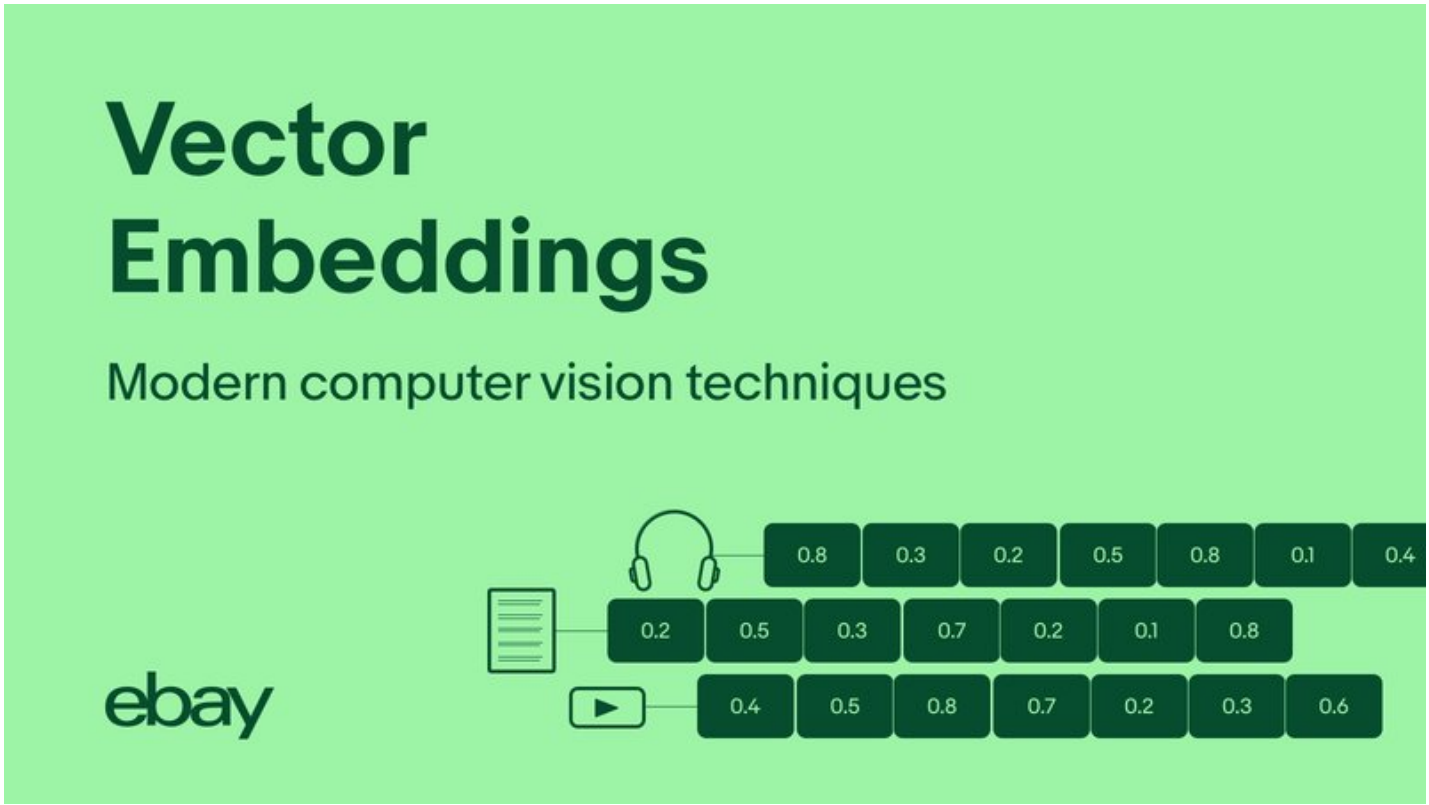
MACHINE LEARNING in ENGINEERING

How eBay's New Search Feature Was Inspired By Window Shopping

Mar 31, 2023

By: Senthilkumar Gopal, Shubhangi Tandon, Christopher Miller, Deepika Srinivasan, Rui Kong, Selcuk Kopru and Srinivas Bhagavathula

A new feature generates customer delight by using modern computer vision techniques to drive new search paradigms through visual discovery.



We live in a world of discovery where visual appetite reigns supreme. Window shopping, infinite scroll lists, and micro engagements using simple visual cues are the norm. Search engines traditionally interpret a textual query input and match items and/or documents ranked by their

relevance to the input query. The relevance of the retrieved results is based on scoring the closeness of the input query to the matching items/documents. This traditional approach heavily constrains the user to provide a language-bounded interpretation of their implicit preferences such as style and aesthetic feel, which are not usually available in a shared vocabulary. Bridging this gap between inspiration and discovery by enabling visual first pivots in search is the goal of our work.

