Learning to Attribute Products to Prices in Retail Shelf Images

Anonymous Author(s)

ABSTRACT

Product-price attribution, the task of retrieving product pricing information directly from an image of a retail display, is critical for monitoring field execution in brick-and-mortar commerce. Poor price compliance, where displayed and point-of-sale prices diverge, can erode consumer trust, reduce revenue, and invite legal risk. Despite growing interest in AI-powered retail compliance solutions, product-price attribution remains underexplored. Existing methods for this task rely on brittle spatial heuristics, rigid shelf-structure assumptions, or high-resolution, close-up imagery that is expensive to obtain. Even state-of-the-art vision-language models (VLMs) struggle with the fine-grained spatial reasoning required. To address these challenges, we present `PriceLens`, an end-to-end system for product-price attribution that combines off-the-shelf object detection and OCR with `PriceNet`, a novel transformer-based association model. `PriceNet` learns to match detected products with price tags by modeling global spatial and semantic context, enabling robust parsing of visually complex retail displays. We introduce the first benchmark dataset for this task and show that `PriceLens` significantly outperforms heuristic, structural, and VLM baselines on challenging real-world display images. To support further research, we release our dataset to the academic community.

ACM Reference Format:

Anonymous Author(s). 2025. Learning to Attribute Products to Prices in Retail Shelf Images. In . ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Retail shelf analysis has emerged as a critical problem in machine learning and data mining. In brick-and-mortar retail, product availability, pricing accuracy, and shelf compliance are essential to operational efficiency and customer satisfaction. In 2024, the North American consumer packaged goods (CPG) retail market was valued at approximately $1.6–2.0 trillion [45]. This market, though sizable, is plagued by implementation errors. For example, a recent study demonstrated that over 90% of retailers reported having significant execution problems [7]. These challenges include incorrect product placement, missing or misaligned price tags, and discrepancies between the prices displayed on the shelf and those charged by the point-of-sale (POS) system. Each of these issues can directly impact revenue, consumer trust, and brand loyalty. In some cases, they have even led to costly lawsuits [37, 51].

In recent years, many retailers have deployed AI-powered monitoring solutions to improve execution. These systems regularly capture photos of store shelves to measure various aspects of compliance. Product-price attribution is the critically important task of determining the current list price(s) of each product from these shelf images. This task is typically performed by *detecting* products and price tags, *extracting* price information from each tag, and then *associating* located products with their corresponding prices.

Significant progress has been made in developing models for the sub-problems of detection (e.g., localizing products and price tags on shelves and classifying detected products [3, 10, 42]) and extraction (e.g., retrieving pricing information from tags using Optical Character Recognition (OCR) techniques [21]). The association component, however, remains an understudied and open challenge. This gap is notable, since product-price attribution cannot be performed without a reliable mechanism for association. Existing methods for this task often rely on handcrafted heuristics. Some approaches assume that a product's corresponding price tag is located directly above or below it. Others simply assign the nearest detected tag based on Euclidean distance [36, 41]. Many industry systems attempt to infer the structure of retail shelves, identifying rows and columns before applying directional association rules [40]. While these methods offer intuitive baselines, they break down in real-world settings where shelf structures are irregular, occlusion is common, and product layouts often violate structural assumptions.

An alternative approach to product-price attribution might forgo explicit handling of each step detailed above and instead run end-to-end inference with a vision-language model (VLM). However, these models have not yet achieved the capacity for reliable performance on this task. Despite remarkable progress in grounding and visual question-answering, studies have repeatedly shown that VLMs struggle with compositional and relational spatial reasoning [2, 24, 39]. Figure 1 illustrates some of the challenges of product-price attribution in the wild.

Rather than rely on brittle rules or general-purpose VLMs, we propose to tackle attribution through a learnable, end-to-end pipeline, which we call `PriceLens`. Our approach consists of three core components: 1) a detection step to identify products and price tags using off-the-shelf object detectors; 2) an extraction step to parse textual content from price tags; and 3) a probabilistic association step powered by `PriceNet`, a novel transformer-based neural network that predicts product-price pairings directly from a set of detections.

In contrast to existing methods for association, `PriceNet` does not rely on heuristics, but instead *learns* a diverse set of relevant spatial association patterns from data. At the core of `PriceNet` is an encoder-only transformer that consumes the coordinates of detected entities in a display and models the global context of the entire scene. We represent candidate product-price pairs as input tokens and predict an association probability for each, allowing the model to learn non-local, compositional structure. This model is trained on a new benchmark dataset, `BRePS` (**B**everage **Re**tail

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(a) Some products have multiple visible price tags, while others have none in the frame.

(b) The "nearest" price tag is not always the correct price tag.

(c) Display layouts vary widely from store to store.

(d) Price tags can be partially occluded by other objects in the scene.

Figure 1: A sample of the challenges that arise when performing product-price attribution in real-world retail settings.

Price Scenes), which we release to support future research in this important domain.

Our contributions can be summarized as follows:

1. We propose a learnable price attribution system. We introduce PriceLens, a general architecture for end-to-end price attribution. This system features PriceNet, a transformer-based model that learns product-price associations from spatial features, circumventing brittle heuristics common in prior work.

2. We achieve a sizable improvement over existing methods. Our experiments demonstrate that PriceNet significantly outperforms heuristic and structural baselines for product-price association, and that our end-to-end system, PriceLens, surpasses frontier vision-language models on a challenging collection of real-world shelf images.

3. We introduce a new benchmark dataset. In conjunction with this paper, we release BRePS, a new, densely-annotated benchmark dataset designed to drive and assess future progress in product-price attribution.

2 PRELIMINARIES

We consider the problem of *product-price attribution*, which can be defined as extracting a variable-length set, $\mathscr{A}$, of product-price tuples (also referred to as *attributions*) from an image, $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, of a retail display. Each tuple links a product to its visually suggested price.

We use $\mathscr{P}$ to refer to any product catalog of interest (e.g., all soft drinks sold at a specific convenience store), while $\mathscr{R}$ represents the set of all possible prices a product could be offered at (such as "$2.99", "4 / $5", or "Buy 2, Get 2 Free"). The *is-priced-at* relation, denoted by $\rho \sim r$, signifies that product ρ is sold at price r. We note that this relation refers exclusively to the *implied pricing structure* of a given display, as perceived by a typical human observer. It does not necessarily reflect the actual price charged for a product at the point of sale, which can differ due to poor price compliance within a store as mentioned in Section 1.

Using this notation, a set of product-price attributions can be formally expressed as $\mathscr{A} = \{(\rho, r) \mid \rho \in \mathscr{P}, r \in \mathscr{R}, \rho \sim r\}$. Our goal is to learn a mapping $f : \mathbb{R}^{H \times W \times C} \to 2^{\mathscr{P} \times \mathscr{R}}$. Given an image of a retail display, this function should output all product-price pairings that are visually suggested by that display.

To clarify our notation, we provide a simple example. Suppose that a store's product catalog is $\mathscr{P}$ = {"cola", "juice", "milk"}, and the set of possible prices is $\mathscr{R}$ = {"$1.00", "$2.00"}. If $\mathbf{X}$ is an image of a display where a price tag reading "$2.00" is prominently centered above shelves containing milk and juice, an oracle model would output $f(\mathbf{X})$ = {("milk", "$2.00"), ("juice", "$2.00")}. Note that in practice, the set $\mathscr{P}$ is typically not made up of human-readable names, but instead consists of universal product codes (UPCs), which are 12-digit numbers that uniquely identify a specific product.

