

What Looks Good with my Sofa: Multimodal Search Engine for Interior Design

Ivona Tautkute^{1,3}, Aleksandra Możejko³, Wojciech Stokowiec^{1,3},
Tomasz Trzciniński^{2,3}, Łukasz Brocki¹ and Krzysztof Marasek¹

Abstract—In this paper, we propose a multi-modal search engine for interior design that combines visual and textual queries. The goal of our engine is to retrieve interior objects, *e.g.* furniture or wall clocks, that share visual and aesthetic similarities with the query. Our search engine allows the user to take a photo of a room and retrieve with a high recall a list of items identical or visually similar to those present in the photo. Additionally, it allows to return other items that aesthetically and stylistically fit well together. To achieve this goal, our system blends the results obtained using textual and visual modalities. Thanks to this blending strategy, we increase the average style similarity score of the retrieved items by 11%. Our work is implemented as a Web-based application and it is planned to be opened to the public.

I. INTRODUCTION

Recent advancements in the development of efficient and effective deep learning methods that rely on multi-layer neural networks have lead to impressive results obtained for many computer vision applications, such as object detection or object classification [1], [2]. Nevertheless, a set of challenges regarding image understanding is still to be solved, for instance training a model which is able not only to detect an object, *e.g.* sofa or chair, in the picture, but based on this detection suggest a table or wallpaper to match their style. This is exactly the topic of this work and the applications of such system are numerous, including but not limited to interior design augmented reality applications or e-commerce recommendation engines.

Although several methods for finding visually similar objects exist [3], [4], they rather focus on the similarities related to the appearance of the objects, not their style or context. On the other hand, recently proposed textual representation called *word2vec* [5] that is used in many text-based search engines is trained mainly using contextual information present in the training corpus. This approach allows to map words describing objects that often appear together, *e.g.* *chair* and *table*, to spaces where their representations are closer to each other than, *e.g.* *table* and *bath tub*. Therefore, one can imagine using *word2vec* representation for finding interior design items that correspond to the same style, as they would often appear together. Nevertheless, textual search often falls short when applied to interior design applications, as the variety of stylistic and aesthetic descriptions, such as *Scandinavian style* or *minimalistic design*, is only known

by a limited number of professional interior designers, and remains cryptic for target users of those applications.

In this paper, we address the above mentioned shortcomings of visual or textual search when applied to interior design by combining the best of both worlds. More precisely, we propose a multi-modal approach to interior design search, dubbed Style Search Engine, which retrieves a list of visually similar objects enhanced with textual input from the user. Fig. 1 shows a high-level overview of our proposed Style Search Engine. The first building block of our engine combines state-of-the-art object detection algorithm YOLO 9000 [6] with visual search engine based on the outputs of deep neural network. The second block allows to further specify search criteria with text and it uses this textual input for context-aware retrieval of stylistically similar objects. At final stage, our method blends the visual and textual search results using similarity score in their respective feature spaces. This leads to 11% performance improvement in terms of style similarity of the retrieved objects.

To summarize, the contributions of this work are threefold:

- Firstly, we propose a multi-modal search framework that combines object detection, visual search and textual query to return a set of results that are visually and stylistically similar.
- Secondly, we propose a new blending method for search models (image and text) that increases the quality of the results.
- Thirdly, we implement our Style Search Engine as a working Web application with the aim of opening it to the public.

The remainder of this paper is organized in the following manner. We begin with a brief overview of the related work and then describe our Style Search Engine along with their building blocks. In Sec. IV, we introduce the datasets that is then used in Sec. V for experiments and validation of our method. We present our Web-based application of Style Search Engine in Sec. VI and in Sec. VII we conclude the paper.

II. RELATED WORK

In this section, we first give an overview of the visual search methods proposed in the literature. We then discuss several approaches used in the context of textual search. Finally, we present works related to defining similarity in the context of aesthetics and style, as it directly pertains to the results obtained using our proposed method.

¹Polish-Japanese Academy of Information Technology, Warsaw, Poland

²Warsaw University of Technology, Warsaw, Poland

³Tooploox, Warsaw, Poland.

