmark datasets used to evaluate the performance of our framework. In this dataset, most text images are curved, rotated, and low-quality text images.

- **ICDAR 2015 (IC15)** [34] this dataset is from ICDAR 2015 Robust Reading Competition. Images in this dataset are captured using Google Glasses without proper positioning and focusing. It includes more than 200 irregular text images. This dataset contains 1500 images, 1000 images for training, and 500 images for testing. It provides word-level annotations. Notably, it contains 17548 text instances.
- **Total-Text** [35] contains 1555 images, 1255 images for training, and 300 for testing. Images in this dataset are collected from various locations, including business-related locations, tourist sites, club logos, formal information, and so on. At the word level, this dataset contains 11,459 cropped word images with more than three different text orientations: horizontal, multi-oriented, and curved text. Total-text provides polygon-shaped ground truths.

- **SCUT-CTW1500** [36] contains 1500 images, 1000 images for training, and 500 images for testing. Images in this dataset are collected from various sources such as Google's open-Image, Internet, and mobile phone cameras. Notably, it contains 10751 cropped word images for testing. At least one curved text appears in each image. There are primarily arbitrary shape texts in text-line instances, but horizontal or multi-oriented text lines also exist in the text images.

Grocery Datasets

we use four publicly available datasets such as GroZi-120 [37], WebMarket [38], Grocery Products [39], and Freiburg Groceries Dataset [40] to train and test our proposed grocery product detection and recognition framework.

- **GroZi-120** [37] is the first publicly released grocery product benchmark dataset. The product images are acquired from grocery web stores such as Froogle, and the text in the product images differs in size, style, and complex background images. There are 120 product categories and 676 product images in the GroZi-120 dataset.
- **WebMarket** [38] consists of 3,153 shelf images of size 2272x1704, which is collected from 18 different product shelves. There are 100 product categories where the products are captured on and off the shelf. Rack images are captured in various scale, pose, and illumination so, it differs from product images. Like GroZi-120, the ground truth of the product is manually identified and annotated for each product located in the rack images.
- **Grocery Products** [39] The Grocery Products dataset is designed to assist with fine-grained object classification and localization. The product images were obtained from the Internet, and the template images were recorded in studio-like conditions. The rack photos were taken using a mobile phone in a real-world retail setting. Various viewing angles, lighting conditions, and magnification settings capture rack images. A rack image can also include anywhere from 6 to 30 products. The ground truth is produced by manually annotating product categories and locations in rack images. There are 80 broad product categories in the dataset. Only 27 of the 80 product categories contain ground truth, including 3235 fine-grained product templates.
- **Freiburg Groceries Dataset** [40] collects pictures of real products and shelves. The Freiburg Groceries Dataset comprises 4947 pictures divided into 25 grocery classes with 97 to 370 images each. The products are captured using four different cameras at Freiburg, including residences, grocery shops, and offices. The text characters present in the product images have various illumination levels and complicated backgrounds in this dataset.

4.1. Implementation

The implementation of our proposed framework is done using PyTorch. All the three experiments such as retail product, text detection and text recognition were carried out on a DELL Precision Tower 7810 workstation, which has Intel(R) Xeon(R) CPU E5-2620 v3 dual processor, 96GB RAM and NVIDIA Quadro K2200 graphics card. We use the YOLOv5 object detector to perform grocery product detection. We

trained and tested our proposed detection network using benchmark grocery datasets. All these datasets do not have an annotation format; we manually annotated them using Labellmg and then placed all the annotated images and text files (as shown in Fig. 9) in the same directory.

The unified annotated format is given below:

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

Where, <object-class> - number of object, represent by an integer number (0) to (total number of class – 1), <x > and < y > represents the center of bounding box rectangle.

<width> = <absolute_x>/<image_width>

<height> = <absolute_height>/<image_height>

Table 4
Parameters used to tune the proposed framework.

Parameters/ Models	Object Detection	Text Detection
Train-test split ratio	Training : Validation : Testing 80 : 10 : 10	Training : Validation : Testing 80 : 10 : 10
Learning rate	$1x10^{-2}$	$1x10^{-3}$
Optimization algorithm	Adam	Adam
Activation function	Hidden Layer: Leaky ReLU Final detection layer: sigmoid	ReLU
Batch size	64	64
No. of epochs	20	20
Loss function	Focal loss	Text classification loss and Bounding Box regression loss.

Table. 4 shows the parameters used to train the proposed famework. For both object and text detection tasks, we divided 80% data for training, 10% data as validation and the remaining 10% for testing (80:10:10). The learning rate is set to be $1x10^{-2}$. The object detector is optimized with the Adam algorithm, with a batch size of 64. Adam combines the best properties of the AdaGrad and RMSProp algorithms. It has a faster computation time and requires fewer parameters for tuning. Adam is relatively easy to configure whereas the default configuration parameters do well on complex problems. Adam is more stable than the other optimizers, it doesn't suffer any major decreases in accuracy. The Adam optimizer is the best among the other optimization algorithms. Hence we employed the Adam optimizer for both object detection and text detection tasks. We use the exact system specification to implement

text detection and recognition algorithms. We trained the text detection model using SynthText under full supervision used as our baseline model. We use the Adam optimizer as an optimization algorithm for the text detection algorithm, the learning rate is set to be $1x10^{-3}$ and the text detection model is trained with a batch size of 64. We chose 299 training images from ICDAR 2011, 229 training images from ICDAR 2013, 1000 training images from ICDAR 2015, 1255 training images from Total-Text, 1,000 training images from SCUT-CTW1500 and 4000 images from SynthText for text detection model training. For a fair comparison, single-scale testing is performed, and a polygonal NMS eliminates redundant detections.

4.2. Performance Metrics

The performance of the framework can be evaluated based on the confusion matrix. The performance metrics are accuracy, precision, specificity, recall or sensitivity, and F1-score. For classification models, accuracy is a critical measure. It's straightforward to comprehend and use for binary and multi-class classification problems. The percentage of true results in the total number of records examined is accurate. Accuracy is useful for evaluating a classification model built only from balanced datasets. If the dataset for classification is skewed or unbalanced, accuracy may provide incorrect results. The percentage of objects or text detected correctly over the total number of detected texts or objects is precision. Another essential metric is recall, which provides more information if all possible positives must be captured. The percentage of objects or text detected correctly over the total ground truth is known as recall. If all positive samples are predicted to be positive, the recall is one. If the best combination of accuracy and recall is needed, these two metrics may be merged to get the F1- score. The F1-score is the harmonic mean of accuracy and recall, ranging from 0 to 1. Eq. (6) to Eq. (9) provides the formulae for evaluating all of these metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

6

$$Precision = \frac{TP}{TP + FP}$$

7

$$Recall = \frac{TP}{TP + FN}$$

8

$$F1 - score = 2X\frac{Precision * Recall}{Precision + Recall}$$

9

In practice, a model should be built with precision and recall of 1, resulting in an F1-score of 1, i.e., 100% accuracy, which is difficult to achieve in a classification problem. As a result, the built classification model should have a better accuracy and recall value.

