AI-Driven Multi-Image Object Recognition and Intelligent Product Discovery System

Saniya R
Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India
USN: 1DB23CS186

Sneha C.K
Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India
USN: 1DB23CS212

Tanushree R Ganguli
Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India
USN: 1DB23CS226

Varsha N
Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India
USN: 1DB23CS233

Dr. Mohamed Najmus Saqhib
Assistant Professor
Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Bangalore, India

***Abstract*—Online shopping normally assumes that the customer can name what they want. That assumption breaks down when a product is unfamiliar, or when a single photograph shows several products at once. This paper presents a web application that takes an image, typed text, or spoken audio and turns whatever it recognises into product searches on Amazon and eBay. Four YOLO-based detection models find the objects in an image, and a custom fusion step, weighted by confidence and based on intersection over union (IoU), merges overlapping boxes of the same class into one. A convolutional neural network (CNN) trained for 20 epochs on selected categories of the Stanford Online Products dataset reaches a reported classification accuracy of 89%. An optical character recognition (OCR) module reads printed text in the image, such as brand names or model numbers, so the search queries can be more specific. The system is written in Python with the Flask framework and an HTML/CSS front end, and a single page shows the uploaded image, the annotated image, the detected objects, the extracted text and the retrieved product listings. The work suggests that joining detection, classification, text reading and marketplace search in one pipeline can make product identification easier for non-expert users.**

***Index Terms*—object detection, YOLO, detection fusion, convolutional neural network, optical character recognition, product discovery, e-commerce, Flask**

I. Introduction

Online marketplaces have become a main route for buying goods, yet their search boxes still expect the shopper to type the right words. A person who notices an unfamiliar gadget, an unusual piece of furniture, or a crowded scene full of objects may not be able to name any of them, and so cannot write a useful query. Visual search gets around this by using the photograph itself as the query, and deep learning has recently been applied to image recognition and recommendation in commercial settings [1]–[3]. Machine-learning-based image recognition is a mature area with broad surveys [4], [5]. The open question today is therefore less whether an image can be recognised and more how the recognition result can be turned into a practical shopping workflow.

A. Problem Statement

Three difficulties come up in practice. First, a photograph often contains more than one product, and a system that outputs a single label throws away useful information. Multi-object recognition has been studied for this reason [6]. Second, one detector may miss an object or draw several overlapping boxes around it, which weakens any search query built from its output. Combining the boxes of several detectors is a known way to gain robustness [7]. Third, appearance alone is sometimes not enough. Brand names, titles and model numbers printed on an item carry information that visual features may miss, which is why reading the text in the scene is worthwhile [8], [9].

B. Contributions

This paper describes a system built to address these difficulties. Its main contributions are:

- a multi-modal front end that accepts image, text and audio input;
- a detection stage that runs four YOLO-based model files and merges their outputs with a custom confidence-weighted, IoU-based fusion procedure, which reduces duplicate predictions;
- a CNN classifier trained on selected categories of the Stanford Online Products dataset [10], with a reported accuracy of 89%;

- an OCR module that extracts printed text from the image to enrich product queries; and
- integration with Amazon and eBay search, so that every recognised object leads to related listings, all presented through a Flask-based web interface.

The rest of the paper is organised as follows. Section II reviews related work, Section III describes the proposed system, Section IV covers the implementation, Section V presents the results and discussion, and Section VI concludes the paper.

TABLE I
DETECTION PARADIGMS REFERRED TO IN THIS WORK

Detector	Stage	Key idea
Faster R-CNN [11]	Two	Region proposal network shares convolutional features with the detection head
YOLO [12]	Single	Detection as one regression problem; real-time speed
YOLOv3 [13]	Single	Multi-scale prediction, an incremental refinement
SSD [14]	Single	Boxes predicted from several feature-map resolutions

II. RELATED WORK

A. Object Detection

Early deep detectors treated region proposal and classification as separate steps. Faster R-CNN introduced a region proposal network that reuses the convolutional features of the detection head, so proposals come at almost no extra cost [11]. Its proposals are built from anchor boxes of several scales and aspect ratios, and the detector then classifies and refines each proposed region. Single-stage detectors skip the proposal step. YOLO casts detection as one regression problem: the image is divided into a grid, and each grid cell predicts a few boxes together with their confidence scores and class probabilities in a single forward pass, which gives real-time speed [12]. YOLOv3 kept this design but added a deeper backbone and predictions at three scales, with independent logistic classifiers for the classes [13]. SSD also predicts boxes in one pass, but attaches default boxes of different shapes to several feature maps of different resolutions [14]. Table I compares these designs. We build on the YOLO family because response time matters in an interactive web application where the user waits after every upload.

