2021 International Conference on Computational Science and Computational Intelligence (CSCI) | 978-1-6654-5841-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSCI54926.2021.00062

Visually Compatible Home Decor Recommendations Using Object Detection and Product Matching

Unaiza Ahsan
The Home Depot
unaiza_ahsan@homedepot.com

Yuanbo Wang
The Home Depot
yuanbo_wang@homedepot.com

Alexander Guo
The Home Depot
alexander_guo@homedepot.com

Kevin D. Tynes Jr.
Georgia Institute of Technology
kdtynes@gatech.edu

Tianlong Xu
The Home Depot
tianlong_xu@homedepot.com

Estelle Afshar
The Home Depot
estelle_afshar@homedepot.com

Xiquan Cui
The Home Depot
xiquan_cui@homedepot.com

***Abstract*—Automatically recommending visually compatible products to Home Decor shoppers is a challenging task for e-commerce companies in the home improvement domain. However, few satisfactory solutions have been proposed to address this problem. In this paper, we propose a novel approach that uses a room scene image as the primary data source to generate visually compatible product recommendations. More specifically, we first detect products shown in the room scene image. Then, we use image retrieval techniques (e.g. color matching and triplet contrastive learning) to find the most similar products, if not the same, from the catalog. The system is designed to scale up to millions of products. To evaluate the performance of the proposed approach under various scenarios and use cases, we test it on several decor datasets, e.g. the catalog of a large home improvement retailer, a sizable public decor dataset, and customer-generated images posted on product reviews. We compare the approaches and determine that the triplet contrastive learning outperforms color matching for image retrieval. When tested live on the retailer's website, this new experience increases by 1.5% the user engagement and by 4.0% the Average Order Value (AOV) of transactions.**

***Index Terms*—recommender systems, contrastive learning, object detection, image retrieval**

I. Introduction

Recommending visually compatible products in e-commerce is a challenging problem, e.g. when recommending furniture, decorative items and finishing touches to create a room. Online retailers have catalogs containing millions of products across hundreds of categories and these products have attributes of many modalities (e.g text, images, etc.). Visual assets associated with these products are typically shown to the customer in the form of product images and products in context such as scene images (see Figure 1). The core insight we use in our solution is that scene images contain multiple *other* products that are visually compatible with each other. This insight allows for the hypothesis that object detection on scene images and retrieving similar (or exact) products from the catalog can result in a visually compatible set of recommendations.

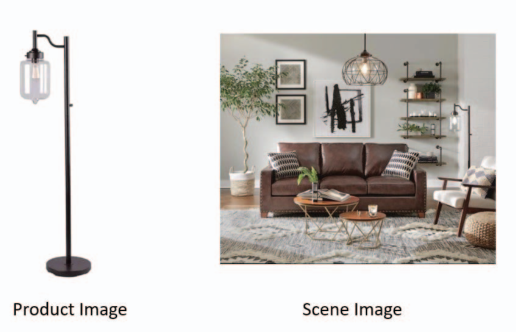

Fig. 1. Example of a product (left) and a scene (right) image of a floor lamp

The first step of the proposed approach is *object detection*. Although it carries its own challenges specific to the decor domain i.e. the occurrence of shades and occlusions in the scene image, we fine-tune pretrained object detection models on categories of interest to increase the granularity of categories (e.g. dining chair, desk chair, etc.) and the detection performance. In this paper, we will focus on the second step of our approach - *visual search*, also called product matching. We define visual search as the process of matching and retrieving exact or similar home decor products from the ones detected in a scene image. This problem is highly challenging because even though object detection gives the broad category of products, e.g a ”chandelier”, other visual aspects i.e. types, styles, shapes, colors and color finishes need to be accurately matched.