5. Results And Discussion

We propose a framework to perform three main tasks: object detection, object text detection and object text recognition. The object detector used is the YOLOv5 algorithm. The object text detection algorithm detects the text present on the grocery product to detect grocery products. Finally, the object text recognition algorithm can recognize the detected text. Once the YOLOv5 object detector detects the grocery products, the obtained result image can be input into the text detection model. The text detection algorithm is specially designed to capture regular and irregular text. Also, it can detect the text with a complex background, affine distorted texts, text with non-uniform spacing, and different text in a single image. The text present in the grocery dataset contains complex shapes, sizes, and orientations. However, the text detection model can detect the complex curved shape texts and multiple texts in an image (Fig. 13). The text recognition algorithm uses the CTC-Attention mechanism to recognize the arbitrary shaped text in the cropped word image. The CTC-Attention-based text recognition model can recognize the curved shaped text, text with non-uniform spacing, and multiple images in an image (Fig. 14). The proposed text detection model and an accurate text recognition model help detect and recognize the grocery products on and off the shelf.

5.1. Grocery Product Detection Result

The performance of grocery product detection is given in Table 5 and Table 6. We used YOLOv5 to train and test the four different benchmark datasets. Pre-trained models greatly support the extraction of features. In the GroZi-120 dataset, the images are minimal; four images per class are not limited to object detection tasks. We performed a data augmentation task to increase the dataset images by scaling, rotating, adding noise, skewing, etc. In the GroZi-120 video dataset, the videos are converted into frames and fed to the YOLOv5 model. The GroZi-120 dataset contains 120 classes; we performed training and testing for all the 120 classes by modifying the fully connected layer. YOLOv5 dramatically improves the detection performance by more than 10% (see Table 5), and it can detect both small and large grocery products (Fig. 10). YOLOv5 provides the most promising for other datasets such as WebMarket, Grocery Products and Freiburg Groceries Dataset, as shown in Tables 5 and 6 (see Fig. 10).

Geng et al. [23] use the GroZi-120 dataset to assess the performance of BRISK and SIFT techniques. The authors used VGG16 and Attention map for feature extraction and classification, respectively. SIFT algorithm is not efficient for many computer vision tasks. Hence used a deep learning-based object detection algorithm, YOLOv5, which completely outperforms BRISK and SIFT techniques, with precision (86.3% vs 46.3% and 49.05%), Recall (77.8% vs 29.50 and 29.37%) and F-Measure (77.04% vs 36.04% and 36.74%). On the GroZi-120 dataset, the YOLOv5 model outperforms other existing models with a

greater margin (+ 30). Franco et al. [25] and Marder et al. [20] use DNN and HOG approaches, respectively, to detect the products of the WebMarket dataset, and achieved F-Measure (46% vs 28.33%).

Santra et al. [40] achieve the F-Measure of 80.21% is the second-highest in the Grocery Products dataset. Ray et al. [24] and Karlinsky et al. [26] were able to achieve a satisfying result with F-Measure (76.20% and 79.05%). Franco et al. [25] and Marder et al. [20] use the BoW approach for product recognition, achieving the F-Measure of 69.30% and 59.91%. Girshick et al. [39] performed semantic segmentation to segment the products from the background which was able to achieve a 78.99% of F-Measure. However, the YOLOv5 achieves the best performance on the Grocery Products dataset.

Table 5
Comparisons of retail product detection performance with existing methods on the GroZi-120 and WebMarket dataset.

Method	GroZi-120			WebMarket		
	Precision (%)	Recall (%)	F-Measure (%)	Precision (%)	Recall (%)	F-Measure (%)
George et al. [22]	13.21	43.03	20.21	-	-	53.33
Merler et al. (CHM) [19]	17	15	15.94	-	-	-
Merler et al. (SIFT) [19]	18	72	28.8	-	-	52.81
Merler et al. (Adaboost) [19]	17	15	15.94	21.3	36.3	26.8
Geng et al. (VGG16) [23]	50.44	30.69	38.16	46.8	35.7	40.5
Geng et al. (VGG16 + $ATmap^{BRISK}$) [23]	46.32	29.50	36.04	49.2	52.4	50.7
Geng et al. (VGG16 + $ATmap^{SIFT}$) [23]	49.05	29.37	36.74	44.9	57.3	50.3
Franco et al. (BoW) [25]	45.70	46.30	46.0	-	-	65.59
Franco et al. (DNN) [25]	45.20	52.70	48.66	-	-	71.13
Ray et al. [24]	-	-	40.10	-	-	67.79
Marder et al. (HOG) [20]	-	-	28.33	-	-	43.03
Marder et al. (BoW) [20]	-	-	26.83	-	-	55.15
Girshick et al. [39]	-	-	40.91	-	-	72.01
Zhang et al. [37]	-	-	31.71	-	-	49.19
Santra et al. [40]	-	-	44.81	-	-	75.50
Karlinsky et al. [26]	62.64	-	-	-	-	72.13
Ciocca et al. [60]	68.4	-	-	71.2	66.8	68.9
Yilmazer and Birant [61]	75.3	67.7	71.3	74.3	71.4	72.8
Santra et al. [62]	80.3	73.7	76.9	70.4	68.4	69.4
Leo et al. [63]	72.1	70.2	71.1	66.3	70.3	68.2
Machado et al. [64]	61.8	54.3	57.8	45.4	58.3	51.0
Domingo et al. [65]	70.2	68.3	69.2	66.4	70.4	68.3
Olóndriz et al. [66]	67.3	66.5	66.9	71.0	63.6	67.1

Method	GroZi-120			WebMarket		
	Precision (%)	Recall (%)	F-Measure (%)	Precision (%)	Recall (%)	F-Measure (%)
Bukhari et al. [67]	78.4	67.3	72.4	79.6	75.6	77.5
Proposed (YOLOv5)	**86.3**	**77.8**	**77.04**	**89.4**	**88.2**	**86.26**

5.2. Text detection result

The performance of our text detection model is examined in this section on ICDAR 2011, ICDAR 2013, ICDAR 2015, Total- Text, and CTW1500. The performance of the text detection model is shown in Table 7. We adopt the most potent backbone network, ResNet50-FPN, to push the text detection performance on different text styles such as horizontal, vertical, and curved text. However, we cropped some of the text from grocery datasets for our text detection task. These images are also used for training and testing purposes. The post-processing algorithm Width Height based Bounding Box Reconstruction (WHBBR) significantly reduces the false detection rate. We compare our backbone network with the Long et al. [29] backbone (VGG16-FPN); our model achieves the best F1-score of 81.1% on SCUT-CTW1500 and 83.3% on Total-Text.