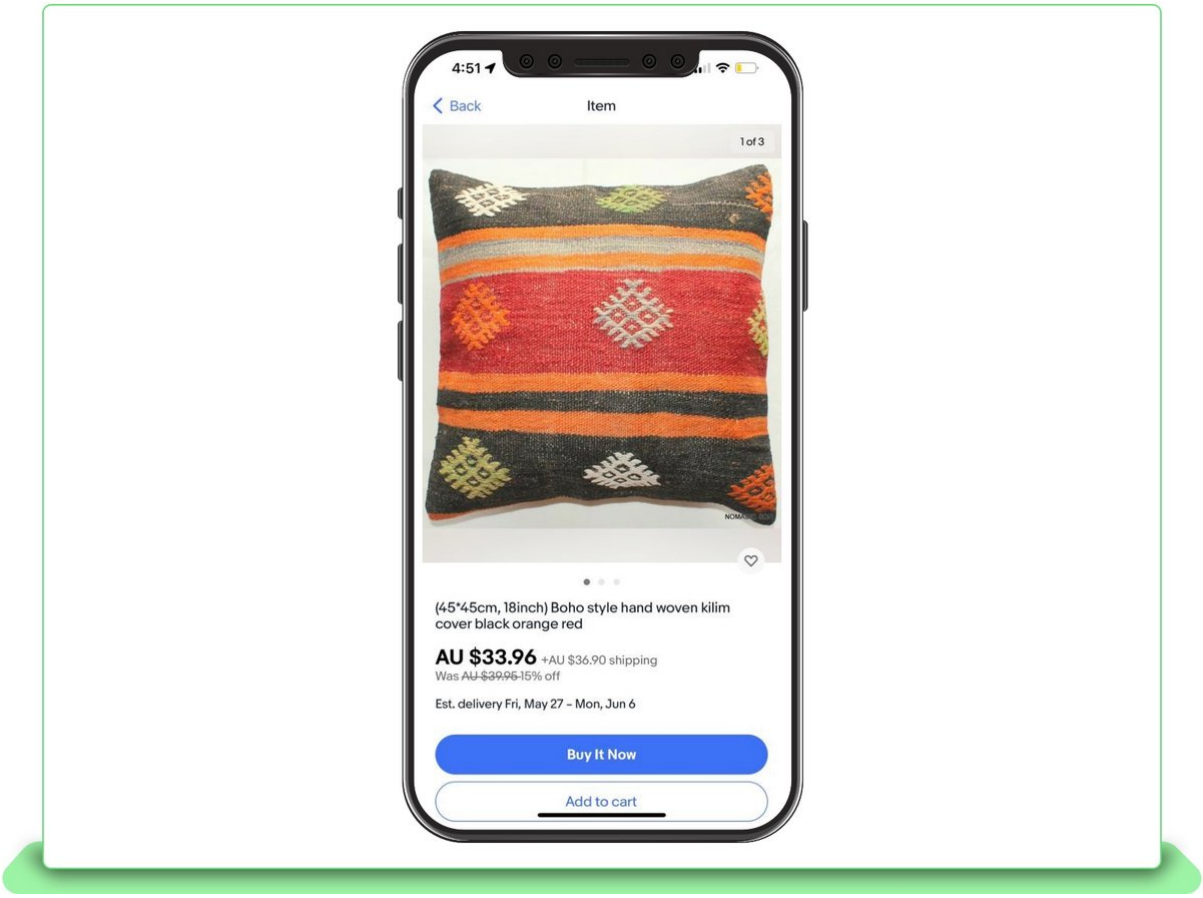


Fig 1: User desired item available in the inventory

Let’s illustrate this with an example. A user has an ethnic Turkish themed living room and is looking to purchase pillows for their recently purchased butterscotch-colored couch. In tune with their personal style, the user starts off their search using “turkish throw pillow,” reviews the results and in the process identifies the specific pattern used in pillows from Turkey as “kilim.”

They attempt to search for this pillow using various combinations of textual queries such as “orange kilim pillows,” “orange throw kilim pillows” or even a broad search “kilim pillows” (results in Figures 1 and 2) which does not yield their desired result in the top results. Though the results were highly relevant and had the best match with the query text which has been provided, the result of matching items varies significantly for each of these queries, as all of the inventory is not being displayed, forcing the user to query over and over again to try and get the correct combination of query text. Thus, words fail to elicit the style preference that the buyer has intended to search for.

