

Transformer-Empowered Multi-Modal Item Embedding for Enhanced Image Search in E-commerce

Chang Liu^{1,2}, Peng Hou¹, Anxiang Zeng², Han Yu²

¹Shopee Pte. Ltd., Singapore

²School of Computer Science and Engineering, Nanyang Technological University, Singapore
{chang.liulc, peng.hou}@shopee.com, {zeng0118, han.yu}@ntu.edu.sg

Abstract

Over the past decade, significant advances have been made in the field of image search for e-commerce applications. Traditional image-to-image retrieval models, which focus solely on image details such as texture, tend to overlook useful semantic information contained within the images. As a result, the retrieved products might possess similar image details, but fail to fulfil the user’s search goals. Moreover, the use of image-to-image retrieval models for products containing multiple images results in significant online product feature storage overhead and complex mapping implementations. In this paper, we report the design and deployment of the proposed Multi-modal Item Embedding Model (MIEM) to address these limitations. It is capable of utilizing both textual information and multiple images about a product to construct meaningful product features. By leveraging semantic information from images, MIEM effectively supplements the image search process, improving the overall accuracy of retrieval results. MIEM has become an integral part of the Shopee image search platform. Since its deployment in March 2023, it has achieved a remarkable 9.90% increase in terms of clicks per user and a 4.23% boost in terms of orders per user for the image search feature on the Shopee e-commerce platform.

Introduction

The image search feature on modern e-commerce platforms has become a convenient and efficient way for users to find desired products without the need to input keywords. When a user uploads an image, this feature automatically identifies items within the picture and returns relevant products on a given e-commerce platform. This functionality not only saves time and effort for the users, but also enhances their shopping experience. The image search feature often relies on an image embedding modeling (Chicco 2021) to extract features from images. Based on these features, K-nearest neighbors (KNN) clustering (Fix and Hodges 1989) is often leveraged to identify similar images within the database.

However, such a single-modal approach has two major limitations. Firstly, it overemphasizes on visual details. The image embedding model tends to retrieve products with similar image details, but not necessarily belonging to the same category. For instance, a query for dental floss in Figure 1(a)

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

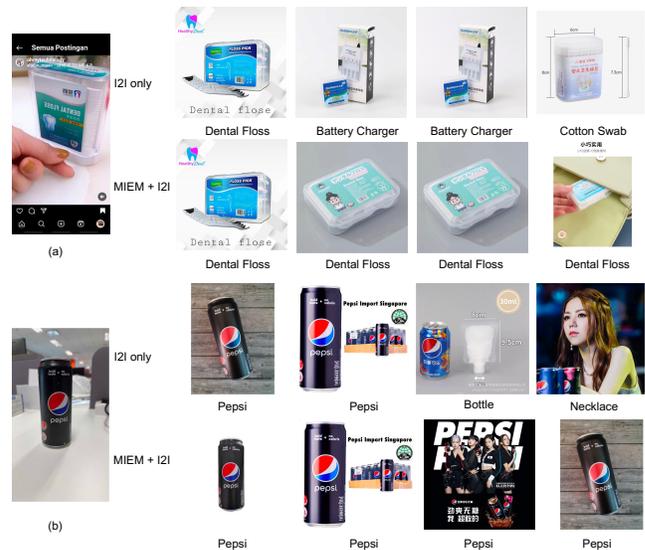


Figure 1: Search results comparison using different models. The left column displays user queries as images, while the right column showcases the top 4 retrieved items. Each item image is accompanied by its respective product name. Notably, employing Image-to-image (I2I) solely emphasizes visual details, disregarding semantic information. However, by integrating MIEM with I2I, these limitations can be effectively addressed.

often results in products with similar packaging (e.g., battery chargers, cotton swabs) being retrieved. Secondly, it neglects semantic information within the image. The presence of detection model biases on the product side may introduce noises, leading to erroneous retrievals when multiple product entities exist within an image. For example, in Figure 1(b), the latter two retrieved products are a plastic bottle and a necklace, with the soda can merely serving as a reference object and not a product itself. Such issues can be mitigated with multi-modal techniques, which incorporate both textual and visual information and are robust to noises.

In addition, image embedding poses engineering challenges. Due to the large number of products in Shopee’s e-

commerce platform¹ and the need for multiple-replications (Zhang et al. 2018), generating an embedding for each image requires a huge index volume. Furthermore, dealing with multiple images per product requires mapping and de-duplication, further increasing engineering complexity.