A product-price attributor f must solve three related problems in order to output its final set of predictions: *detection*, in which products and price tags are located within an image, *extraction*, wherein structured price information is read from each price tag, and *association*, in which products and price tags are connected based on visual context.

3 RELATED WORK

Over the past decade, image-based retail shelf analysis has evolved into a significant area of focus, inspiring both industrial and academic research. This section reviews literature within this domain that is relevant to the product-price attribution problem.

3.1 Product Detection and Classification

Object detection and product classification are foundational to product-price attribution. Early attempts at applying end-to-end object detectors (e.g., Faster R-CNN, SSD, YOLO) in the retail setting largely failed due to the density of products on store shelves. Goldman et al. [10] address this challenge using a RetinaNet-like base detector augmented with a novel Soft-IoU layer to predict the Jaccard overlap for each box, and an EM-based clustering algorithm to merge highly overlapping detections. Their work includes the release of a popular benchmark dataset, SKU-110k, which contains bounding box annotations for over 11,000 densely-packed retail scenes. Follow-up work employs a Cascade R-CNN along with a random crop strategy to control for variation in the input scale, increasing mAP by nearly 10% [38]. Most recently, Xiao et al. [50] achieve state-of-the-art performance on SKU-110k through a "rectify and detect" pipeline that transforms oblique views into frontal views. In parallel, ongoing enhancements to the YOLO family of detectors [15, 19, 46, 49], including anchor-free detection, have

seen significant progress on dense retail scenes, proving capable of real-time or on-edge inference [43].

While many existing approaches perform detection and classification in one forward pass, others separate these tasks into two stages. Under this paradigm, a coarse detector first locates products in an image. These are then identified by a separate model that operates on each product crop. For example, Pietrini et al. [35] use `RetinaNet` for binary product detection, followed by a `MobileNetV3` for feature extraction with a `FAISS` [8] lookup step for final classification.

3.2 Price Detection and OCR

Locating price tags in shelf images and reading accurate pricing information from them is a specialized sub-problem within the larger product-price attribution paradigm. Sikic et al. [42] demonstrate that a `YOLOv8` model with data augmentation can locate price tags with 94% mAP. Once tags are detected, off-the-shelf OCR engines such as `EasyOCR` [18] can be applied to extract the price. Laptev et al. [21] use `EasyOCR` on a small dataset of price tag images and report 95.2% text recognition accuracy.

Recently, many studies have examined the efficacy of using VLMs for OCR. Lamm and Keuper [20] benchmark several commercial and open-source models, including `GPT-4V` [30], `GPT-4o` [31], and `LLaVA-NeXT` [25] on a collection of retail product images, observing acceptable performance on various extraction tasks. Nagaonkar et al. [29] find that VLMs outperform traditional OCR models when reading text from dynamic videos. Meanwhile, Chen et al. [4] train a custom VLM, `Ocean-OCR`, and report state-of-the-art results that improve on leading professional OCR models such as `TextIn` [16] and `PaddleOCR` [6].

3.3 Product-Price Association

Once products and price tags have been located (and labeled) in an image, they must be properly associated with one another. Despite its importance to the product-price attribution problem, this task has received relatively less attention in the academic literature and has been developed largely with an industry focus. Several methods have been proposed, which we detail below.

Heuristic Methods. Typical association heuristics include assuming a price tag is located directly below, directly above, or within an ϵ-neighborhood of a product [36, 41]. Additional geometric rules such as horizontal overlap between price and product bounding boxes can also be enforced [40]. Due to the complexity of real-world scenes, these methods are brittle and often result in poor performance.

Shelf-based, Structural Methods. These techniques seek to exploit the structure inherent in retail shelves (i.e., rows or sections) to associate price tags with products. One approach is to estimate rows via a Deep Hough transform [52], which can be paired with simple geometric heuristics. Several industrial products employ a similar process by detecting shelves, matching products to shelves, and associating products with the nearest price tags directly underneath the shelf [40]. Notably, in common settings with less explicit structure (e.g., full-pallet or promotional displays), shelf-assignment algorithms break down and association becomes significantly more difficult.

Text-based Methods. These methods seek to solve the association problem by reading product information from the price tag using OCR and comparing it to predicted product identities [28, 32]. If the information on the price tag matches the detected product, then the two are deemed to be associated. For such methods to be successful, two assumptions must be satisfied: uniquely-identifying product information must be present in each price tag, and price tag images must be sufficiently high-resolution for extracting fine-grained text details. In practice, we find these conditions are often violated, as price tag informativeness varies drastically from store to store and real-world images are typically captured at a distance (resulting in low-resolution price tag snapshots).

4 METHOD

We design an end-to-end pipeline to perform product-price attribution, which we call `PriceLens`. Our method consists of three main components: *detection*, *extraction*, and *association*. Figure 2 contains a diagram of the `PriceLens` architecture.

4.1 Detection

We assume access to a pre-trained detection module that can locate and classify products and price tags in a given image, such as those presented in [10, 38, 50]. In particular, we employ a `YOLOv8` [47] model that has been fine-tuned on a proprietary dataset containing millions of images of densely-packed beverage displays and 8,700 unique product identities. This model is made available to us through an industry collaboration.

4.2 Extraction

While many methods exist for extracting price information from a detected tag, we observe the best performance by utilizing VLMs (see Section 6.1). Specifically, we pass the image crop defined by a price tag's bounding box into a fine-tuned `Gemini-2.5-Flash` [5], prompting it to return a structured JSON output containing the indicated price and price type (e.g., standard, bulk offer, BOGO, etc.). For details about our fine-tuning procedure, see Section 4.5.

4.3 Association

After running detection and extraction as detailed above, we obtain two sets of labeled bounding boxes: one for products and one for prices. Let $\mathscr{D}_\rho = \{(\mathbf{b}_i, \rho_i)\}_{i=1}^n$ and $\mathscr{D}_r = \{(\mathbf{b}_j, r_j)\}_{j=1}^m$ denote the product and price tag detections, respectively, where each $\mathbf{b}_* \in [0,1]^4$ contains normalized bounding box coordinates in `xywh` format, and ρ_* and r_* are the corresponding product and price labels. These sets define nm possible product-price associations from which we must infer a plausible subset that aligns with the ground truth. For a given i, j, we construct a feature vector $\mathbf{x}_{ij} = \mathbf{b}_i \oplus \mathbf{b}_j$ and define a binary label $y_{ij} = \mathbf{1}\{\rho_i \sim r_j\}$, indicating whether the product and price contained in the respective bounding boxes truly associate with each other.

