



# A Deep Learning Framework for Grocery Product Detection and Recognition

Prabu Selvam<sup>1</sup> · Joseph Abraham Sundar Koilraj<sup>1</sup>

Received: 1 March 2022 / Accepted: 29 July 2022 / Published online: 13 August 2022  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## 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, 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 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-the-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 by replacing the backbone network with ResNet50 + FPN and by introducing a new post-processing technique, Width Height based Bounding Box Reconstruction, to mitigate the problem of inaccurate text detection. In the final module, we used a state-of-the-art text recognition model 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 post-processing approach improves the performance of the existing methods on both regular and irregular text. The robust text detection and 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, and expiring date. The recognized text contexts around the retail products can be used as the identifier to distinguish the product.

**Keywords** Deep learning · Object detection · Text detection · Object recognition · Text recognition · Retail product

## Introduction

Detection and recognition of objects in a static image or video frame are basic and challenging tasks in computer vision. Object recognition and detection have been the subject of much research in the last two decades (Liu et al. 2020; Zhang et al. 2013). 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 (Zhao et al. 2019). 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 flavors of the same brand of Lays chips (Tsai et al. 2010). 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

✉ Prabu Selvam  
prabu@cse.sastra.ac.in

Joseph Abraham Sundar Koilraj  
josephabrahamsundar@it.sastra.edu

<sup>1</sup> School of Computing, SASTRA Deemed to be University, Thanjavur, India



**Fig. 1** Challenges in on-the-shelf retail product recognition

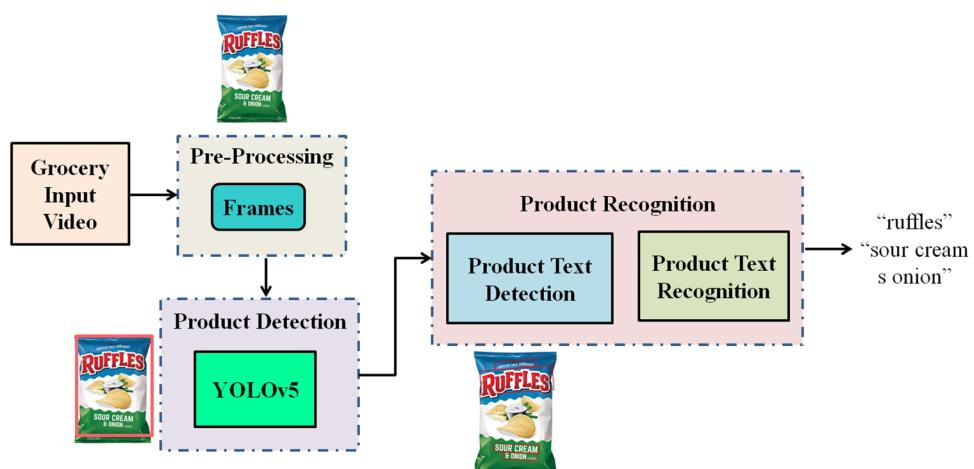
shopping (George et al. 2015). Because product appearance varies significantly due to the substantial changes in pose, perspective, size variations, occlusion, and lighting conditions, product detection/recognition in grocery shops is complicated. Additional peculiar issues are the product's packaging 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 color. Detecting/recognizing specific products is complex, unlike classifying products in macro-categories such as shampoo, chips, and detergent. 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.

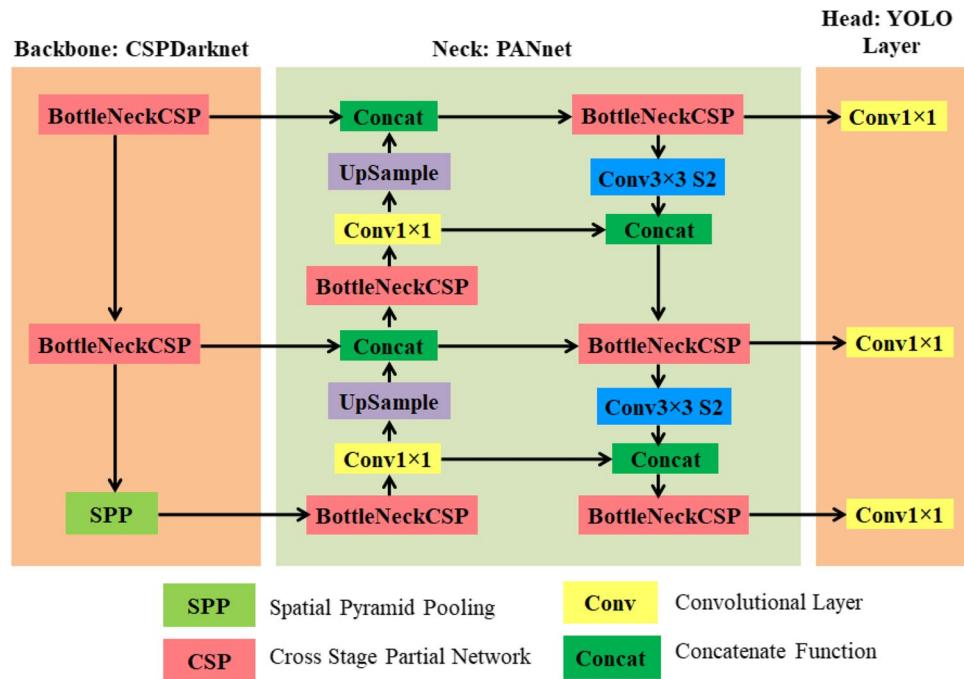
Acquisition and manual annotation of the training images is a time-consuming task. It is not feasible because the products frequently change over time; collecting and annotating new in-store images and retraining the system is not viable. The system must endure cross-domain scenarios where testing images are obtained from different stores and with varying imaging conditions. Since the training and testing images are from varying imaging conditions, it is vital to establish an ideal system that only needs to be trained once and used in various stores and scenarios. López et al. (López-de-Ipiña et al. 2011) developed an automatic product recognition system using radio frequency identification, sensors, or barcodes. The majority of sensor-based systems require manufacturing fabrication, increasing the product's cost and requiring massive investment. The sensor-based methods cannot resolve the planogram compliance problems. Compared to sensor-based approaches, computer vision possesses 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-the-shelf grocery products from shelf images and video frames. We divided our proposed framework into three steps, as shown in Figs. 2 and 3: (1) *Grocery product detection*—we incorporated the YOLO (You Look Only Once) algorithm (Redmon et al. 2016) to perform the grocery product detection task; we trained and tested the YOLO algorithm using benchmark grocery product datasets to detect multiple objects from the shelf image and video frame. (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, and expiry date. (3) *Object text recognition*—detected texts are recognized using an existing text recognition algorithm (Litman et al. 2020). The recognized

**Fig. 2** Block diagram of proposed grocery product detection and recognition



**Fig. 3** Illustration YOLOv5 architecture



texts contain complete information about the corresponding product.

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

- We incorporated the YOLOv5 algorithm to perform the grocery product detection task. We increased the object class size from 80 to more than 120. YOLOv5 algorithm works well for general object detection, where we used and trained it 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 by changing the backbone network from VGG16 to ResNet50. We proposed an algorithm to select the centering point in the text center line instead of picking a random point.
- We proposed an accurate post-processing technique to capture corner characters by combining the 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 products and individual product images. So, we performed a recognition task with only static images.

The organization of the paper is as follows: the “[Related Work](#)” section describes the literature review on object detection, text detection, and retail product detection and recognition. The “[Proposed Framework](#)” section explains the proposed framework and Width Height based Bounding Box Reconstruction (WHBBR) technique with the help of schematic diagrams. The “[Experiment](#)” section presents a description of datasets and implementation details. The “[Results and Discussion](#)” section presents experimental results and a brief discussion of the research outcomes. Finally, the “[Conclusion and Future Work](#)” section draws a conclusion and future work.

## Related Work

This section describes different works 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.

## General Object Detection

Object detection has been a trending area approached by many 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 a building, a human, a dog, a bicycle, or a cat in an image or video frame. 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 color images that can be matched to detect the object. Viola-Jones detectors, histogram of oriented gradients (HOG), principle component analysis (PCA), and Haar-like wavelet transform are the techniques used for object detection in the early stages.

Viola and Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. The approach was used to construct a face detection system. This system experimented on a Real-World MIT + CMU test set. This set consists of 130 images with 507 labeled frontal faces. This system yields detection rates comparable to the best previous systems. The detector runs at 15 frames per second without resorting to image differencing or skin color detection. Hence, it is called the first real-time face detector. Viola and Jones introduced the Haar-like feature in their system, which is a basic rectangular feature used for object identification. The Haar-like feature is also called the Haar wavelet. The Haar-like feature can effectively reflect the local gray change information of the image, and can also be quickly calculated through the integral image. Later it was extended by adding rectangular features with a rotation of 45°. The extended features are roughly divided into three types: edge features, line features, and center-surround features.

The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision. It detects and describes local features in digital images. It also locates certain key points and then furnishes quantitative information. SIFT was used for object detection in an early stage of research works. The major advantage of SIFT features, over HOG, is invariant to rotation, translation, scaling, viewpoint, and illumination. SIFT was 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. 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.