Technologies that have the potential to mitigate these challenges have emerged. Vision-language pretrained models like CLIP (Radford et al. 2021) can align images and texts in the same feature space, thereby supporting user image-based product search. However, these solutions fail to effectively cross-reference information across product images and texts, resulting in low recall score during recall phase. Cheng et al. employed concept-aware modality fusion to integrate information from a single image with texts, but it cannot accommodate multiple images (Cheng et al. 2023). Besides, its reliance on KnnSoftmax (Song et al. 2020) for training also increases implementation complexity during training. A line of previous works (e.g., METER (Dou et al. 2022) and X-VLM (Zeng, Zhang, and Li 2022)) has addressed image-to-text ranking a merge/cross attention transformer to generate matching scores for pairs of images and texts. However, these models have limitations in fusing multi-modal information during the recall phase.

To tackle these issues comprehensively, we propose the Multi-modal Item Embedding Model (MIEM). In this paper, we present the design and deployment experience of MIEM in the Shopee e-commerce platform. MIEM takes multiple images and product titles as the input, and leverages a Merge Attention Transformer Module (Dou et al. 2022) to fuse information from both images and texts, thereby producing an embedding for a given product. The incorporation of product titles injects explicit semantic information into the embedding, effectively alleviating the issue of the image embedding model excessively focusing on image details. MIEM has been deployed in the Shopee e-commerce platform since March 2023 to power its image search business. Compared to the previous adopted solution, MIEM has led to a remarkable 9.90% increase in clicks per user and a 4.23% boost in the orders per user metric.

Application Description

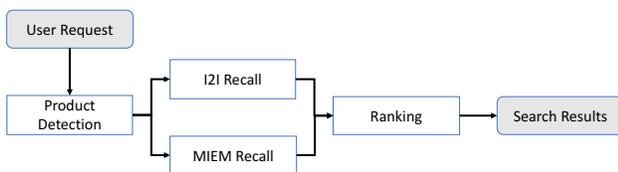


Figure 2: An overview of the Shopee Image Search Engine.

The overall workflow of the Shopee Image Search Engine is depicted in Figure 2. The system takes a user’s request (i.e., an uploaded image) as the input, and outputs a list of products it deems to be relevant. Upon receiving the user’s request, the system performs the following steps:

1. It executes a product detection model to obtain the bounding boxes for the products in the image.
2. The cropped images from the bounding boxes are then passed through two recall models: 1) Image-to-Image (I2I) recall, and 2) MIEM recall.
3. The recalled products are sent to the ranking model, where they are sorted in descending order of their ranking scores, producing the final list of recommended products.

The integration of MIEM into the Shopee Image Search Engine is based on the framework shown in Figure 3, consisting of the online and the offline components. The online component performs the following steps to retrieve the products using MIEM:

1. Upon receiving a user request, the request handler performs the initial task of parsing the request from various sources, such as webpages or different versions of the mobile app, and transforms it into a unified format. Once the request has been standardized, it is subsequently forwarded to the search server for further processing.
2. The search server invokes the feature service module, which then calls the detection service module to obtain the bounding boxes for the products in the image.
3. The cropped query image is used to invoke both the I2I model and the MIEM model services to obtain embeddings for the products.
4. The search proxy uses these embeddings to perform approximate nearest neighbor (ANN) search using the HNSW method (Malkov and Yashunin 2018) in two index databases: 1) the I2I index and 2) the MIEM index. For the I2I index, the corresponding product IDs are obtained for the top- K retrieved images, and then subjected to de-duplication (i.e., products with same IDs will only pertain one in the recalled list). For MIEM, the product IDs can be directly returned. Then, the search proxy merges the two sets of retrieved product IDs and sends them back to the search server.
5. The search server forwards the retrieved product list to the rank service, which scores each product based on relevance, popularity and other factors. The products are then sorted in descending order of their scores and returned as the final output list.
6. The log dumping service records and stores various user-related activities, including user requests, impressions, and interactions with the search results (e.g., click, add-to-cart, order). This helps to facilitate data analysis as well as model improvements to enhance the overall user experience.

The offline component is responsible for building and updating the product indices, and operates as a scheduled task. For the I2I embedding pre-calculation task, the system first perform product detection on the product images, and subsequently passes the cropped products to the I2I model to obtain the embeddings. For the MIEM embedding pre-calculation task, the system directly inputs the associated product images and titles into the model to obtain the embeddings. The embeddings from both tasks are added to their