We use two matching methods. In the first method, we compute the color features on the cropped products detected in the scene image and match based on cosine similarity. This method is our baseline. The second method utilizes triplet networks to learn a distance metric between embeddings. After training the triplet network, we use the model to predict the most similar image to an input cropped product from the scene

978-1-6654-5841-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CSCI54926.2021.00062

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

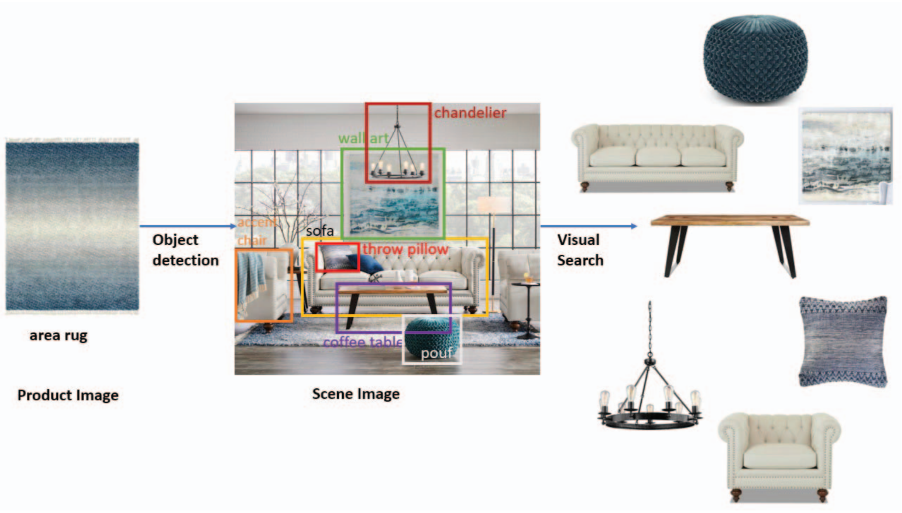

Fig. 2. Given a product, we propose a system to detect objects in its scene images and retrieve either exact or similar *other* products based on detected objects as visually compatible recommendations. For an area rug, the final recommendations are shown on the right.

image. Our main contributions are:

1) A framework composed of object detection and image retrieval to generate complementary recommendations for a given product in the home decor domain by leveraging the room scene image as the primary data source, and using a two-step approach (object detection and matching) to retrieve complementary recommendations for a given product.
2) The comparison of two matching methods to retrieve products; exact products, if available and similar products if exact products are not available in the catalog.
3) The scalability of our system across millions of products and its robustness to catalog updates.
4) The report of significant increase in business performance when the experience is customer facing with A/B test.

II. Related Work

With the phenomenal success and widespread applications of convolutional neural networks in computer vision in the past decade, visual search has powered complementary product recommendation. Zhang *et al.* [1] demonstrate how their model learns product image features for visual search by linking customer clicks data at Alibaba. Yang *et al.* [2] and Zuo *et al.* [3] describe scalable frameworks at eBay and Amazon respectively to train visual search deep learning models and store the representations. Shankar *et al.* [4] present the advantages of adopting a number of triplet VisNet networks to learn product image embeddings across several semantic categories for Ipkart. Hu *et al.* [5] report how they leverage a cascaded learning-to-rank framework to balance relevancy, latency, and scalability while building the visual search engine at Bing. Finally, Kang *et al.* [6] introduce a novel way to extract product information from scene images with attention models in their complete the look project at Pinterest. Our work is similar in intent to the above mentioned approaches but closest to [6]. The main difference in our work is the home decor domain where we cannot exploit side information such as human pose detection in fashion domain. Different from prior approaches, we also do not use any customer behavior data (such as clicks) in our approach, driving it purely on the basis of visual similarity. Shiau *et al.* [7] present their visual search system at Pinterest which also comprises of object detection and embeddings via multi-task learning [8] for retrieval. However, it is hard to compare results directly because the embeddings are not made public.

One of our proposed approaches for visual search uses a triplet network to learn a distance metric between products. Similarly, Wang *et al.* [9] show that triplet networks learn fine-grained image similarity. Their work was further expanded on in FaceNet [10] which experimented with deeper convolutional neural networks with triplet loss. Bell *et al.* [11] propose a Siamese contrastive learning approach for visual search for in-home products. They generate the training set manually. Different from their approach, our framework uses object detection models on room scene images to automatically generate the training set for the triplet network, which is particularly helpful in situations where time and resources are limited. Multiple studies have demonstrated the superiority of using triplet loss for metric learning as opposed to other methods [12], [13].