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. YOLO (Redmon et al. 2016), Single Shot MultiBox Detector (SSD), DetectNet, and SqueezeDet are the typical one-stage algorithms. One-stage methods only localize the object without computing region proposals directly by performing bounding box regression and classification tasks. The two-stage method follows two steps for object detection. The input 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.

Regional Convolutional Neural Network (R-CNN) 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 the convolutional neural network (CNN) to perform classification and bounding box regression tasks. The pace of training and detection was quite weak 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. Region Proposal Network substitutes the selective search approach in Faster R-CNN, which overcomes the issue of significant time overhead in producing the 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. YOLO does not employ multi-scale feature maps. Compared to SSD, generalization capabilities are inferior in YOLO for large-scale changes in an object. It 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. It uses convolutional layers. When a fully connected layer is used to predict bounding boxes, the feature maps can be lost. To over this, 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 from 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 also improved.

Many researchers have widely explored the major problems of object detection in videos and scene images. In these research works many solutions have also been suggested (Liu et al. 2020; Zhang et al. 2013; Zhao et al. 2019). Most of the video object detection algorithms had two networks: Firstly, CNN was used as the backbone network for the feature extraction. Secondly, the detection network classifies the objects and predicts the bound boxes. The methods for detecting objects using deep learning can be classified into LSTM (Long Short Term Memory) based, tracking-based, flow-based, attention-based and other methods. Wang et al. (Wang et al. 2018) proposed MANet to find the optical flow information between adjacent frames, to deliberate the optical flow information in which it extracts the global image features together. This method was efficiently used to extract features by acquiring the instance-level calibration across frames with the optical flows technique and then the pixel-level feature calibration was to improve the performance of video object detection. D&T proposed a ConvNet architecture (Feichtenhofer et al. 2017) to improve object detection and object tracking performance by introducing a multi-task objective frame-based object tracking by adopting techniques such as frame track, regression, correlation features and frame-level detection. The Seq-NMS (Han et al. 2016) proposed a heuristic method composed of sequence selection, re-scoring, and suppression for re-ranking bounding boxes (Qazi et al. 2017) in a video sequence. TSSD (Temporal Single-Shot Detector) method integrates ConvLSTM-based attention used for background and scale suppression.

## Text Detection

A novel method was proposed by Shivakumara et al. (Shivakumara et al. 2013) to detect text from video frames based on neighbor component grouping and GVF (Gradient Vector Flow), which use dominant edge pixels to extract TC (Text Candidates). They also presented two grouping schemes: the first scheme tries to find the nearest neighbors. The second scheme extracts neighbors and restores missing CTC to detect arbitrary text in video frames (Neumann and Matas 2013; Huang et al. 2014; Yin et al. 2014; Jaderberg et al. 2016). Hybrid text detection and text tracking work proposed (Zhong et al. 2016) based on MSER (Maximally Stable Extremal Region), which uses Delaunay Triangulation and multi-scale integration to identify the text candidates. It also utilizes convolving Laplacian with wavelet sub-bands to enhance low-resolution text pixels. The MSERs and SWT (Stroke Width Transform) were combined (Jiang et al. 2017) to obtain arbitrarily shaped text regions from video frames. To detect text from complex video frames, Ye et al. (Ye

et al. 2009) 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 by Zhu et al. (Zhu and Du 2018). In the first stage, video frame features are extracted. Then, text candidates are detected by optimizing RBFNN (Radial Basis Function Neural Network) model, which transforms the input signal into another form, which can be then fed into the network to get linear separability. RBFNN is composed of an input, hidden and output layer. RBFNN is strictly limited to have exactly one hidden layer. The hidden layer is otherwise called a feature vector. A non-linear transfer function can be applied to the feature vector before performing the classification task. When the dimension of the feature vector is increased, the linear separability of the feature vector also increases. RBFNN is structurally the same as perceptron (MLP). RBFNN can be distinguished from other neural networks due to their universal approximation and faster learning speed. Radial basis function networks have many advantages, including function approximation, time series prediction, classification, and system control. RBFNN have characteristics of easy design, good generalization, strong tolerance to input noise, and online learning ability. Finally, a post-processing task was applied to the false detected text candidates. To classify textual and non-textual components, He et al. (He et al. 2016) presented a framework for text detection called Text-Attentional Convolutional Neural Network. They introduced a novel training mechanism to increase the robustness against a complex background.

Liao et al. (Liao et al. 2017) developed a contrast enhancement maximally stable extremal regions detector to improve the video frame's intensity. This method detects complex-shaped text with a high accuracy rate. The false detection rate was reduced by using the loss function and non-maximum suppression (NMS) (Liao et al. 2019) produces final bounding boxed text regions. To detect text regions from low-quality images, the script identification task was introduced. This method extracts low and high-level features using the CNN-LSTM framework and an attention-based patch was used to respective weights which were calculated. Initially, 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 were fused dynamically to perform script identification. Coarse candidate region detection and fine text line detection are effectively used for detecting multi-scale candidate text areas. Candidate text regions are segmented and fed into CNN, which generate 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 developed by Jianqi et al. (Jianqi et al. 2018), which was constructed using a fully convolutional network.

A multi-scale deformable convolution structure was introduced by Mohanty et al. (Mohanty et al. 2018) to extract additional features and spatiotemporal information using a bipartite graph model and the random walk algorithm. Firstly, text candidates and background regions are extracted. Then, shape, motion, and spatial relations between text and background are exploited to refine text candidates. The correlations between text and background regions highly improvise the accuracy of text detection. Zhang et al. (Zhang et al. 2019) proposed a fusion-based detection method which extracts text regions and locates characters. They used tracking trajectories to refine detection results. Liu et al. (Zhang et al. 2019) developed a polygon-based curve text detector by combining R-CNN and transverse and longitudinal offset connection for the precise detection of irregular texts. Post-processing methods, such as non-polygon suppress and polygonal non-maximum suppression, produce accurate text detection result.

## 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 et al. (Merler et al. 2007) introduced a retail product detection problem with a new dataset consisting of rack and product images. It took almost eight years to develop a more comprehensive method to detect and recognize multiple retail products. Marder et al. (Marder et al. 2015) designed two successive layers for a multi-product detection scheme. In the first layer, they followed three different techniques to detect retail products in the rack, i.e., (i) vote map, (ii) HOG, and (iii) BoW (bag of words), based on a sliding-window approach. A saliency map was also used for product recognition and to address the second layer's planogram compliance problem. Beis et al. (Beis and Lowe 1997) used a k-d tree representation and Speeded Up Robust Features (SURF) 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 was used to perform fine-grained recognition.

George et al. (George and Floerkemeier 2014) proposed a three-phase detection and recognition method. 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 Karhunen–Loeve transform in the final phase to track the detected boxes in a video. Geng

et al. (Geng et al. 2018) developed a product detection system to identify the locations of the products in the video frames by creating a saliency map for the shelf images. The saliency map is constructed using SURF key points and Attention and Information Maximization. Finally, a CNN is used to recognize the products. Ray et al. (Ray et al. 2018) presented a conditional random field (CRF)-based method for classifying structured objects. A CNN extracts the visual features and that was linearly fed into a CRF model. Viterbi and forward–backward algorithms were used to generate the labels of the product sequence.

Franco et al. (Franco and Maltoni 2017) divided the product detection and recognition task into three steps: (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 deep neural network (DNN) and a BoW to select the most robust features; (iii) post-processing technique (Kumar et al. 2021; Georgiadis et al. 2021) reduces the false positives by eliminating the multiple overlapped detections of the same products. Karlinsky et al. (Karlinsky et al. 2017) and Zientara et al. (Advani et al. 2017) calculated a homography matrix to identify the grocery products in shelf images by matching SURF key points of product images with corresponding rack images. Goldman et al. (Goldman and Goldberger 2020) use 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. (Bukhari et al. 2021) developed a vision-based Automatic Retail Checkout system, which uses CNNN for object detection. Canny edge detector and hysteresis thresholding are used to perform NMS and generate 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. (Ciocca et al. 2021) introduced a multi-task learning network to extract features from the images. They performed the product classification in both supervised and unsupervised learning methods. Yilmazer and Birant (Yilmazer and Birant 2021) combined two concepts semi-supervised learning and on-shelf availability to identify the empty shelves. Similarly, Santra et al. (Santra et al. 2022) use graph convolutional network (GCN) and Siamese network architecture (SNA) for feature extraction and to capture the similarity of the neighboring superpixels, respectively. Finally, the features extracted from GCN and SNA are fed to SSVM for the identity gaps on the rack. Leo et al. (Leo et al. 2021) assessed the performance of different classification models. Olondriz et al. (Olondriz et al. 2021) introduced the FooDI-ML dataset and Glovo application to recognize the retail product information. Machado et al. (Lima Machado et al. 2021) developed a product recognition system for visually impaired people. The authors inferred

that the ResNet-50-based approach achieves better results than other deep learning-based models. Domingo et al. (Domingo et al. 2022) use cross-validation-voting scheme to classify the retail products.

The existing methods have certain limitations: Santra et al. (Santra et al. 2022) work only in brighter images and dark products on the shelves are assumed as empty shelves. Ray et al. (Ray et al. 2018) cannot differentiate non-identical objects. Geng et al. (Geng et al. 2018) face more partial detections. Franco et al. (Franco and Maltoni 2017) and Karlinsky et al. (Karlinsky et al. 2017) face labeling problem (inaccurate box). Olondriz et al. (Olondriz et al. 2021) require more than 2.73 million parameters, computationally complex.