B. Combining Detections

When several detectors look at the same image, their outputs overlap. Non-maximum suppression (NMS) deals with this by keeping the highest-scoring box in each overlapping group and discarding the rest. Weighted Boxes Fusion (WBF) takes a different route. It clusters overlapping boxes and replaces each cluster with one box whose coordinates are the confidence-weighted average of its members, so every model contributes to the final box [7]. The fusion step used here follows this general idea of confidence-weighted merging, but we wrote it as a custom procedure for our four-model setup.

C. Retail and Product Recognition

Object detection has been used in retail checkout, where products must be identified quickly from camera images [15].

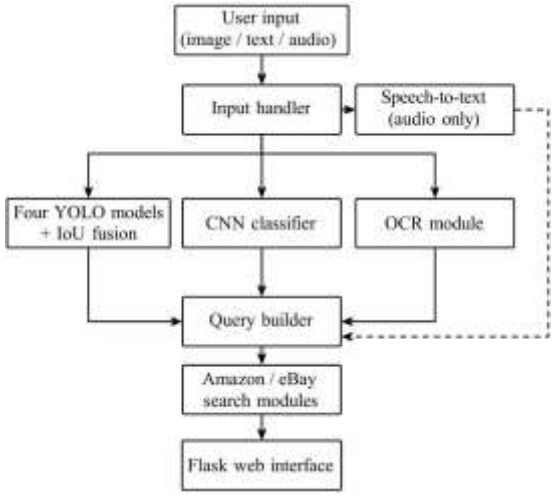


Fig. 1. High-level architecture of the proposed system. Image input flows to the detectors, classifier, and OCR module; typed text, and speech after conversion (dashed path), go directly to the query builder.

Deep learning has also been paired with product recommendation [1], and detection-based recognition has been proposed as a basis for image search [16]. Transfer learning has helped with e-commerce image classification when labelled data is scarce [2], and cross-modal retrieval has been explored as a way of linking product photographs to catalogue entries [3]. For clothing, large annotated collections such as DeepFashion support recognition and retrieval [17]. More recently, contrastive image–text pre-training on hundreds of millions of image–text pairs has made it possible to steer visual models with natural-language descriptions [18].

D. Recognition of Books and Other Objects

Work on library shelves has segmented and recognised individual books from deep visual features [19], and a multi-modal network has been used to predict the genre of a book from its cover image [20]. Surveys and studies of machine-learning-based image recognition and classification give the wider background for these methods [4], [5]. Convolutional approaches to recognising several objects in one image [6] are the closest early precedent for the multi-object setting addressed here.

E. Scene Text Reading

Text on packaging, labels or covers carries information that appearance cannot always supply. Scene-text pipelines usually separate text detection from text recognition. EAST is an efficient detector that predicts word- or line-level text regions directly from the image using a single network [8]. A convolutional recurrent network, in contrast, reads the character sequence inside a cropped region end to end, without segmenting individual characters and for sequences

of any length [9]. These two works are the reason OCR is used here as an extra source of information.

TABLE II
PIPELINE STAGES AND RELATED LITERATURE

Stage	Related work	Use in this system
Detection	[11]–[14]	Four YOLO-based models locate objects
Box merging	[7]	Custom confidence-weighted IoU fusion
Classification	[2], [10], [21]	CNN on Stanford Online Products subset
Text reading	[8], [9]	OCR cues added to queries
Retail / retrieval	[1], [3], [15], [16]	Object-level marketplace search

F. Backbone Networks

Residual connections made very deep CNNs practical to train, because each block learns a correction to its input instead of a complete transformation. Such backbones are widely used for image recognition [21]. Adapting a pretrained backbone to a new task is a common basis for product classification [2].