ICDAR 2011 and ICDAR 2013 datasets focus on the horizontal text. So, we utilize these datasets to assess the robustness of our text detection model for horizontal text and the performance of our text detector for horizontal text is shown in Table 7. Similarly, the performance of our model on ICDAR 2015 dataset are compared with existing methods for detecting the oriented text. The proposed text detection model based on the WHBBR technique (f-measure: 86.03%) performs better than Long et al. [29] (f-measure: 82.60%) with an improvement of 3.4% and meets current state-of-the-art performance on IC15. Table. 7 also compares our performance with existing methods for detecting oriented text on Total-Text and CTW1500. We evaluated the efficiency of the proposed method by detecting arbitrarily shaped texts in Total-Text, were horizontal, orientated, and curved text appears simultaneously in most images.

Table 6
Comparisons of retail product detection performance with existing methods on the Grocery Products and Freiburg Groceries dataset.

Method	Grocery Products			Freiburg Groceries		
	Precision (%)	Recall (%)	F-Measure (%)	Precision (%)	Recall (%)	F-Measure (%)
George et al. [22]	23.5	43.1	30.42	23.8	-	-
Yörük et al. [41]	57.0	41.6	48.10	-	-	34.7
Marder et al. (HOG) [20]	-	-	58.11	-	-	60.6
Marder et al. (BoW) [20]	-	-	59.91	-	-	56.9
Girshick et al. [39]	-	-	78.99	72.4	68.4	70.3
Merler et al. [19]	-	-	51.20	67.9	45.6	54.5
Zhang et al. [37]	-	-	58.39	45.6	56.3	50.4
Santra et al. [40]	-	-	80.21	85.7	-	-
Ray et al. [24]	-	-	76.20	77.4	-	-
Karlinsky et al. [26]	-	-	79.05	80.3	77.8	79.0
Franco et al. (BoW) [25]	73.70	65.40	69.30	72.3	68.1	70.1
Franco et al. (DNN) [25]	73.90	54.70	62.87	76.4	69.4	72.7
Georgiadis et al. [43]	53.1	-	-	66.36	-	-
Kumar et al. (RE) [42]	65.3	68.9	67.1	86.52	-	-
Kumar et al. (SEN) [42]	73.7	-	-	86.39	81.2	83.71
Ciocca et al. [60]	81.4	78.5	79.9	78.4	74.3	76.3
Yilmazer and Birant [61]	80.5	76.2	78.3	72.3	68.4	70.3
Santra et al. [62]	78.3	67.7	72.6	80.6	80.2	80.4
Leo et al. [63]	83.6	77.7	80.5	72.3	71.1	71.7
Machado et al. [64]	86.5	83.7	85.1	78.5	74.3	76.3
Domingo et al. [65]	76.4	81.1	78.7	89.5	83.4	86.3

Method	Grocery Products			Freiburg Groceries		
	Precision (%)	Recall (%)	F-Measure (%)	Precision (%)	Recall (%)	F-Measure (%)
Olóndriz et al. [66]	74.5	78.9	76.6	77.3	73.6	75.4
Bukhari et al. [67]	84.8	85.3	85.0	76.3	68.6	72.2
Proposed (YOLOv5)	**92.1**	**86.8**	**83.31**	**89.6**	**91.5**	**90.54**

Our detection model's performance (f-measure: 84.6%) improves dramatically when the fully annotated training set is used. Similar to ICDAR 2013 and ICDAR 2015, our text detection model outperforms Long et al. [29] by 4.9% and achieves current state-of-the-art performance on the Total-Text dataset. Our text detector achieves an 81.1% F1-score and the best recall of 87.2% outperforming most previous state-of-the-art methods. Our model achieves better than some current methods, such as Zhang et al. [17] (78.4%) and Baek et al. [58] (83.6%), the top performer in F-score, is 2.5% better than ours, although it has a higher computational cost. The CTW1500 dataset has a complex background and includes a variety of multi-oriented texts. The proposed model can handle this text, demonstrating the method's robustness in terms of text appearance and shape (Fig. 11 and Fig. 12).

5.3. Significance of WHBBR

The importance of the WHBBR technique is presented in Table. 8. This proposed technique enhances the detection rate by an average of + 2.3% compared to state-of-the-art methods.

Table 8
Significance of WHBBR technique with state-of-the-art methods.

ICDAR 2015			
	Precision (%)	**Recall (%)**	**F-Measure (%)**
Baseline	84.9	80.4	82.6
Baseline with WHBBR	**87.2**	**84.9**	**86.03**
Total-Text			
Baseline	84.9	80.4	82.6
Baseline with WHBBR	**86.1**	**80.6**	**83.3**
SCUT-CTW1500			
Baseline	**84.9**	80.4	**82.6**
Baseline with WHBBR	75.8	**87.2**	81.1

On the irregular datasets, the WHBBR technique achieved better performance than baseline model. Precision (+ 3%), Recall (+ 4.5%) and F-Measure (+ 4%) on ICDAR 2015 dataset. Precision (+ 1.6%), Recall (+ 0.2%) and F-Measure (+ 0.7%) on Total-Text dataset. Recall (+ 6.5%) on SCUT-CTW1500 dataset.

5.4. Proposed Framework Summary

The proposed framework is proposed to detect and recognize on-shelf and off-shelf retail products by extracting text including, product name, price, quantity, expiry date, etc., from the product label. In order to do that, we are in need of object detection, text detection and a text recognition model. We use a popular object detection model YOLOv5 to perform retail product detection. Individual object region coordinates such as < x>, <y>, <width > and < height > are passed to a current state-of-the-art text detection model TextSnake[29] to obtain product information. However, the text detection model follows a polygon-shaped bounding box construction approach to draw over the detected text, which failed to capture the starting and ending characters in the word. To address this problem and preserve the entire text we proposed a WHBBR technique, which can draw a bounding box on the text accurately. WHBBR greatly improves the performance of the current state-of-the-art methods. The detected texts are cropped and passed to the text recognition model SCATTER [30], which recognizes the text from cropped word images. The proposed framework has various advantages such as assisting visually impaired people, reducing the time taken during checkout, identifying the number of on-shelf products, identifying misplaced products, out-of-stock products and so on.