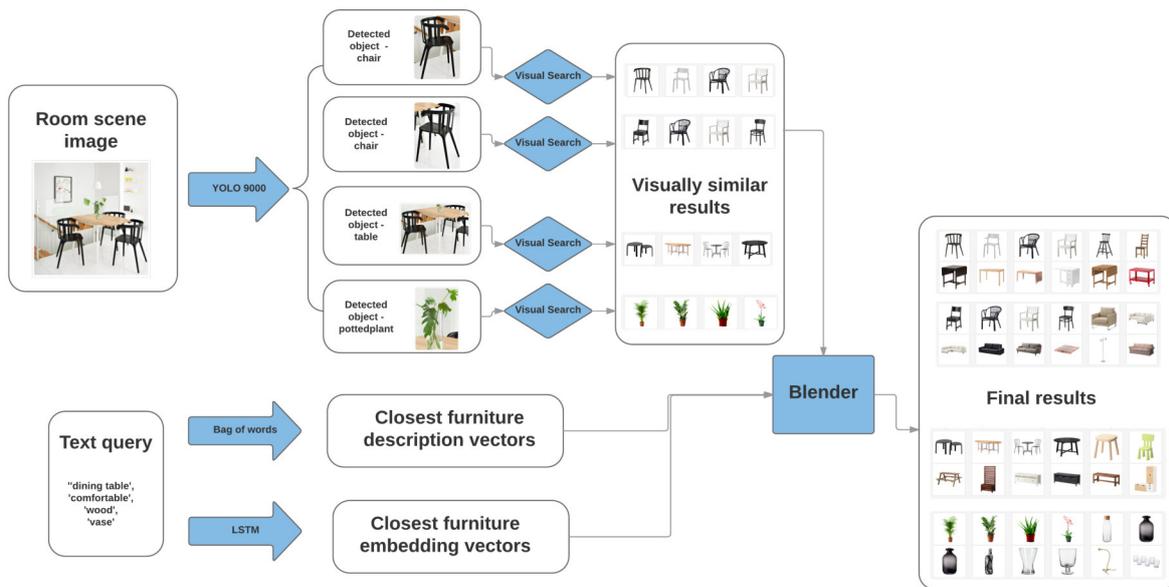


Fig. 1. High-level overview of our proposed Style Search Engine. The visual search block of our engine uses state-of-the-art object detection algorithm YOLO 9000 [6] and the outputs of deep neural network. The textual block allows to further specify search criteria with text and increases the contextual importance of the retrieved results. Finally, by blending the visual and textual search results using similarity score in their respective feature spaces, our method significantly improves the stylistic and aesthetic similarity of the retrieved items.

A. Visual search

Traditionally, image-based search methods drew their inspiration from textual retrieval systems [3]. By using k -means clustering method in the space of local feature descriptors, such as SIFT [7], they are able to mimic textual word entities with the so-called *visual words*. Once the mapping from image salient keypoints to visually representative *words* was established, typical textual retrieval methods, such as Bag-of-Words [8] could be used. Video Google [9] was one of the first visual search engines that relied on this concept. Several extensions of this concept were proposed, e.g. spatial verification [4] that checks for geometrical correctness of initial query and eliminates the results that are not geometrically plausible. Other descriptor pooling methods were also proposed, e.g. Fisher Vectors [10] or VLAD [11].

Successful applications of deep learning techniques in other computer vision applications have motivated researchers to apply those methods also to visual search. Although preliminary results did not seem promising due to lack of robustness to cropping, scaling and image clutter [12], later works proved potential of those methods in the domain of image-based retrieval. For instance, by incorporation of R-MAC technique [13] image representation based on the outputs of convolutional neural networks could be computed in a fixed layout of spatial regions. Many other deep architectures were also proposed, such as siamese networks, and proved successful when applied to content-based image retrieval [14].

Nevertheless, all of the above mentioned methods suffer from an important drawback, namely they do not take into

account the contextual and stylistic similarity of the retrieved objects, which yields their application to the problem of interior design items retrieval infeasible.

B. Textual Search

First methods proposed to address textual information retrieval have been based on token counts, e.g. *Bag-of-Words* [8] or *TF-IDF* [15]. Despite being conceptually simple and adequate to small-scale search problems, the scalability of those methods is very limited. This is due to the fact that the representation size grows with the indexed *corpus* size and, in turn, causes problems with less frequent tokens. Additionally, when using such representations long sequences (documents) tend to have similar token distributions which results in lower discriminative power of the representation and lower retrieval precision. One way to avoid those problems is to apply a SVD decomposition of the token co-occurrence matrix and, hence, reduce the dimensionality of a representation vector [16], [17]. This, however, does not address another problem commonly occurring in token-based representations, namely the fact that they are insensitive to any sequence (token) permutation. Moreover, it is not straightforward to obtain a good representation of single tokens using above mentioned methods.