4.3.1 Spatial Context. Rather than predicting each y_{ij} in isolation, we condition its probability on all candidate associations in the scene, i.e., $\hat{p}_{ij} \triangleq p(y_{ij} \mid \{\mathbf{x}_*\}_{\forall i,j})$. This allows the model to use the

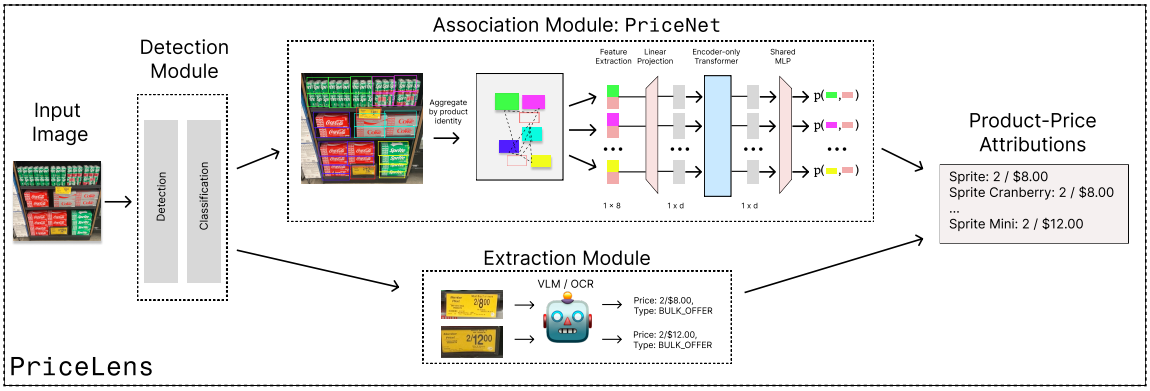


Figure 2: Overview of PriceLens, our end-to-end pipeline for price attribution. Our novel association module, PriceNet, aggregates candidate product-price associations by product identity, extracts spatial features from detected bounding boxes, and uses an encoder-only transformer to model global context and predict an association probability for each pair.

spatial context of the entire display when making predictions. We capture this dependency structure using an encoder-only Transformer without positional embeddings or causal masking (similar to the `Set Transformer` [22]). The encoder jointly processes the full set of candidate associations, and the resulting representations are passed through a shared MLP and sigmoid to predict $\hat{p}_{ij}$ for each pair. Given some threshold $\tau \in [0, 1]$, we then translate our predicted bounding box associations into a set of price attributions: $\hat{\mathcal{A}} = \{(\rho_i, r_j) \mid \hat{p}_{ij} > \tau\}$. We refer to this association network as `PriceNet` throughout the rest of the paper. The training procedure for `PriceNet` is detailed in Section 4.5.

4.3.2 Aggregation. When displays contain a large number of products and price tags, `PriceNet`'s self-attention incurs a high cost, with $O(n^2m^2)$ complexity. In these scenarios, we have also observed that prediction performance can degrade, potentially due to redundant detected product facings (e.g., repeated instances of the same item), which reduce the signal-to-noise ratio in the training data. To address this problem, we introduce an inductive bias: assuming all facings of a product on a display share the same price(s), we reduce the number of potential associations by only predicting on one product-price pair per unique product and price tag. We select the product detection whose centroid is nearest to the price tag (by Euclidean distance). This reduces the size of the candidate set to km (for k unique product identities), making our new complexity $O(k^2m^2)$, which can be much more tractable (especially if $k << n$). To obtain the full set of nm association probabilities, we simply propagate the predictions from each representative pair to all related associations. For a visual depiction of this process, see Figure 12 in Appendix B.

While we group products by identity (UPC) before predicting association probabilities, other groupings (e.g., by brand and packaging) may also be valid. We leave the specific choice of grouping to practitioners, but strongly recommend aggregating to reduce the set of candidate associations before prediction.

4.4 Overview

In summary, `PriceLens` obtains $\mathcal{D}_\rho$ and $\mathcal{D}_r$ from some image $\mathbf{X}$ using an object detector and price extractor. These sets define nm possible *associations*, which are pruned by `PriceNet` and combined with labels to form a final set of predicted *product-price attributions*, $\hat{\mathcal{A}}$. See Figure 2 for a visual depiction of our method.

4.5 Training Details

For the extraction task, we adapt `Gemini-2.5-Flash` [5] using supervised fine-tuning [14, 33]. This is performed through Google Cloud's Vertex AI service, which offers a tuning API [11]. During fine-tuning, the VLM repeatedly receives system instructions and an image of a price tag. Its expected output is simply a JSON representation of the true price contained in that tag. Our fine-tuning data is formed from the training split of BRePS, (Section 5), which contains labels for 18,648 price tags.

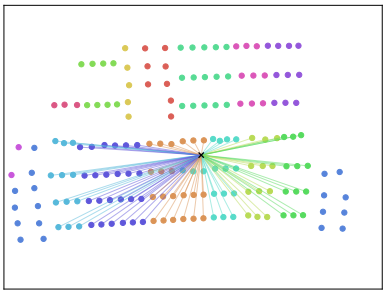
We train `PriceNet` on 3,904 annotated "price scenes", each containing product and price tag bounding boxes with ground-truth associations (see Section 5). The data is preprocessed into $(\mathbf{x}_{ij}, y_{ij})$ pairs as outlined in Section 4.3. We use focal loss [23] for our objective, and implement all training in PyTorch [34]. Code is available online [1]. See Appendix A.4 for full training details.

5 DATASET

In this section, we introduce BRePS (**B**everage **Re**tail **P**rice **S**cenes), a dataset of 4,881 annotated images that is used in all experiments. The data is canonically split into 3,904 training, 488 validation, and 489 test instances. Each scene includes dense annotations: a set of normalized `xywh` bounding boxes for products (with UPCs) and price tags, ground-truth product-price associations, and structured JSON price labels to support parsing of complex prices (e.g., "Buy 2, get 2 free"). Figure 3 shows a scene from this dataset (see also Appendix B). We present summary statistics for each data split in Table 1. BRePS is released for academic use under the PolyForm Noncommercial License [44] [2].

[1] https://anonymous.4open.science/r/price-net-E510
[2] The dataset will be made available upon publication.

(a) Display image, with color-coded bounding boxes indicating products and price tags in the scene.



(b) Depiction of the associative mapping between product and price tag bounding boxes that is implied by the display. For simplicity, only centroids are shown.

(c) Color key indicating the identities (UPCs) of all products in the scene.

Figure 3: Example of an annotated price scene from BRePS. All visible beverages are marked with a bounding box and labeled by identity (UPC). Price tags are also given a bounding box (outlined in white) and a structured price string indicating their contents. A mapping is stored for each scene that specifies which product and price tag bounding boxes are associated.

	Train	Validation	Test
# images	3,904	488	489
# distinct products	4,329	1,618	1,563
# price tags / image (IQR)	[1, 5]	[1, 5]	[1, 4]
# product facings / image (IQR)	[30, 180]	[34, 174]	[28, 168]
# unique products / image (IQR)	[2, 7]	[2, 7]	[2, 8]
# associations / image (IQR)	[19, 69]	[18, 69]	[19, 62]
% possible associations verified	32.6%	31.2%	34.5%

Table 1: Summary statistics for BRePS.

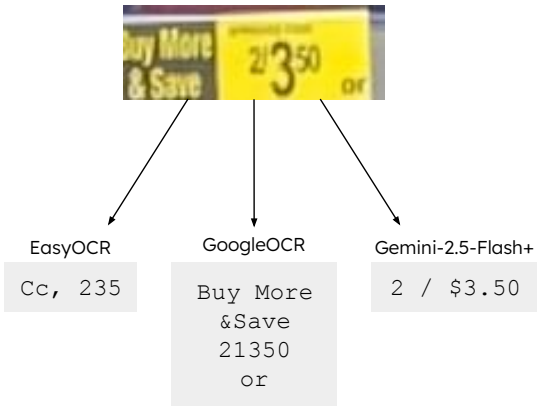


Figure 4: Outputs from different price extractors.