The problem of retrieving products from real world images is also studied in the domain of fashion to retrieve similar [14] or exact [15] clothing items detected in the image. Our work is different in that we do not assume that bounding boxes are given. We train object detection models and then detect the objects in the scene images.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

III. Approach

Our approach (see Figure 2) consists of two main steps: Object Detection on Scene Images and Visual Search for matching detected objects to recommend similar or exact ones.

A. Object Detection on Scene Images

Object detection is a well established field in Computer Vision and the goal is to detect objects in the image via bounding boxes. The challenge here is that our home scene images may be cluttered and occluded and hence, furniture and decor products may be partially visible (e.g. the scene image in Figure 2).

1) Generating Training Data: We sample scene images from 8 different rooms (bedroom, bath, kitchen, dining, entryway, patio, office and living room) and leverage a crowd-sourced annotation tool to label these images. Specifically, we provide contributors with a set of object category labels per room and ask them to draw tight bounding boxes for each occurrence of each category in the images. The resulting images along with the bounding box coordinates make up our training set. We train one object detection model per room.

2) Predicting Bounding Boxes: Once we train the object detection models for each room, we pass all the scene images of the different categories through the models. We obtain bounding box predictions for the images and begin the process of matching the bounding boxes and predicted label to the catalog to retrieve the best match. Our object detection pipeline is shown in Figure 3.

B. Visual Search

In this section we describe the process of visual search or matching which maps predicted objects in scene images to product images or vice versa. We describe two main approaches: a color matching approach and a triplet network contrastive approach for visual search.

1) Matching via Color Features: As a baseline, we compute color features for each product image in our dataset. Our goal is to retrieve the same or closest matching product image to the cropped bounding box in the scene image. Specifically, we generate the RGB color histograms for each image. Because product images often have a clear white background and the object of interest centered, computing the pixel color distribution seems like a natural first choice. Having said that, the predicted bounding box crop in the scene image consists of the detected object but also the background and/or occluding objects. Therefore, we need to separate the foreground from the background and compute the color histogram on the resulting foreground mask. To isolate the foreground from the background, we apply mean adaptive thresholding [17].

Given a grayscale image G which has value $G_{old}(p,q)$ at pixel location (p,q), the threshold $thresh(p,q)$ is computed as the average of a $m \times m$ window neighboring (p,q) where z is a constant subtracted from the mean and m is the block size. The threshold value $thresh(p,q)$ is calculated for each pixel location (p,q) and the image is threshold accordingly.

Hence Equation 1 gives the formula to adaptively compute the threshold, and the new intensity values of the pixel based on the threshold. If the pixel value is above the threshold, then the maximum value is given by max. We set $max = 255$ and the resulting image is a binary image with a mask on the foreground region. The color histogram is then computed for only the pixels under this masked region.

$$G_{new}(p,q) = \begin{cases} max, & \text{if } G_{old}(x,y) > thresh(p,q) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

2) Matching via Triplet Networks: While using the color distribution of the cropped images can be a fast and efficient way to match the crops to products, it ignores other important visual cues and semantic information i.e. properties like shape and style. Since our objects of interest can occur in scenes at different angles and orientations, producing an embedding through traditional means (such as hand-tuned features) can be inaccurate and make it difficult to scale to larger, more noisy sets.

Thus, we use a deep CNN, specifically a pretrained ResNet-50 [18], to map images into a semantic space which captures high level information. The network is a function f with learned parameters Θ, and maps an image I to an embedding $x = f(I;\Theta)$ in $\mathbb{R}^D$ (in our case, $D = 2048$). Ideally, the learned parameters Θ enable features from images containing the same product to be mapped closer in the embedding space than features of dissimilar images.