To handle those shortcomings, a new type of representation called *word2vec* has been proposed by Mikolov *et. al* [5]. The proposed instances of *word2vec*, namely continuous Bag of Words (CBOW) and Skip-Grams, allow the token representation to be learned based on its local context. To grasp also the global context of the token, later extension of *word2vec* called GLoVe [18] has been introduced. GLoVe

takes advantage of information both from local context and the global co-occurrence matrix, therefore providing a powerful and discriminative representation of textual data.

C. Stylistic Similarity

Comparing the style similarity of two objects or scenes is one of the challenges that has to be answered when training a machine learning model for interior design retrieval application. This problem is far from being solved mainly due to the lack of a clear metric defining how to measure style similarity. Various approaches have been proposed for defining style similarity metric. Some of them focus on evaluating similarity between shapes based on their structures [19], [20] and measuring the differences between scales and orientations of bounding boxes. Other approach propose structure-transcending style similarity measure that accounts for element similarity, element saliency and prevalence [21]. In this work, we follow [22], and define style as *a distinctive manner which permits the grouping of works into related categories*. Nevertheless, instead of using hand-crafted features and predefined styles, we take data-driven probabilistic approach to determine stylistic similarity measure that we define in Sec. V-B.

III. STYLE SEARCH ENGINE

In this section, we present the pipeline of our multi-modal Style Search Engine. As an input, it takes two types of query information: an image of an interior, *e.g.* a picture of a dining room, and a textual query used to specify search criteria, *e.g.* *cozy and fluffy*. Then, an object detection algorithm is run on the uploaded picture to detect objects of classes of interest such as chairs, tables or sofas. Once the objects are detected, their regions of interest are extracted as picture patches and submitted to visual search method. Simultaneously, the engine retrieves the results for a textual query. With all visual and textual matches retrieved, our *blending algorithm* ranks them depending on the similarity in the respective features spaces and serves the resulting list of stylistically and aesthetically similar objects. Fig. 1 shows a high-level overview of our Style Search Engine. Below, we describe each part of the engine in more details.

A. Visual search

Instead of using an entire image of the interior as a query, our search engine applies an object detection algorithm as a pre-processing step of. This way, not only can we retrieve the results with higher precision, as we search only within a limited space of same-class pictures, but we do not need to know the object category beforehand. This is in contrast to other visual search engines proposed in the literature [14], [23], where the object category is known at test time or inferred from textual tags provided by human labeling.

As our object detection method, we use the state-of-the-art detection model YOLO 9000 [6]. It is based on DarkNet-19 model [24], [6] with 19 convolutional layers and 5 max-pooling levels. YOLO 9000 is able to detect multiple furniture classes along with their bounding boxes.

The bounding boxes are then used to generate Regions of Interest (ROIs) in the pictures and visual search is performed on the extracted ROIs.

In a set of initial experiments, we optimized the parameters of YOLO 9000 detection algorithm, mainly focusing on the detection confidence threshold. We set this threshold to 0.1, although in case of overlapping bounding boxes returned by the model, we take the one with the highest confidence score.

Once the ROIs are extracted, we compute their representation using the outputs of pre-trained deep neural networks. More precisely, we use the outputs of fully connected layers of neural networks pre-trained on ImageNet dataset [2]. We then normalize the extracted vectors of outputs, so that their L_2 norm is equal to 1 and search for similar images within the dataset using this representation. To determine the neural network architecture providing the best performance, we conducted several experiments described in details in Sec. V-A.

B. Text query search

To extend the functionality of our Style Search Engine, we implement a text query search that allows to further specify the search criteria. This part of our engine is particularly useful when trying to search for interior items that represent abstract concepts, such as *minimalism* or *Scandinavian style*.