¹<https://shopee.sg/>

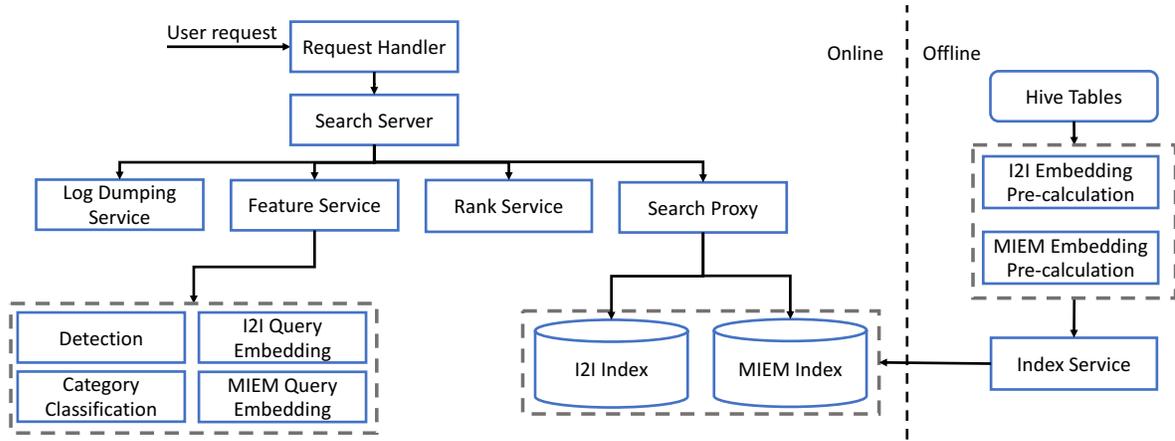


Figure 3: The integration of MIEM into the Shopee Image Search Engine. The arrows indicate the request triggering process.

respective index database using the index service. In addition, the index service manages the removal of indices corresponding to unavailable products.

Use of AI Technology

The proposed MIEM approach primarily utilizes user queries and their corresponding clicked products as the training data. In this section, we introduce the model design of MIEM and how to train this model.

The Design of MIEM

MIEM, as depicted in Figure 4, adopts a dual-tower structure comprising two distinct components: the query tower and the item tower. The query tower takes the customer’s uploaded query image as input and generates the corresponding query embedding as output. On the other hand, the item tower processes text and multiple product images as input and produces the item embedding. Notably, the image encoder is shared between the query and the item towers, thereby reducing the number of parameters required and improving the recall score. The details of the model design are as follows:

1. *Image Encoder*: the Swin Transformer (Liu et al. 2021b) is adopted to build the image encoder. It is a novel vision transformer model that employs hierarchical feature representations with linear computational complexity relative to input image size. Compared to the previous Vision Transformer (ViT) (Dosovitskiy et al. 2020), Swin Transformer has achieved state-of-the-art results in ImageNet classification, COCO object detection and semantic segmentation tasks. The Swin Transformer base is used in MIEM to strike a balance between accuracy and speed.
2. *Title Encoder*: following (Zeng, Zhang, and Li 2022), we design the title encoder based on the multilingual BERT base (Devlin et al. 2018).
3. *The Fusion Module*: the fusion module consists of 6 layers of merge attention networks (Dou et al. 2022). It concatenates patch-level embeddings from images and token-level embeddings from texts, and feeds them into a vanilla transformer model. Empirical investigations show

that this structure achieves better performance than using 6 layers of cross attention (which is commonly used in image-text pretrained models). While cross attention is suitable for ranking tasks, merge attention can better preserve information from both modalities, leading to improved recall scores in modality fusion.

4. *Projection Layer*: a single fully connected (FC) layer is used as the project layer. For image encoder projection, the average of patch-level embeddings is used as the input. For fusion encoder projection, the embedding of the [CLS] token is used as the input, thereby generating the final 128-dimensional feature.

In the training phase, we utilize search click logs to train the model. User queries are directed to the query tower, and the text and images of the clicked product are inputted into the item tower. To jointly train both towers, we employ the proposed loss functions and training methods, which will be described in detail below.

In the inference phase, the item tower computes the item embedding offline, while the query tower computes the query embedding in real-time during online queries. This setup enables image-to-product recall through vector-based approximate nearest neighbor (ANN) retrieval.

The Training Process of MIEM

A MIEM is trained separately for each country/region using PyTorch, and the training process involves three major steps. Following previous works (Zeng, Zhang, and Li 2022; Li et al. 2021, 2022, 2023), we use AdamW (Loshchilov and Hutter 2019) as the optimizer for all three steps.