In order to accomplish this, we use a triplet network architecture and a triplet loss for training, similar to FaceNet [10]. In this way, we can learn all of our parameters end-to-end from the image to the output embedding. Specifically, consider three images (I_a, I_p, I_n) in our training set where I_a, I_p contain the same product and I_a, I_n contain different products. Note that I_a is a cropped image of a product that occurs in a scene, whereas I_p and I_n are product images with white backgrounds. The loss function is given by:

$$\mathcal{L}(\Theta) = \sum_{(I_a,I_p,I_n)} \left[d(I_a,I_p) - d(I_a,I_n) + \alpha\right]_+ \tag{2}$$

$$\text{where} \quad d(I_x,I_y) = \|f(I_x,\Theta) - f(I_y,\Theta)\|_2 \tag{3}$$

The term α represents the margin enforced between the positive and negative pairs, and serves as a penalty to the loss. Additionally, if we were to loop through all possible triplets, many of them would be useless to training because they would already fulfill the margin requirement, contributing zero loss. Intuitively, the model would better learn to associate images with the same product if we fed it **hard** triplets where the negative image looks similar to the product in question, as shown in Figure 4. Hermans *et al.* [12] show that choosing semi-hard triplets (batch hard) outperform other sampling methods when training.

We employ a modified but similar technique where we sample from the hardest percentile of triplets. Given an anchor image I_a in a batch, we randomly choose the positive example

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

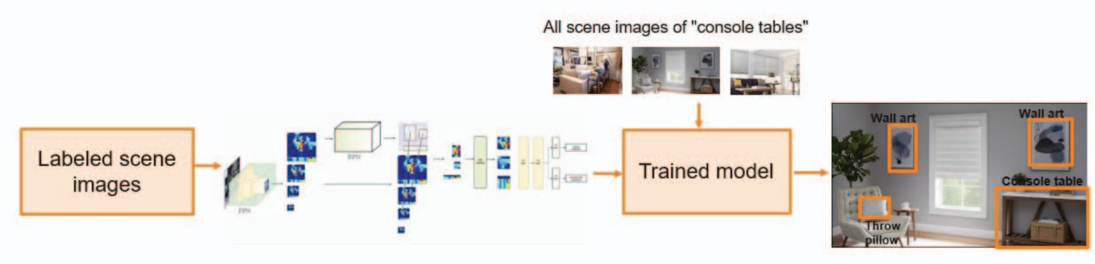

Fig. 3. Object Detection and Prediction on Scene Images. Network diagram taken from [16]

I_p since each product typically only has a couple of images. Then, we sample a random negative image that is among the top 25% hardest. In other words we choose I_n such that $d(I_a, I_n)$ from Equation (3) is sufficiently small. An alternative sampling technique is to simply take the argmin of a smaller subset or batch. Although it is more efficient, we decided against it because we preferred greater variance in our training triplets.

Once the model is trained, we determine the best match by mapping a query scene crop to its corresponding embedding using the learned parameters. Then, we use our distance metric to run k-nearest neighbors on our catalogue's pre-computed embeddings to return the top k matches. This technique is shown in Figure 5.

3) Exact Match vs Similar Match: Our goal for this paper is to retrieve the exact product that is shown in a scene image from the database. In Figure 2, the example shown is of an exact match. However, e-commerce databases are usually large and constantly evolving. Therefore, it is possible that the exact product may be out of stock or discontinued. If that is the case, our goal is to show the next best product, which should be similar to the product in the scene image. We call retrieval of such a product from the database: Similar Match. The challenge in retrieving similar products is to define a metric for visual compatibility; while keeping in mind, that objects may be occluded or partially in view in the scene image.

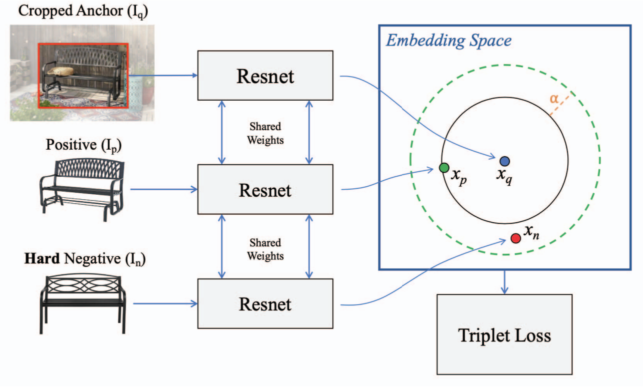

Fig. 4. Triplet Network training technique. We sample our triplets such that the negative embedding (x_n) is close to the anchor (x_q). Ideally, x_n should fall within the margin denoted by the dotted green circle. In this way, it contributes loss which enables learning.