In order to perform such a search, we need to find the mapping from textual information to vector representation of the interior item. The resulting representation should live in a multi-dimensional space, where stylistically similar objects reside close to each other. We formulate this problem in the following manner. Let us first define $\mathbf{f} \in \mathbb{R}^n$ to be a vector representation of an item stored in the database and $(t_1, t_2, \dots, t_i) = \mathbf{t} \in \mathfrak{T}$ be a variable length sequence that represents a textual query. We are interested in finding a mapping $m : \mathfrak{T} \rightarrow \mathbb{R}^n$ from the space of queries to the vector space of interior items, such that $dist(m(\mathbf{t}), \mathbf{f})$ is small, when \mathbf{f} are relevant to the query \mathbf{t} . Having found such a mapping, we can perform search by returning k -nearest neighbors of transformed query in interior item space using cosine similarity as a distance measure.

To obtain the above defined space embedding, we use a state-of-the-art Continuous Bag-of-Words (CBOW) model that belongs to word2vec model family [5]. We use the descriptions of various household parts, such as living rooms or kitchens, to infer the contextual information about interior items. Such descriptions are available as part of the IKEA dataset which we describe in details in Sec. IV. It is worth noticing that our embedding is trained without relying on any linguistic knowledge since the only information that the model sees during training is whether given objects appeared in the same room.

In order to optimize hyper-parameters of CBOW for furniture embedding, we run a set of initial experiments on the validation dataset and use cluster analysis of the embedding results. We select the parameters that minimize intra-cluster distances at the same maximizing inter-cluster distance. Fig. 2 shows the obtained feature embeddings using

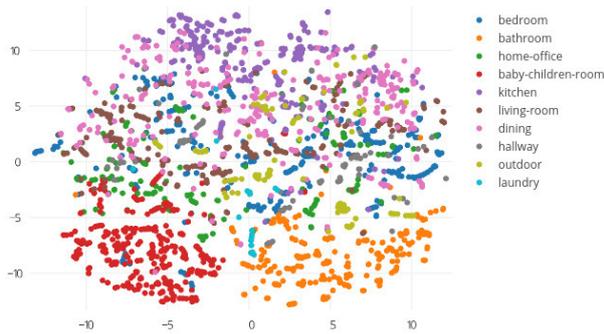


Fig. 2. t-SNE visualization of interior items' embedding. Distinctive classes of objects, *e.g.* those that appear in bathroom or baby room, are clustered around the same region of the space.

t-SNE dimensionality reduction algorithm [25]. One can see that some classes of objects, *e.g.* those that appear in bathroom or baby room, are clustered around the same region of the space.

After obtaining the furniture embedding, we need a model to find an appropriate mapping $m : \mathcal{T} \rightarrow \mathbb{R}^n$ from query space to the space of furniture embeddings.

To this end, we train a Long Short-Term Memory (LSTM) deep neural network architecture that has been successfully applied in several other natural language processing applications such as language modeling [26], machine translation [27] or on-line content popularity prediction [28].

We formulate the question of finding $m : \mathcal{T} \rightarrow \mathbb{R}^n$ as a regression problem. To be more explicit, let $\mathbf{t} = (t_1, t_2, \dots, t_i) \in \mathcal{T}$ be a furniture description from IKEA Dataset and $\mathbf{f} \in \mathbb{R}^n$ denote its furniture embedding. We train our model to minimize the MSE between the predicted item embedding based on its description $\hat{\mathbf{f}} = LSTM(\mathbf{t})$ and the ground-truth furniture embedding \mathbf{f} .

Due to the fact, that vocabulary of IKEA Dataset products description is rather limited and may possibly not contain words from user-generated queries, we initialized the LSTM's query embedding layer with word embeddings trained on dump of English Wikipedia with CBOW model. Additionally, to avoid overfitting, we froze the query embedding layer during training.

IV. DATASET

In order to evaluate our proposed Style Search Engine, we collected a dataset of interior items along with their textual description and the context in which they appear. Although several datasets for standard visual search methods exist, *e.g.* Oxford 5K [4] or Paris 6K [29], we could not use them in our work, as our multi-modal approach requires additional type of information to be evaluated. More precisely, our dataset that can be used in the context of multi-modal interior design search engine should fulfill the following conditions:

- It should contain both images of individual objects as well as room scene images with those objects present.

- It should have a ground truth defining which objects are present in a given room scene photo.
- It should also have a textual description for each room scene image.