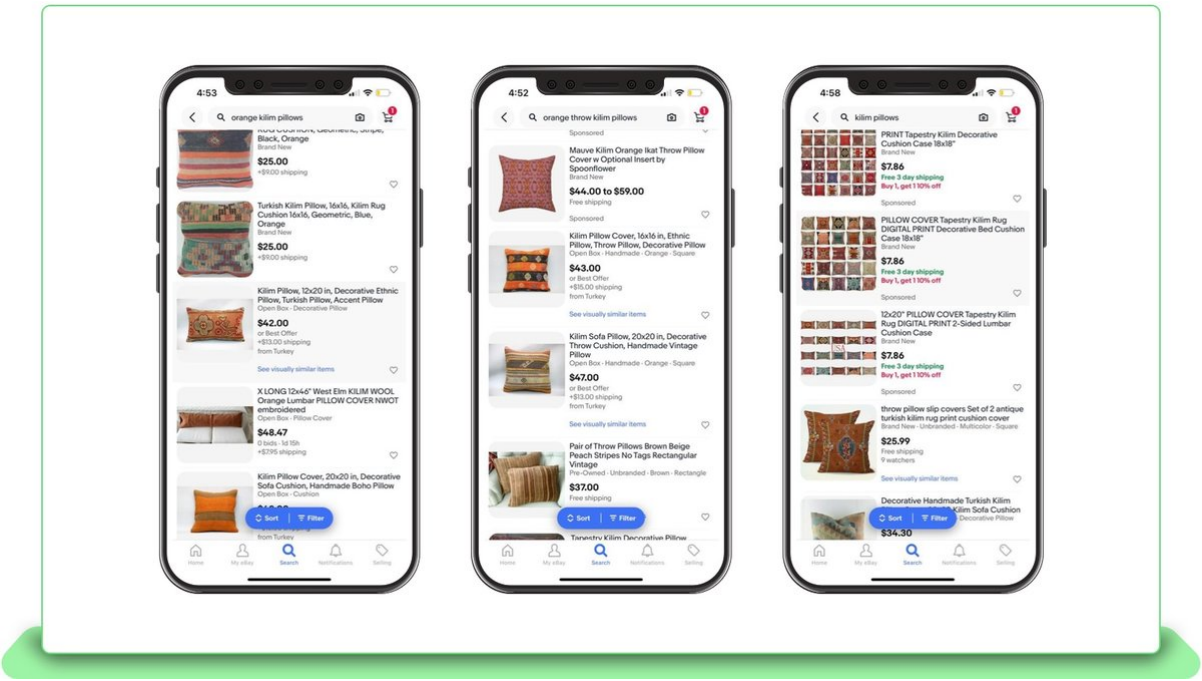


Fig 2: Various attempts by the user to search for the particular item

In such situations, we need to provide capabilities to help engage enthusiast customers by inviting them to browse their assortment in a more visually appealing methodology allowing them to zero down on the item they love.

The following sections delve deeper into how the search team embraced this visual discovery and the mechanisms powering this new experience.

Search Query and Retrieval Methodology

The following describes how a query that is submitted by a user is matched against the listings created by the sellers. At a very high level, once the seller creates the listings, the metadata associated with the listing, such as its title, description, image and all its other attributes alongside interpreted data to help retrieve the listing efficiently, are all extracted and indexed for Cassini, eBay's search engine [6].

When a query is submitted, for example "pillows," the query is submitted through the application frontend (mobile/desktop) to the search service, which in turn internally utilizes multiple supporting mechanisms for query rewrite, expansion and more, along with session contextualization, to retrieve the best matching listings possible.

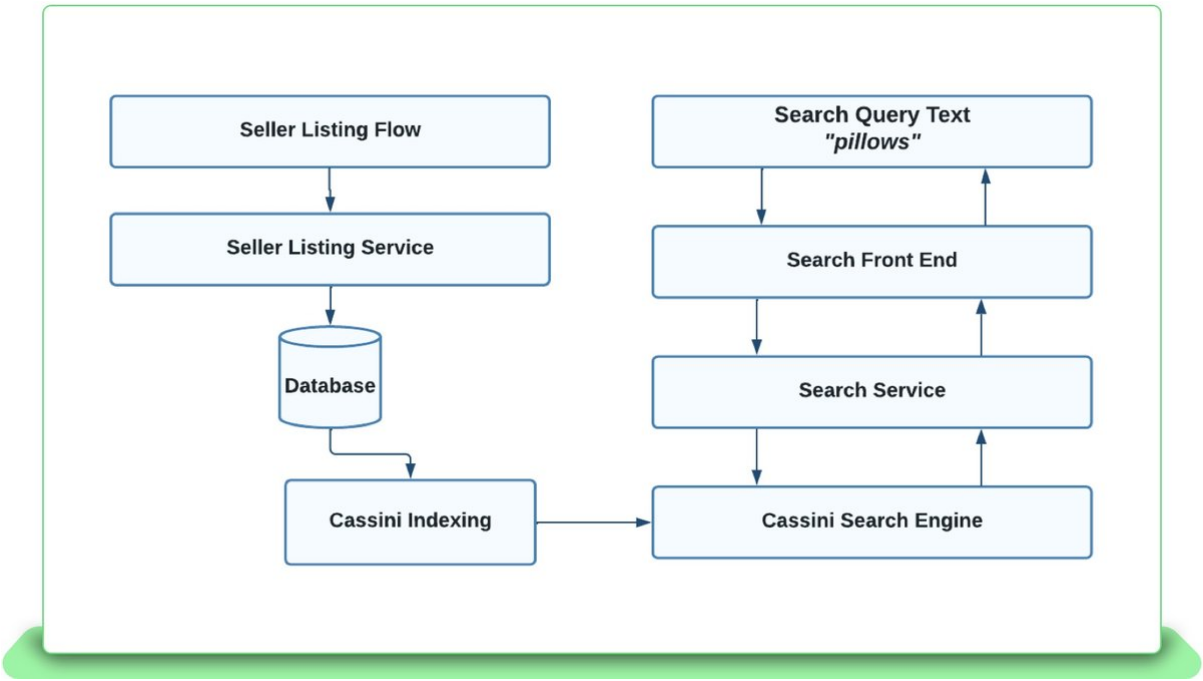


Fig 3: A generic representation of the search query flow

With a large inventory and millions of buyers spanning across the globe, this approach scales to provide sub-second response times with great relevance. This has been the traditional approach for information retrieval problems for all types of search engines. The goal is to interpret the query as efficiently and accurately as possible to match the intent and the expectation of the user, solely based on the input query being provided. There have been numerous interesting problems and innovations in the past to improve the retrieval accuracy. However, they have always been limited by the input purely being a text based query. How can we provide users the best possible ability to express their desired outcome if it cannot be expressed in words?