6. Conclusion And Future Work

We proposed a new framework composed of three models (Fig. 15); product detection, product text detection, and product text recognition to detect and recognize the retail products from the supermarket shelves. Generally, the text present on the retail products (e.g. product name, brand name, price, expiring date and so on) has unique information about the corresponding product. To acquire that precious text information from the retail products, we enhanced the Textsnake text detection model by adding an accurate post-processing technique named Width Height based Bounding Box Reconstruction (WHBBR). The text detection model's modified backbone and post-processing technique greatly eliminate the false detection and inaccurate bounding boxes. The Attention-based text recognition model can accurately detect and recognize the arbitrary shaped text. The proposed framework has the practical application of assisting visually impaired people during shopping. Our framework is computationally expensive during training but can detect and recognize objects promptly and accurately during testing. Our product recognition model completely depends on the text present on the retail products. If the text is occulted or missing from the product, the product recognition model gets failed. Still, our product detection model based on the YOLOv5 algorithm can detect the product. The limitations of this paper can be addressed in our future work. The retail product can be detected and recognized based on shape and colour features. In addition, we intend to address the out-of-stock problem, product count and misplaced items.

References

1. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep Learning for Generic Object Detection: A Survey, International Journal of Computer Vision (IJCV), 128, 2020, pp. 261–318.
2. X. Zhang, Y.H. Yang, Z. Han, H. Wang, C. Gao, Object Class Detection: A Survey, ACM Computing Surveys, 46 (1), 2013, pp. 1–53.
3. Z. Q. Zhao, P. Zheng, S. T. Xu, X. Wu, Object Detection with Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems, 30 (11), 2019, pp. 3212–3232.
4. S. S. Tsai, D. Chen, V. Chandrasekhar, G. Takacs, N. Cheung, R. Vedantham, R. Grzeszczuk, B. Girod, Mobile product recognition, Proceedings of the 18th ACM international conference on Multimedia, Association for Computing Machinery, 2010, pp. 1587–1590.
5. M. George, D. Mircic, S. Gabor, C. Floerkemeier, F. Mattern, Fine-Grained Product Class Recognition for Assisted Shopping, Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 2015, pp. 546–554.
6. D. López-de-Ipiña, T. Lorido, U. López, Indoor Navigation and Product Recognition for Blind People Assisted Shopping. In: Bravo J., Hervás R., Villarreal V. (eds) Ambient Assisted Living. IWAAL 2011. Lecture Notes in Computer Science, (6693). Springer, Berlin, Heidelberg, 2011, pp. 33–40.
7. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.
8. S. Wang, Y. Zhou, J. Yan, Z. Deng, Fully motion-aware network for video object detection, Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 2018, pp. 542–557.
9. C. Feichtenhofer, A. Pinz, A. Zisserman, Detect to Track and Track to Detect, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 3057–3065.
10. W. Han, P. Khorrami, T. L. Paine,P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, T. S. Huang, Seq-NMS for Video Object Detection, 2016, pp. 1–9, arXiv preprint arXiv:1602.08465.
11. H. A. Qazi, U. Jahangir, B. M. Yousuf, A. Noor, Human action recognition using SIFT and HOG method, Proceedings of the International Conference on Information and Communication Technologies (ICICT), Moscow, Russia, 2017, pp. 6–10.
12. P. Shivakumara, T. Q. Phan, S. Lu, C. L. Tan, Gradient Vector Flow and Grouping-Based Method for Arbitrarily Oriented Scene Text Detection in Video Images, IEEE Transactions on Circuits and Systems for Video Technology, 23 (10), 2013, pp. 1729–1739.
13. J. Ye, L. Huang, X. Hao, Neural Network Based Text Detection in Videos Using Local Binary Patterns, Proceedings of the 2009 Chinese Conference on Pattern Recognition, Nanjing, China, 2009, pp. 1–5.
14. T. He, W. Huang, Y. Qiao, J. Yao, Text-Attentional Convolutional Neural Network for Scene Text Detection, IEEE Transactions on Image Processing, 25 (6), 2016, pp. 2529–2541.
15. M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: A fast text detector with a single deep neural network, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San

Francisco, California, USA, 2017, pp. 4161–4167.

16. S. Mohanty, T. Dutta, H. P. Gupta, Recurrent Global Convolutional Network for Scene Text Detection, Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 2750–2754.
17. C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, X. Ding, Look more than once: An accurate detector for text of arbitrary shapes, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 10552–10561.
18. Y. Liu, L. Jin, S. Zhang, C. Luo, S. Zhang, Curved scene text detection via transverse and longitudinal sequence connection, Pattern Recognition, 90, 2019, pp. 337–345.
19. M. Merler, C. Galleguillos, S. Belongie, Recognizing Groceries in situ Using in vitro Training Data, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 2007, pp. 1–8.
20. M. Marder, S. Harary, A. Ribak, Y. Tzur, S. Alpert,A. Tzadok, Using image analytics to monitor retail store shelves, IBM Journal of Research and Development, 59 (2/3), 2015, pp. 3:1–3:11.
21. J. S. Beis, D. G. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, Proceedings of the Computer Vision and Pattern Recognition, San Juan, PR, USA, 1997, pp. 1000–1006.
22. M. George, C. Floerkemeier, Recognizing products: A per-exemplar multi-label image classification approach, Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, Springer, 2014, pp. 440–455.
23. W. Geng, F. Han, J. Lin, L. Zhu, J. Bai, S. Wang, L. He, Q. Xiao, Z. Lai, Fine-Grained Grocery Product Recognition by One-Shot Learning, Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 1706–1714.
24. A. Ray, N. Kumar, A. Shaw, D. P. Mukherjee, U-PC: Unsupervised Planogram Compliance, Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 2018, pp. 586–600.
25. Franco, D. Maltoni, S. Papi, Grocery product detection and recognition, Expert Systems with Applications, 81, 2017, pp. 163–176.
26. L. Karlinsky, J. Shtok, Y. Tzur, A. Tzadok, Fine-grained recognition of thousands of object categories with single-example training, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4113–4122.
27. S. Advani, P. Zientara, N. Shukla, I. Okafor, K. Irick, J. Sampson, S. Datta, V. Narayanan, A Multitask Grocery Assist System for the Visually Impaired: Smart glasses, gloves, and shopping carts provide auditory and tactile feedback, IEEE Consumer Electronics Magazine, 6 (1), 2017, pp. 73–81.
28. E. Goldman, J. Goldberger, Large-Scale Classification of Structured Image Classification from Conditional Random Field with Deep Class Embedding, Computer Vision and Image Understanding 191, 2020, pp. 1–11.
29. S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision –

ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, 11206. Springer, Cham, 2018, pp. 19–35.