To our knowledge, no such dataset is publicly available. Hence, we collected our own dataset by recursively scrapping the website of one of the most popular interior design distributor - IKEA¹. We were able to download 298 room photos with their description and 2193 individual product photos with their textual descriptions. A sample image of the room scene and interior item along with their description can be seen in Fig. 3. We have also grouped together some of the most frequent object classes (*e.g.* chair, table, sofa) for more detailed analysis. In addition, we also divided room scene photos into 10 categories based on the room class (kitchen, living room, bedroom, children room, office). This kind of classification can be useful, *e.g.* for qualitative analysis of embedding results, as shown in Fig. 2. We plan to release our IKEA dataset to the public.

V. EXPERIMENTS

In this section, we present the results of the experiments conducted using our Style Search Engine to evaluate its performance with respect to the baseline methods. We first show how incorporating object detection algorithm and deep neural network architectures within our visual search engine improves the search accuracy. We then present our method for blending the results of multi-modal search and prove that using this approach we can increase the system performance by 11%.

A. Visual Search with Object Detection and Neural Networks

In this experiment, we analyze the results of our visual search when using various neural network architectures combined with YOLO 9000 object detection algorithm. The goal of this experiment is to select the right configuration of deep neural network used as the descriptor extractor for our interior design images, as well as to quantify the improvement obtained when adding a pre-processing step of object detection. To that end, we evaluate two neural network architectures that were successfully applied to object recognition task on ImageNet dataset: ResNet [30] and VGG[31]. We use VGG network with 3×3 convolutional filters in two configurations, with 16 and 19 weight layers. We analyze the outputs of the first (*fc6*) and the second fully connected layer (*fc7*) of the VGG network. For ResNet, we take the average pooling layer. In all experiments, we use normalized outputs of the networks pre-trained on ImageNet dataset and we compute the similarity measure with Euclidean distance. The networks were implemented using Keras [32] with Theano backend for deep feature extraction.

Baseline: As our baseline, we take the conventional Bag-of-Visual-Words search engine [9]. It is based on the SIFT feature extraction algorithm [7]. We extract the descriptors and cluster them using *k*-means clustering [3] into $k = 1000$

¹<https://ikea.com/>

Room images:	Object images:	Description:
		You sit comfortably thanks to the armrests.
		There's a natural and living feeling of wood, as knots and other marks remain on the surface.
		This lamp gives a pleasant light for dining and spreads a good directed light across your dining or bar table.
		Extendable, so it can be pulled out as your child grows.
		You can adjust the height of the clothes rail and shelves as your child grows.
		Made of plastic which makes it easy to carry and move for children.

Fig. 3. Example entries from IKEA dataset contain room images, object images and their respective text descriptions.

visual words. We use SIFT implementation available in OpenCV for Python [33] with contrast threshold set to 0.05, edge threshold to 11 and L_2 norm.

Evaluation metric: To measure the performance of our system, we use Hit@ k metric [34]. We define it in the following manner. Let \mathcal{F} denote a set of all possible interior items available in the dataset. We define a room $\mathcal{R} \in \mathfrak{R}$ as a set that contains elements $f \in \mathcal{F}$. Hit@ k is therefore defined as the fraction of retrieved items that contain at least one of the ground truth objects in the top k predictions. More formally, if $rank_{f,\mathcal{R}}$ is the rank of furniture f in the room \mathcal{R} (the highest scoring furniture having rank 1) and $G_{\mathcal{R}}$ is the set of ground-truth objects for \mathcal{R} , then Hit@ k is defined as:

$$\frac{1}{|\mathfrak{R}|} \sum_{\mathcal{R} \in \mathfrak{R}} \vee_{f \in G_{\mathcal{R}}} \mathbb{I}(rank_{f,\mathcal{R}} \leq k), \quad (1)$$

where \vee is logical OR operator.

