SERGI: Similar Entity Retrieval using Grouped Images

Akshit Sarpal*, Raviteja Uppalapati, Sayan Biswas, Rajesh N. Reddy and
Samrat Kokkula

Walmart Global Tech, 860 W California Ave, Sunnyvale, CA 94086

Abstract

Image data is frequently organized in semantic groups for entities such as e-commerce products, social media users or hotels on travel websites. There are a wide range of applications for retrieving entities based on their images, yet its exploration remains limited. Signals from images are commonly infused with other attributes in the form of embeddings, but purely leveraging groups of images for retrieval is relatively unexplored. Drawing inspiration from natural language literature, we developed an efficient and scalable method, SERGI (Similar Entity Retrieval using Grouped Images), for retrieving entities similar to given image groups. For practical implementation, we apply SERGI to an e-commerce use-case, aiming to identify products with brand misrepresentation. Despite the scarcity of benchmark methods for comparison, our system demonstrates superior performance compared to a baseline and a commonly used representation-based method, showing high precision in this relatively uncharted domain.

Keywords

Image retrieval, CLIP, grouped images, late-interaction

1. Introduction

Image retrieval, a process which involves searching and retrieving images from an extensive digital database, is an integral component of various digital solutions. Traditional methods typically rely on metadata, such as captions, keywords, or descriptions, to facilitate text-based searchability. An alternative approach is content-based image retrieval [1] (CBIR), which preferentially employs the visual content of an image over its metadata for search and retrieval. Instance-based image retrieval (IIR) is a more specific approach, which seeks to identify images from a database that depict the same object or scene as a reference image.

Despite the extensive research conducted on instance-based image retrieval within the computer vision community, retrieval of entities represented by collections of images remains a largely uncharted territory. In many real-world scenarios, entities are frequently represented by groups of images, such as product listings on an e-commerce website, hotel rooms or destinations on a travel portal, or user-visited locations on social media platforms. There are numerous practical applications that necessitate the identification of similar products, hotels, or places based on a given query. Existing methodologies attempt to amalgamate signals from an entity's images and metadata into embeddings, utilizing strategies such as

eCom'24: ACM SIGIR Workshop on eCommerce, July 18, 2024, Washington, DC, USA

*Corresponding author.

Akshit.Sharpal@walmart.com (A. Sarpal); Raviteja.Uppalapati@walmart.com (R. Uppalapati); sayan.biswas@walmart.com (S. Biswas); Rajesh.Narasimha_Red@walmart.com (R. N. Reddy); samrat.kokkula@walmart.com (S. Kokkula)

0009-0008-2907-3285 (A. Sarpal)

summing, averaging, or concatenating the feature embeddings (where a feature could be an individual image). However, these techniques encounter an array of challenges like variable or increased dimensionality resulting from concatenation, and potential loss of information, semantic meaning, and sensitivity to noise when summing or averaging embeddings.

In this study, we delve into the complex problem of entity retrieval based on groups of images. Here, 'entity' is a broad term used to denote a set of either query or reference images. Given the nascent state of research in the area of grouped image retrieval, our approach draws inspiration from methodologies employed in natural language processing for information retrieval. Consequently, we have developed a method we call SERGI (Similar Entity Retrieval using Grouped Images), and have applied it to a brand categorization use-case within the context of Walmart's e-commerce marketplace. We adapt the retrieval system to perform fine-grained visual categorization using signals across multiple images. The observations and insights from this application are subsequently discussed.

In an e-commerce marketplace setting, a product listing is composed of numerous characteristics. However, maintaining the accuracy of these attributes can be challenging when listings are managed by individual sellers. In this study, we focus specifically on brand name as an attribute. We observe that a notable percentage of listings can contain misrepresented brand names, which can lead to poor customer experiences and adversely impact sales. Consequently, we utilize our retrieval system to identify products with erroneous brand names and recommend appropriate corrections. The fundamental assumption of this work is that text attributes such as brand names are more prone to misrepresentation, whereas manipulating images is more complex. Image manipulation not only demands more time but can also adversely affect product sales since customers heavily rely on images when deciding to purchase. Therefore, images can be used as an anchor to validate text attributes.

We have constructed an instance of our method, SERGI, to ascertain whether a new product listing has been misbranded, utilizing groups of images for this purpose. Brands are categorized into two distinct segments in this study. The first segment encompasses brands that boast an established reputation and widespread recognition, referred to herein as 'trusted brands'. The second segment comprises lesser-known brands, designated as 'unverified brands'. For the purpose of this study, we have constructed a scalable index that stores unique representations of all item images associated with these trusted brands. In addition, we have established a real-time image retrieval system that persistently monitors all images derived from items linked to unverified brands. This system is engineered to identify entities (groups of product images) that demonstrate a substantial degree of resemblance with the indexed images of trusted brands. Any matches of this nature are flagged in real-time, and the corresponding trusted brand name is suggested as the correct brand name. This enhances the efficiency and accuracy of our brand verification process, thereby providing a robust solution to the challenge of brand misrepresentation. We call this use-case Brand Protection.

Our methodology employs a late-interaction architecture [2] to compare images from unverified brand items to those of trusted brand items. The unique contributions of this study, compared with previously published work, are as follows:

- **We propose a generalizable approach to conduct grouped matches.** Our research diverges from existing image retrieval methods, which typically define an entity as a single image. Instead, we center our work on image groups. This shift in focus presents its own set of unique challenges, which we will explore in-depth in this paper.