Solution Motivation

Taking inspiration from the analogies of window shopping at a furniture store, where different models or brands of the same furniture are available within a co-located space, we can enable the users to explore similar items with variations based on a cue that they provide as their input. This input can be in the form of an item that they find interesting and similar to what they are looking for. This visual cue allows us to provide the user with a way to identify their perfect item.

The user may then find something which is closer to their intended item in the result and continue to look for something similar once again and continue the journey. This enables them to also discover newer inventory which may be complementary to their original item and has the potential to improve customer happiness. Using semantic similarity [1] is not new to eBay, with the introduction of Image search in the eBay app, which allows the user to find identically looking items based on a picture taken from their phone camera. However, the usage of visual cue for fueling visual discovery and inventory browsing enables the user to efficiently navigate search results based on their style and preferences without the need for textually enforced filters such as aspects.

On a more technical note, let us dive into how a simple k-nearest neighbors algorithm (knn) works. A knn algorithm stores the item as a multi-dimensional vector based on varying factors during its learning phase. Then, in the classification phase, a query retrieves the k (for example: 500 results) closest items nearest to that query item.

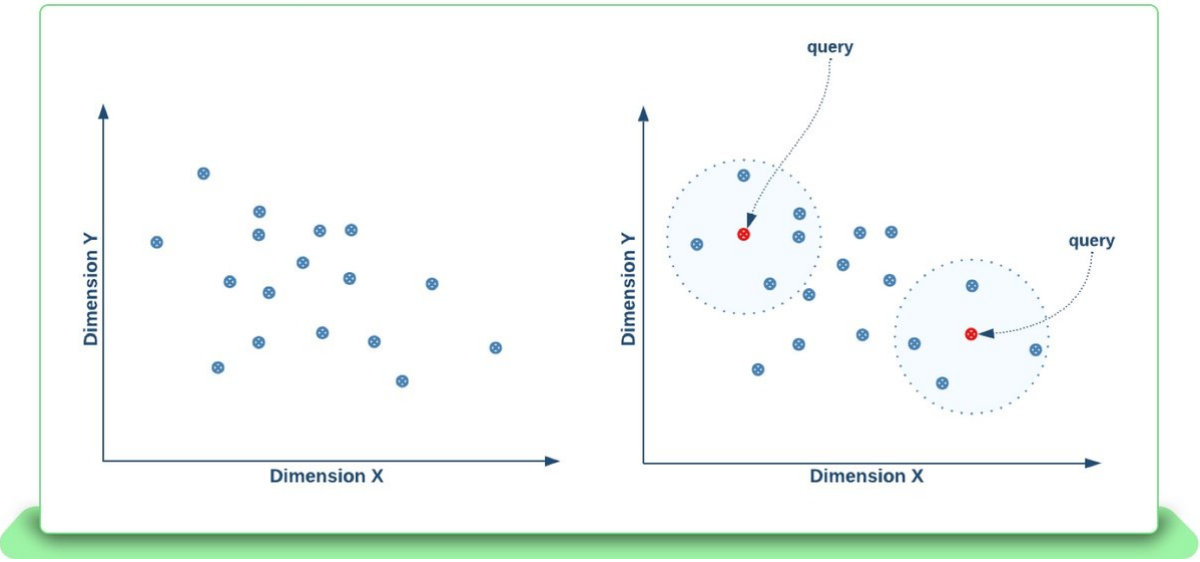


Fig 4: Simple representation of item title with 2 dimensions to illustrate nearest neighbors search

Due to the scale involved (1.7 billion listings as of Q4 2022), training and enabling a full-fledged knn search for high dimensional data is inefficient and time consuming. To overcome this, Approximate Nearest Neighbors (ANN) search is utilized instead and the specifics are explained below. They provide a more optimal means to efficiently retrieve neighbors with a minimal loss of accuracy.

The above description for an item title as a 2-dimensional vector (Fig. 4) is a very simplistic representation, as an item has high dimensionality of attributes, along with large dimensions of data to uniquely represent the title text, aspects and the images associated with the item.

So where exactly do these multi-dimensional vectors come from, and what do they represent?

Learning a Vector Embedding for an Item

In machine learning, vectors are some of the key components that act as representations for more complicated objects like images or text. It's easier to think about a vector as a compressed encoding or hash of the most salient information contained in the encoded object. Vectors are practical for this purpose because they can be expressive across many different dimensions, and also because operations on vectors scale well inside GPUs.

From here, we will refer to vectors as representations. One of the more traditional representations is called a word embedding. As the name suggests, this is a distinct embedding assigned to each word of interest. There are many ways to create a word embedding, but one of the most popular is through a method called "co-occurrence." Co-occurrence asserts that words which appear near to the same text should themselves have similar word embeddings. For example, "My pet cat drinks

milk" and "My pet dog drinks water" suggests that "cat" and "dog" should have similar embeddings because they are both neighbored by the words "pet" and "drinks."

The exact vectors that are associated with each word are not individually tailored. Instead our machine learning models look at a large corpus of text data and attempt to produce them for us. For the purposes of this project, we did not need embeddings for single words. Instead we focus on getting embeddings for images and titles.