30. R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor and R. Manmatha, SCATTER: Selective Context Attentional Scene Text Recognizer, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11959–11969.
31. A. Gupta, A. Vedaldi, A. Zisserman, Synthetic Data for Text Localisation in Natural Images, Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2315–2324.
32. Shahab, F. Shafait, A. Dengel, ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images, Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 2011, pp. 1491–1496.
33. D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, L. P. de las Heras, ICDAR 2013 Robust Reading Competition, Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 2013, pp. 1484–1493.
34. D. Karatzas, L. G. Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, E. Valveny, ICDAR 2015 competition on robust reading, Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 2015, pp. 1156–1160.
35. K. Ch'ng, C. S. Chan, Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition, Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 935–942.
36. Y. Liu, L. Jin, S. Zhang, C. Luo, S. Zhang, Curved scene text detection via transverse and longitudinal sequence connection, Pattern Recognition, 90, 2019, pp. 337–345.
37. Y. Zhang, L. Wang, R. Hartley, H. Li, Where's the weet-bix?, Proceedings of the Asian Conference on Computer Vision, Springer, Tokyo, Japan, 2007, pp. 800–810.
38. P. Jund, N. Abdo, A. Eitel, W. Burgard, The Freiburg Groceries Dataset, 2016, pp. 1–7, arXiv preprint arXiv:1611.05799.
39. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580–587.
40. Santra, A. K. Shaw, D. P. Mukherjee, Graph-based non-maximal suppression for detecting products on the rack, Pattern Recognition Letters, 140, 2020, pp. 73–80.
41. Yörük, K. T. Öner, C. B. Akgül, An efficient Hough transform for multi-instance object recognition and pose estimation, Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016, pp. 1352–1357.
42. M. Kumar, B. Moser, L. Fischer, B. Freudenthaler, Membership-Mappings for Data Representation Learning: Measure Theoretic Conceptualization. In: Kotsis G. et al. (eds) Database and Expert

Systems Applications - DEXA 2021 Workshops. DEXA 2021. Communications in Computer and Information Science, 1479. Springer, Cham, 2021, pp. 127–137.

43. K. Georgiadis, G. K. Zilos, F. Kalaganis, P. Migkotzidis, E. Chatzilari, V. Panakidou, K. Pantouvakis,S. Tortopidis, S. Papadopoulos, S. Nikolopoulos,I. Kompatsiaris, Products-6K: A Large-Scale Groceries Product Recognition Dataset, Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference, 2021, pp. 1–7.
44. L. Neumann, J. Matas, On Combining Multiple Segmentations in Scene Text Recognition, Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 2013, pp. 523–527.
45. W. Huang, Y. Qiao, X. Tang, Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014, Lecture Notes in Computer Science, 8692, Springer, Cham, pp. 497–511.
46. X. C. Yin, X. Yin, K. Huang, H. W. Hao, Robust Text Detection in Natural Scene Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (5), 2014, pp. 970–983.
47. M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, International Journal of Computer Vision, 116, 2016, pp. 1–20.
48. M. Buta, L. Neumann, J. Matas, FASText: Efficient Unconstrained Scene Text Detector, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1206–1214.
49. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 2550–2558.
50. Y. Zhu, J. Du, Sliding Line Point Regression for Shape Robust Scene Text Detection, Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 3735–3740.
51. Z. Zhong, L. Jin, S. Zhang, Z. Feng, DeepText: A Unified Framework for Text Proposal Generation and Text Detection in Natural Images, arXiv 2016, pp. 1–12, arXiv:1605.07314.
52. S. Tian, S. Lu, C. Li, WeText: Scene Text Detection under Weak Supervision, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 1501–1509.
53. Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection, arXiv 2017, pp. 1–8, arXiv:1706.09579.
54. M. Jianqi, W. Shao, H. Ye, W. Li, H. Wang, Y. Zheng, X. Xue, Arbitrary-Oriented Scene Text Detection via Rotation Proposals, IEEE Transactions on Multimedia, 20 (11), 2018, pp. 3111–3122.
55. X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: An efficient and accurate scene text detector, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 5551–5560.
56. Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, TextField: Learning a Deep Direction Field for Irregular Scene Text Detection, IEEE Transactions on Image Processing, 28 (11), 2019, pp. 5566–5579.

57. M. Liao, P. Lyu, M. He, C. Yao, W. Wu, X. Bai, Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (2), 2019, pp. 532–548.
58. Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 9365–9374.
59. W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019,pages 9336–9345.
60. Ciocca, G., Napoletano, P., Locatelli, S.G. (2021). Multi-task Learning for Supervised and Unsupervised Classification of Grocery Images. In:, et al. Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science(), vol 12662. Springer, Cham. https://doi.org/10.1007/978-3-030-68790-8_26.
61. Yilmazer, R., & Birant, D. (2021). Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. Sensors, 21(2), 327.
62. Santra, B., Ghosh, U., & Mukherjee, D. P. (2022). Graph-based modelling of superpixels for automatic identification of empty shelves in supermarkets. Pattern Recognition, 127, 108627.
63. Leo, M., Carcagnì, P., & Distante, C. (2021, January). A Systematic Investigation on end-to-end Deep Recognition of Grocery Products in the Wild. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 7234–7241). IEEE.
64. de Lima Machado, A., Aires, K., Veras, R., & Neto, L. B. (2021, November). Grocery Product Recognition to Aid Visually Impaired People. In Anais do XVII Workshop de Visão Computacional (pp. 94–99). SBC.
65. Domingo, J. D., Aparicio, R. M., & Rodrigo, L. M. G. (2022). Cross Validation Voting for Improving CNN Classification in Grocery Products. IEEE Access, 10, 20913–20925.
66. Olóndriz, D. A., Puigdevall, P. P., & Palau, A. S. (2021). FooDI-ML: a large multi-language dataset of food, drinks and groceries images and descriptions. arXiv preprint arXiv:2110.02035.
67. Bukhari, S. T., Amin, A. W., Naveed, M. A., & Abbas, M. R. (2021). ARC: A Vision-based Automatic Retail Checkout System. arXiv preprint arXiv:2104.02832.