Training the Image-Text Embedding Model In this step of training, the fusion module is not involved. The projected query embedding and the projected text embedding are directly aligned. The widely adopted InfoNCE loss approach (Oord, Li, and Vinyals 2018) is used for this alignment:

$$\mathcal{L}'(Q, T) = -\mathbb{E}_{i \in [0, N]} \left[\log \frac{e^{s(Q_i, T_i)}}{\sum_{j \in [0, N]} e^{s(Q_i, T_j)}} \right]. \quad (1)$$

Here, Q_i denotes the query image embedding. T_i represents the item title embedding. $S(Q_i, T_i)$ is the cosine similarity

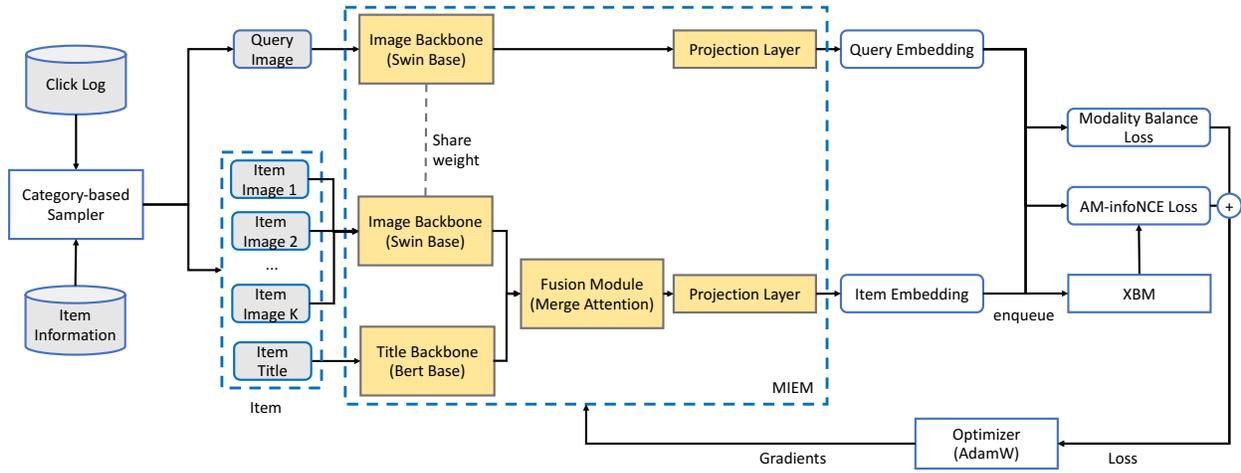


Figure 4: The system architecture of the AI Engine (MIEM). The yellow components are the trainable parts of the entire framework. The gray components are the training data.

between Q_i and T_i . N is the batch size.

Inspired by Cosface (Wang et al. 2018), we propose AM-InfoNCE to further enhance the model capability. AM-InfoNCE employs two hyperparameters, γ and m , to optimizing the decision boundary in the angular space by minimizing intra-class variance and maximizing inter-class variance. γ controls the overall size of the logits, while m controlling the margin between classes.

$$\mathcal{L}(Q, T) = -\mathbb{E}_{i \in [0, N]} \left[\log \frac{e^{\gamma(s(Q_i, T_i) - m)}}{e^{\gamma(s(Q_i, T_i) - m)} + \sum_{j \in [0, N], j \neq i} e^{\gamma s(Q_i, T_j)}} \right]. \quad (2)$$

It is important to note that this loss is different from Cosface (Wang et al. 2018). Cosface adopts a classification loss, computing the similarity between samples in a batch and the weights of each class (product) to obtain the loss. In contrast, AM-InfoNCE uses the query as a positive sample with clicked item titles and other items in the batch as negative samples to calculate the similarity.

Training with the Fusion Module In this step, the fusion module is involved in the training process. Only a single item image is used for training the entire model. Class-based sampling and Cross Batch Memory (XBM) (Wang et al. 2020) are used to improve performance. Class-based sampling ensures that each minibatch contains products of the same class, providing more hard negative samples to improve model performance over random sampling. XBM (Wang et al. 2020) is a mechanism that remembers embeddings from past iterations, thereby allowing the model to collect sufficient hard negative sample pairs across multiple batches. By connecting each anchor in the current batch with the closest embeddings in the nearest minibatch, XBM provides a large number of hard negative samples. Although XBM effectively mines hard negative samples, it is still limited by the small batch size training (Wang et al. 2020). Thus, we avoid direct training with multiple item images.

To ensure the preservation of alignment between the image and text feature space, we propose the modality balance loss. Under this loss, the fusion embedding without the item image and texts is separately aligned with the query embedding. In this way, the model can learn more robust feature information from individual modalities. The modality balance loss is computed as follows:

$$\begin{aligned} \mathcal{L}^{\text{balance}}(Q, I, T) = & \mathcal{L}(Q, F(I^{\text{def}}, T)) + \mathcal{L}(Q, F(I, T^{\text{def}})) \\ & + \mathcal{L}(F(I^{\text{def}}, T), Q) + \mathcal{L}(F(I, T^{\text{def}}), Q) \end{aligned} \quad (3)$$

where F is the fusion module. I^{def} and T^{def} are default item images and title embedding. Their corresponding tokens have all-zero attention values within the fusion module.