G. Positioning of This Work

Most earlier studies address a single stage of the problem: detection [11]–[15], classification [2], [21], text reading [8], [9], or retrieval [3], [16]. This work joins several stages in one web application: multi-model detection with fusion, CNN-based classification, OCR, and live marketplace search. Table II shows how each stage of the pipeline relates to the literature.

III. PROPOSED SYSTEM

A. Overview

The system takes a user input, runs it through the AI modules, turns the results into search queries, and shows everything on one page. Fig. 1 gives the data flow and Table IV lists the modules. We had two goals: a user who cannot name an item should still reach relevant listings, and a photograph that contains several items should produce several separate sets of results.

B. Multi-Modal Input

Three kinds of input are accepted. Images are the main input and go to the vision modules. Typed text is used directly as a search query. Audio is for users who would rather say the name of an item aloud; once the speech has been converted to text, it is handled exactly like typed input. Since typed text and recognised speech both reduce to a query string, the search modules downstream do not need to tell them apart.

C. YOLO-Based Multi-Object Detection

Every uploaded image is passed to four YOLO-based model files. Each model returns a set of detections, and each detection consists of a bounding box, a class label and a confidence score. Using several models raises the chance that an object missed by one model is found by another, but it also produces many overlapping boxes for the same object, which is why a fusion step is needed. We chose the YOLO family because single-stage detection offers real-time speed [12], [13], which suits an interactive web application.

D. Confidence-Weighted IoU Fusion

Let a detection be $d_i = (\mathbf{b}_i, c_i, s_i)$ with box $\mathbf{b}_i$, class c_i and confidence s_i. The overlap between two boxes is measured by the intersection over union,

$$\mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_j) = \frac{|\mathbf{b}_i \cap \mathbf{b}_j|}{|\mathbf{b}_i \cup \mathbf{b}_j|}. \tag{1}$$

Detections from all four models are pooled and sorted by confidence. A detection joins an existing cluster when it has the same class and its IoU with the fused box of that cluster exceeds a threshold τ, which is a tunable hyperparameter. For a cluster C, the fused box is the confidence-weighted average of its member boxes, computed coordinate by coordinate,

$$\hat{\mathbf{b}}_C = \frac{\sum_{i \in C} s_i \mathbf{b}_i}{\sum_{i \in C} s_i}, \tag{2}$$

The cluster is then reported as one detection with the fused box and an aggregated confidence, which we take as the mean of the member confidences,

$$\hat{s}_C = \frac{1}{|C|} \sum_{i \in C} s_i. \tag{3}$$

Because detections with higher confidence get larger weights, the fused box leans toward the more reliable predictions, and the output has one box per object instead of several near-duplicates. The procedure follows the principle of Weighted Boxes Fusion [7], adapted to the four models used here, and is summarised in Algorithm 1.

Algorithm 1 Confidence-Weighted IoU Fusion

Require: Detections $D = \{d_i = (\mathbf{b}_i, c_i, s_i)\}$ pooled from four models; threshold τ
Ensure: Fused detections F
Sort D by s_i in descending order
$C \leftarrow \varnothing$ {list of clusters}
for each $d_i \in D$ **do**
  find a cluster $C \in C$ with $c_C = c_i$ and $\mathrm{IoU}(\hat{\mathbf{b}}_C, \mathbf{b}_i) > \tau$
  if such a cluster exists **then**
    add d_i to C and recompute $\hat{\mathbf{b}}_C$ by (2)
  else
    create the cluster $\{d_i\}$ and add it to C
  end if
end for
for each cluster $C \in C$ **do**
  append $(\hat{\mathbf{b}}_C, c_C, s_C)$ to F, with s_C from (3)
end for
return F

Complexity: If n detections are pooled, sorting costs $O(n \log n)$ and the worst-case cluster search costs $O(n^2)$ IoU evaluations. The number n is limited to the boxes that the four models emit for one image, which is small compared with the cost of running the four networks themselves, so fusion adds little to the response time.

Relation to WBF: In the original WBF, the score of a cluster is the average of its member scores, which is then rescaled by the ratio of the number of boxes in the cluster, capped at the number of models M, to M: $s' = s_C \cdot \min(|C|, M)/M$ [7]. The rescaling lowers the score of an object that only a few models agree on. The plain mean in (3) leaves out this correction, so an object found by a single model keeps that model's confidence. Adding the correction is a possible refinement of the present procedure.