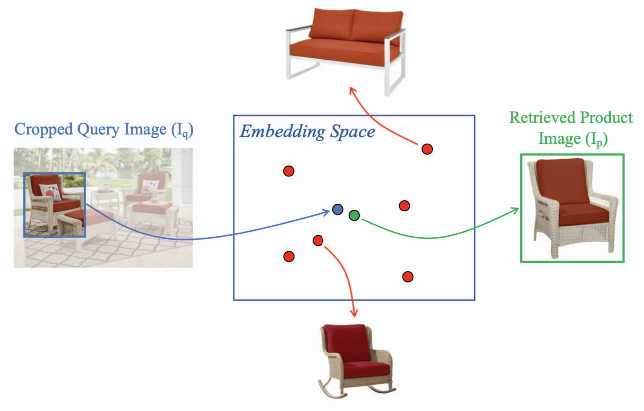

Fig. 5. Returning an item's best match using the triplet network. The neighboring embeddings correspond to products that are stylistically similar to the query object.

IV. Experiments

In this section, we describe the datasets we use in our experiments, the experimental setup and relevant details.

A. Datasets

Dataset Name	**Source**	**# Scenes**	**# Products**	**# Scene-Product Pairs**
Triplet Dataset	Online Retailer	$\approx 300k$	$\approx 500k$	$\approx 100k$
STL Dataset [19]	Pinterest	24,022	41,306	93,274
Customer Review Dataset	Online Retailer	3,137	2,248	3,137

TABLE I
Datasets we use to evaluate Visual Search

a) Object Detection Dataset: We sample scene images from the dataset of a large home improvement retailer. High level category information is provided in large e-commerce datasets e.g Bedroom. Using this information, we divide the scene images into eight different categories: bedroom, bath, kitchen, dining, entryway, patio, office and living room. We manually curate labels of objects that belong to each room. We upload these images to a third party vendor platform, where we ask contributors to manually generate bounding boxes with labels. To maintain the quality of this process, we generate hidden test questions. In other words, we check that their labeled boxes fall within a certain intersection over

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

union (IoU) threshold and their contributions are removed if they fail to achieve a certain accuracy. In the end, we have approximately 200k labeled bounding boxes with over 100 different categories spread across the eight rooms.

b) Triplet Dataset

Training: In order to create the training set for the triplet network to do visual search, we use the dataset provided by a large home improvement retailer. For this, we need positive scene-product pairs (a cropped product from scene image and a white background actual product image) as well as negative pairs. First, we sample all products that have both a white background product image and an associated scene image. We use our trained object detection models to crop the product from the scene image. We further divide this entire dataset into smaller datasets based on the category of the product. As mentioned before, high level category information is available in e-commerce datasets. For example, all products with categories related to chairs (i.e. accent chairs, rocking chairs, benches) would belong to a single dataset. These datasets are then used to train individual triplet networks that are specialized for matching products within a certain group. Our motivation is to take advantage of the pre-existing category names on the e-commerce website and produce more accurate results. At the same time, we are able to make the groupings broad enough such that scaling the system to new categories can be done easily. On average, the number of pairs of product-scene images in each of these datasets is $\approx 10k$.

c) Evaluation: We use the hold-out test set from the triplet training data to evaluate visual search performance. We randomly sample 10 different categories and 100 scene-product image pairs per category for evaluation. Note that we calculate accuracy scores on these categories individually instead of averaging. This is because our typical search space for retrieval is within the (broad) category predicted by object detection models. Including unrelated categories in the search space is not only inefficient but also decreases retrieval performance.

d) STL Dataset: The Shop The Look (STL) dataset provided by [19] is obtained from Pinterest and is composed of STL-Fashion (fashion scene-product image pairs), and STL-Home (interior design and home decor scene-product image pairs) [19]. The dataset also contains product categories and bounding box coordinates of products in each scene. However, while many product images match exactly with their counterparts in the scene images, some product images only resemble the cropped products from the scene images (see Figure 6).