6 EXPERIMENTS

In this section, we seek to empirically validate our design of PriceLens. Given its modular structure, we first conduct isolated experiments on the extraction (Section 6.1) and association modules (Section 6.2). We then assess end-to-end performance (Section 6.3), followed by a few illustrative case studies (Section 6.4).

6.1 Extraction

We evaluate seven potential variants of the extraction module. Two are traditional OCR systems: the open-source Python package EasyOCR [18], and Google Cloud's commercial API [12] (which we term GoogleOCR). The remaining five are different iterations of Gemini models: 2.0-Flash, 2.5-Flash, 2.5-Pro, and fine-tuned versions of the Flash models (denoted with a trailing "+"). Fine-tuning details are provided in Section 4.5.

Each extractor is run on 2,112 labeled price tags from the test split of BRePS. Tags are cropped from display snapshots using bounding boxes, then fed to each extractor. Outputs are compared to the ground-truth labels using two metrics: exact-match accuracy and mean intersection over union (mIoU) of predicted vs. true tokens (using both word and character bi-gram tokenizations). Tokenization examples are shown in Figures 8 and 9 (Appendix B). Sample extractor outputs appear in Figure 4. Results are summarized in Table 2.

We observe that VLMs significantly outperform traditional OCR on price extraction. Many BRePS tags are low-resolution and cluttered with non-price text (e.g., sale dates, product info), which degrades OCR accuracy. OCR systems often misread currency symbols, mishandle decimal points, or interleave price digits with other text on the tag, compromising efforts to recover the true price. In contrast, VLMs, guided by our system instructions, are much better at ignoring irrelevant context and structuring their responses. This improves both accuracy and usability by reducing the need for extensive post-processing.

Gemini-2.5-Flash+, which was fine-tuned on BRePS price tags, performs best among VLMs (although this gain is not mirrored in Gemini-2.0-Flash+). However, despite promising results, overall accuracy remains below 90%, which could hinder reliability in deployment settings. We mark this as an open challenge for future work.

	Accuracy	mIoU (words)	mIoU (bi-grams)
EasyOCR	0.015	0.022	0.023
GoogleOCR	0.171	0.198	0.212
Gemini-2.0-Flash	0.793	0.803	0.832
Gemini-2.5-Flash	0.793	0.802	0.834
Gemini-2.5-Pro	0.797	0.805	0.832
Gemini-2.0-Flash+	0.794	0.802	0.831
Gemini-2.5-Flash+	<u>0.832</u>	<u>0.840</u>	<u>0.873</u>

Table 2: Quantitative comparison of various price extraction methods. We indicate the best performer in each metric with an <u>underline</u>. Our finetuned VLM, `Gemini-2.5-Flash+`, significantly outperforms all other approaches.

	Precision	Recall	F1
epsilon ($\epsilon = 0.4$)	0.502	0.714	0.590
nearest	0.755	0.533	0.625
nearest-below	0.768	0.355	0.485
nearest-on-shelf	0.756	0.377	0.504
nearest-per-group	0.776	0.548	0.642
nearest-below-per-group	0.800	0.454	0.580
text-based	0.608	0.589	0.598
PriceNet ($\tau = 0.5$)	<u>0.946</u>	<u>0.983</u>	<u>0.964</u>

Table 3: Association performance metrics. We indicate the best performer in each metric with an <u>underline</u>. `PriceNet` outperforms all baselines by a wide margin.

6.2 Association

In this section, we compare the performance of `PriceNet` to various heuristics drawn from prior work [28, 32, 36, 40, 41] (see Section 3). We also conduct a series of ablations to investigate the impact of specific modeling choices.

6.2.1 Comparison with Existing Methods. We evaluate `PriceNet` on the association task, comparing it against a range of heuristic methods:

- `epsilon`: Associates products with tags that are within an ϵ radius (ϵ is chosen by maximizing F1 on a holdout set).
- `nearest`: Associates each product with the closest price.
- `nearest-below`: Same as `nearest`, but restricts tags to those located below the product.
- `nearest-on-shelf`: Restricts to tags on the same shelf region, defined via Deep Hough lines [52].
- `nearest-per-group`: Groups product detections by identity (UPC) and assigns the price tag that is closest to a member of the group.
- `nearest-below-per-group`: Same as above, but only considers tags below the group.
- `text-based`: Uses `Gemini-2.5-Pro` [5] to extract text and match tags to products by name similarity.

As outlined in Section 4.3, we treat association as a binary classification problem on product-price candidate pairs. We measure precision, recall, and F1 on the test split of BRePS. Any relevant thresholds (ϵ for `epsilon`, τ for `PriceNet`) are tuned on the validation split (maximizing F1). We run three trials for `PriceNet` and `text-based` due to non-determinism and report the mean of each metric (other methods are evaluated once). Standard errors across runs are small and are listed in Appendix A.3. Results are shown in Table 3.

Our experiments clearly demonstrate the superiority of learnable association over predefined heuristics. `PriceNet` surpasses all baselines, achieving 18.3% higher precision, 37.7% higher recall, and 50.2% higher F1 than the next-best-method. Many factors contribute to its strong performance. We highlight a select few:

1. Many-to-many associations. Heuristics often enforce one-to-one or many-to-one mappings, which fail in real scenarios where products may have multiple prices (or no visible tag due to occlusion or poor compliance). `PriceNet` imposes no such restrictions.

2. Robustness to background noise. Price tags picked up in the background can mislead distance-based methods. `PriceNet` adapts to these scenarios by dynamically considering the context of the entire display.

3. Minimal assumptions. While the `text-based` method often underperforms due to inconsistent price tags and insufficient image resolution, `PriceNet` does not make implicit assumptions about the availability of fine-grained information at test time.

Unlike rigid heuristics, `PriceNet` learns from context, handles natural layout variability, and avoids common failure modes. See Section 6.4 for a few example predictions from this model.

6.2.2 PriceNet Ablation Studies. We evaluate three core design choices for our `PriceNet` architecture: jointly encoding all associations in a scene, identity-based candidate aggregation, and full bounding box featurization. To do so, we replicate the association experiments of Section 6.2 with various architectural configurations of `PriceNet`. In addition to precision, recall, and F1 (all measured with a common threshold $\tau = 0.5$), we report AUPR and AUROC by varying $\tau \in [0, 1]$. We train each model three times with seeds 0, 1, and 2. Table 4 reports the mean and standard error (in parentheses) of each metric across all ablations.

Joint vs. Marginal Prediction. `PriceNet` jointly encodes all candidate associations before predicting probabilities, incorporating scene-wide context. We compare this strategy to `PriceNet-marginal`, which considers each product-price pairing in isolation (see Appendix A.4 for more details). As shown in Table 4, joint modeling yields a substantial performance gain, highlighting the benefit of reasoning over all associations simultaneously.

Candidate Set Reduction. In Section 4.3, we describe how to group products by identity to reduce the number of candidate associations. We assess the utility of this step by comparing against `PriceNet-no-agg`, which does not aggregate the set of candidates. Results show that aggregation improves both efficiency and accuracy by shortening input sequences and emphasizing hierarchical priors. Figure 6 in Section 6.4 further illustrates this effect.