- **We have established a highly scalable content-based image retrieval system for e-commerce marketplaces**. Image retrieval has proven successful in various applications such as recommender systems [3], healthcare [4, 5], remote-sensing [6], and search engines [7]. Our work navigates the unique challenges associated with e-commerce catalog product images, like the presence of swatches (an image with a uniform pattern, such as the color of a dress), and generic images like placeholder images and nutrition labels which can be highly similar across unrelated products and lead to false matches. An instance of our proposed method, SERGI, was set up on e-commerce data, and the findings from this implementation are shared within this study.
- **We show a unique application of grouped image retrieval to detect products with misrepresented brands by adapting our retrieval system for fine-grained classification task.** Identifying products that are broadly similar to a target item, such as a blender, shoe, or toothpaste, is relatively straightforward. However, pinpointing products that belong to the same brand can be considerably more demanding. Products within the same brand often possess highly similar local features, as can be seen in multiple models of blenders from a leading kitchenware brand. Instead of striving for improvements in image-to-image matches using local features, our methodology employs groups of images for fine-grained visual categorization of products to a set of trusted brands.

We study the performance of SERGI using multiple sets of real e-commerce data from Walmart's marketplace and share the details of our deployment for the classification use-case. Our results underscore the effectiveness and scalability of our approach in detecting misrepresented brands. Beyond enhancing customer trust in e-commerce marketplaces, our solution offers a generalizable approach for other use-cases that require similarities between groups of images.

2. Related Work

The domain of Content-Based Image Retrieval (CBIR), an extensively researched area focusing on image matching, facilitates the retrieval of visually similar images from a specified database with respect to a user-provided query image [8]. This process, in essence, involves a user submitting a query image, following which, the system retrieves images from the database that bear a visual resemblance to the query image. Images can be represented through a variety of visual features such as color, texture, gradient, among others. However, the use of deep-learning based representations have proven to be superior to traditional feature descriptors [9].

The development of dense image representations benefits from a plethora of pretrained models. The use of Convolutional Neural Network (CNN) based architectures, including VGG [10], ResNet [11], Inception [12], and EfficientNet [13], has become widespread. Recently, transformer-based architectures have demonstrated superior performance, surpassing previous state-of-the-art models. Among these, CLIP (Contrastive Language-Image Pre-training) [14], a transformer-based machine learning model, has been instrumental in bridging the chasm between vision and language. CLIP is designed to understand and generate representations of images and their accompanying textual descriptions within a joint embedding space. During training, the model utilizes contrastive learning to associate images and texts, thereby maximizing the similarity between accurate pairs while minimizing the similarity between incorrect pairs. One of CLIP's defining features is its zero-shot capabilities, which underscores its ability to understand a broad spectrum of vision and natural language tasks without the need for finetuning. Consequently, CLIP embeddings are employed in this study to represent images.

For performing grouped similarity between the query product and the indexed products, we inspire our work from natural language based neural retrieval. In natural language application, there are broadly three types of matching paradigms for neural information retrieval: (a) representation-focused rankers that independently calculate query and document representations and calculate vector similarity [15]; (b) interaction-based rankers that model relationships across query and document tokens and match them using a neural network [16], or rankers that model interactions across and within tokens [17]; and (c) rankers based on late-interaction that delay the connection between query and document terms [2]. We adapt the highly efficient late-interaction architecture for comparing groups of images, which enables us to use precomputed candidate representations for image retrieval.

Previous research and applications in e-commerce are either focused on single image retrieval, such as [18] where only the primary product image is used; or retrieval using multimodal representations [19][20]. We explore retrieval in grouped image settings.

3. Methodology

In this section, we cover our SERGI method in detail. In context of image retrieval, a query is a user's image-based request to locate particular images from a database. In our generalized definition, a query is an entity represented by group of images. Retrieval refers to the process of searching for and obtaining entities that are coarsely similar to the query. We call these candidate entities, which represent a small and relevant subset from millions of entities. Reranking refers to the process of reordering the initially retrieved candidates based on their fine-grained relevance to the query. We first provide an overview of the image representations used, followed by image retrieval and the novel reranking approach. We conclude this section by summarizing the system design for Brand Protection use-case. Table 1 summarizes common annotations used in this section.

3.1. Image Representation

In traditional content-based image retrieval, a query image is provided to the system, and gets represented in form of a feature descriptor. In neural retrievers, the representation is computed as deep-learning based embeddings. We use Contrastive Language-Image Pre-training (CLIP) [14], which is a simplified version of the unsupervised strategy proposed in ConVIRT [21]. The model consists of two parts: a vision transformer and a text transformer which are trained to perform a contrastive prediction task. The model is trained to maximize the cosine similarity between an image and its corresponding text in the same minibatch while minimizing the cosine similarity with other texts or images in the minibatch.