Table 7

Table 7 is available in the Supplementary Files section

Figures

Figure 1

Challenges in on-the-shelf retail product recognition

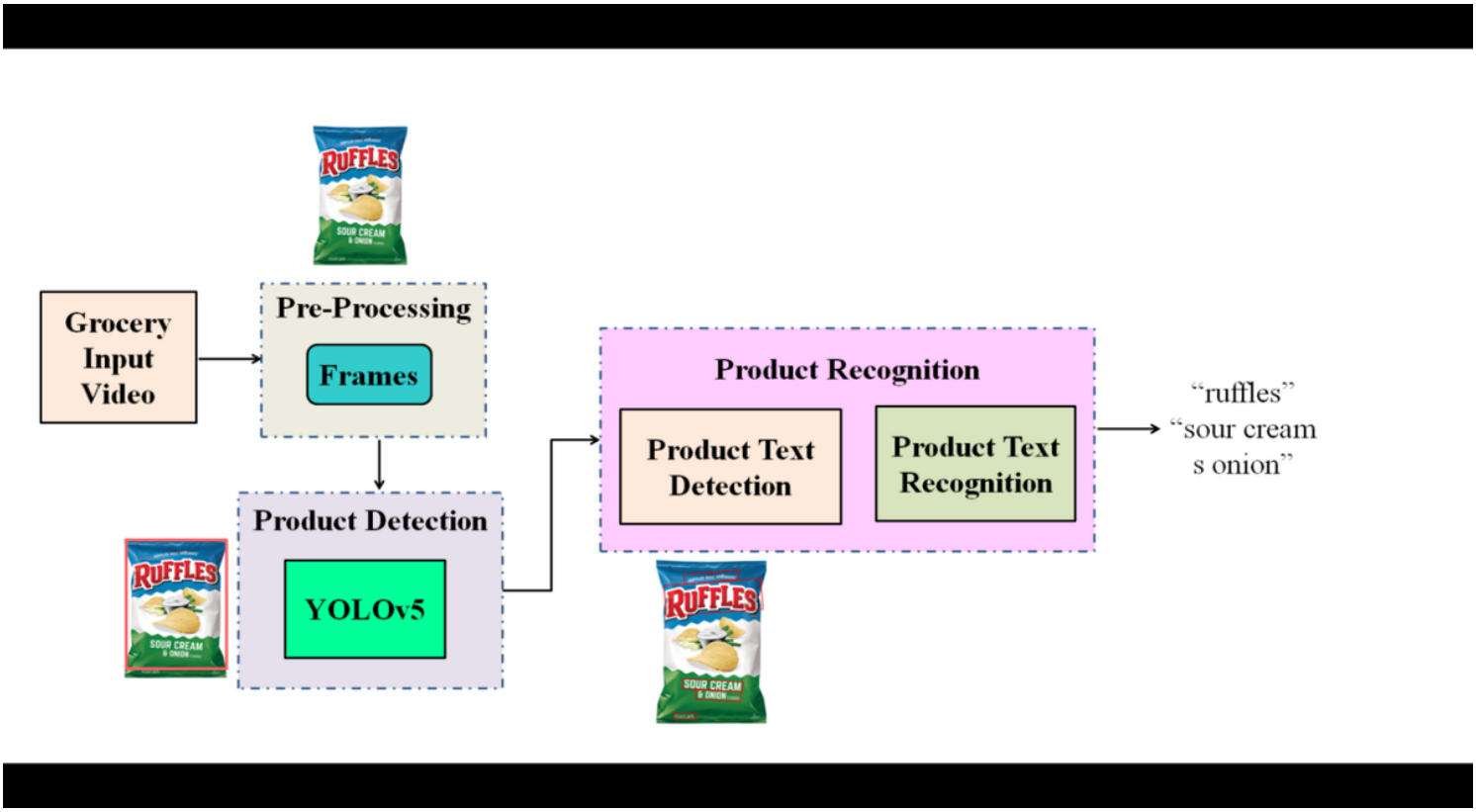

Figure 2

Block diagram of proposed grocery product detection and recognition