Results: Tab. I displays the results obtained for this experiment. Adding object detection algorithm as a pre-processing step significantly increases the number of correctly retrieved results across all evaluated configurations. We have illustrated the results for Hit@6 as we retrieved visually similar objects for six distinct object classes - chair, table, sofa, bed, wall clock and pottedplant. For Hit@6 the performance gains reach up to 175% (in the case of ResNet) and 238% (for VGG-19 with fc7). Feature extraction with ResNet and object detection pre-processing yields the highest

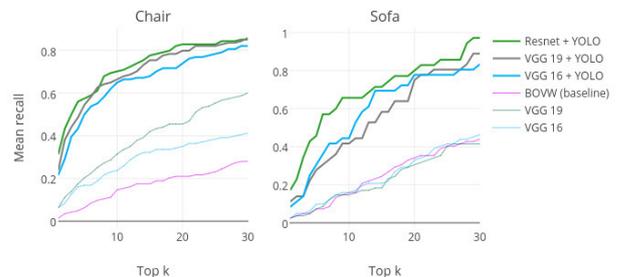


Fig. 4. Quantitative evaluation of various feature extraction methods combined with object detection algorithm YOLO 9000. We use recall as an evaluation metric that shows whether or not a single item present in the room picture was returned by the search engine. The recall is plotted as a function of the number of returned items k .

Hit@ k score, retrieving correct results for almost half of all queries. To further analyze the performances of the proposed methods, in Fig. 4 we also plot recall curves for two sample object classes. Again, ResNet combined with object detection step remains the best performing configuration. One can also notice that all methods based on deep network architectures significantly outperform baseline BoVW method.

B. Results blending

In order to use the full potential of our multi-modal interior design search engine, we introduce a blending method to

TABLE I

RESULTS FOR CONTENT BASED IMAGE RETRIEVAL EXPERIMENT FOR DIFFERENT MODELS AND ALL OBJECT CLASSES. CONFIGURATION OF RESNET NEURAL NETWORK WITH YOLO 9000 OBJECT DETECTION AS A PRE-PROCESSING STEP SIGNIFICANTLY OUTPERFORMS BOTH THE BASELINE BOVW MODEL AND OTHER DEEP NEURAL NETWORK ARCHITECTURES.

Model	Layer	Hit@6	
		whole image	with object detection
BoVW	N/A	0.066	0.26
VGG-16	fc6	0.126	0.392
	fc7	0.153	0.314
VGG-19	fc6	0.141	0.43
	fc7	0.136	0.445
ResNet	avg pool	0.167	0.458

combine the retrieval results of visual and textual search engines and present them to the user. To that end, we use *feature similarity blending* approach. More precisely, the search engine returns an initial set of results for each modality, extracts visual features (normalized outputs of pre-trained deep neural network) and then re-ranks them using the distance from the query to the item in visual features' space for each modality independently (visual search results do not need to be re-ranked). A set of closest items is returned as a final result.

Simple blending: As an alternative method for blending the results, we blend k best results from each modality and return them as a final result.

Evaluation metric: As mentioned in Sec. II-C, defining a similarity metric that allows to quantify the stylistic similarity between interior design objects is a challenging task and an active area of research. In this work, we propose the following similarity measure that is inspired by [22] and based on a probabilistic data-driven approach. Similarly to Hit@ k metric, let us first define \mathcal{F} as a set of all possible interior items available in our dataset and a room $\mathcal{R} \in \mathcal{R}$ as a set containing elements $f \in \mathcal{F}$. Our proposed similarity metric between two items $f_1, f_2 \in \mathcal{F}$ that determines if they fit well together can be computed as:

$$C(f_1, f_2) = |\{\mathcal{R} : f_1 \in \mathcal{R} \wedge f_2 \in \mathcal{R}\}|. \quad (2)$$

We defined the style similarity as:

$$s(f_1, f_2) = \frac{C(f_1, f_2)}{\max_{f_i, f_j \in \mathcal{F}} C(f_i, f_j)}. \quad (3)$$

In fact, it is the fraction of the number of rooms, in which both f_1 and f_2 appear and total number of rooms in which any of those items co-occur. This metric can be interpreted as empirical probability for two objects f_1 and f_2 to appear in the same room.

Results: Tab. II shows the results of the blending methods in terms of mean value of our similarity metric. *Text query*

= *object class name* means that detected object class, *i.e.* the one with the highest detection confidence, was used as a text query.

Vanilla visual search without text query achieves an average value of 0.2295 where similarity is calculated over visually similar results to the query object, all belonging to the same object class. For text search average similarity was slightly lower - 0.2243.

When analyzing the results of the evaluated blending approaches, both of them have a score that is higher than the ones obtained for vanilla visual and text search. Our proposed blending method outperforms both the visual search and simple blending, yielding an improvement of 11% and 4% respectively. It is worth noticing that simply adding a name of detected object class as a text query improves the search results already. Providing additional information such as color or style (*e.g. white* or *decorative*) yields further performance improvement.