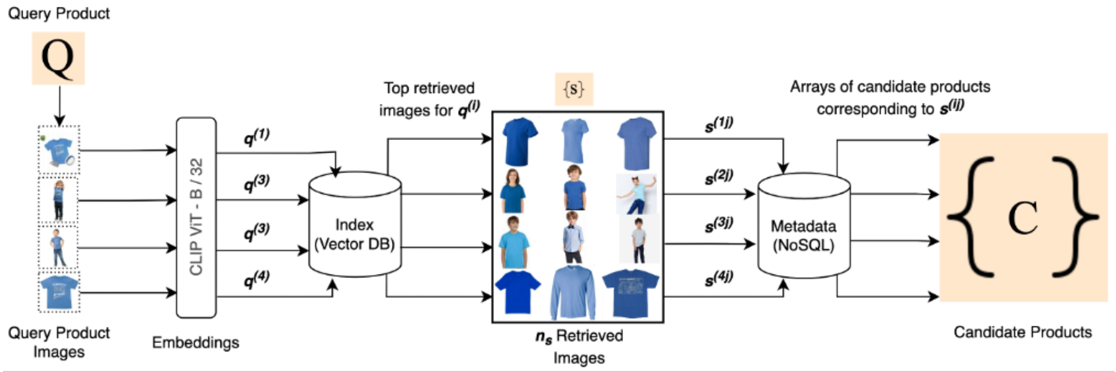

Figure 1: Product candidate generation process with an example. Each query product image is used to issue concurrent retrieval queries. The corresponding retrieved images are aggregated to get distinct candidate products.

Given its zero-shot abilities and the fact that it can be used to generate image embeddings that carry visual semantics, we use a pretrained CLIP checkpoint (CLIP ViT-B/32) to generate product image representations. Image representations/embedding are normalized vectors in a 512-dimensional space. We use $\mathbf{q}$ to describe the query image representation.

$$\mathbf{q}^{(\mathbf{i})} = \mathit{Embeddings}(\mathbf{img}^{(\mathbf{i})}) \quad \forall \quad i \in \{1, \ldots, \mathbf{n_q}\} \tag{1}$$

where $\mathbf{n_q}$ represents the number of images belonging to the query product $\mathbf{Q}$.

3.2. Candidate Generation

The representation is used to retrieve most similar images from an index (typically a vector database in modern applications) that uses approximate nearest neighbor search to return candidate images. Given a query product $\mathbf{Q}$ with $\mathbf{n_q}$ images, similar images corresponding to each query image are retrieved. This set of $\mathbf{n_s}$ similar images across all query images is denoted by $\{\mathbf{s}\}$. Using a metadata database, each $\mathbf{s}^{(\mathbf{ij})}$ is mapped to its product identifiers, and this set of candidate products is denoted as $\{\mathbf{C}\}$. Figure 1 displays the candidate generation process. We summarize the process symbolically below.

$$\mathbf{s}^{(\mathbf{ij})} = \mathit{Retrieve}(\mathbf{q}^{(\mathbf{i})}, \mathbf{I}, \mathbf{N}) \quad \forall \quad i \in \{1, \ldots, \mathbf{n_q}\}; j \in \{1, \ldots, \mathbf{n_r}\} \tag{2}$$

$$\{\mathbf{C}\} \leftarrow \mathit{ProductID}(\mathbf{s}^{(\mathbf{ij})}) \quad \forall \quad i \in \{1, \ldots, \mathbf{n_s}\}; j \in \{1, \ldots, \mathbf{n_r}\} \tag{3}$$

where $\mathbf{I}$ denotes the product index and $\mathbf{n_q}$ represents the number of images belonging to the query product $\mathbf{Q}$.

3.3. Reranking

The objective of reranking is to ensure that from all candidate items, the ones that are most visually similar get the highest similarity scores. Typically, neural re-rankers condition the

Table 1
Summary of notations.

Symbol	Description
$\mathtt{Q}$	Query item with $\mathbf{n_q}$ images
$\mathtt{C}$	Candidate item with $\mathbf{n_c}$ images
$\mathtt{C_{rep}}$	Array of padded candidate representations
$\mathtt{D}$	Array of query – candidate similarity matrices
$\mathtt{S}$	Array of calculated similarities
$\mathtt{q}$	Query image representations
$\{\mathtt{s}\}$	Set of retrieved similar images to $\mathbf{q}$
$\mathtt{n_s}$	Number of nearest neighbors retrieved
$\mathtt{n_q}$	Number of query images
$\mathtt{n_c}$	Number of candidate images
$\mathtt{n_r}$	Number of images retrieved per query image
$\mathtt{n_m}$	Maximum images across all $\mathbf{N_c}$ candidates
$\mathtt{N_c}$	Number of candidate products
$\{\mathtt{C}\}$	Set of candidate products/entities
$\mathtt{I}$	Index of products
$\mathtt{V}$	Default vector for padding
$\mathtt{d}$	Embedding dimension – 512 for our use-case

bulk of their computations on the joint query–indexed image pair, which can be expensive in practice. We leverage learnings from late-interaction based architecture and adapt an efficient re-ranker that achieves state-of-the-art for natural language, to our image use-case.

All images from the candidate product are queried from a low latency database. These images are denoted by $\mathbf{c}$, such that $\mathbf{c^{(ij)}}$ represents $\mathbf{i^{th}}$ image from $\mathbf{j^{th}}$ product. As the number of images can vary across products, a default vector, $\mathbf{V}$ is used to make each product in a batch uniform in dimensions. If $\mathbf{n_c}^{(j)}$ represents the number of images for the $\mathbf{j^{th}}$ candidate product, then all products with $\mathbf{n_c}^{(j)}$ lower than $(\mathbf{max(n_c}^{(j)}))$ among all $\mathbf{N_C}$ candidates are padded by $\mathbf{V}$. Maximum number of images across candidate products is represented by $\mathbf{n_m}$. The choice of $\mathbf{V}$ can be arbitrary as long as it guarantees highest distance from any possible image vector.