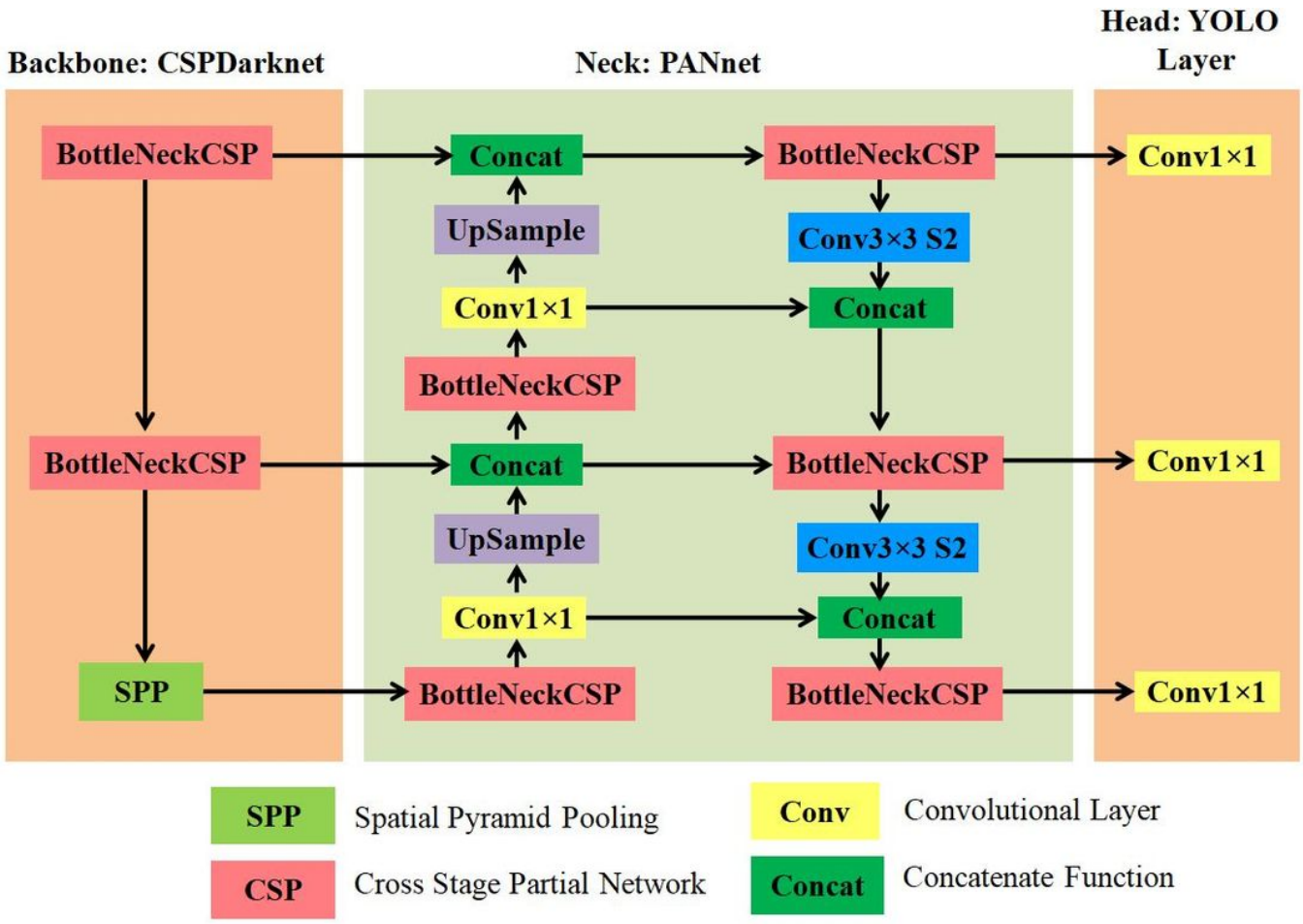

Figure 3

YOLOv5 Architecture

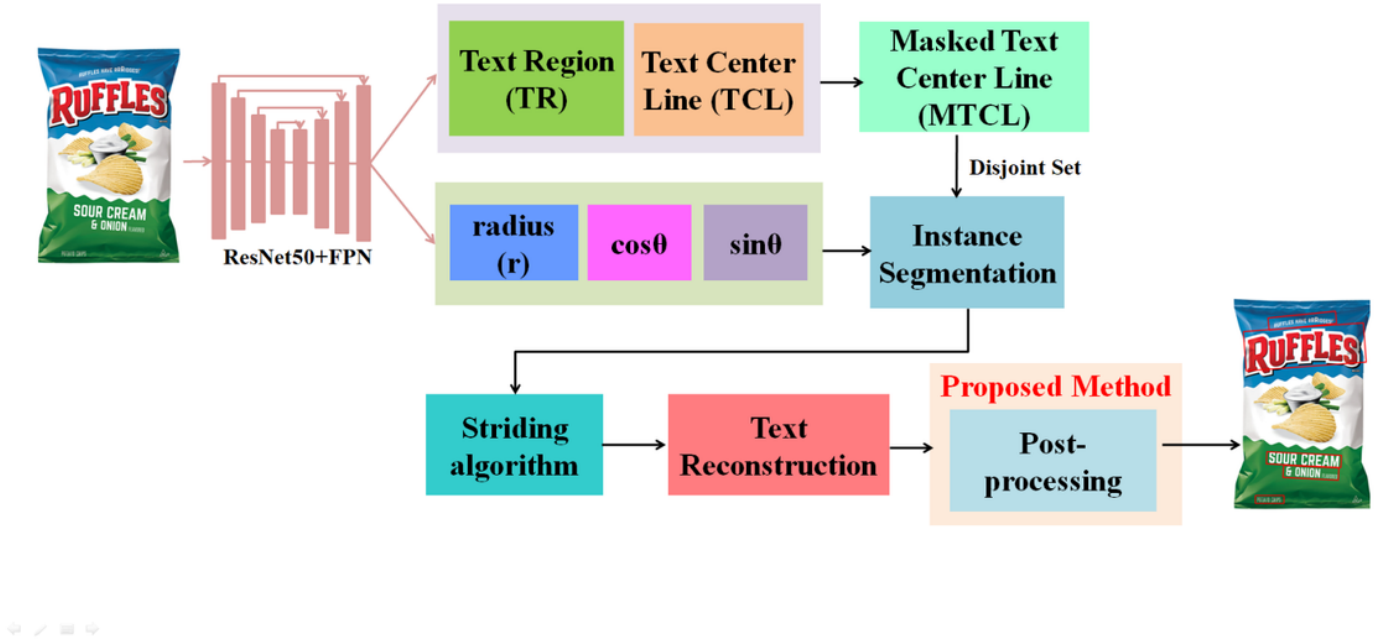

Figure 4

The overall architecture of the text detection model.

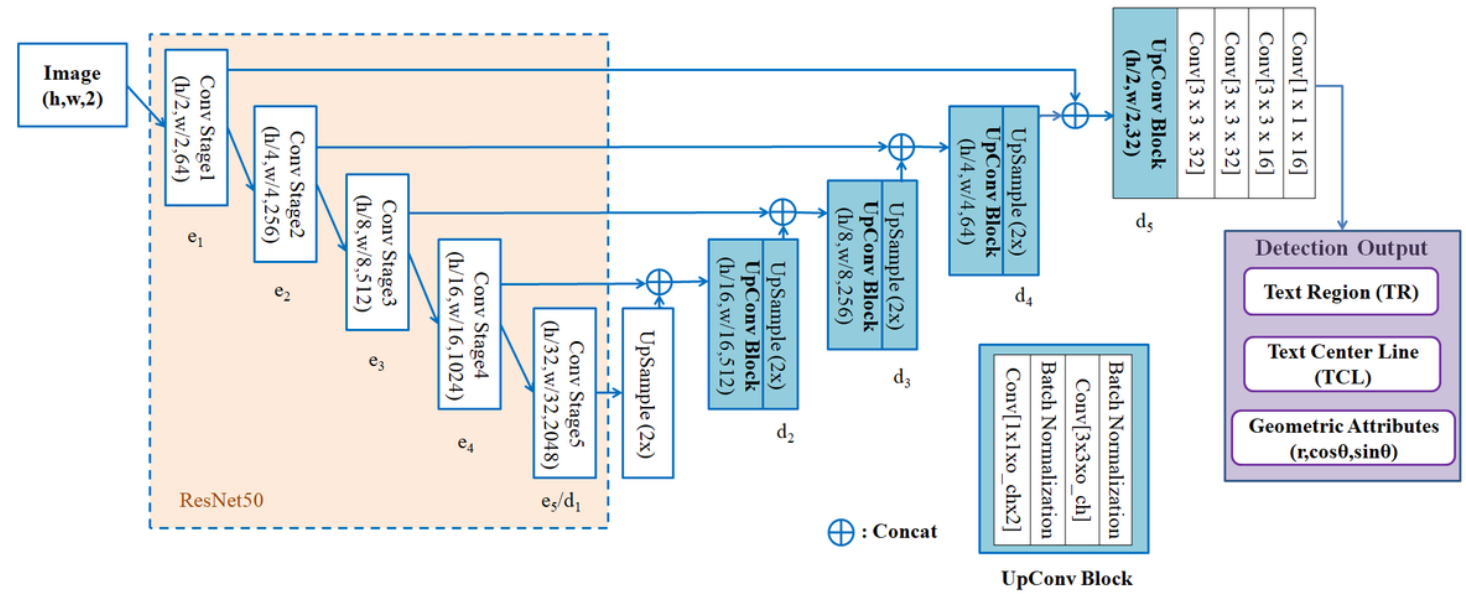

Figure 5

Schematic overview of the text detection backbone network.