VI. WEB APPLICATION

To enable dissemination of our work, we implemented a Web-based application of our Style Search Engine. The application allows the user either to choose the query image from a pre-defined set of room images or to upload his/her own image. The application was implemented using Python Flask² - a lightweight server library. It is currently available for restricted use only³ and we plan to open it to the public, once it passes the initial tests with trial users. Fig. 5 shows a set of screenshots from the working Web application with Style Search Engine.

VII. CONCLUSIONS

In this paper, we proposed a multi-modal search engine for interior design applications dubbed Style Search Engine. By combining textual and visual information, it can successfully and with high recall retrieve stylistically similar images from a dataset of interior items. Thanks to the object detection pre-processing step, the results of our visual search component improved by over 200%. Using feature similarity blending approach to combine the results of visual and textual search engines, we increased the overall similarity score of the retrieved results by 11%. We also implemented working prototype of a Web application that uses our Style Search Engine.

In our future research, we plan to explore various approaches towards common latent space mapping that could allow to map both textual and visual queries to a common space and perform similarity search there.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," pp. 1097–1105, 2012.
- [2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, p. 211252, Nov 2015.

²<http://flask.pocoo.org/>

³<http://style-search.us-west-2.elasticbeanstalk.com>

TABLE II
MEAN SIMILARITY RESULTS AVERAGED FOR ALL ROOM PICTURES IN IKEA DATASET AND SAMPLE TEXT QUERIES.

Text query	Visual search	Text search	Simple blending	Feature similarity blending
-	0.2295	-	-	-
object class name	-	-	0.2486	0.2374
<i>decorative</i>	-	0.1358	0.2316	0.2517
<i>black</i>	-	0.1538	0.2493	0.2244
<i>white</i>	-	0.2036	0.2958	0.2793
<i>smooth</i>	-	0.3520	0.2415	0.3052
<i>cosy</i>	-	0.2419	0.2126	0.2334
<i>fabric</i>	-	0.0371	0.1269	0.1344
<i>colourful</i>	-	0.4461	0.3032	0.3215
Average	0.2295	0.2243	0.2387	0.2484

- [3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*.
- [4] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR abs/1301.3781*, Sep 2013.
- [6] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, p. 91110, 2004.
- [8] Z. S. Harris, "Distributional structure," *Papers on Syntax*, p. 322, 1981.
- [9] J. Sivic and A. Zisserman, "Video google: Efficient visual search of videos," *Toward Category-Level Object Recognition Lecture Notes in Computer Science*, p. 127144, 2006.
- [10] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [11] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.
- [12] A. Gordo, J. Almazn, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," *Computer Vision ECCV 2016*, p. 241257, 2016.
- [13] G. Toliás, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *CoRR*, vol. abs/1511.05879, 2015.
- [14] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics*, vol. 34, no. 4, 2015.
- [15] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, p. 613620, Jan 1975.
- [16] C. H. Q. Ding, "A similarity-based probability model for latent semantic indexing," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, 1999.
- [17] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '88*, 1988.
- [18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [19] M. E. Yumer and L. B. Kara, "Co-constrained handles for deformation in shape collections," *ACM Transactions on Graphics*, vol. 33, no. 6, p. 111, 2014.
- [20] O. V. Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or, "Co-hierarchical analysis of shape structures," *ACM Transactions on Graphics*, vol. 32, p. 1, Jan 2013.
- [21] Z. Lun, E. Kalogerakis, and A. Sheffer, "Elements of style," *ACM Transactions on Graphics*, vol. 34, no. 4, 2015.
- [22] "Art history and its methods: a critical anthology," *Choice Reviews Online*, vol. 33, Jan 1996.
- [23] Y. Jing, D. Liu, D. Kisyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel, "Visual search at pinterest," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 2015.
- [24] J. Redmon, "Darknet: Open source neural networks in c.," 2016.
- [25] G. Hinton and L. Van der Maaten, "Visualizing data using t-sne," *Journal of Machine Learning Research*.
- [26] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *CoRR*, vol. abs/1602.02410, 2016.
- [27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014.
- [28] W. Stokowiec, T. Trzcinski, K. Wolk, K. Marasek, and P. Rokita, "Shallow reading with deep learning: Predicting popularity of online content using only its title," *International Symposium on Methodologies for Intelligent Systems, (ISMIS)*, 2017.
- [29] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [32] F. Chollet, "Keras," 2015.
- [33] G. Bratski, "Opencv," *Dr. Dobb's Journal of Software Tools*, 2000.
- [34] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *CoRR*, vol. abs/1609.08675, 2016.