$$\mathbf{n_m} = max(\mathbf{n_c}^{(j)}) \quad \forall \quad j \in \{1, \dots, \mathbf{N_C}\} \tag{4}$$

$$\mathbf{C^{(j)}} = Pad(\mathbf{C^{(j)}}, \mathbf{V}, \mathbf{n_m} - \mathbf{n_c}^{(j)}) \quad \forall \quad j \in \{1, \dots, \mathbf{N_C}\} \tag{5}$$

$$\mathbf{C_{rep}} = Concat([\mathbf{C^{(j)}}]) \quad \forall \quad j \in \{1, \dots, \mathbf{N_C}\} \tag{6}$$

This enables us to marshal each candidate product $\mathbf{C^{(j)}}$ in set $\mathbf{\{C\}}$ as a 2-d array of size (d, n_m), where $\mathbf{d}$ represents the embedding dimension and is taken to be 512 for our use-case. Consequently, each $\mathbf{C^{(j)}}$ is concatenated to form a 3-d array of size (N_c, d, n_m) which represents the corpus of all candidate product representations and is denoted as $\mathbf{C_{rep}}$.

A Hadamard product between $\mathbf{Q}$ and $\mathbf{C_{rep}}$ is performed, and the resultant (N_C,n_q,n_m) array is denoted as $\mathbf{D}$. Each of the $\mathbf{N_C}$ matrices with dimensions (n_q,n_m) represents a similarity matrix

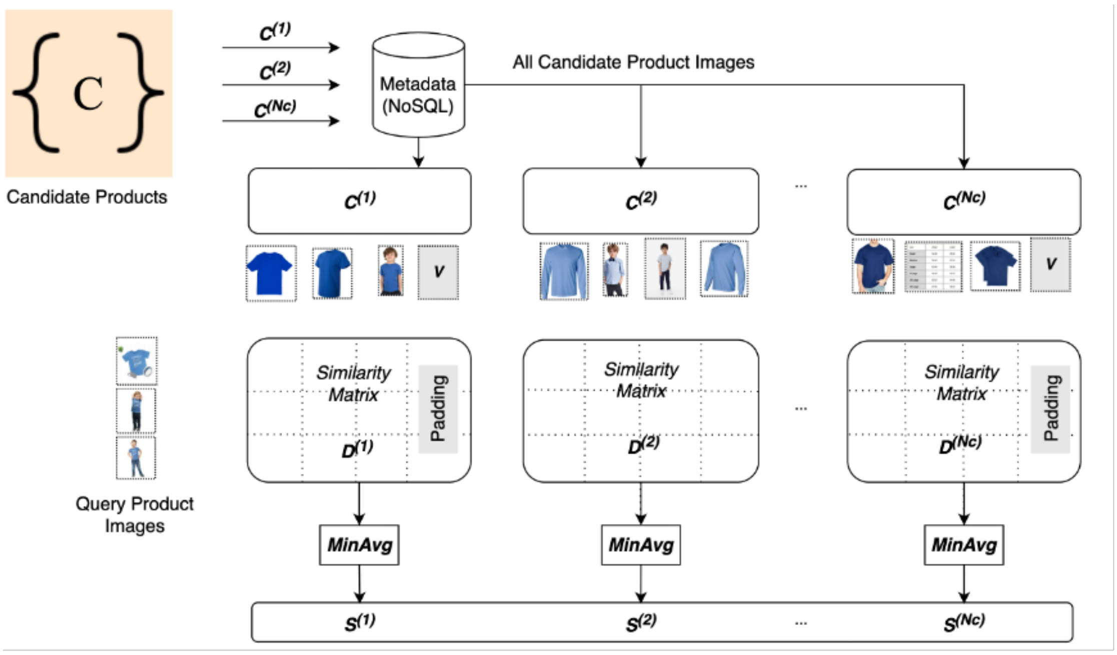

Figure 2: An illustration of the reranking methodology.

between **Q** and the corresponding candidate. Next, a *MinAvg* operator is applied to each matrix to reduce each matrix to a singular similarity score. The *MinAvg* operator, inspired by *ColBERT* [2] takes a row-wise minimum and averages the result. This is equivalent to identifying the closest image to each query image, with replacement, calculating similarity score and averaging across all query images to get a unified product-level score. The resultant array is denoted as **S**.

$$\mathbf{D} = \mathbf{Q} \odot \mathbf{C_{rep}} \tag{7}$$

$$\mathbf{S} = [MinAvg(\mathbf{D^{(j)}})] \quad \forall \quad j \in \{1, \dots, \mathbf{N_C}\} \tag{8}$$

Similarity scores can be directly used for reranking the candidates. For our use-case as we configured our retrieval index to use Euclidean distance, we also convert our reranking results from similarity vector to a vector of Euclidean distances. A summary of the reranking process is provided in Figure 2.