## 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 text on grocery product packing (brand name, product name, quantity, and other information). Finally, the third module recognizes the text using the current state-of-the-art text recognition algorithm. The recognized text has unique information about the corresponding product.

### 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 s, i.e., two frames are extracted per second. We use cv2.VideoCapture() and vidcap.read() pre-defined function to capture the video frames.

### Grocery Product Detection using YOLOv5

YOLOv5 incorporates CSPDarknet and Path Aggregation Network (PANet), which makes it easier to train the object detection model and reduces the computation cost, respectively. When compared to other models, YOLOv5 performs better at detecting smaller objects or far away objects, and inference speed is good when compared to Faster-RCNN, Fast-RCNN, and SSD. Unlike R-CNN and SPP-net, there were no overlapping boxes around the objects.

First, 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 which significantly improves the CNN's learning ability. In the case of accuracy, it was found to lag due to being lightweight and simultaneously it reduces the 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. Detection speed, model size, and accuracy are imperative in our grocery product detection task. The resultant efficiency of product detection on low-resource edge devices is determined based on model size.

Second, 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 features. Similarly, large receptive fields capture richer context information on a high-level feature to produce accurate prediction. A fully connected fusion is used for mask prediction that differentiates instances and recognizes 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.

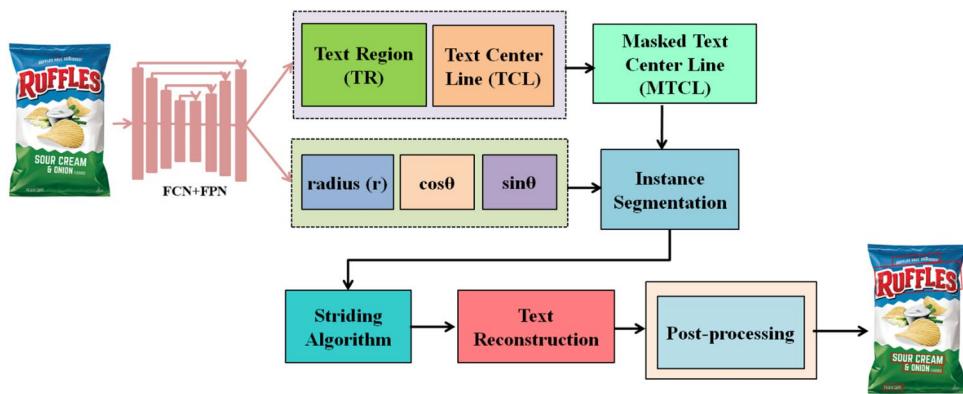
Finally, to achieve multi-scale prediction, the head of the YOLOv5 algorithm adapts the YOLO layer and produces feature maps of different sizes such as  $19 \times 19$ ,  $38 \times 38$ , and  $76 \times 76$  helping the model to handle and detect small, medium, and oversized objects. It also predicts 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 can detect the grocery product even if the size changes during the detection process.

### Text Detection

We propose a text-detecting method for detecting text information in grocery items such as product name, brand name, and amount using an efficient post-processing technique. Figure 4 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 proposed a curve-shaped text detection method. 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  $O(t) = \{W_0, W_1, \dots, W_i, \dots, W_n\}$  consisting of multiple characters, where " $W_i$ " and " $n$ " in the ordered list represent the  $i$ th disk and

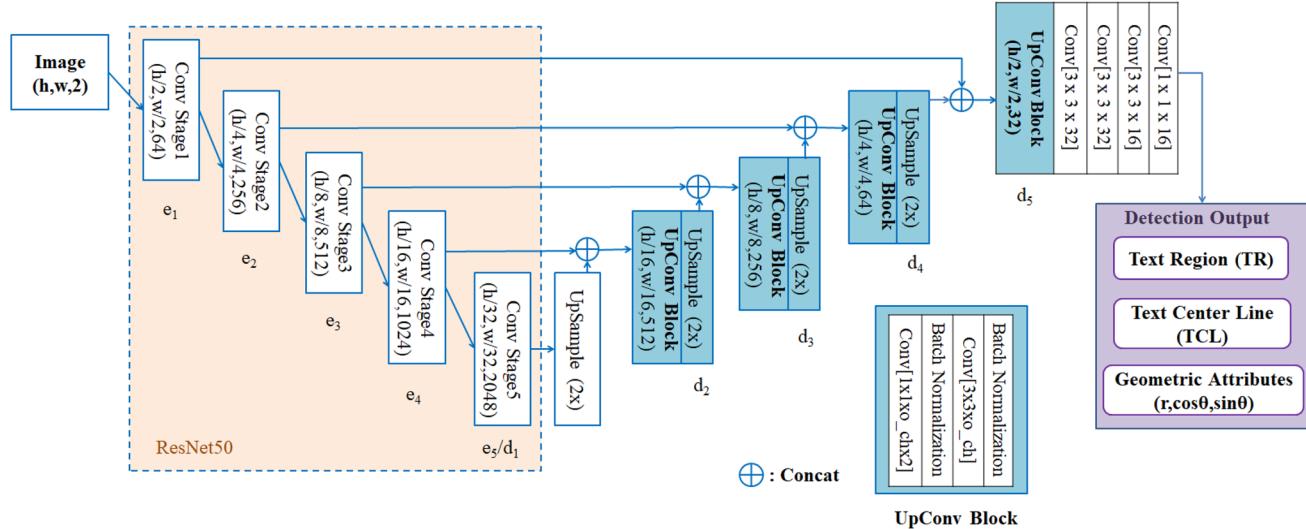
**Fig. 4** The overall architecture of the text detection model



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.,  $W=(c,r,\theta)$ . In the center, 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  $\theta$ . 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 (TRs), text center line (TCL), and its geometric attributes such as radius ( $r$ ),  $\sin \theta$ , and  $\cos \theta$ . 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 instance reconstruction and mitigates false text detection.

## 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. 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



**Fig. 5** Schematic overview of the text detection backbone network

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

Procedure Centralizing	
<b>Input:</b>	x- axis Leftmost point ( $x_1$ ), Rightmost point ( $x_2$ ), y- axis topmost point ( $y_1$ ), bottommost point ( $y_2$ )
<b>Output:</b>	x- axis ( $x_{cp}$ ),y- axis ( $y_{cp}$ ), text center point ( $t_{cp}$ )
1.	$x_{cp} = \text{median}(x_1, x_2)$
2.	$y_{cp} = \text{median}(y_1, y_2)$
3.	$t_{cp} = (x_{cp}, y_{cp})$

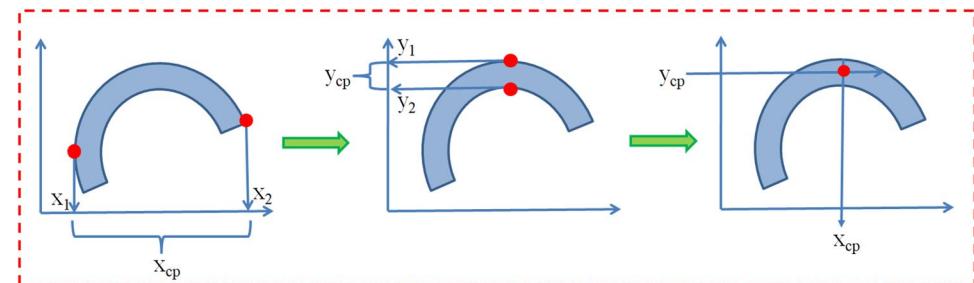
step. In a merging network, several stages are piled one on top of the other. Each stage has its merging unit that extracts feature maps from its previous stage. The following Eqs. (1–3) interpret the merging branch.

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

$$d_i = \text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}[e_i - 1; \text{UpSample}_{\times 2}(d_i - 1)]), \text{for } i \geq 2 \text{ and } i \leq 4 \quad (2)$$

$$d_5 = \text{Conv}_{3 \times 3}(\text{Conv}_{3 \times 3}(\text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}[e_i - 1; \text{UpSample}_{\times 2}(d_i - 1)]))), \\ \text{for } i = 5 \quad (3)$$

where  $e_i$  and  $d_i$  represent feature maps of the  $i$ th stage and the corresponding upsampling and merging units, respectively. After 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  $r$ ,  $\sin\theta$ , and  $\cos\theta$ . The backbone network generates TCL, TR, and geometry maps after feed-forwarding. TR is a binary mask with 1 for foreground pixels (those inside the polygon annotation) and 0 for background pixels. TCL is computed using the sequencing process.

**Fig. 6** Calculating center point computation to perform the striding operation

## 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.

**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.

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

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

Equations (4) and (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 ( $\theta$ ) is determined. Since the TCL is a straight line, it is simple to compute it using algebraic triangles and quadrangles. But it is not 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 1.

Text instances ( $t$ ) represented as a set of vertices ( $v_0, v_1, v_2, \dots, v_n$ ). 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  $M(e_{i,i+1}) = c \cos(e_{i-1,i,i+1,i+2})$ , and 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  $\frac{1}{4}$  of the radius of control points so that most of the TCL pixels remain within TR. If we take  $\frac{1}{2}$  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.