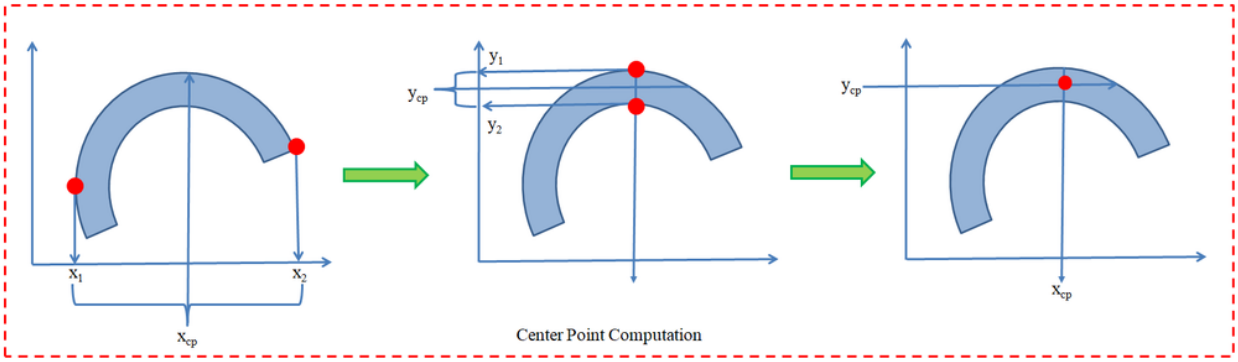

Figure 6

Calculating center point computation to perform the striding operation

Figure 7

Schematic overview of TCL extraction and TCL expansion.

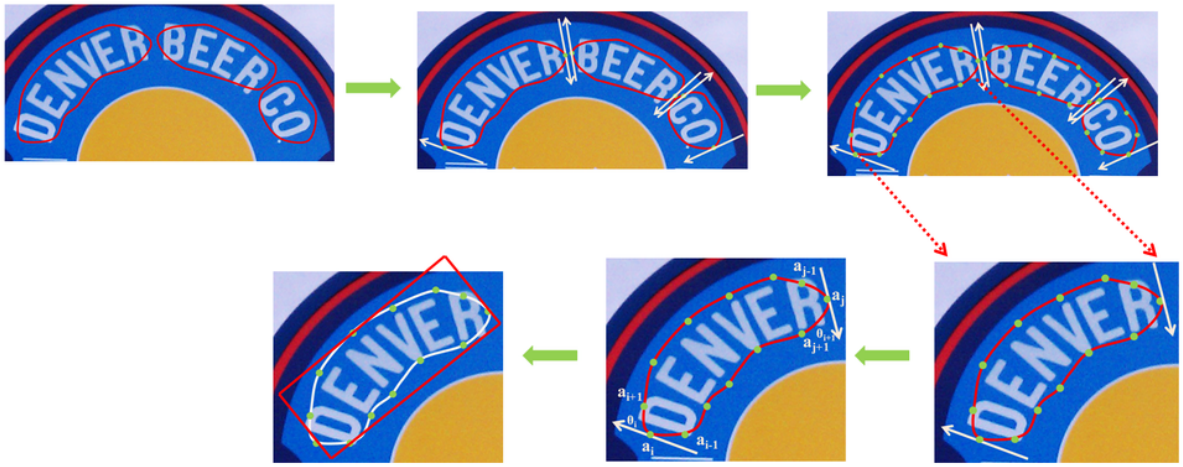

Figure 8

Schematic overview post-processing using the Width Height based Bounding Box Reconstruction (WHBBR) algorithm.

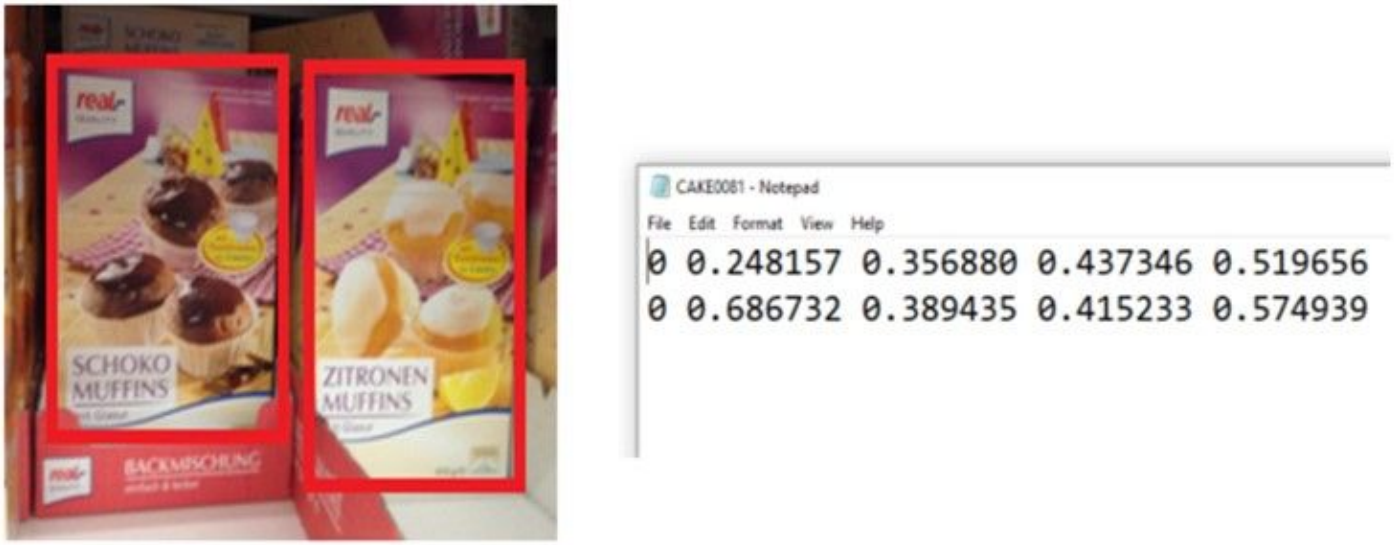

Figure 9

Illustration of bounding box annotation and its format

Figure 10

Visualization results of text detection model on the Grozi 120 dataset

Figure 11

Visualization results of grocery product detection by the YOLOv5 algorithm on the public benchmark retail product datasets.

Figure 12

Visualization results of text detection by the proposed model on the public benchmark text detection datasets.

Figure 13

Visualization results of text detection by the proposed model on the public benchmark retail product datasets.

Figure 14

The text recognition results on the public benchmark retail product dataset.

Figure 15

Product detection and product recognition are based on product text information

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Table7.docx