e) Customer Review Dataset: Visual matching methods are often trained and tested on images of manually curated scenes which eliminates the noise and complexity found in a real world visual search scenario. To simulate a more realistic application, we evaluate our visual search framework on an image dataset submitted by customers when reviewing products from a home improvement retailer. Although customers may have attached several images to their product review, we use the first image with the empirical assumption that customers are more likely to first include a holistic image of the product or scene followed by images of smaller details. We then take a random uniform sample across 10 categories and use object detection (previously trained) models to remove images not containing home decor products. Unlike other curated datasets, this evaluation set has a larger diversity of image properties such as lighting, distortion, and pose. An example scene-product pair from this dataset is shown in Figure 6.

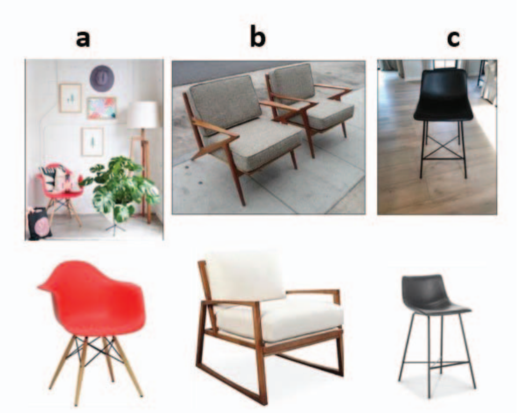

Fig. 6. Example of a scene (top) product (bottom) pair with exact match (a) and similar match (b) from STL Dataset [19] and an example from customer review dataset (c).

B. Training Object Detection Models

We use *detectron2* [20] by Facebook AI Research to train object detection models. We choose Faster R-CNN [21] with ResNet-101 [18] with FPN [22] backbone. This model is pretrained on MS COCO dataset [23], and we finetune this model using our in-house dataset. Our training runs for 50,000 epochs with a base learning rate of 0.00025.

C. Training Triplet Networks

We train the triplet networks using triplet loss and semi-hard triplets. When choosing a margin value for the loss function, we do not want the model to converge slowly or diverge during training. We choose a margin value of $\alpha = 1$. Our model is ResNet50 [18] pretrained on ImageNet [24]. We remove the final classification layer so the output is a 2048 dimensional vector. We train the model using Stochastic Gradient Descent using 1e-3 learning rate, 1e-3 weight decay, and 0.9 momentum.

V. Results

In this section we provide quantitative results for both similar and exact match retrieval tasks.

A. Retrieving Exact Matching Products

Recall that in visual search our goal is to retrieve the exact match in the dataset given a product cropped bounding box from the scene image. Our evaluation metric is the top-k accuracy, which is expressed as the percentage of times the exact product is returned in at most the kth position. We select both $k = 1$ and $k = 5$ during evaluation.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

	Color		Pretrained		Triplet	
Category	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
Bar Stools	26	45	53	76	**72**	89
Sofas	45	70	10	24	**69**	91
Bathroom Vanities	21	41	32	44	**58**	79
Bathroom Faucets	5	22	35	60	**56**	78
Accent Chairs	9	29	27	56	**53**	75
End Tables	5	19	32	54	**52**	84
Table Lamps	1	11	29	57	**48**	84
Ceiling Fans	21	37	24	48	**45**	62
Beds	26	41	25	31	**39**	66
Kitchen Faucets	4	13	25	55	**37**	61

TABLE II
PERFORMANCE OF OUR PROPOSED VISUAL SEARCH TECHNIQUES ON THE TRIPLET TEST DATASET

	Color		Pretrained		Triplet	
Category	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
Chairs	2.63	9.02	19.17	48.50	**33.08**	66.54
Ceiling Light Fixtures	2.78	6.11	10.55	23.33	**17.22**	48.33
Lamps	4.46	14.73	5.36	24.11	**16.52**	45.98
Table & Bar Stools	1.03	11.03	8.97	35.52	**12.07**	25.52
Faucets	0.72	6.45	3.60	20.14	**6.12**	17.27
Sofas	1.01	9.12	3.38	14.19	**3.72**	11.49