### Width Height Based Bounding Box Reconstruction Algorithm

Table 1 and Fig. 8 show the algorithm and working mechanism of the proposed post-processing technique, respectively. The final resultant bounding box after the text reconstruction step is in the form of an arbitrary shape that fails to capture the corner characters (starting and ending characters) in the processed image. We propose an accurate post-processing technique named Width Height based Bounding Box Reconstruction (WHBBR) algorithm to

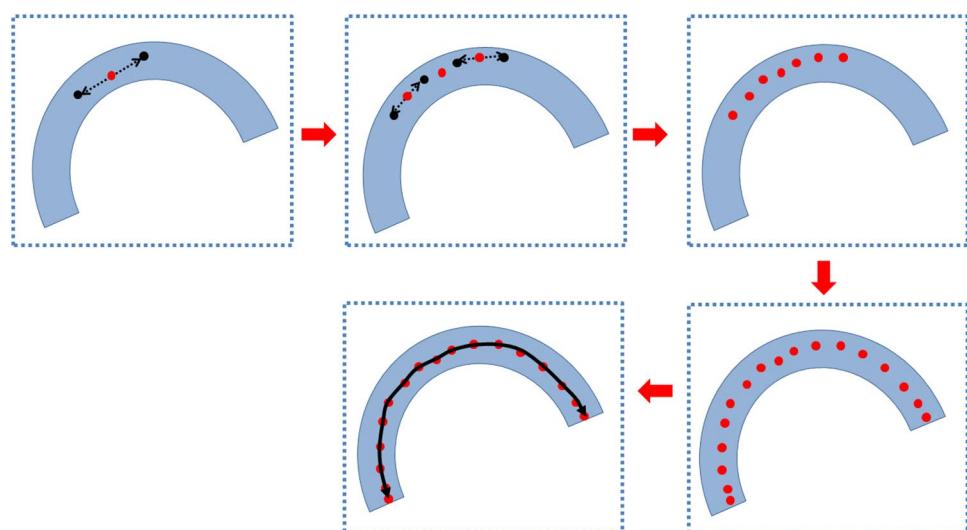
enclose the starting and ending characters. Firstly, the set of arbitrary-shaped bounding box coordinates is given as input to the Graham Scan Algorithm. We adopted this algorithm to perform the search operation on the top, left, right, and bottom-most coordinates and so it facilitates us to choose the outmost extreme points by sorting all the points. The main aim of using the Graham Scan Algorithm in this work is to find those coordinates. 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, two directed tangent lines of support were drawn at antipodal points. Next, at each antipodal point position, these two parallel lines are used to compute the distance measures between the antipodal points of the arbitrary bounding box. These two parallel lines visit all pairs of antipodal vertices by rotating clockwise. This process continues until all the vertices are visited at least once. At each iteration, the height and width are computed. Finally, the peak height and width with their respective coordinate are considered for detection of the arbitrary-shaped text that will be then enclosed with an accurate bounding box.

The WHBBR algorithm identifies the corner characters accurately and converts the arbitrarily shaped bounding box into a rectangular bounding box.

### Text Recognition

We adapt the context attentional network (Litman et al. 2020) as our text recognizer. Cropped text images are fed into a text recognition model. A four-step mechanism was followed to process the detected text image. Firstly, the cropped text image has been transformed into the normalized image using thin-plate spline transformation to reduce the burden for the subsequent feature extraction stage. Secondly, a 29-layer ResNet is used as the convolutional neural

**Fig. 7** Schematic overview of TCL extraction and TCL expansion



**Table 2** Procedure for Mask to TCL conversion and the proposed WHBBR Algorithm

<b>Procedure</b> mask_to_tcl	<b>Algorithm 1:</b> Width Height based Bounding Box Reconstruction (WHBBR)
<b>Input:</b> pred_sin, pred_cos, pred_radii, tcl_contour, direction, initial_x, initial_y <b>Output:</b> text center line <pre> <b>initialize</b> H,W, flag = 1, x_shift = initial_x, y_shift = initial_y <b>initialize</b> result = [], max = 200, iteration = 0 <b>while</b> in_contour(tcl_contour,( x_shift, y_shift)) <b>do</b>   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) <b>while</b> !contour_end <b>do</b>   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 * </pre>	<b>Input:</b> Set of points A={ a <sub>0</sub> ,a <sub>1</sub> ,a <sub>2</sub> ,..., a <sub>n</sub> } polygon bounding box vertices. <pre> <b>Initialize</b> count = 0, Detected = 0. <b>Compute</b> x<sub>min</sub>, x<sub>max</sub> y<sub>min</sub>, and y<sub>max</sub> <b>Draw</b> two vertical parallel lines of support on A through y<sub>min</sub> and y<sub>max</sub> <b>While</b> visited edges[] !=NULL <b>do</b>   <b>if</b> one    both parallel lines tangents with   an edge, <b>then</b>     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   <b>end if</b>   <b>if</b> Detected edge pair ⊥ to the x-axis, <b>then</b>     W_Detected = Detected   <b>else</b>     H_Detected = Detected   <b>end if</b>   <b>Rotate</b> the parallel lines until one is   connected to the next   polygon edge is detected.   <b>if</b> new antipodal pair is detected, <b>then</b>     Add edge pair in the visited edges list     Temp_distance = new edge length     flag=flag+1   <b>if</b> Temp_distance &gt;Max_distance <b>then</b> </pre>

network's backbone to extract essential features from the input image. The final feature map is 512 channels. Thirdly, CTC-Attention-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

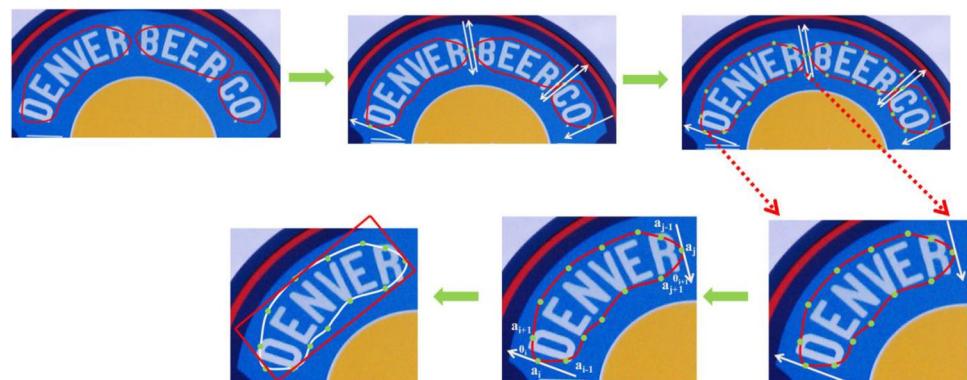
**Table 2** (continued)

<pre>         stride * flag         y_shift_back = y_center - sin_center *         stride * flag <b>if</b> size_of_result = 1 <b>then</b>         final_x = x_shift_front         final_y = y_shift_front <b>else</b>         compute the distance_front and         distance_back <b>if</b> distance_front &gt; distance_back <b>then</b>         final_x = x_shift_front         final_y = y_shift_front <b>else</b>         final_x = x_shift_front         final_y = y_shift_front <b>endif</b> <b>end while</b> <b>if</b> final_x ≥ W    final_x &lt; 0    final_y ≥ H    y_shift &lt; 0 <b>then</b>         break <b>endif</b> <b>end while</b> <b>return</b> result      #contains the coordinates of text center line </pre>	<pre> Max_distance = Temp_distance W_Detected = an antipodal-edge or edge- edge pair <b>else</b> Min_distance = Temp_distance H_Detected = an antipodal-edge or edge- edge pair <b>end if</b> <b>end if</b> <b>if</b> visited edges[] contains all the edges &amp; parallel lines reach their original position, <b>then</b> <b>break</b> <b>end if</b> <b>end while</b> <b>return</b> W_Detected, H_Detected, Max_distance, Min_distance </pre>
--	--

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. The

**Fig. 8** Schematic overview post-processing using the Width Height Based Bounding Box Reconstruction (WHBBR) algorithm



recognized text information such as product name, quantity, expiration date, and price are displayed in a separate file. This text recognizer also supports our framework to obtain the on the shelf and off-shelf product information, extending to which it helps the visually impaired people to hear the product information through an audio device.

## Experiment

This paper uses a synthetic dataset SynthText (Gupta et al. 2016) to train our text detection 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 has been evaluated using standard benchmark datasets such as ICDAR 2011 (Shahab et al. 2011) and ICDAR 2013 (Karatzas et al. 2013). The majority of the text images in these datasets are almost horizontal text images:

- **ICDAR2011 (IC11)** (Shahab et al. 2011) dataset is inherited from previous ICDAR contest 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)** (Karatzas et al. 2013) 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 (Karatzas et al. 2015), Total-Text (Ch'ng and Chan 2017), and SCUT-CTW1500 (Liu et al. 2019) are the benchmark datasets used to evaluate the performance of our framework. In this dataset, most text images are curved and rotated, and of low quality:

- **ICDAR 2015 (IC15)** (Karatzas et al. 2015) 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 17,548 text instances.
- **Total-Text** (Ch'ng and Chan 2017) 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, and formal information. 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** (Liu et al. 2019) 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 10,751 cropped word images for testing. At least one curved text appears in each image. There are primarily arbitrary-shaped 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 (Zhang et al. 2007), WebMarket (Jund et al. 2016), Grocery Products (Girshick et al. 2014), and Freiburg Groceries Dataset (Santra and Shaw 2020) to train and test our proposed grocery product detection and recognition framework.