Feature Representation. `PriceNet` uses full bounding box features (`xywh`) for both products and price tags in its initial representations. Is it possible to achieve the same performance while only using centroid coordinates? We investigate by training `PriceNet-no-wh`,

	Precision	Recall	F1	AUROC	AUPR
PriceNet-marginal	0.816 (0.005)	0.657 (0.012)	0.728 (0.006)	0.882 (0.001)	0.830 (0.001)
PriceNet-no-agg	0.874 (0.005)	0.921 (0.011)	0.897 (0.007)	0.939 (0.002)	0.941 (0.007)
PriceNet-no-wh	0.925 (0.004)	0.985 (0.006)	0.954 (0.005)	0.936 (0.008)	0.986 (0.001)
PriceNet	0.946 (0.003)	0.983 (0.001)	0.964 (0.001)	0.950 (0.004)	0.988 (0.002)

Table 4: Results of ablation experiments with different configurations of PriceNet. We indicate the best performer in each metric (and any statistical ties) with an underline.

a variant of our model which only uses xy features. While the gap is smaller than other ablations, including wh information provides a modest performance boost. It is possible that these additional coordinates provide richer spatial context, allowing for more complex reasoning on irregular display layouts.

6.3 End-to-End Comparison with VLMs

	Precision	Recall	F1
Gemini-2.0-Flash	0.268 (0.003)	0.189 (0.001)	0.222 (0.002)
Gemini-2.5-Pro	0.466 (0.010)	0.398 (0.003)	0.429 (0.006)
GPT-4.1	0.243 (0.057)	0.190 (0.007)	0.212 (0.023)
GPT-4o-Latest	0.267 (0.005)	0.129 (0.004)	0.174 (0.003)
PriceLens	0.642 (0.005)	0.606 (0.006)	0.623 (0.003)
PriceLens-oracle	*0.858 (0.004)*	*0.833 (0.010)*	*0.845 (0.005)*

Table 5: End-to-end evaluation results comparing PriceLens with leading-edge commercial VLMs. To marginalize out known shortcomings from our detection and extraction modules, we also report results for the theoretical setting where these steps are assumed to be perfect (thus the association module is the only source of error). Excluding this "partial oracle" model, we indicate the best performer in each metric with an underline.

Finally, we test the full PriceLens pipeline against frontier vision-language models. In this experiment, we seek to validate that our decompositional approach is superior to the end-to-end inference offered by a modern VLM. Section 4 details the full sequence of steps performed by PriceLens to obtain product-price attributions from a shelf image. For VLMs, we provide the full display image along with detailed system instructions[3], and parse responses into a structured format for comparison.

For each method, we compute precision, recall, and F1 by comparing predicted attributions $\hat{\mathscr{A}}$ to the ground truth $\mathscr{A}$ on the BRePS test set. We mark each predicted tuple $(\rho_i, r_i) \in \hat{\mathscr{A}}$ as a true or false positive depending on its presence in $\mathscr{A}$, and count unmatched elements in $\mathscr{A}$ as false negatives. Both PriceLens and VLMs are evaluated over 3 runs, and we report the mean and standard error. Confidence-based metrics like AUROC and AUPR are omitted due to non-probabilistic VLM outputs. Results are listed in Table 5.

Despite strong general capabilities, current VLMs fall well short of PriceLens when applied to product-price attribution. This gap likely stems from needing to solve multiple sub-tasks—localization, identification, text extraction, association, and structured generation—without dedicated components or supervision. This is consistent with contemporaneous research showing that VLMs are bottlenecked by visual reasoning [13]. In contrast, PriceLens leverages specialized modules for each of these steps, achieving higher overall accuracy.

Current Limitations. While PriceLens outperforms VLMs, errors can still arise. Typical failure modes include missed price tags, product misclassification, and price reading errors (see Figures 13 and 14 in Appendix B). However, the association module remains robust, as demonstrated in Section 6.2. To control for the impact of upstream errors, we evaluate PriceLens-oracle, which accesses ground-truth detections and extractions but still uses PriceNet for association. This boosts F1 from 0.623 to 0.845, with similar improvements in precision and recall (Table 5), thus validating the associator's strength when given correct inputs[4]. Despite current limitations, PriceLens offers a general, modular architecture that allows practitioners to use detectors or OCR systems better suited to their specific application. Combined with the strong PriceNet associator, we believe this paves the way for reliable price compliance monitoring systems.

6.4 Case Studies

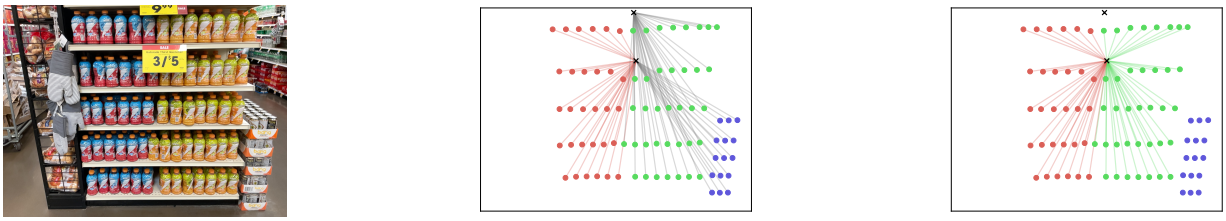
In this section, we qualitatively compare predictions from different associators and attribution pipelines to highlight the advantages of our proposed method.

Figure 5 contrasts PriceNet with the best-performing heuristic associator, nearest-per-group. The scene depicted in this figure mostly contains facings of two flavors of Gatorade, with a central price tag reading "3 / $5" that corresponds to these products. However, a pallet of Bang energy drinks is also present to the side of the main display, and a separate "$9.99" price tag is visible on the top shelf (its corresponding products are out of sight). Since nearest-per-group forces all products to match a price, it incorrectly associates the energy drinks with the Gatorade tag. Another association error is caused by the misleading spatial proximity of the "$9.99" tag to the lemon-lime Gatorades. Meanwhile, PriceNet reasons over the global layout and correctly ignores these out-of-context elements.

Figure 6 highlights the importance of identity-based aggregation for association (Section 4.3). The display in this figure contains two columns of Mountain Dew 12-packs on either side of a central stack of 20oz bottles. Only the left column of 12-packs has a price tag. Without aggregating by product identity, it is difficult to

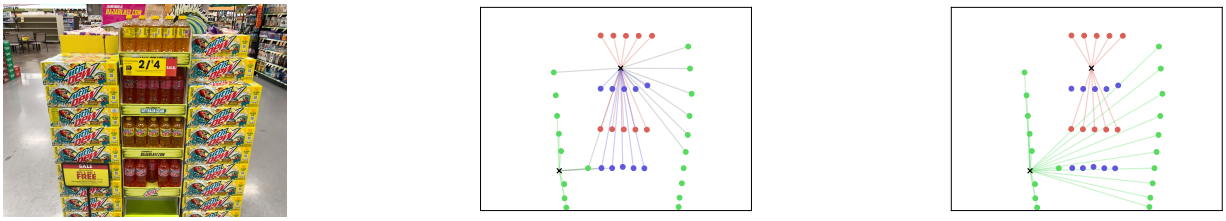
[3]All prompts can be found in our source code. See footnote 1.