TABLE III
PERFORMANCE OF OUR PROPOSED VISUAL SEARCH TECHNIQUES ON THE STL DATASET

a) Triplet Test Dataset: On the Triplet test dataset, table II shows that the proposed triplet contrastive learning approach substantially outperforms the two baselines. Note that the pretrained network baseline method refers to matching of pretrained ResNet-50 embeddings for the scene crops and associated product images using cosine similarity. Predictably, there is a large difference in performance between using color features and a trained triplet network for matching the cropped images to the products. Interestingly enough, we have observed that for some categories, such as sofas and beds, the color matching approach outperforms pretrained network feature matching in retrieval accuracy. This is probably because larger furniture is relatively unoccluded which may help the color feature approach as it is more sensitive to occlusions. Therefore, smaller decor categories such as table lamps and faucets suffer from this issue, resulting poor performance when using the color feature based retrieval method.

b) STL Dataset: Table III shows the evaluation results on the STL dataset. For each category, we evaluate on scene-product image pairs taken from 100 distinct products. The number of pairs across each category varies, since there are multiple scene images associated with individual products. Overall, the trained triplet network outperforms the baselines in top-1 accuracy. As previously stated, some matches in the dataset are not necessarily exact matches, which may explain why the pretrained network had comparable accuracy in some categories.

B. Retrieving Similar Products

When exact product does not exist in the catalog, visual search can still be performed by retrieving the most visually similar product. It is non-trivial to quantitatively evaluate matching techniques to retrieve similar products with top-k accuracy. However, we can qualitatively evaluate our results. To do this, we conduct a user study in which contributors select the best stylistic match. We provide them with a cropped image of an item in a scene, and they must select one of four product matches. The four choices are: trained triplet network, pretrained network, color matching, and random matching. Then, we collect 3-5 responses for each individual question and aggregate the responses to get a final majority vote. Our results for triplet matching achieve **46.6%** approval, pretrained network based matching gets 32.4% approval, color matching gets 13.8% and random match achieves 7.2% approval. These results demonstrate that network-based matching outperforms the other methods for not only matching of exact products, but also similar products.

C. Visual Matching of Customer Images

Comprehensive evaluation of any machine learning model requires an unseen, yet challenging test set that can provide insight into the model's ability to generalize to new scenarios and conditions. For this work, we utilize a dataset of customer generated review images described in Section IV-A to evaluate our visual search pipeline on example scenes that are not manually curated. Scene images from this dataset may contain poor lighting, unrelated items in the background, and a different pose or angle from the product image. Here, we benchmark our matching method with the highest performing one, the triplet network from Section 3.2.2.

The triplet network struggles on all ten categories when compared to the performance on the in-house dataset from Table II. Top-1 accuracies on the customer review dataset using the triplet network range from 0.0% to 5.8% while top-5 accuracies range from 0.0% to 14.8%. This varied performance suggests that a category-based strategy is necessary for home improvement recommendations as different decor categories warrant different visual search strategies. Further study is required to find the optimal matching strategy for each category.

D. Online Performance with A/B Test

We conducted an A/B test with real traffic on a large home improvement retailer's website where we compared the experience of the scene images without tags (control) and the scene images with the tags. When hovering over a tag, an overlay window appeared with the best match product. The tag positions were determined by our object detection model inference and the best match product is the result of our visual search algorithm. From there, users could move away from the tag and overlay window, click on and/or buy the recommended product corresponding to the best match. This test was conducted over multiple weeks to smooth out periodic user behavior. It resulted in +1.5% increase in user engagement and +4.0% increase in Average Order Value

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.