- **GroZi-120** (Zhang et al. 2007) 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. Image of dimension  $256 \times 256$  with 96 dpi.
- **WebMarket** (Jund et al. 2016) consists of 3153 shelf images of size  $2272 \times 1704$  with 96 dpi, 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** (Girshick et al. 2014) 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-

**Fig. 9** Illustration of bounding box annotation and its format



CAKE0081 - Notepad  
File Edit Format View Help  
0 0.248157 0.356880 0.437346 0.519656  
0 0.686732 0.389435 0.415233 0.574939

grained product templates. Image of dimension  $348 \times 348$  with 96 dpi.

- **Freiburg Groceries Dataset** (Santra and Shaw 2020) 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. Image of dimension  $256 \times 256$  with 96 dpi.

## 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, 96 GB 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 LabelImg 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 1 shows the parameters used to train the proposed framework. 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  $1 \times 10^{-2}$ . 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 does not 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 used 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 used the Adam optimizer as an optimization algorithm for the text detection algorithm; the learning rate is set to be  $1 \times 10^{-3}$  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.

## 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 is 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 termed accuracy. It is useful for evaluating a classification model built only from balanced datasets. If the dataset for classification is skewed

**Table 3** 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	$1 \times 10^{-2}$	$1 \times 10^{-3}$
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

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. Equations (6) to (9) provide the formulae for evaluating all of these metrics.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (7)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

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.

## Results and Discussion

We propose a framework to perform three main tasks: object detection, object text detection, and object text recognition. The YOLOv5 algorithm is used as our object detector. 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 texts 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 support the proposed framework by detecting and recognizing the on-and off-the-shelf grocery products.

## Grocery Product Detection Result

The performance of grocery product detection is given in Table 2 and Table 3. 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 120 classes. YOLOv5 dramatically improves the detection performance by more than 10% (see Table 2), and it can detect both small and large grocery products (see Fig. 10). YOLOv5 provides the most promising for other datasets such as WebMarket, Grocery Products, and Freiburg Groceries Dataset, as shown in Table 2 and Table 3.

Geng et al. (Geng et al. 2018) 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, we used a

**Table 4** Comparisons of retail product detection performance with existing methods on the GroZi-120 and WebMarket dataset

Method	GroZi-120			WebMarket		
	Precision (%)	Recall (%)	F1 score (%)	Precision (%)	Recall (%)	F1 score (%)
George et al. (George and Floerkemeier 2014)	13.2	43.0	20.2	–	–	53.3
Merler et al. (CHM) (Merler et al. 2007)	17	15	15.9	–	–	–
Merler et al. (SIFT) (Merler et al. 2007)	18	72	28.8	–	–	52.8
Merler et al. (Adaboost) (Merler et al. 2007)	17	15	15.9	21.3	36.3	26.8
Geng et al. (VGG16) (Geng et al. 2018)	50.4	30.7	38.2	46.8	35.7	40.5
Geng et al. (VGG16 + ATmap <sup>BRISK</sup> ) (Geng et al. 2018)	46.3	29.5	36.0	49.2	52.4	50.7
Geng et al. (VGG16 + ATmap <sup>SIFT</sup> ) (Geng et al. 2018)	49.1	29.4	36.7	44.9	57.3	50.3
Franco et al. (BoW) (Franco and Maltoni 2017)	45.7	46.3	46.0	–	–	65.6
Franco et al. (DNN) (Franco and Maltoni 2017)	45.2	52.7	48.7	–	–	71.1
Ray et al. (Ray et al. 2018)	–	–	40.10	–	–	67.8
Marder et al. (HOG) (Marder et al. 2015)	–	–	28.3	–	–	43.0
Marder et al. (BoW) (Marder et al. 2015)	–	–	26.8	–	–	55.2
Girshick et al. (Girshick et al. 2014)	–	–	40.9	–	–	72.0
Zhang et al. (Zhang et al. 2007)	–	–	31.7	–	–	49.2
Santra et al. (Santra and Shaw 2020)	–	–	44.8	–	–	75.5
Karlinsky et al. (Karlinsky et al. 2017)	62.6	–	–	–	–	72.1
Ciocca et al. (Ciocca et al. 2021)	68.4	–	–	71.2	66.8	68.9
Yilmazer and Birant (Yilmazer and Birant 2021)	75.3	67.7	71.3	74.3	71.4	72.8
Santra et al. (Santra et al. 2022)	80.3	73.7	76.9	70.4	68.4	69.4
Leo et al. (Leo et al. 2021)	72.1	70.2	71.1	66.3	70.3	68.2
Machado et al. (Lima Machado et al. 2021)	61.8	54.3	57.8	45.4	58.3	51.0
Domingo et al. (Domingo et al. 2022)	70.2	68.3	69.2	66.4	70.4	68.3
Olondriz et al. (Olondriz et al. 2021)	67.3	66.5	66.9	71.0	63.6	67.1
Bukhari et al. (Bukhari et al. 2021)	78.4	67.3	72.4	79.6	75.6	77.5
<b>Proposed (YOLOv5)</b>	<b>86.3</b>	<b>77.8</b>	<b>77.0</b>	<b>89.4</b>	<b>88.2</b>	<b>86.3</b>

The bold values represent the possible best value in each column

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 F1 score (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. (Franco and Maltoni 2017) and Marder et al. (Marder et al. 2015) use DNN and HOG approaches, respectively, to detect the products of the WebMarket dataset and achieved an F1 score of 46% and 28.33%.

Santra et al. (Santra and Shaw 2020) achieved the F1 score of 80.21% which is the second highest in the Grocery Products dataset. Ray et al. (Ray et al. 2018) and Karlinsky et al. (Karlinsky et al. 2017) were able to achieve a satisfying result with F1 score of 76.20% and 79.05%. Franco et al. (Franco and Maltoni 2017) and Marder et al. (Marder et al. 2015) use the BoW approach for product recognition, achieving the F1 score of 69.30% and 59.91%. Girshick et al. (Girshick et al. 2014) performed semantic segmentation to segment the products from the background which was able to achieve a 78.99% of F1 score. However, the

YOLOv5 achieves the best performance on the Grocery Products dataset.

## 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 4. We adopt the most potent backbone network, ResNet50-FPN, to enhance 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, WHBBR, significantly reduces the false detection rate. We compare our backbone network with the Long et al. (Long et al. 2018) backbone (VGG16-FPN); our model achieves the best F1 score of 87.8% on SCUT-CTW1500 and 87.7% on Total-Text.

**Table 5** Comparisons of retail product detection performance with existing methods on the Grocery Products and Freiburg Groceries dataset

Method	Grocery Products			Freiburg Groceries		
	Precision (%)	Recall (%)	F1 score (%)	Precision (%)	Recall (%)	F1 score (%)
George et al. (George and Floerkemeier 2014)	23.5	43.1	30.4	23.8	—	—
Yörük et al. (Yörük et al. 2016)	57.0	41.6	48.1	—	—	34.7
Marder et al. (HOG) (Marder et al. 2015)	—	—	58.1	—	—	60.6
Marder et al. (BoW) (Marder et al. 2015)	—	—	59.9	—	—	56.9
Girshick et al. (Girshick et al. 2014)	—	—	79.0	72.4	68.4	70.3
Merler et al. (Merler et al. 2007)	—	—	51.2	67.9	45.6	54.5
Zhang et al. (Zhang et al. 2007)	—	—	58.4	45.6	56.3	50.4
Santra et al. (Santra and Shaw 2020)	—	—	80.2	85.7	—	—
Ray et al. (Ray et al. 2018)	—	—	76.2	77.4	—	—
Karlinsky et al. (Karlinsky et al. 2017)	—	—	79.1	80.3	77.8	79.0
Franco et al. (BoW) (Franco and Maltoni 2017)	73.7	65.4	69.3	72.3	68.1	70.1
Franco et al. (DNN) (Franco and Maltoni 2017)	73.9	54.7	62.9	76.4	69.4	72.7
Georgiadis et al. (Georgiadis et al. 2021)	53.1	—	—	66.4	—	—
Kumar et al. (RE) (Kumar et al. 2021)	65.3	68.9	67.1	86.5	—	—
Kumar et al. (SEN) (Kumar et al. 2021)	73.7	—	—	86.4	81.2	83.7
Ciocca et al. (Ciocca et al. 2021)	81.4	78.5	79.9	78.4	74.3	76.3
Yilmazer and Birant (Yilmazer and Birant 2021)	80.5	76.2	78.3	72.3	68.4	70.3
Santra et al. (Santra et al. 2022)	78.3	67.7	72.6	80.6	80.2	80.4
Leo et al. (Leo et al. 2021)	83.6	77.7	80.5	72.3	71.1	71.7
Machado et al. (Lima Machado et al. 2021)	86.5	83.7	85.1	78.5	74.3	76.3
Domingo et al. (Domingo et al. 2022)	76.4	81.1	78.7	89.5	83.4	86.3
Olondriz et al. (Olondriz et al. 2021)	74.5	78.9	76.6	77.3	73.6	75.4
Bukhari et al. (Bukhari et al. 2021)	84.8	85.3	85.0	76.3	68.6	72.2
<b>Proposed (YOLOv5)</b>	<b>92.1</b>	<b>86.8</b>	<b>83.3</b>	<b>89.6</b>	<b>91.5</b>	<b>90.5</b>

The bold values represent the possible best value in each column

**Fig. 10** Visualization results of text detection model on the Grozi 120 dataset



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 4. Similarly, the performance of our model on ICDAR 2015 dataset is compared with existing methods for detecting the oriented text. The proposed text detection model based on the WHBBR technique (F1 score: 90.0%) performs better than Long et al. (Long et al. 2018) (F1 score: 82.6%) with an improvement of 7.4% and meets the current

state-of-the-art performance on ICDAR15. Table 4 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, where horizontal, orientated, and curved text appears simultaneously in most images.