The final loss is the sum of modality balance loss (Eq. 3), AM-InfoNCE (Eq. 2) and XBM-enhanced AM-InfoNCE loss:

$$\begin{aligned} \mathcal{L}^{\text{final}}(Q, I, T) = & \mathcal{L}(Q, F(I, T)) + \mathcal{L}(F(I, T), Q) \\ & + \mathcal{L}^{\text{balance}}(Q, I, T) \\ & + \mathcal{L}(Q, F_m) + \mathcal{L}(F(I, T), Q_m), \end{aligned} \quad (4)$$

where F_m is the fusion embedding in the XBM and Q_m is query embedding in XBM (Wang et al. 2020).

Training with Multiple Item Images In this step, the entire model is trained with multiple item images. The loss function and training techniques used in Step 2 are maintained during this process. To facilitate batch training and inference, we set a fixed number of item images to K ($K > 1$). For items with an image count below K , we provide empty images to reach the required number of K . Conversely, for items with an image count exceeding K , we truncate the images to match the fixed number K .

After conducting experiments, we determined that setting $K = 4$ strikes a balance between inference speed and recall accuracy. This choice allows us to efficiently process batches

during training and inference while still achieving satisfactory recall performance, resulting in a practical and effective configuration for the model.

Acquisition of Training Data

We leverage user click logs to train MIEM. Each log entry is represented as a triplet (query image, clicked item title, clicked item image list). Here, the clicked item title and clicked item image list correspond to the titles and images of the products that users clicked after using the query image for search. While theoretically, users may click on completely unrelated products, through manual reviewing and analysis, we found that approximately 45% of the triplets consist of query-item pairs that are identical, 50% are on similar items, and less than 5% are on unrelated noisy products. Thus, the dataset is useful for our purpose.

We conjecture that MIEM should be able to learn even from the similar item triplets. The reasons are as follows:

1. From the perspective of the entire system, these similar items are products of interest to users. Even if they are not identical, they should be retrieved in the recall stage.
2. From a learning perspective, although similar item triplets do not all perfectly match, often the (query image, clicked item title) pairs correspond correctly, with only differences in images. Thus, the model can still learn useful associations between the queries and the products from these examples.

Thus, we retain this portion of the data containing similar item triplets. As for the 5% of completely unrelated data, we have not conducted specific cleansing for them at present. This could be a potential point for future extension.

Application Development and Deployment

MIEM has been deployed in the Shopee Image Search Engine, serving customers from Southeast Asia, Taiwan and Brazil since March, 2023. The deployment follows the process described in Figure 3. Throughout the development and deployment process, a collection of technologies has been developed to meet practical requirements.

Online Query Feature Extraction

To improve the online service delivery efficiency, the image encoder (a PyTorch (Paszke et al. 2019) model) is converted into a TensorRT (Vanholder 2016) model and integrated into the online feature extraction service. This service takes images and bounding boxes predicted by the detection service as inputs, processes the images using the TensorRT model, and outputs the projected image embeddings. TensorRT optimizes the neural networks based on the GPU, leading to reduced service latency.

Offline Item Feature Extraction

Similarly, the item tower service is built on a TensorRT model. In this case, the item images no longer require bounding boxes. In addition, the service takes the item titles as inputs and outputs the projected item embeddings.

To enhance online service efficiency, we pre-calculate the embeddings for items in the database. To handle the vast size of the product database and the high number of daily addition/removal of items, we design the following pipeline (as shown in Figure 5). We maintain a full-feature Hive table for all items, partitioned by day. Eventually, the features are inputted into the MIEM Elasticsearch (ES) (Gormley and Tong 2015) Index. This process is executed daily as follows:

1. The full-feature partition from the previous day is first copied into the current day partition. Deleted items and items with changed information (e.g., images, titles) are not copied. This is achieved by comparing the upstream product information table with the full-feature table from the previous day.
2. The information of newly added or updated items is collected and sent to a PySpark (Drabas and Lee 2017) task. This task efficiently obtains embeddings through gRPC calls (Wang, Zhao, and Zhu 1993) to the item embedding service, and writes them into the current day partition. Items with unreadable images or titles containing only punctuation marks or spaces are not written into the partition.
3. We generate commands to update the ES index by comparing the current day and previous day full-feature item tables. We define two actions: delete and update. The delete action removes items that exist in the previous day partition, but not in the current day partition. The update action includes newly added or updated items for which the service successfully obtained features. These commands are sent to the Index service to update the ES index (Gormley and Tong 2015).

It is worth noting that although the full-feature Hive table appears redundant with the ES index, it is essential for two reasons. Firstly, we cannot view item features in the ES index, making debugging difficult. Secondly, due to limitations of the HNSW algorithm, the ES index requires periodic reconstruction. Otherwise, the recall score declines with the increasing number of updated items. Maintaining the full-feature Hive table facilitates the reconstruction process.