3.4. System Design

We provide a high-level overview of the Brand Protection pipeline. Processing millions of products a day requires the system to be highly efficient, and we therefore choose to use the popular retrieval – reranking framework, which enables us to perform fast candidate generation, while effectively capturing similarities between groups of images. We have segmented the design into two sections – indexing and inference. Indexing workflow helps onboard brands

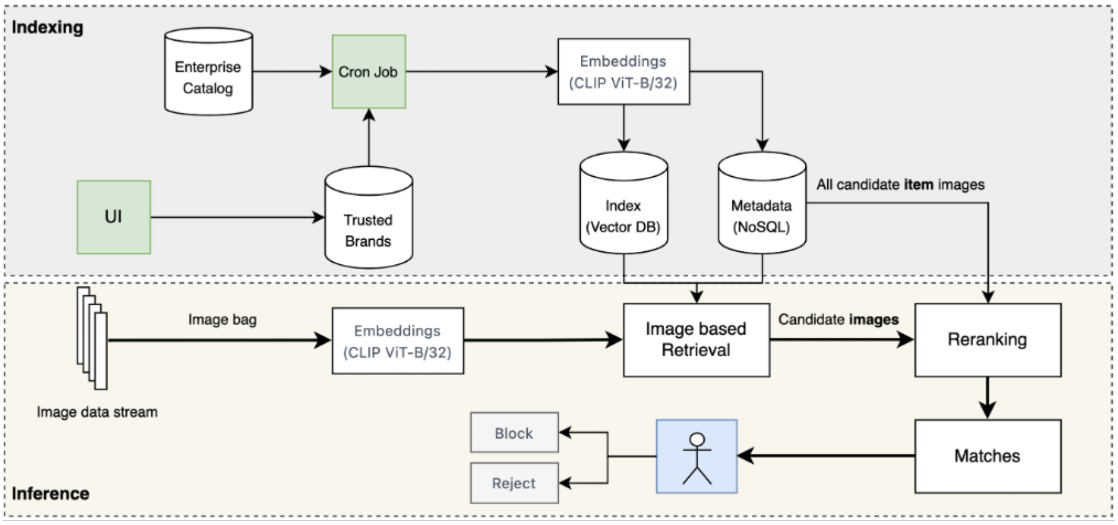

Figure 3: System Architecture.

and cover relevant products, whereas inference pipeline proactively monitors newly set up listings for brand misrepresentation. Figure 3 shows the design of our system.

Indexing refers to the process of translating raw product images to dense vector representations and storing them to a low search-latency vector database. First, we expose a user interface to the business stakeholders to enable onboarding "trusted" brands to the brand verification pipeline. Users select brands which are to be onboarded on-demand, and these brand names are stored in a database. A daily cron job selects any new brands that have been added and triggers the indexing workflow. All items mapping to the onboarded brands are selected and each of their images are embedded using CLIP. Product image embeddings are indexed in a high-scale low-latency vector database that perform ultra-fast similarity matching using an approximate nearest neighbors search. Additional product metadata required for reranking is saved in a low-latency NoSQL database.

Inference is performed on newly set up products and begins with consuming images from an upstream image stream. Products are read in batches where each product likely has multiple images. Image embeddings are generated for each product, and each product is now represented by a bag of embedding vectors. There can be a variable number of images in each bag/product. We retrieve the closest images to each query image from the index, and map them back to their product IDs using metadata database. Unique product IDs are taken as candidate matches for reranking.

After an initial retrieval using the IIR system, reranking aims to improve the ranking of the retrieved images by considering additional features or similarity measures. This process helps to ensure that the most relevant images are presented at the top of the search results, enhancing the overall performance and effectiveness of the IIR system. In our adaptation, we perform reranking at a product level (represented by a bag of image embeddings) rather than single image level. If the closest reranked candidate is below a preconfigured distance threshold, a match record gets generated. An operations team reviews the match to determine if it is a truly misrepresented brand and blocks true positives from the website.

4. Experiments

We describe our experimental setup evaluating various retrieval approaches and their results. We assess the performance of three approaches on different subsets of e-commerce data with distinguishing characteristics.

4.1. Data and Preprocessing

We perform experiments on items from a subset of popular brands. A sample of 1.3M products is selected, and all unique images are onboarded to a vector database and the relevant metadata is inserted to a low-latency NoSQL database.
Tasks. We analyze the performance of SERGI for two tasks -general purpose image *retrieval* and *classification.*
Index. Two vector databases are set up for experiments – (i) *image-level index* containing image representations to be used for baseline image-to-image match and *SERGI*; and (ii) *product-level index* with average of image representations for a product. Both indexes use Euclidean distance and perform brute force retrieval which guarantees finding nearest neighbor candidates for experimentation.
Datasets. Four segments of product data from Walmart are prepared – (i) **Eval** represents an unbiased sample of products from the same brands as indexed items; (ii) **Eval-C** represents a subset of **Eval** which belongs to brands whose product listings are known to have low noise and highly reliable product-to-brand mappings; (iii) **Eval-N** denotes a subset of **Eval** which is known to contain noisy images; and (iv) **Eval-Cls** is prepared by adding an independent sample of products from brands which are not indexed to **Eval** – this dataset is used for evaluating classification performance using *SERGI.* Table 2 provides a summary of indexes and datasets.
Representations. CLIP embeddings are used for representing query and indexed images.
Preprocessing. In context of our work, noise represents signals which lead to increased similarity between unrelated items, for example swatch images or nutrition labels. In real e-commerce data, noise is expected and therefore we do not remove noisy images from our data. Rather, we explore products with noisy images in more detail so as to assess robustness of the studied approaches. Products with single images are removed as they reduce all the three approaches to a trivial image-to-image match and may regress the results. Considering nearly 85% of our listings have multiple images, our inferences generalize to majority of the catalog. To avoid potential leakage of the evaluation products in the index, we use a cutoff date for indexing, such that any items set up before the date get indexed. For validation, we sample from items that are newly set up after the cutoff date to avoid any leakage.

4.2. Evaluation Metrics

We assess the results of our work from two perspectives – *retrieval* and *classification.* While the primary application for our system is to be used as an IIR system for image groups, it can also be used for certain classification tasks. Since our business use case – Brand Protection is a high

Table 2
Index and dataset summary.