Our detection model's performance (F1 score: 89.1%) improves dramatically when the fully annotated ICDAR2011 training set is used. Similar to ICDAR 2013 and ICDAR 2015, our text detection model outperforms Long et al.

**Table 6** Comparisons with existing methods on text detection datasets

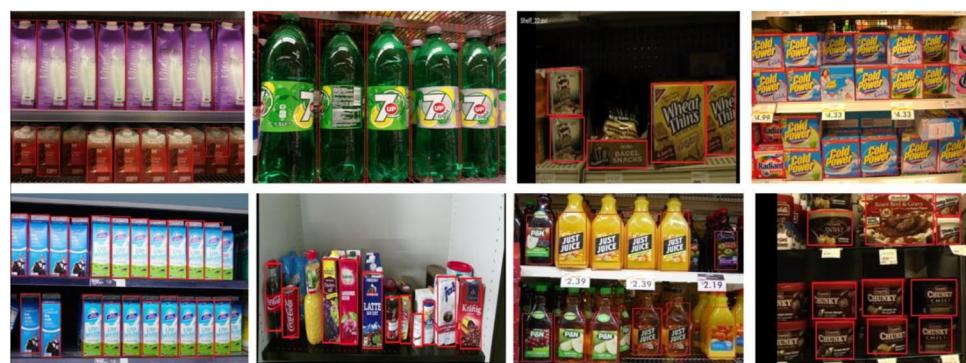
Method	ICDAR 2011			ICDAR 2013			ICDAR 2015			Total-Text			SCUT-CTW1500		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
Neumann et al. (Neumann and Matas 2013)	85.4	67.5	75.4	84.3	72.4	77.9	88.4	76.5	82.0	73.5	76.9	75.2	67.3	45.3	50.4
Huang et al. (Huang et al. 2014)	88.0	71.0	78.0	82.1	80.4	81.2	78.9	76.5	77.7	84.5	80.6	82.5	68.9	65.6	66.4
Liao et al. (Liao et al. 2017)	89.0	82.0	86.0	89.0	83.0	86.0	87.2	76.7	82.5	69.0	55.0	61.3	71.4	56.8	62.3
Yin et al. (Yin et al. 2014)	78.0	69.0	73.2	76.4	83.4	79.7	66.3	56.4	61.0	75.6	78.6	77.1	56.4	65.4	68.3
Jadeberg et al. (Jaderberg et al. 2016)	88.0	86.7	87.3	80.1	76.4	78.2	76.4	77.8	77.1	83.4	67.2	74.4	81.3	78.9	65.8
Busta et al. (Buta et al. 2015)	65.4	67.4	66.4	84.0	69.3	76.8	67.6	78.8	72.8	74.2	87.5	80.3	87.9	78.7	80.0
Shi et al. (Shi and Bai 2017)	78.3	67.2	72.3	87.7	83.0	85.3	89.5	78.9	83.9	30.3	23.8	26.7	42.3	40.0	40.8
Zhu et al. (Zhu and Du 2018)	73.2	70.2	71.7	90.0	72.0	80.0	80.2	84.2	82.2	78.3	81.3	79.8	77.6	75.4	76.5
Mohanty et al. (Mohanty et al. 2018)	67.5	71.2	69.3	89.0	77.0	83.0	78.6	81.2	79.9	67.7	77.2	72.1	74.5	78.9	76.6
Zhong et al. (Zhong et al. 2016)	81.2	84.5	82.8	87.0	83.0	85.0	82.1	88.5	85.2	81.2	80.3	80.7	82.3	84.5	83.4
Tian et al. (Tian et al. 2017)	76.4	77.4	76.9	91.1	83.1	86.9	76.7	83.5	80.0	75.6	78.5	77.0	83.4	81.0	82.2
Jiang et al. (Jiang et al. 2017)	65.7	82.1	73.0	93.5	82.6	87.7	85.6	79.7	82.5	87.5	86.5	87.0	77.9	74.5	76.2
Zhang et al. (Zhang et al. 2019)	87.3	81.4	84.2	81.2	84.5	82.8	87.8	87.6	87.7	88.6	75.7	81.6	89.2	69.6	78.4

**Table 6** (continued)

Method	ICDAR 2011			ICDAR 2013			ICDAR 2015			Total-Text			SCUT-CTW1500		
	Precision	Recall (%)	F-score (%)	Precision	Recall (%)	F1-score (%)									
Long et al. (Long et al. 2018)	78.9	78.6	78.7	76.5	82.1	79.2	84.9	80.4	82.6	82.7	74.5	78.4	67.9	85.3	75.6
Jianqi et al. (Jianqi et al. 2018)	87.3	86.4	86.8	73.4	79.6	76.4	82.2	73.2	77.4	73.4	69.4	71.3	77.5	73.3	82.3
Zhou et al. (Zhou et al. 2017)	76.9	76.4	76.6	81.6	80.5	81.0	83.3	78.3	80.7	50.0	36.2	42.0	78.7	49.1	60.4
Xu et al. (Xu et al. 2019)	90.2	82.6	86.2	71.0	82.1	76.1	84.3	83.9	84.1	81.2	79.9	80.6	83.0	79.8	81.4
Liao et al. (Liao et al. 2019)	78.5	77.4	77.9	67.4	77.4	72.1	91.6	81.0	86.0	70.1	69.0	71.2	79.9	76.6	83.4
Baek et al. (Baek et al. 2019)	82.1	80.1	81.1	75.3	67.8	71.4	89.8	84.3	86.9	87.6	79.9	83.6	87.6	79.9	83.6
Wang et al. (Wang et al. 2019)	83.4	74.3	78.6	83.4	84.5	83.9	69.1	73.2	65.4	81.8	75.1	78.3	80.6	75.6	78.0
Liu et al. (Liu et al. 2019)	82.3	73.3	77.5	86.4	78.9	82.5	74.8	77.0	72.7	70.1	67.3	73.2	77.4	69.8	73.4
<b>Proposed Method</b>	<b>91.6</b>	<b>86.7</b>	<b>89.1</b>	<b>94.4</b>	<b>88.3</b>	<b>91.2</b>	<b>92.2</b>	<b>87.9</b>	<b>90.0</b>	<b>89.1</b>	<b>86.3</b>	<b>87.7</b>	<b>88.4</b>	<b>87.2</b>	<b>87.8</b>

The bold values represent the possible best value in each column

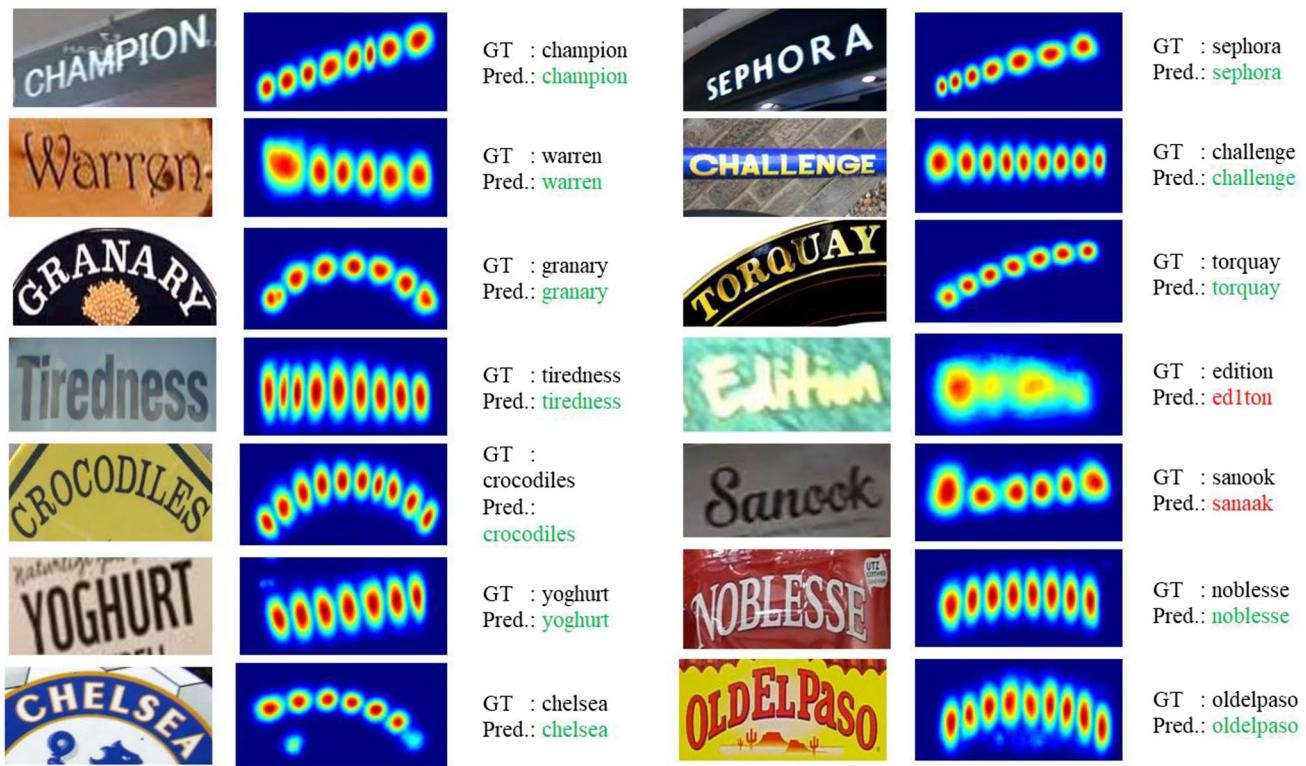
**Fig. 11** Visualization results of grocery product detection by the YOLOv5 algorithm on the public benchmark retail product datasets