Online Same Item Rate Analysis

To analyze the online same item rate, we periodically use outsourced annotation to detect changes in the same item rate. The annotation task provides a query image, the top 5 product images from the search results, and the product

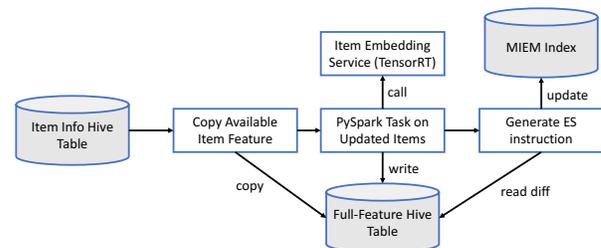


Figure 5: MIEM item embedding pre-calculation.

titles. The annotators determine whether the query and the products are the same, similar or dissimilar. To facilitate annotation, we use a machine translation model to translate the product titles into the native languages of the annotators, along with the original titles. “Same” indicates that the query and the product are the same, while “similar” means that they belong to the same category (e.g., both are sweaters or dresses). Since user queries may include products that do not exist in the Shopee product catalog, evaluating the number of similar items allows us to assess whether the system includes similar products in search results when identical items are not available, thereby better satisfying users’ search needs.

Application Use and Payoff

Offline Evaluation

Offline evaluation serves as a useful evidence collection step to help Shopee management make the decision to replace the previous item embedding model with MIEM. To evaluate the performance of MIEM at this stage, a dataset based on real-world user queries collected from the Indonesia market was constructed, containing user query images on 3 million Shopee products. Since some products may have identical items (same physical products with different IDs), a same item merging strategy was employed. For each product image a , the top- K product images b were obtained using an ANN approach. An MLP model was used to determine whether a and b are the same item. This MLP model takes the concatenation of the features of a and b as the input, and outputs the probability of them being the same item. The training data for the MLP model was annotated by external annotators. Moreover, since the products the users click on are not necessarily the same items, the features of the user queries in the 3 million product database were retrieved using I2I and MIEM. These features were then annotated by external annotators to obtain accurate same item information to improve the accuracy of evaluation. A total of 9,000 query images were retained after removing different-item queries.

Two metrics were used in the offline evaluation:

- *Recall@k*: commonly used in deep metric learning related papers (Liu et al. 2021a; Sun et al. 2020; Wang et al. 2019; Cakir et al. 2019; Qian et al. 2019; Ibrahim et al. 2022) to determine whether the correct results are included in the top- K retrieved results, Recall@k directly reflects model performance in terms of same item rate.
- *Category Accuracy*: evaluates whether the most common category among the top 10 results matches the correct product category.

MIEM was compared against the following models:

- **I2I**: Shopee’s previously used image representation model, based on the ViT base. To handle the issue of multiple images per product, the KNNs of the images were obtained with the duplicates removed, ensuring that only the most relevant image was kept as the representative of the product.
- **I2T**: A CLIP (Radford et al. 2021) model that performs image-to-text retrieval. It is the output of the first step in

MIEM training, without the fusion encoder.

The following variants of MIEM were involved in offline evaluation:

- **MIEM (1 emb/img)**: An embedding is generated for each $\langle \text{image}, \text{text} \rangle$ pair about a product. This is a fair comparison with I2I. However, this strategy did not achieve the desired storage reduction.
- **MIEM+I2I**: Both models were used for multi-modal recall, and the scores were fused as $(\text{I2I score} + \text{weight} \times \text{MIEM score})$, with the weight adjustable based on performance on the test set.

The retrieval scores are reported in Table 1. Compared to I2I, I2T slightly improved category accuracy with significantly reduced recall score. This is due to the ambiguity of text descriptions, making it difficult to accurately describe product textures, patterns and styles. MIEM achieved slightly lower Recall@1 than I2I, but outperformed it under Recall@5 and beyond. In the context of recall stage, the slight drop in Recall@1 is acceptable. MIEM (1 emb/img) achieved more significant improvements in Recall@5 and beyond compared to MIEM, with comparable Recall@1 to I2I. It also achieved the highest category accuracy among all comparison approaches. However, since this strategy did not achieve the desired storage savings, it was not eventually selected for online deployment. Using both I2I and MIEM for multi-modal recall yielded the best results across the board, demonstrating that I2I and MIEM have complementary strengths. Combining both models maximized the same item rate improvement. Based on these results, the decision on deploying MIEM+I2I in the Shopee Image Search Engine was eventually made.