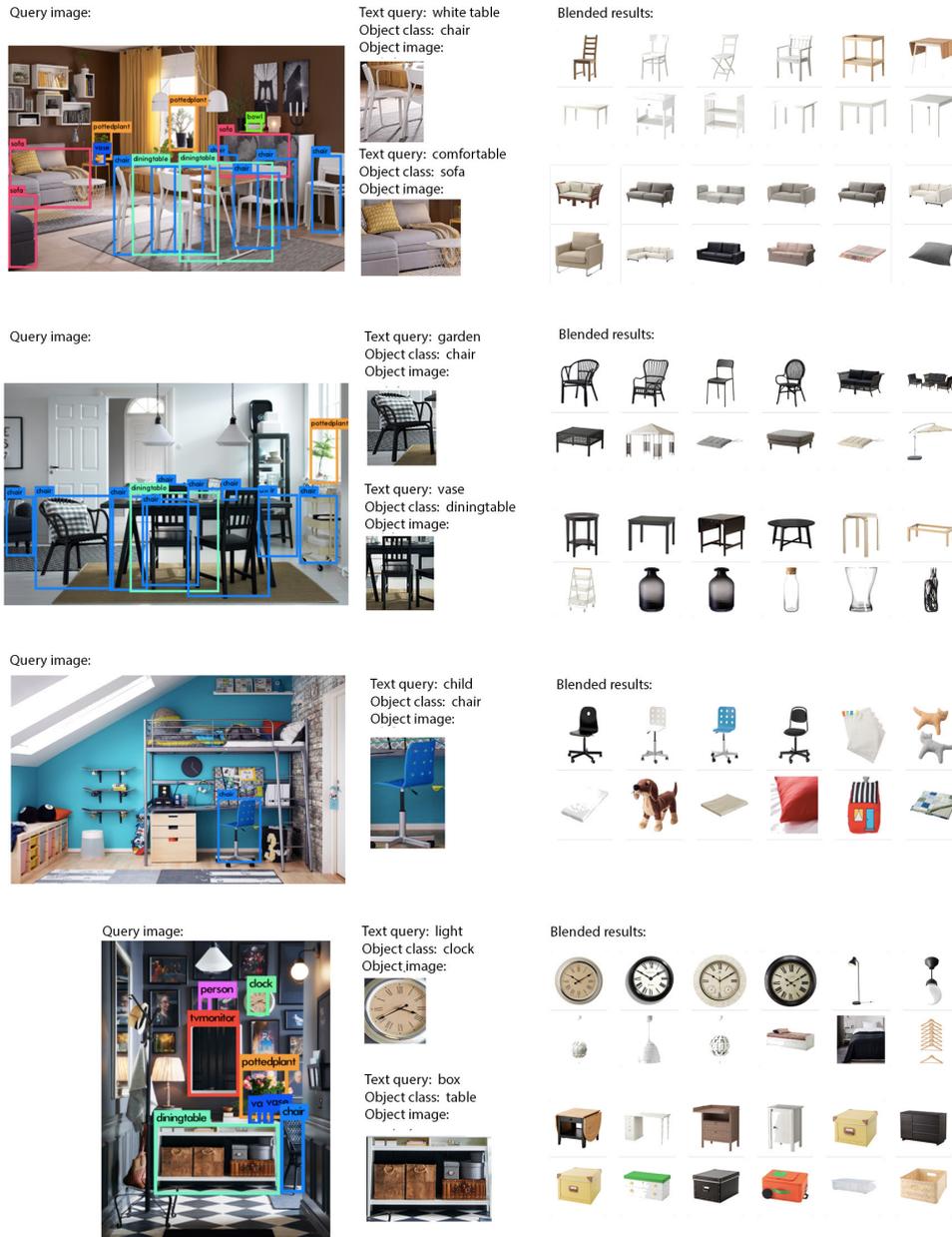


Fig. 5. Screenshots of the Web application of our Style Search Engine. Sample results retrieved for room images from the IKEA dataset and combined with custom text queries.