**Fig. 12** Visualization results of text detection by the proposed model on the public benchmark text detection datasets



**Fig. 13** Visualization results of text detection by the proposed model on the public benchmark retail product datasets



**Fig. 14** The text recognition results on the public benchmark retail product dataset

(Long et al. 2018) by 13.3% and achieves current state-of-the-art performance on the Total-Text dataset. Our text detector achieves an 87.8% F1 score and the best recall of 87.2% outperforming most previous state-of-the-art methods. Our model achieves better than most of the existing methods, including Zhang et al. (Zhang et al. 2019) (precision: 89.2%) and Zhong et al. (Zhong et al. 2016) (recall: 83.6%), which are the top performers in irregular, although these methods have higher computational costs. The CTW1500 dataset has a complex background and includes a variety of multi-oriented texts. The proposed framework can detect multiple on-the-shelf retail objects (see Fig. 11), detect both word image (see Fig. 11) and object text (see Fig. 12), recognize irregular text accurately (see Fig. 14). Experiment results demonstrates that the proposed method is robust to both regular and irregular text.

### Significance of WHBBR

The importance of the modified backbone, centralizing, and the WHBBR technique is presented in Table 5. This proposed technique enhances the detection rate by an average of +2.3% compared to state-of-the-art methods.

On the irregular datasets, the modified backbone, centralizing, and the WHBBR technique achieved better performance than baseline model: precision (+7.3%), recall

(+7.5%), and F1 score (+7.4%) on ICDAR 2015 dataset; precision (+6.4%), recall (+11.8%), and F1score (+9.3%) on Total-Text dataset; precision (+17.5%), recall (+11.9%), and F1 score (+14.8%) on SCUT-CTW1500 dataset.

### Proposed Framework Summary

The proposed framework is proposed to detect and recognize on-shelf and off-shelf retail products by extracting text information including, product name, price, quantity, and expiry date from the product's particulars. In order to do that, we require an object detection, text detection, and text recognition model. We use a popular object detection model YOLOv5 to perform retail product detection. Individual object region coordinates such as  $(x_1, y_1)$  and  $(x_2, y_2)$  are passed to a text detection model to obtain product information. However, the text detection model follows an arbitrarily shaped bounding box construction approach to draw over the detected text, which failed to capture the starting and ending characters of the text. To address this problem, we proposed a WHBBR technique, which can draw a rectangular bounding box around the text accurately. It also helps the text recognition algorithm to predict the correct text. The WHBBR algorithm greatly improves the performance of the state-of-the-art methods, which follow arbitrary-shaped text reconstruction methods. The detected texts are cropped

**null 5** Significance of WHBBR technique with state-of-the-art methods

ICDAR 2015	Precision (%)	Recall (%)	F-measure (%)
Baseline	84.9	80.4	82.6
Modified backbone + WHBBR	85.2	80.7	82.9
Centralizing + WHBBR	86.0	81.2	83.5
Modified backbone + centralizing + WHBBR	92.2	87.9	90
Total-Text			
Baseline	82.7	74.5	78.4
Modified backbone + WHBBR	85.2	79.5	82.3
Centralizing + WHBBR	85.6	80.1	82.8
Modified backbone + centralizing + WHBBR	89.1	86.3	87.7
SCUT-CTW1500			
Baseline	70.9	75.3	73.0
Modified backbone + WHBBR	73.8	84.2	78.7
Centralizing + WHBBR	84.3	85.9	85.1
Modified backbone + centralizing + WHBBR	88.4	87.2	87.8



**Fig. 15** Product detection and product recognition are based on product text information

and passed to the text recognition model SCATTER (Litman et al. 2020), which recognizes the text from cropped word images. The proposed framework has many advantages such as assisting visually impaired people, reducing the time taken during checkout, identifying the number of on-shelf products, misplaced products, out-of-stock products, and so on. Figure 15 shows the overall result obtained after performing the series of steps. The proposed framework is a generalized model based on a deep learning algorithm; it can be used in any text-based applications, such as vehicle license plate recognition and road sign recognition. Object

text detection and recognition models are trained with only English texts. So the proposed framework is limited to only the English language.

## Conclusion and Future Work

We proposed a new framework composed of three models: 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) has unique information about the corresponding product. To acquire that precious text information from the retail products, we enhanced the state-of-the-art 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 fails. 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 color features. In addition, we intend to address the out-of-stock problem, product count, and misplaced items.

**Acknowledgements** We thank the editor and reviewers for their feedback and suggestions.

**Data Availability** All data generated or analyzed during this study are included in this published article.

## Declarations

**Ethics Approval** Not applicable.

**Consent to Participate** Not applicable.

**Consent for Publication** Not applicable.

**Conflict of Interest** S.Prabu declares that he has no conflict of interest. K. Joseph Abraham Sundar declares that he has no conflict of interest.

## References

- Advani S, Zientara P, Shukla N, Okafor I, Irick K, Sampson J, Datta S, Narayanan V (2017) A multitask grocery assist system for the visually impaired: smart glasses, gloves, and shopping carts provide auditory and tactile feedback. *IEEE Consum Electron Mag* 6(1):73–81. <https://doi.org/10.1109/MCE.2016.2614422>
- Baek Y, Lee B, Han D, Yun S, H. Lee G (2019) Character region awareness for text detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Long Beach, CA, USA. IEEE, pp 9365–9374. <https://doi.org/10.1109/CVPR.2019.00959>
- Beis JS, Lowe DG (1997) Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Proceedings of the computer vision and pattern recognition, San Juan, PR, USA. IEEE, pp 1000–1006. <https://doi.org/10.1109/CVPR.1997.609451>
- Bukhari ST, Amin AW, Naveed MA, Abbas MR (2021) ARC: a vision-based Automatic Retail Checkout system. arXiv preprint arXiv:2104.02832, pp 1–8. <https://doi.org/10.48550/arXiv.2104.02832>
- Buta M, Neumann L, Matas J (2015) FASText: efficient unconstrained scene text detector. IN: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile. IEEE, pp 1206–1214
- Ch'ng K, Chan CS (2017) Total-Text: a comprehensive dataset for scene text detection and recognition. In: Proceedings of the 2017 14<sup>th</sup> IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan. pp 935–942
- Ciocca G, Napoletano P, Locatelli SG (2021) Multi-task learning for supervised and unsupervised classification of grocery images. In: 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](https://doi.org/10.1007/978-3-030-68790-8_26)
- Domingo JD, Aparicio RM, Rodrigo LMG (2022) Cross validation voting for improving CNN classification in grocery products. *IEEE Access* 10:20913–20925
- Feichtenhofer C, Pinz A, Zisserman A (2017) Detect to track and track to detect, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp 3057–3065
- Franco D, Maltoni S (2017) Papi, Grocery product detection and recognition. *Expert Syst Appl* 81:163–176
- Geng W, Han F, Lin J, Zhu L, Bai J, Wang S, He L, Xiao Q, Lai Z (2018) Fine-grained grocery product recognition by one-shot learning, Proceedings of the 26th ACM international conference on Multimedia, pp 1706–1714
- George M, Floerkemeier C (2014) Recognizing products: a per-exemplar multi-label image classification approach, Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, Springer, pp. 440–455
- George M, Mircic D, Gabor S, Floerkemeier C, Mattern F (2015) Fine-grained product class recognition for assisted shopping. In: Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, pp 546–554. <https://doi.org/10.1109/ICCVW.2015.77>
- Georgiadis K, Zilos GK, Kalaganis F, Migkotzidis P, Chatzilari E, Panakidou V, Pantouvakis K, Tortopidis S, Papadopoulos S, Nikolopoulos S, Kompatsiaris I (2021) Products-6K: a large-scale groceries product recognition dataset, Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference, 2021, 1–7
- Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp 580–587
- Goldman E, Goldberger J (2020) Large-scale classification of structured image classification from conditional random field with deep class embedding. *Comput vis Image Underst* 191:1–11
- Gupta A, Vedaldi A, Zisserman A (2016) 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, pp 2315–2324
- Han W, Khorrami P, Paine TL, Ramachandran P, Babaeizadeh M, Shi H, Li J, Yan S, Huang TS (2016) Seq-NMS for video object detection, pp. 1–9, arXiv preprint <https://arXiv:1602.08465>
- He T, Huang W, Qiao Y, Yao J (2016) Text-Attentional convolutional neural network for scene text detection. *IEEE Trans Image Process* 25(6):2529–2541