[4]Note that metrics on the *attribution* task differ from *association*, since we are counting unique product-price pairs instead of individual bounding box links. This is why PriceLens-oracle's numbers differ from what is reported for PriceNet in Section 6.2.



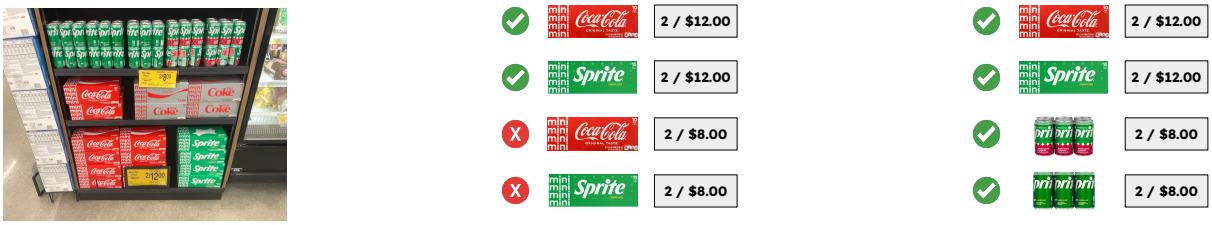
(a) Display image (b) `nearest-per-group` (c) `PriceNet`

Figure 5: The `nearest-per-group` association heuristic fails on this complex scene due to unrelated products and extraneous price tags. `PriceNet`, leveraging global layout cues, recovers the correct mapping.



(a) Display image (b) `PriceNet-no-agg` (c) `PriceNet`

Figure 6: Without identity-based aggregation, spatial features alone are insufficient to resolve associations in this multi-column display. Grouping by identity enables `PriceNet` to overcome this ambiguity.



(a) Display image (b) `Gemini-2.5-Pro` (c) `PriceLens`

Figure 7: `PriceLens` consistently outperforms VLM attributors by leveraging specialized components, including `PriceNet` for accurate product-price association.

infer that both columns share the same price, since spatially, the association between the right column of 12-packs and the leftmost price tag appears unlikely. `PriceNet` resolves this ambiguity, while `PriceNet-no-agg` fails to do so.

Finally, Figure 7 compares end-to-end attribution outputs from `PriceLens` and `Gemini-2.5-Pro` (the best-performing VLM). In this scene, despite perfect product/price recognition (we do not observe any misclassifications or price reading errors), `Gemini-2.5-Pro` erroneously connects the "2 / $8.00" price with Coke and Sprite 10-packs, overlooking the 6-packs on the top shelf that are truly sold at this number. In contrast, `PriceLens` outputs a fully valid attribution set. This can largely be ascribed to `PriceNet`, which is specifically trained to parse complex displays and is less prone to association errors.

7 CONCLUSION

In this paper, we proposed `PriceLens`, a modular architecture for extracting fine-grained pricing information from retail shelf images. We formalized the problem of *product-price attribution* and introduced a new dataset, BRePS, to support future research in automated price compliance monitoring.

At the core of our approach is `PriceNet`, a transformer-based associator that leverages global context to match products with price tags. Our experiments show that `PriceNet` significantly outperforms heuristic association methods, and that `PriceLens` as a whole exceeds the performance of frontier vision-language models. While detection and extraction remain potential bottlenecks, improvements in these areas could further enhance system reliability.

Overall, `PriceLens` represents a sizable step toward scalable, practical solutions for price compliance in brick-and-mortar retail.

REFERENCES

[1] Matthew Ashman, Cristiana Diaconu, Eric Langezaal, Adrian Weller, and Richard E. Turner. 2025. Gridded Transformer Neural Processes for Large Unstructured Spatio-Temporal Data. In *International conference on machine learning*.

[2] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14455–14465.

[3] Fangyi Chen, Han Zhang, Zaiwang Li, Jiachen Dou, Shentong Mo, Hao Chen, Yongxin Zhang, Uzair Ahmed, Chenchen Zhu, and Marios Savvides. 2022. Unitail: detecting, reading, and matching in retail scene. In *European Conference on Computer Vision*. Springer, 705–722.

[4] Song Chen, Xinyu Guo, Yadong Li, Tao Zhang, Mingan Lin, Dongdong Kuang, Youwei Zhang, Lingfeng Ming, Fengyu Zhang, Yuran Wang, et al. 2025. Ocean-ocr: Towards general ocr application via a vision-language model. *arXiv preprint arXiv:2501.15558* (2025).

[5] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).

[6] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. 2025. PaddleOCR 3.0 Technical Report. arXiv:2507.05595 [cs.CV] https://arxiv.org/abs/2507.05595

[7] Deloitte. 2016. *Retail Execution: The New Differentiator*. Technical Report. Deloitte. https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/consumer-business/retail-retail-execution.pdf Accessed: 2025-06-11.

[8] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss library. arXiv:2401.08281 [cs.LG] https://arxiv.org/abs/2401.08281

[9] Kunihiko Fukushima. 2007. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics* 5, 4 (2007), 322–333.

[10] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. 2019. Precise detection in densely packed scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5227–5236.

[11] Google. 2025. Tune Gemini models by using supervised fine-tuning. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini-use-supervised-tuning. Accessed: 23 July 2025.

[12] Google Cloud. 5. Vision AI: Image and visual AI tools. https://cloud.google.com/vision?hl=en. Accessed: 24 July 2025.

[13] Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. 2025. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444* (2025).

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[15] Muhammad Hussain. 2024. Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988* (2024).

[16] INTSIG. 2025. TextIn OCR Cloud Service. https://www.textin.ai/.

[17] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*. PMLR, 4651–4664.

[18] Jaided AI. 2020. EasyOCR: Ready-to-use OCR with 80+ Languages Supported. https://github.com/JaidedAI/EasyOCR. Accessed: 2025-06-11.

[19] Rahima Khanam and Muhammad Hussain. 2024. What is YOLOv5: A deep look into the internal features of the popular object detector. *arXiv preprint arXiv:2407.20892* (2024).

[20] Bianca Lamm and Janis Keuper. 2024. Can visual language models replace ocr-based visual question answering pipelines in production? a case study in retail. *arXiv preprint arXiv:2408.15626* (2024).

[21] Pavel Laptev, Sergey Litovkin, Sergey Davydenko, Anton Konev, Evgeny Kostyuchenko, and Alexander Shelupanov. 2022. Neural network-based price tag data analysis. *Future Internet* 14, 3 (2022), 88.

[22] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*. PMLR, 3744–3753.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[24] Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics* 11 (2023), 635–651.

[25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[26] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[27] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*.

[28] Emmanuel F Morán, Boris Xavier Vintimilla, and Miguel Realpe. 2023. Towards a Robust Solution for the Supermarket Shelf Audit Problem.. In *VISIGRAPP (4: VISAPP)*. 912–919.

[29] Sankalp Nagaonkar, Augustya Sharma, Ashish Choithani, and Ashutosh Trivedi. 2025. Benchmarking vision-language models on optical character recognition in dynamic video environments. *arXiv preprint arXiv:2502.06445* (2025).

[30] OpenAI. 2023. *GPT-4V(ision) Technical Work and Authors*. Technical Report. OpenAI. https://cdn.openai.com/contributions/gpt-4v.pdf

[31] OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/.