Online Evaluation

Since its deployment in March 2023, A/B testing was conducted at Shopee’s Indonesia and Brazil e-commerce marketplaces. In the online evaluation, four types of metrics were utilized:

- *Same & Similar Item Recall@4*: This metric determines whether the correct results are present among the top-4 retrieved items. The determination of whether an item is considered the same or similar to the query is made by annotators, rather than relying on ground truth data used in offline evaluation. The top-4 items were assessed because, in the Shopee App, the first page of search results displays these top 4 items.
- *Top4 Irrelevant Item Rate*: This metric evaluates the rate of irrelevant items found in the top 4 search results. Annotators are responsible for determining the relevance of items. Any items that are neither considered the same nor similar to the query are classified as irrelevant.
- *Clicks/User*: This metric measures the number of items clicked per user within the search results. It provides insights into user engagement with the presented search results.
- *Orders/User*: This metric measures the number of orders placed per user within the image search. It helps assess

Model	Category Accuracy	Recall@1	Recall@5	Recall@10	Recall@50	Recall@100
I2I	0.874	0.820	0.922	0.951	0.979	0.985
I2T	0.876	0.395	0.632	0.714	0.856	0.896
MIEM	0.881	0.812	0.927	0.952	0.984	0.988
MIEM (1 emb/img)	0.894	0.819	0.939	0.964	0.988	0.990
MIEM+I2I	0.879	0.872	0.953	0.969	0.984	0.985

Table 1: The offline evaluation results on a Shopee product test set with 3 million items. The latest proposed solution of combining I2I with MIEM achieved the best performance.

	A/B Test Period	Group	Clicks/User	Δ Clicks/User	Orders/User	Δ Orders/User
Shopee Indonesia	31/03/2023	I2I	3.84		0.0804	
	-10/04/2023	MIEM+I2I	4.22	+9.90%	0.0838	+4.23%
Shopee Brazil	20/04/2023	I2I	4.00		0.0364	
	-05/05/2023	MIEM+I2I	4.24	+6.00%	0.0382	+4.95%

Table 2: A/B test results of the Shopee Image Search Engine in Indonesia and Brazil. Δ refers to the relative improvements achieved by MIEM+I2I compared to I2I.

Model	Same Item Recall@4	Similar Item Recall@4	Top 4 Irrelevant Rate
I2I	50.55%	93.26%	13.58%
MIEM+I2I	53.66%	95.60%	10.71%

Table 3: A/B test results on Same & Similar Item Recall@4 and Irrelevant Rate for the top 4 products from Shopee Indonesia.

the effectiveness of the image search in driving user purchases.

The results are shown in Table 2. It can be observed that the new I2I+MIEM solution for multi-modal image search recall significantly increased clicks per user and orders per user compared to the previous deployed solution of using I2I alone. Meanwhile, according to the online similarity rate analysis method, we collected 3,000 queries from Shopee Indonesia, and evaluated the top4 same and similarity item rates. The evaluation results in Table 3 show that MIEM significantly increases the same and similarity item rates, while reducing the proportion of the top4 irrelevant products. Furthermore, when compared to the image embedding model, MIEM significantly reduces the storage space requirements. In particular, for the online ES index, MIEM only requires a approximately 61.17% less storage space compared to the previously adopted I2I index.

Qualitative Analysis

Figure 6 illustrates examples of the top 4 online cases. In the first example, the query is for Indonesian traditional attire. The I2I model mistakenly returns results for protective clothing. However, after deploying MIEM, this issue is significantly improved, and more relevant results are obtained. In the second example, the query is for a body spray. The I2I model returns results that resemble boxes of cakes. MIEM,

focusing not only on image details but also on semantic information, accurately identifies the correct product, resulting in more precise search results. In the third case, the query is for a speaker. MIEM successfully delivers top 4 results that match the appearance of the query image. In the last example, without the multi-modal approach, the I2I model retrieves an irrelevant product (the third item), mistaking the cake box for the actual product which is bottled water. The multi-modal capability of MIEM helps resolve such issues by considering both text and image information, leading to more accurate and contextually relevant recommendations.

Lessons Learned During Development, Deployment, and Maintenance

Throughout the development and deployment process, several lessons were learned:

1. Some products are challenging for MIEM to handle. Products with many SKUs (e.g., phone cases with different patterns) may be difficult to cover with a single embedding. One potential solution is to create multiple embeddings for such products based on the number of SKUs or the degree of variation in the product images. Methods like softtriple (Qian et al. 2019) are promising solutions.
2. In terms of deployment, achieving a balance between speed and accuracy requires the use of TensorRT. However, attention should also be paid to the pre-processing speed. In addition, the overall system outputs should be consist with the offline evaluation results.
3. In terms of model maintenance, there are a large number of new products added to the Shopee database daily. The previously adopted I2I may be less affected by new products due to its focus on image details alone. In contrast, MIEM, which focuses on both the semantic information and the images, may struggle to establish relationships between texts and images for products not seen during training (e.g., newly released iPhones). Regularly retraining the model can help mitigate this issue. Nevertheless,



Figure 6: Qualitative comparison of top4 search results using different queries under MIEM+I2I vs. I2I.

the improvement has been deemed to outweigh the maintenance overhead incurred.