If representations are compressed encodings/hashes, then models are the hash functions. Their purpose is to take in the raw data and output the vectors. We produce image and text representations using a deep learning model that was developed at eBay to solve the eProduct [2] challenge described more in detail here. This deep learning model is trained to represent multiple modalities that can independently produce embeddings for each modality.

In Fig. 5, on the one side, there is an image encoder using any foundational CNN model, such as Resnet-50 [4] and on the other side, there is a text encoder, such as BERT [3]. The model is trained by considering a batch of fixed number of listings, where each listing is composed of an image and a title. Each listing is encoded using their respective component of the multi-modal model. The model then attempts to push the vectors coming from the same listing closer together, and to push apart all other pairs. The corresponding loss function is called matching or contrastive loss (Fig. 6).

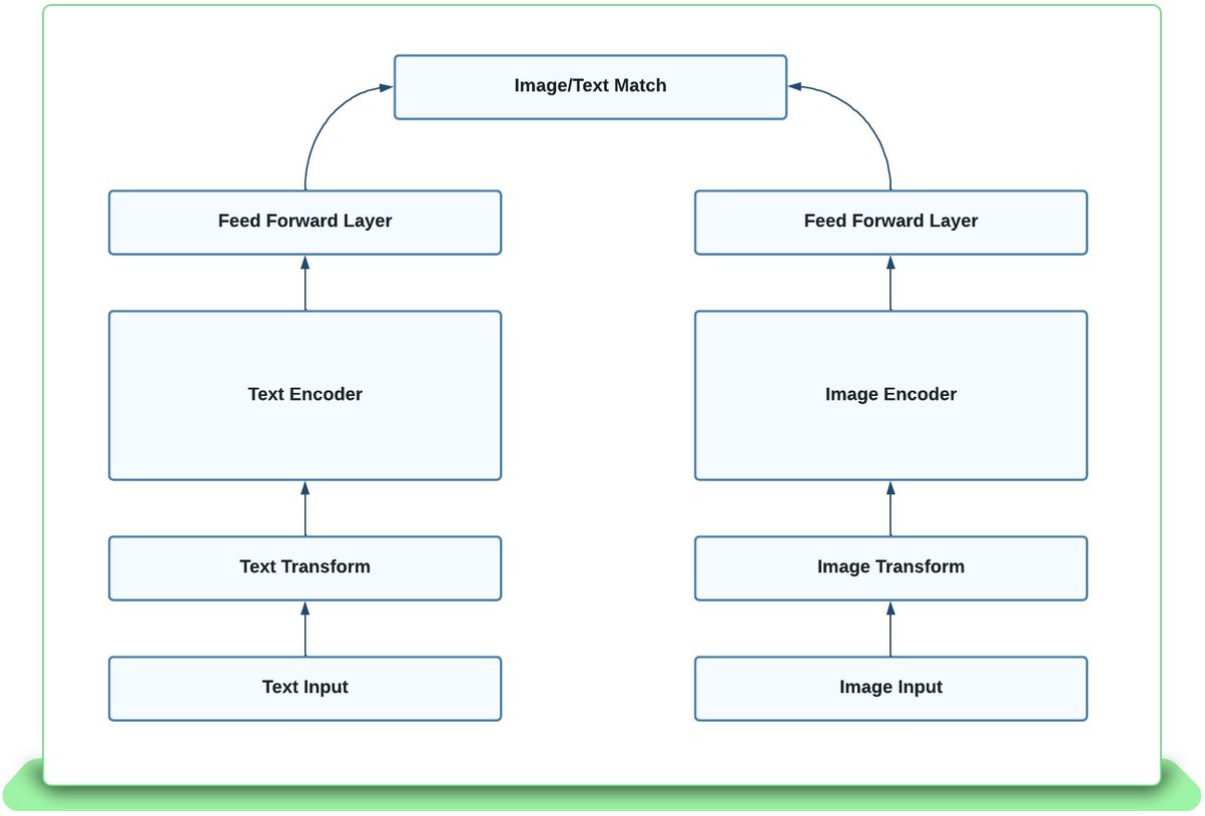


Fig 5: A simplified representation of the model architecture to produce multimodal embeddings

This is a practical way to train this sort of model, because it does not require gathering any additional human-labeled data and is inspired by the CLIP [5] model. This type of learning is called

self-supervised learning. We can measure the model performance by seeing how well the images predict the title from the same listing.