Type	Description	Images	Products
`Index`	*Image Level*	2.5M	1.3M
`Index`	*Product Level*	-	1.3M
`Dataset`	*Eval*	23,435	4,809
`Dataset`	*Eval-C*	4,893	1,187
`Dataset`	*Eval-N*	649	131
`Dataset`	*Eval-Cls*	54,820	9,795

cardinality classification task, we also report on the classification performance of *SERGI*. For measuring *retrieval* performance, we define *relevance* of reranked candidates as 1 if query and candidate products are from the same brand:

$$relevance = \begin{cases} 1 & \text{if } brand(\mathbf{Q}) = brand(\mathbf{C}^{(j)}) \quad \forall \quad j \in \{1, \ldots, \mathbf{N_C}\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Three metrics are used to summarize retrieval performance – **Precision@K** which is the ratio of relevant candidates from the retrieved results; *Mean Average Precision@K* (***MAP@K***) which considers the order of the returned relevant candidates and provides higher scores if relevant candidates are ranked lower; and *Mean Reciprocal Rank* (***MRR***) which is the mean of multiplicative inverse of the rank of the first relevant candidate. For the classification task corresponding to Brand Protection use-case, we report ***precision***, ***recall*** and ***F_1 score***.

4.3. Comparable Approaches

As the application of grouped image retrieval is relatively unexplored, we use two alternative approaches for comparison.

Image to Image Match (*I2I*). An image-to-image match system is initially implemented as a baseline. Given a group of images representing a query product, we perform top-K (K=20) retrieval across each query image. The resulting candidate images are reranked based on the Euclidean distance from their closest query image. We observe that while the baseline performs reasonably well on ***Eval***, it lacks robustness to noisy and generic product images. This is expected because this approach prioritizes the closest image from a group of product images, which are often generic. *I2I* works well if we can identify and exclude generic images, but that is often not viable with complex and large-scale catalogs that take inputs from multiple internal sources as well as third party sellers. Some examples of such matches are provided in Figure 4. Therefore, we formulate an approach that is robust to generic images.

Representation-focused learning (REP). Inspired by representation-focused rankers in information retrieval literature, we construct a consolidated representation for products to be indexed. For each image associated with a product, we generate image embedding based on the CLIP model. We then calculate the element-wise mean across all images to build consolidated product embeddings. This technique mirrors the Average Query Expansion (AQE) [22] strategy, which involves modifying the query by averaging representations of the top retrieved images.

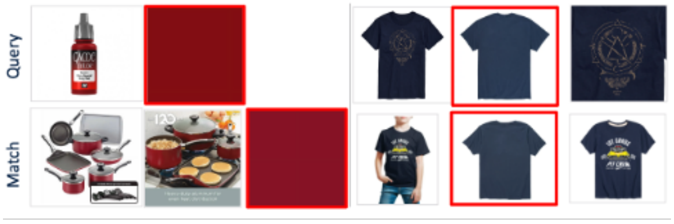

Figure 4: Two examples demonstrating limitations of image-to-image retrieval. Left hand side example shows an incorrect match between unrelated products based on swatch image. Right hand side shows an incorrect match based on a generic image.

These embeddings are incorporated into the *product-level index*. During inference a consolidated representation of the query product is generated in a similar manner. The final retrieval step is conducted using the *product-level index* and the consolidated representation of the query product.

4.4. Retrieval Performance Comparison

We compare performance of the three approaches on three datasets. For *Eval* data, we see that ***SERGI*** performs slightly better than ***I2I*** baseline. We hypothesize that the lack of meaningful lift is because *Eval* data is reasonably clean and has low generic image count. Difference between ***SERGI*** and ***I2I*** becomes more pronounced in *Eval-C* data. This is because *Eval-C* contains products from a set of brands which have highly pronounced characteristics. Moreover, difference between ***I2I*** and ***REP*** diminishes because average product representations for ***REP*** become more discriminative. Each approach sees a significant lift when compared with *Eval*.

The performance of each approach drops when using *Eval-N* data due to the presence of generic images that lead to lower quality retrieval. ***SERGI*** significantly outperforms the other two approaches, especially at lower values of K, which is important for the classification use-case. Retrieval results are summarized in Table 3.

Table 3
Retrieval Results.

Dataset	**Method**	P@1	P@3	P@5	MAP@1	MAP@3	MAP@5	MRR@5
Eval	I2I	0.650	0.628	0.613	0.650	0.667	0.666	0.691
Eval	REP	0.569	0.570	0.570	0.569	0.607	0.612	0.658
Eval	**SERGI**	**0.667**	**0.629**	**0.609**	**0.667**	**0.683**	**0.682**	**0.706**
Eval-C	I2I	0.868	0.844	0.835	0.868	0.878	0.880	0.898
Eval-C	REP	0.853	0.858	0.854	0.853	0.884	0.883	0.915
Eval-C	**SERGI**	**0.895**	**0.880**	**0.874**	**0.895**	**0.905**	**0.906**	**0.922**
Eval-N	I2I	0.496	0.483	0.479	0.496	0.517	0.530	0.569
Eval-N	REP	0.458	0.489	0.510	0.458	0.500	0.530	0.560
Eval-N	**SERGI**	**0.550**	**0.519**	**0.511**	**0.550**	**0.571**	**0.577**	**0.610**