Conclusions and Future Work

The proposed Multi-modal Item Embedding Model (MIEM) complements Shopee’s image search by providing semantic information for both images and product texts. By considering both images and text for a product and extracting valuable information, MIEM constructs more context-aware feature representations for products. Compared to the previously deployed single modal image search model, MIEM achieves higher the recall score while requiring significantly less feature storage space compared to using Image-to-image only. The model has become an essential part of the Shopee Image Search Engine. Since its deployment in March 2023, it has help the Shopee e-commerce platform achieve 9.90% higher clicks per user and a 4.23% higher orders per user, thereby bringing significant benefits to its businesses.

In subsequent research, we will explore how to enhance MIEM from the perspective of data privacy preservation via federated learning (Yang et al. 2020; Goebel et al. 2023) to further improve user experience.

Acknowledgements

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); and the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore.

References

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’19)*, 1861–1870.

Cheng, Z.; Xiao, S.; Zhai, Z.; Zeng, X.; and Huang, W. 2023. Mixer: Image to Multi-Modal Retrieval Learning for Industrial Application. *arXiv preprint arXiv:2305.03972*.

Chicco, D. 2021. Siamese neural networks: An overview. *Artificial Neural Networks*, 73–94.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dou, Z.-Y.; Xu, Y.; Gan, Z.; Wang, J.; Wang, S.; Wang, L.; Zhu, C.; Zhang, P.; Yuan, L.; Peng, N.; et al. 2022. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’22)*, 18166–18176.

Drabas, T.; and Lee, D. 2017. *Learning PySpark*. Packt Publishing Ltd.

Fix, E.; and Hodges, J. L. 1989. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3): 238–247.

- Goebel, R.; Yu, H.; Faltings, B.; Fan, L.; and Xiong, Z. 2023. *Trustworthy Federated Learning*, volume 13448. Springer, Cham.
- Gormley, C.; and Tong, Z. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine.* ” O’Reilly Media, Inc.”.
- Ibrahimi, S.; Sors, A.; de Rezende, R. S.; and Clinchant, S. 2022. Learning with label noise for image retrieval by selecting interactions. In *Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV’22)*, 2181–2190.
- Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 12888–12900. PMLR.
- Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; and Hoi, S. C. H. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705.
- Liu, C.; Yu, H.; Li, B.; Shen, Z.; Gao, Z.; Ren, P.; Xie, X.; Cui, L.; and Miao, C. 2021a. Noise-resistant deep metric learning with ranking-based instance selection. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’21)*, 6811–6820.
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV’21)*, 10012–10022.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled weight decay regularization. In *ICLR*.
- Malkov, Y. A.; and Yashunin, D. A. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4): 824–836.
- Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- Qian, Q.; Shang, L.; Sun, B.; Hu, J.; Li, H.; and Jin, R. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV’19)*, 6450–6458.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML’21)*, 8748–8763.
- Song, L.; Pan, P.; Zhao, K.; Yang, H.; Chen, Y.; Zhang, Y.; Xu, Y.; and Jin, R. 2020. Large-scale training system for 100-million classification at alibaba. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD’20)*, 2909–2930.
- Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’20)*, 6398–6407.
- Vanholder, H. 2016. Efficient inference with tensorsrt. In *GPU Technology Conference*, volume 1.
- Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 5265–5274.
- Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’19)*, 5022–5030.
- Wang, X.; Zhang, H.; Huang, W.; and Scott, M. R. 2020. Cross-batch memory for embedding learning. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’20)*, 6388–6397.
- Wang, X.; Zhao, H.; and Zhu, J. 1993. GRPC: A communication cooperation mechanism in distributed systems. *ACM SIGOPS Operating Systems Review*, 27(3): 75–86.
- Yang, Q.; Liu, Y.; Cheng, Y.; Kang, Y.; Chen, T.; and Yu, H., eds. 2020. *Federated Learning*. Springer, Cham.
- Zeng, Y.; Zhang, X.; and Li, H. 2022. Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts. In *Proceedings of the 39th International Conference on Machine Learning (ICML’22)*, 25994–26009.
- Zhang, Y.; Pan, P.; Zheng, Y.; Zhao, K.; Zhang, Y.; Ren, X.; and Jin, R. 2018. Visual search at alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD’18)*, 993–1001.