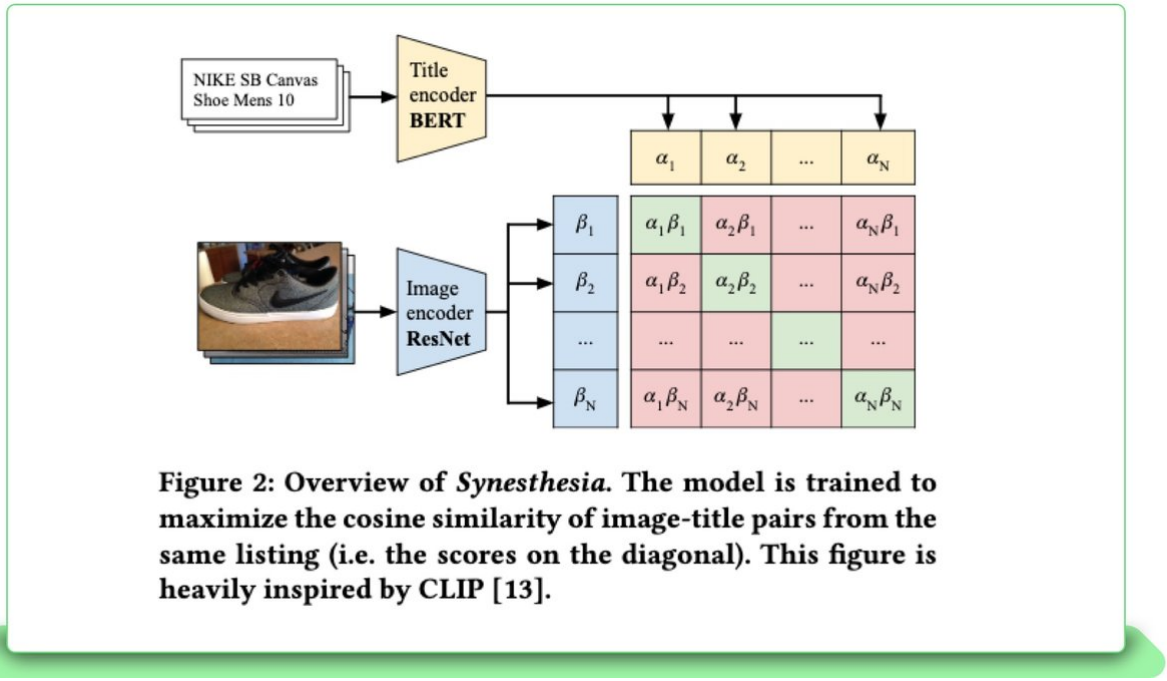


Figure 2: Overview of *Synesthesia*. The model is trained to maximize the cosine similarity of image-title pairs from the same listing (i.e. the scores on the diagonal). This figure is heavily inspired by CLIP [13].

Fig 6: The pairwise dot product of the matching image and title within a batch and negative sampling are represented using the off-diagonal combinations. This maps the fine-tuning of BERT [3] and ResNet-50 [4] jointly to embed into a shared space by learning the representations using contrastive loss.

To measure the performance of this model, using a mini-batch, the titles were compared to only one image within the batch ie., a single image embedding is compared against the title embeddings within the mini-batch. The trial is successful only if the correct title embedding is nearest to the query image. These trials are carried out for many batches, and the overall accuracy is computed.

Training and Inferencing Flows

Model training requires a large dataset of image-title pairs, and recently sold listings were used as the training dataset which acts as high fidelity and cohesive data. The multi-modal model trains by randomly and iteratively dividing the data into packets called batches and computing the matching loss. Each pass over the data is called an epoch. Multiple models were evaluated using hyper parameter tuning to optimize for image to title accuracy, and then the most performant model was selected.

The embedding dimension plays a critical role in accuracy as kNN search gets slower for high dimensional vectors. Based on the tradeoff between image to title accuracy performance and embedding size, we determined the embedding dimension which provides adequate performance speed without much loss in accuracy.

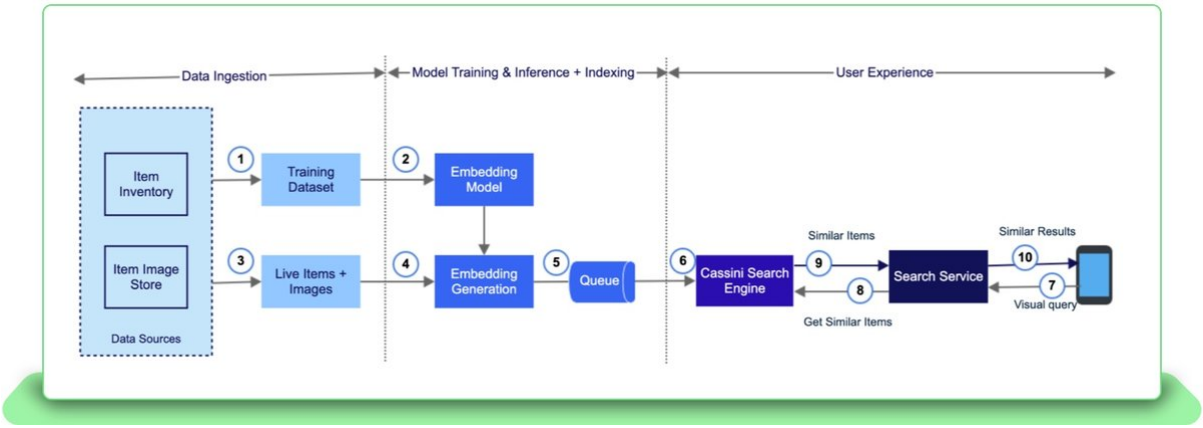


Fig 7: Simplified design flow for productionizing the data pipeline for Visual Discovery

Once the model is trained, it is time to run inference on the data which will then be posted to the site index. Due to the large data size involved, the data is passed to the multi-modal model in batches for easier reading from the Hadoop File System. Once the vectors are produced, there is an additional indexing step which converts the vectors into a compressed format that can be used in production. Finally the compressed vectors are passed to the data systems for indexing into the search engine.