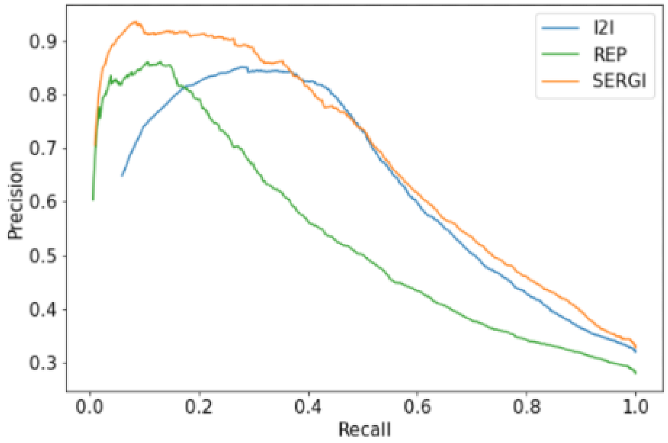

Figure 5: Precision-Recall characteristics for the three approaches at normalized Euclidean distance based thresholds.

4.5. The Classification Use-case

Image retrieval systems have been explored in context of image classification by various studies. Some studies explore a unified framework for both tasks, highlighting that both involve measuring the similarity between the query and training or candidate images [23]. We leverage ***SERGI*** for classification for our Brand Protection use-case. The primary objective is to classify whether a newly set up product from an unfamous brand actually belongs to an indexed popular brand. Consider that a candidate product belongs to a brand $\mathbf{B_s}$. For adapting the retrieval results for classifications, we tune a distance threshold **T** such that if Euclidean distance between the query and a candidate product is below **T** then the query product is classified as belonging to brand **Bs**.

The precision-recall (PR) characteristics of the three approaches are illustrated in Figure 5. Given the non-uniform range of Euclidean distance, it is normalized between 0 and 1 for consistent representation. Our classification results on *Eval-Cls* data are presented in Table 4. Of the three, ***SERGI*** demonstrates the maximum area under the PR curve, followed by ***I2I***, and ***REP***. We observe that while ***SERGI*** significantly outperforms other methods for low recall regions, its precision is relatively comparable to ***I2I*** for higher recall values. This implies when a lower, more restrictive **T** is selected, ***SERGI*** is expected to be relatively more precise whereas for use-cases requiring higher recall, benefits of ***SERGI*** tend to wane.

The choice of $\mathbf{n_r}$, and indirectly $\mathbf{n_m}$ can also influence the nature of the PR curve. Additionally using *MinSum* operator instead of the currently used *MinAvg* can reduce ***SERGI*** to ***I2I***. We plan to analyze the influence of these parameters on the PR characteristics through ablation studies in future. Although we report the results at a recall threshold of 70% – the long-term production target, our initial deployment prioritizes higher precision to foster stakeholder confidence. Consequently, we reduce our recall requirement to 50%, which is expected to enhance precision to 0.735.

Table 4
Classification performance at selected thresholds.

	T	Precision	Recall	F_1 Score	AUC
I2I	0.277	0.501	0.700	0.584	0.595
REP	0.346	0.377	0.700	0.490	0.535
SERGI	0.415	**0.534**	0.700	**0.606**	**0.677**

4.6. Deployment

In this section, we delineate the deployment of the SERGI system for the purpose of Brand Protection and subsequently discuss the results post-implementation. The SERGI system actively monitored new products from a selection of lesser-known brands. If a monitored product corresponded with an indexed product from a reputable and trusted brand, it was earmarked for manual review. Our business stakeholders conducted an individual review of each flagged product to ascertain the accuracy of the match, specifically, whether the highlighted item truly belonged to the suggested trusted brand. The outcomes from these manual reviews are subsequently reported.

Progressive deployment. In the initial phase, we indexed products from a limited range of fewer than 100 brands. The selection of these brands was based on internally curated databases to ensure the inclusion of popular and reliable brands. To ensure the integrity of our database, we used a combination of historical claims data, manual brand curation, and data from gated brands to select a clean set of brands. As we began receiving reviews, we identified patterns of false positives, which we promptly addressed. After a meticulous two-month period of monitoring and active manual reviews, we deemed it appropriate to expand the system to include additional brands. For the onboarding of these additional brands, we offer two options – *ad-hoc onboarding* for a few brands through a user interface, which helps take prompt action for brands with recent IP related escalations; and *bootstrapping process* to onboard thousands of brands and achieve scale.

Prioritizing brands for onboarding. As there are millions of brands in the catalog, it is imperative to filter trustworthy and reliable brands for indexing. Since it can be impractical to manually annotate each brand as fit or unfit, we exploit LLMs' parametric memory and general understanding of natural language, and hence brand popularity to shortlist suitable brands. A set of 20,000 brands is initially shortlisted based on business-provided criteria and a sample from these brands is passed through a LLM powered chat interface. A suitable prompt is tuned, which takes batches of 25 brands selected randomly, and provides relative ranking of each brand based on its general understanding of a brand's reach, recognition, and usage among consumers. A few batches are sent for manual review by business, and after confirming the reliability of the results, all 20,000 brands are passed to the LLM. We select a rank threshold (of 10) based on manually reviewed batches, and all brands with rank equal to or below this threshold are shortlisted for onboarding.

Filtering noisy products. Despite our method proving more robust to noisy and generic images, we observed a few instances where items escaped the robustness of our system. For instance, we noticed products that only contain swatch images or those with only plain white blocks, and led to a high volume of false matches. We developed a heuristic-based approach to coarsely detect and filter such images. Filtering is performed at two stages – during indexing, we remove noisy images; and post-inference, we exclude products for which the primary image is noisy. We plan to undertake an ablation study to optimize the placement of this approach in future.