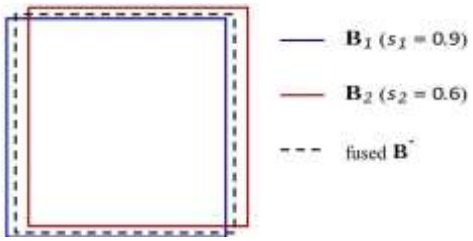


Fig. 2. Illustrative fusion of two same-class boxes. The fused box (dashed) lies nearer to the higher-confidence box. Values are illustrative.

Worked example: Take two detections of the same class from two different models, with illustrative (not measured) values $\mathbf{b}_1 = (100, 100, 200, 200)$, $s_1 = 0.9$ and $\mathbf{b}_2 = (110, 105, 210, 205)$, $s_2 = 0.6$, given as (x_1, y_1, x_2, y_2) in pixels. The intersection is $90 \times 95 = 8550$ px^2 and the union is $10000 + 10000 - 8550 = 11450$ px^2, so IoU ≈ 0.747. For any τ below this value the two boxes form one cluster. By (2) the weights are $0.9/1.5 = 0.6$ and $0.6/1.5 = 0.4$, which gives the fused box (104, 102, 204, 202), closer to the more confident box. By (3), the aggregated confidence is 0.75. Fig. 2 illustrates this case.

E. CNN-Based Classification

Alongside detection, a CNN was trained on selected categories of the Stanford Online Products dataset [10]. The dataset was released together with a deep metric-learning method. It holds product images collected from an online marketplace and grouped into broad super-categories (Table III). Residual backbones are widely used for image recognition [21], and transfer learning is a common way to adapt pretrained networks to commerce data [2]. The model was trained for 20 epochs and reached a reported classification accuracy of 89%. Its predictions help assign product-level categories to the objects found in the image. Classification accuracy is the fraction of test images whose predicted category matches the true category,

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{4}$$

TABLE III
STANFORD ONLINE PRODUCTS DATASET [10] AND ITS USE HERE

Item	Description
Source	Product images from an online marketplace
Scale (full dataset)	120,053 images of 22,634 products
Super-categories	12: bicycle, cabinet, chair, coffee maker, fan, kettle, lamp, mug, sofa, stapler, table, toaster
Used in this work	Selected categories only
Training	20 epochs
Reported accuracy	89%

F. Optical Character Recognition

The OCR module reads any text that appears in the uploaded image, such as brand names, titles or model numbers. The text is shown to the user and can be added to a search query to make it more specific, in line with earlier work on scene-text reading [8], [9]. Such methods usually split the task into locating text regions [8] and recognising the character sequence in each region [9]. The OCR stage of this system fills the same two roles and passes the recognised string to the query builder.

G. Query Builder and Product Retrieval

For each recognised object, the object name serves as the search query. When OCR returns useful text, it is added so that the query becomes more specific. The Amazon and eBay search modules return related product listings, which are collected for display. Because queries are formed per object, one photograph containing several items produces several separate sets of results.

TABLE IV
SUMMARY OF SYSTEM MODULES

Module	Role
Input handler	Accepts image, text, or audio from the user
YOLO detectors	Four model files that locate objects with bounding boxes
Fusion	Merges overlapping boxes by confidence-weighted IoU clustering
CNN classifier	Classifies product categories (Stanford Online Products subset)
OCR	Extracts text from the uploaded image
Shopping modules	Fetch related listings from Amazon and eBay
Web interface	Shows images, detections, text, and listings

IV. IMPLEMENTATION

A. Software Stack

The back end is written in Python with the Flask framework, and the front end uses HTML and CSS. The Flask application ties together the uploaded inputs, the detection and fusion code, the CNN classifier, the OCR module and the shopping-platform modules. Table IV summarises the modules.

B. Processing Workflow

The workflow runs in the following order (Fig. 3):

1) The user uploads an image, or provides text or audio.
2) The image is passed to the four YOLO models and to the OCR module.
3) The detections are fused to remove duplicates.
4) Object names, together with any useful extracted text, are turned into search queries.
5) The Amazon and eBay modules return related listings.
6) The web page displays the original image, the annotated image, the detected objects, the extracted text, and the listings.