Operationalization of Embedding Generation

Apache Spark was the preferred choice for implementing the item and image downloader, thanks to its high speed data processing and transformation for large data sets, especially for iterative operations, efficient reads from and writes to the disk and efficient multithreading within the JVM.

The item downloader is implemented as a Spark job that reads from the eBay item data warehouse, stored on HDFS. The downloader filters active items from the inventory and selects the relevant fields required for Visual Search for the active items. The resulting item data set is stored on HDFS as parquet files. The image downloader takes this list of items and downloads their images as binaries from the eBay item image store and a typical checksum based system is utilized to avoid redundant downloads and to verify the integrity of the downloaded image.

Model Inference

Krylov is the scalable and multi-tenant, cloud-based AI platform within eBay which allows Machine learning research to grow, iterate and scale rapidly. PyKrylov is the pythonic interface to Krylov that is used by researchers and engineers company-wide to access this AI platform and all of the below tasks identified are orchestrated using PyKrylov tasks.

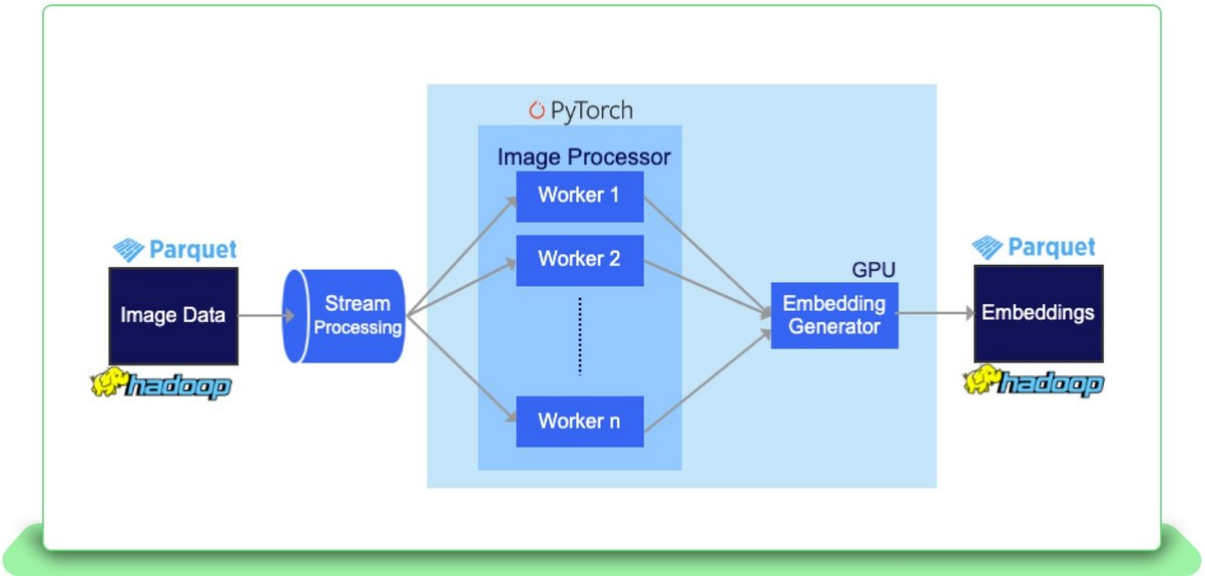


Fig 8: Simplified view of embedding generation

The Inference pipeline reads item and image data from the data ingestion step. The image binary is read as raw pixels and transformed to batch tensors for inference along with checksum validation. The image embedding vectors generated by model inference are published to the data systems that use a large columnar database such as HBase. HBase is a massively scalable and columnar storage and is a good fit for storing item related data for billions of items [6]. The item indexer reads item records from HBase, applies required business transformations and generates an index that is distributed to the Cassini search engine grid [6].

Workflow Orchestration using Airflow

We use Apache Airflow to automate the workflows described earlier. A workflow is a DAG (directed acyclic graph) of a sequence of tasks, with each component representing a task. Airflow provides the flexibility to support multiple tasks using a combination of Spark and Python based ML tasks. We choose Airflow since it scales easily irrespective of the number of tasks involved and complexity of the workflow DAG. The support for built in operators for Spark, Hadoop, Python, etc. allows easy onboarding and adoption for any new workflows.

Given this workflow, we support multiple processing modes.

- **Bulk** mode generates embeddings for the entire inventory when a new model is onboarded.
- **Delta** mode runs daily and processes items that are new or updated since the last run.

Some of the challenges that we faced while building this workflow are:

1. Orchestrating tasks using multiple versions of Spark
2. Tasks running on different platforms and availability zones
3. Handling DAG failures due to environment instabilities

How does this all connect together?