Results from release. During the initial two-month activation period of the system, it identified 1,424 products requiring review, of which 1,399 were subsequently assessed. Of these, 1,062 were confirmed as true positives, indicating a precision rate of 0.759. This precision rate was fairly close to the expected precision rate of 0.735 at 50% recall from our experiments. The minor difference between expected and observed precision may be attributed to the difference in brand distribution between experiments and deployment, and the use of post-inference filters.

The true positives represented products inaccurately listed under incorrect brand names, adversely impacting customer experience. As a result, the review team promptly blocked these products while notifying the marketplace sellers. If the sellers update their listings with the accurate brand name, the product is then eligible for republishing. While we outline a specific use-case for detecting incorrect brands, learnings from our work can be generalized to other use-cases which require finding similar entities to a query entity.

5. Conclusions

In this study, we thoroughly investigated an array of strategies for entity retrieval rooted in image groupings. We introduced a unique method, drawing inspiration from the late-interaction architecture prevalent in natural language literature, which facilitates precise and highly efficient retrieval. We created a specialized method, SERGI, designed for the retrieval of entities similar to the given image groups. This paper also showcases an internal application of our system tailored for a fine-grained visual categorization, demonstrating its adaptation for tasks involving high cardinality classification. The innovative concept of grouped entity retrieval, along with the application of SERGI, holds significant potential for a wide range of other domains.

6. Acknowledgements

We would like to thank the entire Walmart Item & Inventory data science team for their contributions to this project. We thank Sankalp Jain, Elizabeth Snukst, Binwei Yang, Brian Seaman and Prakhar Mehrotra for their unwavering support and review for the paper. We are grateful to Mohammad Qurashi for his guidance on understanding the business problem, data landscape and working with stakeholders to identify key business opportunities. We thank Dion Tang, Robert Caya, Kendel Jackson and William Zuniga for proactively providing business feedback, annotations and reviews that helped with iterative enhancements of the deployed system. We thank Arup Das for his help in detecting generic and noisy images used for experimentation. We thank Ishan Bhatt, Yi-Sheng Yang and Quoc Tran for helpful discussions and suggestions about addressing common catalog imagery challenges.

References

[1] I. M. Hameed, S. H. Abdulhussain, B. M. Mahmmod, Content-based image retrieval: A review of recent trends, Cogent Engineering 8 (2021) 1927469.

[2] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.

[3] Z. Kurt, K. Özkan, An image-based recommender system based on feature extraction techniques, in: 2017 International Conference on Computer Science and Engineering (UBMK), IEEE, 2017, pp. 769–774.

[4] P. Shamna, V. Govindan, K. A. Nazeer, Content based medical image retrieval using topic and location model, Journal of biomedical informatics 91 (2019) 103112.

[5] S. Kalra, H. R. Tizhoosh, C. Choi, S. Shah, P. Diamandis, C. J. Campbell, L. Pantanowitz, Yottixel–an image search engine for large archives of histopathology whole slide images, Medical Image Analysis 65 (2020) 101757.

[6] Y. Li, J. Ma, Y. Zhang, Image retrieval from remote sensing big data: A survey, Information Fusion 67 (2021) 94–115.

[7] K. Smelyakov, D. Sandrkin, I. Ruban, M. Vitalii, Y. Romanenkov, Search by image. new search engine service model, in: 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), IEEE, 2018, pp. 181–186.

[8] S. R. Dubey, A decade survey of content based image retrieval using deep learning, IEEE Transactions on Circuits and Systems for Video Technology 32 (2021) 2687–2704.

[9] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, Deep learning vs. traditional computer vision, in: Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1, Springer, 2020, pp. 128–144.

[10] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[13] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.

[15] H. Zamani, M. Dehghani, W. B. Croft, E. Learned-Miller, J. Kamps, From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 497–506.

[16] B. Mitra, F. Diaz, N. Craswell, Learning to match using local and distributed representations

of text for web search, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 1291–1299.

[17] S. K. Addagarla, A. Amalanathan, Probabilistic unsupervised machine learning approach for a similar image recommender system for e-commerce, Symmetry 12 (2020) 1783.

[18] T. Stanley, N. Vanjara, Y. Pan, E. Pirogova, S. Chakraborty, A. Chaudhuri, Sir: Similar image retrieval for product search in e-commerce, in: Similarity Search and Applications: 13th International Conference, SISAP 2020, Copenhagen, Denmark, September 30–October 2, 2020, Proceedings 13, Springer, 2020, pp. 338–351.

[19] A. Baldrati, M. Bertini, T. Uricchio, A. Del Bimbo, Conditioned and composed image retrieval combining and partially fine-tuning clip-based features, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4959–4968.

[20] M. Hendriksen, M. Bleeker, S. Vakulenko, N. Van Noord, E. Kuiper, M. De Rijke, Extending clip for category-to-image retrieval in e-commerce, in: European Conference on Information Retrieval, Springer, 2022, pp. 289–303.

[21] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, C. P. Langlotz, Contrastive learning of medical visual representations from paired images and text, in: Machine Learning for Healthcare Conference, PMLR, 2022, pp. 2–25.

[22] O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.

[23] L. Xie, R. Hong, B. Zhang, Q. Tian, Image classification and retrieval are one, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 3–10.