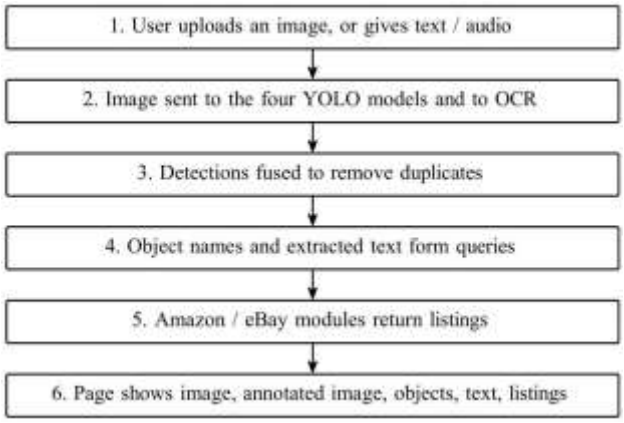


Fig. 3. Processing workflow of the application.

C. Web Interface

The interface is a single page that shows all results together (Fig. 4). The uploaded image and the annotated image are placed side by side, so the user can see directly which regions were detected. Below them the page lists the detected objects, the text returned by OCR, and the product listings retrieved for each object.

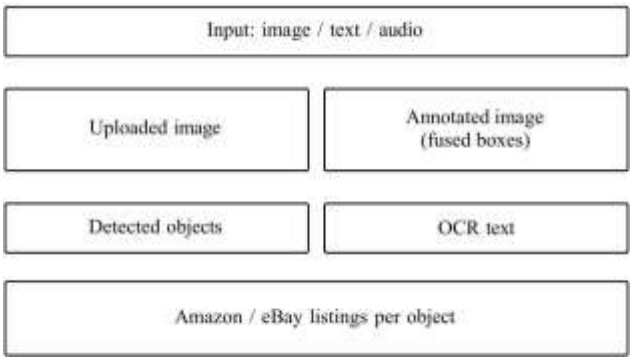


Fig. 4. Schematic layout of the result page (not a screenshot).

V. Results and Discussion

A. Classification Performance

The CNN trained for 20 epochs on the selected Stanford Online Products categories reached a reported classification accuracy of 89%. This figure applies to the chosen subset of categories and not to every product class sold online, so it should be read as an indication of performance on that subset only. The dataset was also designed for metric learning and retrieval [10], so an accuracy computed over a classification split cannot be compared directly with retrieval results reported for the same dataset.

B. System Output

The interface shows the uploaded image next to its annotated version, in which each fused detection is drawn as a labelled bounding box. Below them the page lists the detected objects, the text returned by OCR, and the product listings retrieved for each object. Seeing these items together lets the user check that the recognised objects match what was intended before opening a listing.

C. Effect of Fusion

Pooling four detectors produces several boxes for the same object. The fusion step replaces each group of overlapping same-class boxes with a single box, which cuts down duplicate predictions and keeps the number of search queries equal to the number of distinct objects. This matters for retrieval, because every duplicate box would otherwise trigger a redundant query to the shopping platforms. A quantitative comparison of the fused output against each individual model, for example precision and recall on a labelled test set, is left for future evaluation.

D. Planned Evaluation Protocol

To complete the evaluation, the metrics in Table V are planned. Precision, recall, and the F_1 score are

$$P = \frac{TP}{TP + FP}, \tag{5}$$

$$R = \frac{TP}{TP + FN}, \tag{6}$$

$$F_1 = \frac{2PR}{P + R} \tag{7}$$

where a detection counts as a true positive when its IoU with a ground-truth box of the same class exceeds a chosen threshold. Reporting these values for every individual model and for the fused output on the same labelled set would show directly what fusion contributes.

TABLE V
PLANNED EVALUATION METRICS (FUTURE WORK)

Component	Planned metric
Detection and fusion	Precision, recall, and F_1 per model versus the fused output
CNN classifier	Accuracy (reported 89%) and per-category accuracy
OCR	Fraction of text correctly read, by image clarity
Retrieval	Relevance of the top listings for each object
Response time	End-to-end time per uploaded image

E. Limitations

Retrieval quality depends on the object name used as the query, so a wrong label leads to unrelated listings. Detection accuracy may fall for small, occluded, or unusual objects. OCR accuracy depends on image clarity and text size.