[32] ParallelDots. 2023. OCR and Image Recognition for Price Tag Detection: A Comparative Analysis. https://www.paralleldots.com/resources/blog/ocr-price-tag-detection-image-recognition Accessed: 2025-06-11.

[33] Aldo Pareja, Nikhil Shivakumar Nayak, Hao Wang, Krishnateja Killamsetty, Shivchander Sudalairaj, Wenlong Zhao, Seungwook Han, Abhishek Bhandwaldar, Guangxuan Xu, Kai Xu, et al. 2024. Unveiling the secret recipe: A guide for supervised fine-tuning small llms. *arXiv preprint arXiv:2412.13337* (2024).

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[35] Rocco Pietrini, Marina Paolanti, Adriano Mancini, Emanuele Frontoni, and Primo Zingaretti. 2024. Shelf Management: A deep learning-based system for shelf visual monitoring. *Expert Syst. Appl.* 255 (2024), 124635. https://doi.org/10.1016/j.eswa.2024.124635

[36] Product AI. 2021. Retail shelf image recognition — price and SKU data extraction. Medium. https://medium.com/product-ai/retail-shelf-image-recognition-price-and-sku-data-extraction-5a73ec19256e Accessed: 2025-07-24.

[37] Dennis Romero. 2024. Home Depot to Pay Nearly $2 Million to Settle Suit Alleging It Overcharged Shoppers. *CNBC* (18 September 2024). https://www.cnbc.com/2024/09/18/home-depot-to-pay-nearly-2-million-to-settle-suit-alleging-it-overcharged-shoppers.html

[38] Tianze Rong, Yanjia Zhu, Hongxiang Cai, and Yichao Xiong. 2020. A solution to product detection in densely packed scenes. *arXiv preprint arXiv:2007.11946* (2020).

[39] Julia Rozanova, Deborah Ferreira, Krishna Dubba, Weiwei Cheng, Dell Zhang, and Andre Freitas. 2021. Grounding Natural Language Instructions: Can Large Language Models Capture Spatial Information? *arXiv preprint arXiv:2109.08634* (2021).

[40] Ashutosh Saxena, Aadil Kazmi, Daniel D. Herrington, Mark A. Bowers, and Cameron E. Browne. 2016. Realogram scene analysis of images. https://patents.google.com/patent/US20160171429A1/en Assignee: Ricoh Company, Ltd..

[41] Bernd Schoner and Dimitri Granovskii. 2015. Shelf monitoring using image recognition. https://patents.google.com/patent/US20150262116A1/en Assignee: Trax Technology Solutions Pte Ltd.

[42] Franko Sikic, Branimir Filipovic, Zoran Kalafatic, Marko Subasic, and Sven Loncaric. 2023. Multi-Class Price Tag Detection in Images of Supermarket Shelves. In *2023 International Symposium on Image and Signal Processing and Analysis (ISPA)*. 1–6. https://doi.org/10.1109/ISPA58351.2023.10279709

[43] Jacob Solawetz. 2022. Retail Store Item Detection using YOLOv5. https://blog.roboflow.com/retail-store-item-detection-using-yolov5/ Accessed: 2025-06-11.

[44] The Polyform Project. [n. d.]. Polyform Noncommercial License, Version 1.0.0. https://polyformproject.org/licenses/noncommercial/1.0.0/.

[45] Towards Packaging. 2024. North America's CPG Industry Commands a $2 Trillion Valuation. https://www.towardspackaging.com/insights/consumer-packaged-goods-cpg-market-sizing Accessed: 2025-06-11.

[46] Ultralytics. 2021. YOLOv5: A state-of-the-art real-time object detection system. https://docs.ultralytics.com. Accessed: 2025-07-16.

[47] Rejin Varghese and M Sambath. 2024. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*. IEEE, 1–6.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[49] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. 2024. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems* 37 (2024), 107984–108011.

[50] Junhao Xiao, Yi Chen, Xiao Feng, Ruoyu Wang, and Zhiyu Wu. 2025. RecNet: Optimization for Dense Object Detection in Retail Scenarios Based on View Rectification. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[51] Marian Zboraj. 2023. Dollar General Pays Hefty Settlement in Wisconsin Over Pricing Discrepancies. *Progressive Grocer* (27 November 2023). https://progressivegrocer.com/dollar-general-pays-hefty-settlement-wisconsin-over-pricing-discrepancies

[52] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. 2021. Deep hough transform for semantic line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 4793–4806.

A PRICELENS—ADDITIONAL DETAILS

In this section, we provide additional details about our method, along with key clarifications and connections. This is intended to facilitate understanding and reproducibility of our `PriceLens` system.

A.1 Attribution vs. Association

In our problem formulation, we employ two related but distinct terms: *attribution* and *association*. *Attribution* refers to linking product identities (UPCs) with specific prices (e.g, "$2.99" or "Buy 1, Get 1 Free"). In contrast, *association* involves defining a mapping between product bounding boxes and corresponding price tag bounding boxes on a retail display. Attribution is downstream of association. Our proposed neural network `PriceNet` performs the association task, while the overall system `PriceLens` performs attribution.

A.2 Connection to Pseudo-Tokens

The reduction strategy we employ in our association module (Section 4.3) may be viewed as a special case of pseudo-token-based transformers [17], where instead of utilizing cross-attention with learned latents to reduce our token set, we hand-engineer an informative subset using subject matter expertise, similar to [1]. While a deeper analysis of this connection is not essential for motivating our method or demonstrating its efficacy, we highlight it to encourage further research.

A.3 Standard Errors from Associator Comparison Experiment

As mentioned in Section 6.2, both the `text-based` baseline and `PriceNet` are non-deterministic. Thus, in Table 3, we report the mean precision, recall, and F1 of these methods across three separate trials. The corresponding standard errors can be found below:

	Precision	Recall	F1
`text-based`	0.008	0.008	0.005
`PriceNet` ($\tau = 0.5$)	0.003	0.001	0.001

Table 6: Standard errors for non-deterministic associators.

A.4 Training Specifications

All `PriceNet` training runs were performed on a single machine with an NVIDIA GeForce RTX 3080 GPU (16 GB VRAM). Models were trained for 30 epochs with a batch size of 16 using the AdamW [27] optimizer, with linear warmup to a learning rate of 3e-4 during the first 3 epochs of training, followed by cosine annealing [26] until epoch 30. We set weight decay to 1e-5. We use `Focal Loss*` [23] as our objective, with $\beta = 0$ and $\gamma = 2$. For numerical stability, we clip logits to a maximum value of 10 before computing loss.

For our joint encoder, we project each 8-dimensional association candidate $\mathbf{x}_{ij}$ in a price scene to 256 dimensions ($d_{\text{model}} = 256$). We then pass the set of candidates through 3 stacked Transformer blocks [48], with 8 attention heads each and $d_{\text{feedforward}} = 512$. We set dropout to 0.3. As mentioned in Section 4.3, we do not use positional encodings or causal attention, since our method is permutation-invariant. After this encoding step, all representations are fed through the same multi-layer perceptron (layer widths `[256, 128, 64]`) and transformed into a 1-dimensional logit that can be converted into an association probability $p(y_{ij})$ via the sigmoid function. Between each intermediate layer, the MLP applies the ReLU activation [9] followed by dropout (with $p = 0.3$).

The `PriceNet-marginal` method introduced in Section 6.2.2 does not use a Transformer, but instead passes all association candidate representations directly to the MLP described above.