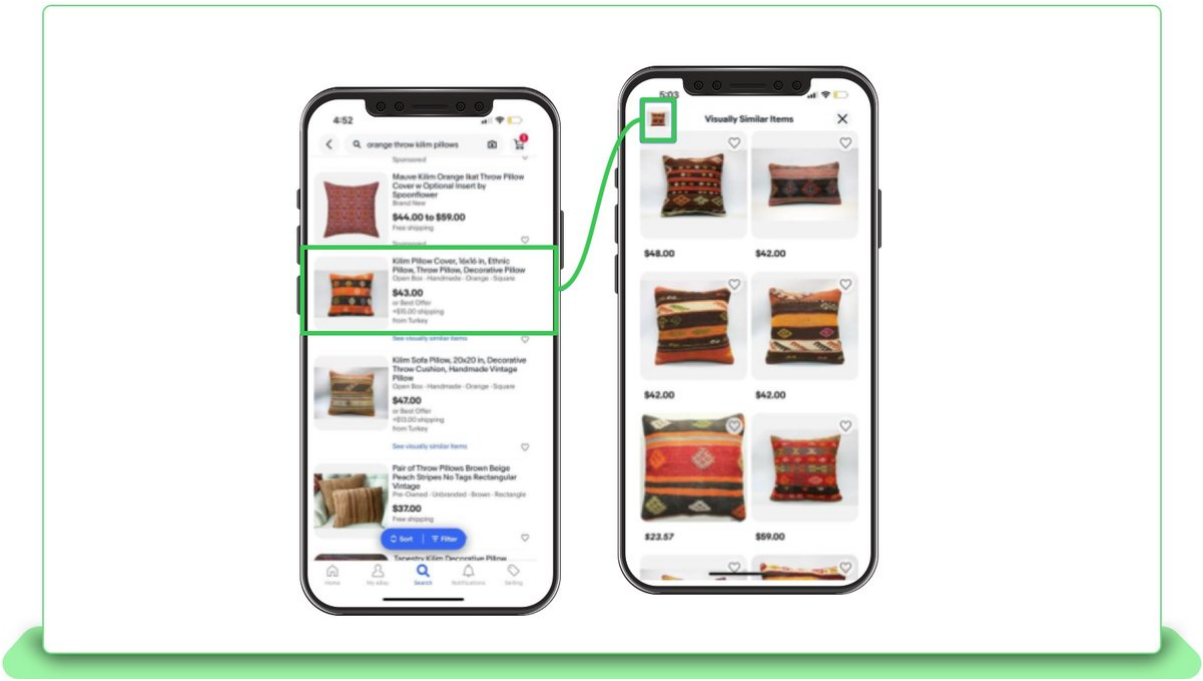


Fig 9: A simplified experience to search for visually similar items from search results

Looking back at the original example of the “kilim pillows,” the user now has the ability to browse through the search results and provide a visual cue for further exploration with a single click using the “See visually similar items” link. This brings the vast inventory of similar style and preferences within the control of the user’s choice and helps them easily identify their intended item without the need to provide textual representation of their preferences. This allows them to hone in on a specific set of closely related items in terms of style, aesthetics, fashion trends and more, opening the doors for completely novel discoveries of undiscovered inventory. This functionality has already been enabled for a few categories where visual search experiences prove a key enabler, such as Furniture and Home Decor, and will be available for other categories — and all users — in the next few months.

What's in the future?

Enabling this unique search experience provides capabilities to help engage enthusiast customers by showcasing style and choice and inviting them to browse in a more visually appealing and efficient way. Future iterations would allow users to pivot from their choices to other discoveries which also match their unique tastes, such as moving from pillows to blankets, tablecloths and more. The user could even pursue discovery journeys by simply cueing another interesting item within the result, providing an experience catered uniquely to their choice and taste.

Alongside their similarity to the pivot/cue item, the results would also be ranked based on quality measures to bring the best of both worlds, similarity and non query based ranking features, to produce the best blend of results. This improved experience helps users navigate through the entire inventory of choices matching their preferences, without the need to rewrite their search query at all — and with a single click.

Citations

[1] Yang, Fan, et al. "Visual search at eBay." Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.

[2] Yuan, Jiangbo, et al. "eProduct: A Million-Scale Visual Search Benchmark to Address Product Recognition Challenges." arXiv preprint arXiv:2107.05856 (2021).

[3] Devlin, Jacob, et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, arXiv, 24 May 2019. arXiv.org.

[4] He, Kaiming, et al. Deep Residual Learning for Image Recognition. arXiv:1512.03385, arXiv, 10 Dec. 2015. arXiv.org.

[5] Radford, Alec, et al. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020, arXiv, 26 Feb. 2021. arXiv.org, http://arxiv.org/abs/2103.00020.

[6] Trotman, Andrew, Jon Degenhardt, and Surya Kallumadi. "The architecture of ebay search." eCOM@ SIGIR. 2017.

TAGS: Computer Vision, Deep Learning, Machine Learning, Recommender Systems, Search Science

Previous Article:
How eBay Made Its New Accessibility Tool — And Made It Available to All

Next Article:
How eBay Modernized the Most Important Page on Our Platform

Artificial Intelligence

Making online commerce smarter and more efficient.

More Articles

Related Articles

eBay Execs Talk Generative AI and Computer Vision at VentureBeat Transform Conference
Jul 21, 2023

eBay Chief Technology Officer Mazen Rawashdeh Talks AI, Embracing Tech Disruption on Bloomberg Podcast

Jun 27, 2023

eBay's Blazingly Fast Billion-Scale Vector Similarity Engine

May 1, 2023

eBay Announces Winners of 4th Annual Machine Learning Challenge

Jan 11, 2023

How eBay Created a Language Model With Three Billion Item Titles

Jan 9, 2023

Follow Us

Subscribe to our RSS feed and follow us on social media