- Huang W, Qiao Y, Tang X (2014) 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, vol 8692, Springer, Cham, pp 497–511
- Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading text in the wild with convolutional neural networks. *Int J Comput Vision* 116:1–20
- Jiang Y, Zhu X, Wang X, Yang S, Li W, Wang H, Fu P, Luo Z (2017) R2CNN: Rotational Region CNN for orientation robust scene text detection, arXiv 2017, pp. 1–8, arXiv:1706.09579
- Jianqi M, Shao W, Ye H, Li W, Wang H, Zheng Y, Xue X (2018) Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans Multimedia* 20(11):3111–3122
- Jund P, Abdo N, Eitel A, Burgard W (2016) The Freiburg Groceries Dataset, 2016, pp. 1–7, arXiv preprint arXiv:1611.05799
- Karatzas D, Shafait F, Uchida S, Iwamura M, Bigorda LG, Mestre SR, Mas J, Mota DF, Almazán JA, de las Heras LP (2013) ICDAR 2013 Robust Reading Competition, Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, pp. 1484–1493
- Karatzas D, Bigorda LG, Nicolaou A, Ghosh SK, Bagdanov AD, Iwamura M, Matas J, Neumann L, Chandrasekhar VR, Lu S, Shafait F, Uchida S, Valveny E (2015) ICDAR 2015 competition on robust reading, Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, pp. 1156–1160
- Karlinsky L, Shtok J, Tzur Y, Tzadok A (2017) Fine-grained recognition of thousands of object categories with single-example training, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4113–4122
- Kumar M, Moser B, Fischer L, Freudenthaler B (2021) 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, vol 1479. Springer, Cham, pp 127–137
- Leo M, Carcagni P, Distante C (2021) 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
- Liao M, Lyu P, He M, Yao C, Wu W, Bai X (2019) Mask TextSpotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans Pattern Anal Mach Intell* 43(2):532–548
- Liao M, Shi B, Bai X, Wang X, Liu W (2017) Textboxes: a fast text detector with a single deep neural network. In: Proceedings of the 31<sup>st</sup> AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, California, USA. pp 4161–4167. <https://doi.org/10.1109/aaai.v31i1.11196>
- Lima Machado de A, Aires K, Veras R, Neto LB (2021) Grocery product recognition to aid visually impaired people. In: Anais do XVII Workshop de Visão Computacional. pp 94–99. <https://doi.org/10.5753/wvc.2021.18896>
- Litman R, Anschel O, Tsiper S, Litman R, Mazor S, Manmatha R (2020) SCATTER: selective context attentional scene text recognizer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA. IEEE, pp 11959–11969. <https://doi.org/10.1109/CVPR42600.2020.01198>
- Liu Y, Jin L, Zhang S, Luo C, Zhang S (2019) Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recogn* 90:337–345
- Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. *Int J Comput Vision (IJCV)* 128:261–318
- Long S, Ruan J, Zhang W, He X, Wu W, Yao C (2018) 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, pp 19–35
- López-de-Ipiña D, Lorido T, López U (2011) 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, pp 33–40
- Marder M, Harary S, Ribak A, Tzur Y, Alpert S, Tzadok A (2015) Using image analytics to monitor retail store shelves. *IBM J Res Dev* 59(23):3:1–3:11
- Merler M, Galleguillos C, Belongie S (2007) Recognizing groceries in situ using in vitro training data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA. IEEE, pp 1–8. <https://doi.org/10.1109/CVPR.2007.383486>
- Mohanty S, Dutta T, Gupta HP (2018) Recurrent global convolutional network for scene text detection. In: Proceedings of the 25<sup>th</sup> IEEE International Conference on Image Processing (ICIP), Athens, Greece. IEEE, pp 2750–2754. <https://doi.org/10.1109/ICIP.2018.8451058>
- Neumann L, Matas J (2013) On combining multiple segmentations in scene text recognition. In: Proceedings of the 2013 12<sup>th</sup> International Conference on Document Analysis and Recognition, Washington, DC, USA. IEEE, pp 523–527. <https://doi.org/10.1109/ICDAR.2013.110>
- Olónriz DA, Puigdevall PP, Palau AS (2021) FooDI-ML: a large multi-language dataset of food, drinks and groceries images and descriptions. arXiv preprint arXiv:2110.02035. pp 1–14. <https://doi.org/10.48550/arXiv.2110.02035>
- Qazi HA, Jahangir U, Yousuf BM, Noor A (2017) Human action recognition using SIFT and HOG method. In: Proceedings of the International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan. IEEE, pp 6–10. <https://doi.org/10.1109/ICICT.2017.8320156>
- Ray A, Kumar N, Shaw A, Mukherjee DP (2018) U-PC: unsupervised planogram compliance. In: Ferrar, V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11214. Springer, Cham, pp 598–613. [https://doi.org/10.1007/978-3-030-01249-6\\_36](https://doi.org/10.1007/978-3-030-01249-6_36)
- Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA. IEEE, pp 779–788. <https://doi.org/10.1109/cvpr.2016.91>
- Santra AK, Shaw DP (2020) Mukherjee, Graph-based non-maximal suppression for detecting products on the rack. *Pattern Recogn Lett* 140:73–80
- Santra B, Ghosh U, Mukherjee DP (2022) Graph-based modelling of superpixels for automatic identification of empty shelves in supermarkets. *Pattern Recogn* 127:108627
- Shahab A, Shafait F, Dengel A, ICDAR (2011) Robust Reading Competition Challenge 2: reading text in scene images. In: Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China. IEEE, pp 1491–1496. <https://doi.org/10.1109/ICDAR.2011.296>
- Shi B, Bai X, Belongie S (2017) Detecting oriented text in natural images by linking segments. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA. IEEE, pp 2550–2558. <https://doi.org/10.1109/CVPR.2017.371>
- Shivakumara P, Phan TQ, Lu S, Tan CL (2013) Gradient vector flow and grouping-based method for arbitrarily oriented scene text detection in video images. *IEEE Trans Circuits Syst Video Technol* 23(10):1729–1739

- 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
- Tsai SS, Chen D, Chandrasekhar V, Takacs G, Cheung NM, Vedantham R, Grzeszczuk R, Girod B (2010) Mobile product recognition. In: Proceedings of the 18<sup>th</sup> ACM international conference on multimedia, association for computing machinery, Firenze, Italy, pp 1587–1590. <https://doi.org/10.1145/1873951.1874293>
- Wang S, Zhou Y, Yan J, Deng Z (2018) Fully motion-aware network for video object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany. pp 542–557. [https://doi.org/10.1007/978-3-030-01261-8\\_33](https://doi.org/10.1007/978-3-030-01261-8_33)
- Wang W, Xie E, Li X, Hou W, Lu T, Yu G, Shao S (2019) Shape robust text detection with progressive scale expansion network, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, pp 9336–9345
- Xu Y, Wang Y, Zhou W, Wang Y, Yang Z, Bai X (2019) TextField: learning a deep direction field for irregular scene text detection. *IEEE Trans Image Process* 28(11):5566–5579
- Ye J, Huang L, Hao X (2009) Neural network based text detection in videos using local binary patterns, Proceedings of the 2009 Chinese Conference on Pattern Recognition, Nanjing, China, pp 1–5
- 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
- Yin XC, Yin X, Huang K, Hao HW (2014) Robust text detection in natural scene images. *IEEE Trans Pattern Anal Mach Intell* 36(5):970–983
- Yörük KT, Öner CB Akgül (2016) An efficient Hough transform for multi-instance object recognition and pose estimation, Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, pp 1352–1357
- Zhang X, Yang YH, Han Z, Wang H, Gao C (2013) Object Class detection: a survey. *ACM Comput Surv* 46(1):1–53
- Zhang Y, Wang L, Hartley R, Li H (2007) Where's the weet-bix?, Proceedings of the Asian Conference on Computer Vision, Springer, Tokyo, Japan, pp 800–810
- Zhang C, Liang B, Huang Z, En M, Han J, Ding E, Ding X (2019) 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, pp 10552–10561
- Zhao ZQ, Zheng P, Xu ST, Wu X (2019) Object detection with deep learning: a review. *IEEE Trans Neural Netw Learn Syst* 30(11):3212–3232
- Zhong Z, Jin L, Zhang S, Feng Z (2016) DeepText: a unified framework for text proposal generation and text detection in natural images, arXiv pp. 1–12, arXiv:1605.07314
- Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J (2017) East: an efficient and accurate scene text detector, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp 5551–5560
- Zhu Y, Du J (2018) Sliding line point regression for shape robust scene text detection, Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, pp 3735–3740

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.