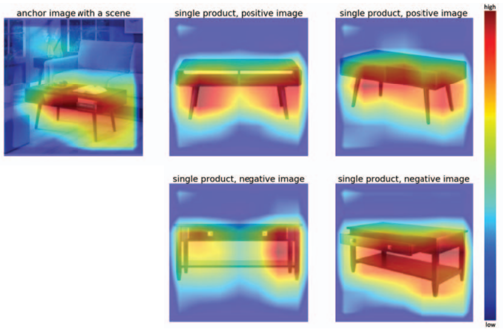
anchor image with a scene
single product, positive image
single product, positive image
single product, negative image
single product, negative image
high
low

Fig. 7. The overlaying heatmaps are generated based on the top convolutional layer's output values (feature maps). The size of the tensor is 2048*7*7, and the values shown here are averaged across the 2048 dimensions.

(AOV). The increase in AOV strongly suggests that users have purchased additional products in their transactions when tags were displayed on scene images, thus validating the relevance of our visual search results.

E. Network Visualization

We are interested in visualizing how the triplet network adapts its weights so that it leads to good results. For classification, a class activation mapping analysis is usually done to highlight the regions the network focuses on when making the classification decision. In our case, we do not have the fully connected or the softmax layers, so our focus is the output of top convolutional layer of the Triplet network. Figure 7 illustrates how the Triplet network captures the arc shaped characteristics of the input product (intensive focus on the legs of table) and its corresponded positive product images. In contrast, such characteristics are diluted in the negative examples. These will cause significant difference in the embedding layers, thus determining network output.

VI. Conclusion

In this paper, we propose to leverage room scene images with object detection and product matching to generate complementary recommendations in the home decor domain. Our triplet contrastive product matching approach outperforms strong baselines in matching exact as well as similar products across multiple datasets and hundreds of product categories. The system was deployed live on the website of a major home improvement retailer and resulted in improved revenue metrics.

References

[1] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin, "Visual search at alibaba," 2018.

[2] F. Yang, A. Kale, Y. Bubnov, L. Stein, Q. Wang, H. Kiapour, and R. Piramuthu, "Visual search at ebay," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2101–2110. [Online]. Available: https://doi.org/10.1145/3097983.3098162

[3] Z. Zuo, L. Wang, M. Momma, W. Wang, Y. Ni, J. Lin, and Y. Sun, "A flexible large-scale similar product identification system in e-commerce," 2020.

[4] D. Shankar, S. Narumanchi, H. Ananya, P. Kompalli, and K. Chaudhury, "Deep learning based large scale visual recommendation and search for e-commerce," *arXiv preprint arXiv:1703.02344*, 2017.

[5] H. Hu, Y. Wang, L. Yang, P. Komlev, L. Huang, X. Chen, J. Huang, Y. Wu, M. Merchant, and A. Sacheti, "Web-scale responsive visual search at bing," 2018.

[6] W.-C. Kang, E. Kim, J. Leskovec, C. Rosenberg, and J. McAuley, "Complete the look: Scene-based complementary product recommendation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] R. Shiau, H.-Y. Wu, E. Kim, Y. L. Du, A. Guo, Z. Zhang, E. Li, K. Gu, C. Rosenberg, and A. Zhai, "Shop the look," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, Jul 2020. [Online]. Available: http://dx.doi.org/10.1145/3394486.3403372

[8] A. Zhai, H.-Y. Wu, E. Tzeng, D. H. Park, and C. Rosenberg, "Learning a unified embedding for visual search at pinterest," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2412–2420.

[9] J. Wang, Y. song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," 2014.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[11] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015. [Online]. Available: https://doi.org/10.1145/2766959

[12] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017.

[13] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," 2018.

[14] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3330–3337.

[15] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3343–3351.

[16] J. Hui, "understanding," Mar 2018. [Online]. Available: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

[17] R. C. Gonzalez and P. Wintz, "Digital image processing(book)," *Reading, Mass., Addison-Wesley Publishing Co., Inc.(Applied Mathematics and Computation*, no. 13, p. 451, 1977.

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

[19] W.-C. Kang, E. Kim, J. Leskovec, C. Rosenberg, and J. McAuley, "Complete the look: Scene-based complementary product recommendation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 532–10 541.

[20] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[23] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on September 04,2025 at 20:16:30 UTC from IEEE Xplore. Restrictions apply.