Finally, the availability and format of results from external shopping platforms are outside the control of the system. Table VI pairs each limitation with a direction for improvement taken from the future work below.

TABLE VI
LIMITATIONS AND POSSIBLE DIRECTIONS

Limitation	Possible direction
Wrong label gives unrelated listings	Show detections to the user for confirmation; use image–text embeddings [18]
Small or occluded objects	Quantitative evaluation of fusion; more diverse training data
Limited CNN categories	Extend the CNN to more product categories
Unclear text for OCR	Use text only when it is legible
External platform changes	Support additional shopping platforms

VI. CONCLUSION AND FUTURE WORK

This paper presented an AI-driven application that identifies multiple objects in user images and links them to product listings. The system combines four YOLO-based detectors with a custom confidence-weighted IoU fusion step, a CNN classifier with a reported accuracy of 89% on selected Stanford Online Products categories, an OCR module, and Amazon and eBay search integration, all delivered through a Flask web interface. By letting users start from a picture instead of a product name, the system makes product identification and online product discovery simpler.

Future work includes a quantitative evaluation of the fusion step against the individual models, extending the CNN to more product categories, applying image–text embeddings to retrieval [18], ranking listings by visual similarity, and supporting additional shopping platforms.

REFERENCES

[1] W. Diwu, R. Zhang, L. Feng, and Q. Mu, "Image recognition and product recommendation algorithms based on deep learning," in *Proc. 3rd Int. Conf. Distrib. Comput. Electr. Circuits Electron. (ICDCECE)*, 2024, doi: 10.1109/ICDCECE60827.2024.10549427.

[2] S. Bhoir and S. Patil, "Transfer learning with deep neural networks for image classification in the e-commerce industry," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT)*, 2022, doi: 10.1109/I2CT54291.2022.9824903.

[3] C. Zhang and J. Feng, "Cross-modal product image retrieval for e-commerce recommendation systems via deep learning," *Traitement du Signal*, vol. 42, no. 4, pp. 2289–2299, Aug. 2025, doi: 10.18280/ts.420438.

[4] L. Liu, Y. Wang, and W. Chi, "Image recognition technology based on machine learning," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3021590.

[5] A. M., G. Parasa, et al., "Image recognition & classification with machine learning," in *Proc. 7th Int. Conf. Intell. Syst. Control (ISCON)*, 2025, doi: 10.1109/ISCON65210.2025.11341468.

[6] H. Shi, X. Mu, and S. Wang, "BVCNN: A multi-object image recognition method based on the convolutional neural networks," in *Proc. Int. Conf. Virtual Reality Vis. (ICVRV)*, 2015, doi: 10.1109/ICVRV.2015.28.

[7] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image Vis. Comput.*, vol. 107, Art. no. 104117, Mar. 2021, doi: 10.1016/j.imavis.2021.104117.

[8] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, doi: 10.1109/CVPR.2017.283.

[9] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017, doi: 10.1109/TPAMI.2016.2646371.

[10] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 4004–4012.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.

[14] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016. [Online]. Available: https://arxiv.org/abs/1512.02325

[15] V. M., P. B. Patil, D. K., and S. Mugada, "Real-time object detection for an intelligent retail checkout system," in *Proc. Int. Conf. Intell. Innov. Technol. Comput. Electr. Electron. (IITCEE)*, 2025, doi: 10.1109/IITCEE64140.2025.10915293.

[16] T. A. Nayeem, S. M. Motaharuzzaman, A. T. Hoque, et al., "Computer vision based object detection and recognition system for image searching," in *Proc. IEEE Int. Conf. Electr. Comput. Commun. Eng. (ICECE)*, 2023, doi: 10.1109/ICECE57408.2022.10089019.

[17] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1096–1104, doi: 10.1109/CVPR.2016.124.

[18] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[19] S. Zhou et al., "Library on-shelf book segmentation and recognition based on deep visual features," *Inf. Process. Manage.*, vol. 60, no. 4, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457322002023

[20] C. Kundu and L. Zheng, "Deep multi-modal networks for book genre classification based on its cover," arXiv:2011.07658, 2020.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.