B SUPPLEMENTARY FIGURES

In this section, we provide additional figures to supplement the content in the main body of the paper.

Figures 8 and 9 illustrate how we compute the mIoU measures mentioned in Section 6.1. Figures 10 and 11 show more examples of annotated price scenes from our new benchmark dataset `BRePS` (Section 5). Figure 12 shows how our proposed aggregation scheme (Section 4.3) works. Figures 13 and 14 illustrate a handful of representative `PriceLens` failures that are caused exclusively by the detection and/or extraction modules (and not by `PriceNet`). See Section 6.3 for the related discussion.

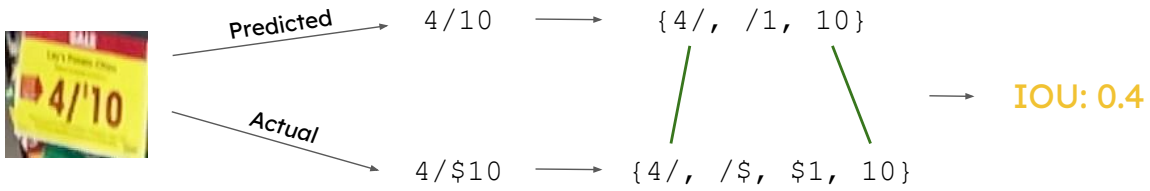


Figure 8: How we compute mIoU between an extracted and ground-truth price using character bi-grams as tokens.

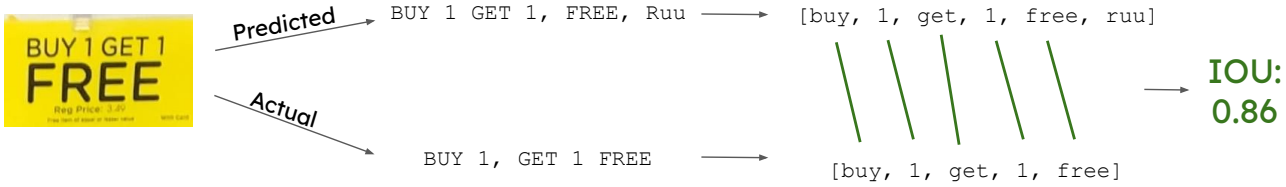
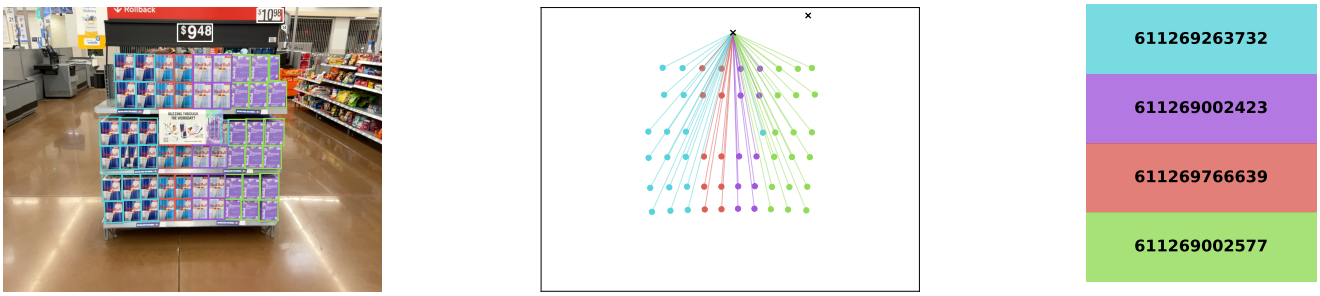
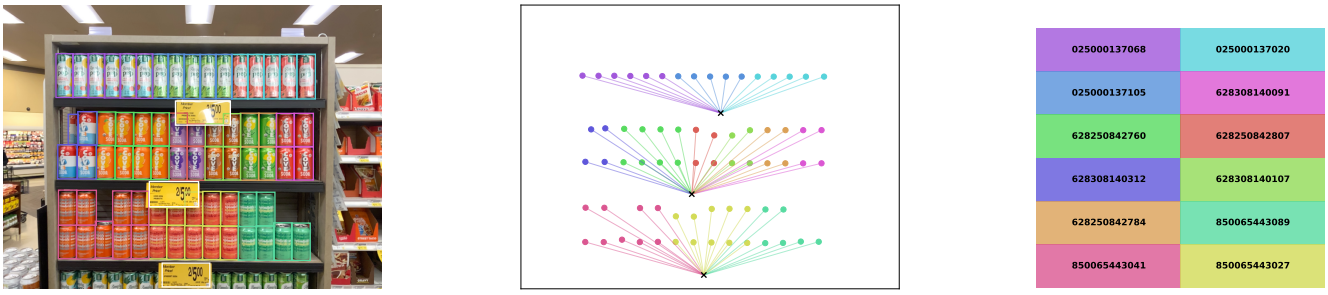


Figure 9: How we compute mIoU between an extracted and ground-truth price using individual words as tokens.



(a) Display image **(b) Product-price associations** **(c) UPC color key**

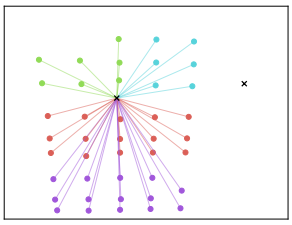
Figure 10: An annotated price scene from BRePS. Compare Figure 3.



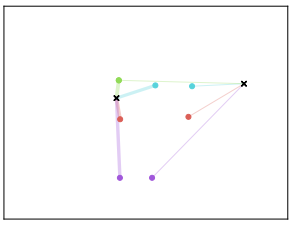
(a) Display image **(b) Product-price associations** **(c) UPC color key**

Figure 11: An annotated price scene from BRePS. Compare Figure 3.

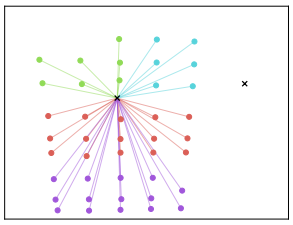
(a) Display image



(b) Ground-truth associations



(c) PriceNet's predictions on the reduced candidate set. Line width represents the predicted probability.



(d) PriceNet's final predicted associations.

Figure 12: A depiction of how PriceNet aggregates the set of candidate associations before making predictions (Section 4.3). For each price tag, the closest bounding box per product group is chosen as a representative. We then predict the probability that this product and price tag are associated, and propagate the result to all products in the group.

(a) Display image

(b) PriceLens attributions

Figure 13: Classification mistakes propagate to the attributions output by PriceLens. In this image, our detection module misidentified the 6-packs of "Ensure High Protein Milk Chocolate Bottles" as "Ensure Original Milk Chocolate Bottles". Despite correct bounding box association, this error corrupted the final attribution set.

(a) Display image. Detected price tags are marked with a green box. Note that the tag on the top shelf was not identified by our detection module.

(b) `PriceLens` attributions. The gray question mark indicates which ground-truth attributions were missing from the predicted set.

Figure 14: Mistakes from both detection and extraction can affect the final results. In this image, the detector ignores the top-most price tag, leading to missed attributions for all Prime energy drinks. The extractor then misreads the bottom-most price tag as "2 / $5.00" instead of "4 / $5.00". This leads to an attribution set that is invalid despite correct associations.