IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 1, 2019, accepted June 6, 2019, date of publication June 17, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923552

DeepStyle: Multimodal Search Engine for Fashion and Interior Design

IVONA TAUTKUTE[1,2], TOMASZ TRZCI SKI[2,3], (Member, IEEE), ALEKSANDER P. SKORUPA[2], ŁUKASZ BROCKI[1], AND KRZYSZTOF MARASEK[1]

[1]Institute of Multimedia, Polish-Japanese Academy of Information Technology, 02-008 Warsaw, Poland
[2]Tooploox Sp. z o.o., 00-001 Warsaw, Poland
[3]Institute of Computer Science, Warsaw University of Technology, 00-661 Warsaw, Poland

Corresponding author: Ivona Tautkute (s16352 at pjwstk.edu.pl)

ABSTRACT In this paper, we propose a multimodal search engine that combines visual and textual cues to retrieve items from a multimedia database aesthetically similar to the query. The goal of our engine is to enable intuitive retrieval of fashion merchandise such as clothes or furniture. Existing search engines treat textual input only as an additional source of information about the query image and do not correspond to the real-life scenario, where the user looks for "the same shirt but of denim". Our novel method, dubbed DeepStyle, mitigates those shortcomings by using a joint neural network architecture to model contextual dependencies between features of different modalities. We prove the robustness of this approach on two different challenging datasets of fashion items and furniture where our DeepStyle engine outperforms baseline methods by more than 20% on tested datasets. Our search engine is commercially deployed and available through a Web-based application.

INDEX TERMS Multimedia computing, multi-layer neural network, multimodal search, machine learning.

I. INTRODUCTION

Multimodal search engine allows to retrieve a set of items from a multimedia database according to their similarity to the query in more than one feature spaces, e.g. textual and visual or audiovisual (see Fig. 1). This problem can be divided into smaller subproblems by using separate solutions for each modality. The advantage of this approach is that both textual and visual search engines have been developed for several decades now and have reached a certain level of maturity. Traditional approaches such as Video Google [2] have been improved, adapted and deployed in industry, especially in the ever-growing domain of e-commerce. Major online retailers such as Zalando, Alibaba and ASOS already offer visual search engine functionalities to help users find products that they want to buy [3]. Furthermore, interactive multimedia search engines are omnipresent in mobile devices and allow for speech, text or visual queries [4]–[6].

Nevertheless, using separate search engines per each modality suffers from one significant shortcoming: it prevents the users from specifying a very natural query such as 'I want this type of dress but made of silk'. This is mainly due to the fact that the notion of similarity in separate spaces of different modalities is different than in one multimodal space. Furthermore, modeling this highly dimensional multimodal space requires more complex training strategies and thoroughly annotated datasets. Finally, defining the right balance between the importance of various modalities in the context of a user query is not obvious and hard to estimate a priori. Although several multimodal representations have been proposed in the context of a search for fashion items, they typically focus on using other modalities as an additional source of information, e.g. to increase classification accuracy of compatible and non-compatible outfits [7].

To address the above-mentioned shortcomings of the currently available search engines, we propose a novel end-to-end method that uses neural network architecture to model the joint multimodal space of database objects. This method is an extension of our previous work [9] that blended multimodal results. Although in this paper we focus mostly on the fashion items (clothes, accessories) and furniture, our search engine is in principle agnostic to object types and we see no limitations from applying it to other domains. We call our method DeepStyle and show that thanks to its ability to jointly model both visual and textual modalities, it allows for a more intuitive search queries, while providing higher accuracy than the competing approaches. We prove the superiority of our method over single-modality approaches and state-of-the-art

The associate editor coordinating the review of this manuscript and approving it for publication was Dezhong Peng.

FIGURE 1. Example of a typical multimodal query sent to a search engine for fashion items. By modeling common multimodal space with a deep neural network, we can provide a more flexible and natural user interface while retrieving results that are semantically correct, as opposed to the results of the search based on the state-of-the-art visual search embedding model [8].

multimodal representation using two large-scale datasets of fashion and furniture items. Finally, we deploy our DeepStyle search engine as a web-based application.

To summarize, the contributions of our paper are threefold:

- We introduce a novel DeepStyle-Siamese method for retrieval of stylistically similar product items that could be applied to a broad range of domains. To the best of our knowledge, this is the first system for joint learning of stylistic context as well as semantic regularities of both image and text. The proposed method outperforms the baselines on diversified datasets from fashion and interior design domains by 18 and 21%, respectively.
- Our system is deployed in production and available through a Web-based application.
- Last but not least, we introduce a new interior design dataset of furniture items offered by IKEA, an international furniture manufacturer, which contains both visual and textual meta-data of over 2 000 objects from almost 300 rooms. We plan to release the dataset to the public.

The remainder of this work is organized in the following manner. In Sec. II we discuss related work. In Sec. III we present a set of methods based on blending single-modality search results that serve as our baseline. Finally, in Sec. IV, we introduce our DeepStyle multimodal approach as well as its extension. In Sec. V we present the datasets used for evaluation and in Sec. VI we evaluate our method and compare its results against the baseline. Sec. VIII concludes the paper.

II. RELATED WORK

In this section, we first give an overview of the current visual search solutions proposed in the literature. Secondly, we discuss several approaches used in the context of a textual search. We then 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. Finally, we present an overview of existing search methods in fashion domain as this topic is gaining popularity.

A. VISUAL SEARCH

Traditionally, image-based search methods drew their inspiration from textual retrieval systems [10]. By using k-means clustering method in the space of local feature descriptors such as SIFT [11], 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 [12] could be used. Video Google [2] was one of the first visual search engines that relied on this concept. Several extensions of this concept were proposed, *e.g.* spatial verification [13] that checks for geometrical correctness of initial query or fine-grained image search [14] that accounts for semantic attributes of visual words.

Successful applications of deep learning techniques in other computer vision applications have motivated researchers to apply those methods also to visual search. Preliminary results proved that applications of convolutional neural networks [15] (image-based retrieval), as well as other deep architectures such as Siamese networks [16] (content-based image retrieval) may be successful. New methods have been proposed to bridge the gap between real-shot images from users, that often contain a lot of clutter, and online shop images [17]. New ranking and indexing methods have been proposed to deal with large scale data, often containing billions of images [17], [18].

Nevertheless, all of the above-mentioned methods suffer from one important drawback, namely they do not take into account the stylistic similarity of the retrieved objects, which is often a different problem from visual similarity. Items that are similar in style do not necessarily have to be close in visual features space.

B. TEXTUAL SEARCH

First methods that proposed to address textual information retrieval were based on token counts, *e.g. Bag-of-Words* [12] or *TF-IDF* [19].

Later, a new type of representation called *word2vec* was proposed by Mikolov *et al.* [20]. The proposed models in *word2vec* family, 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, GloVe [21] has been introduced. GloVe takes advantage of information both from the local context and the global co-occurrence matrix, thus providing a powerful and discriminative representation of textual data. Similarly, not all queries can be represented with a text only. There might be a clear textual definition missing for style similarities that are apparent in visual examples. Also, the same concepts might be expressed in synonymical ways.

C. STYLISTIC SIMILARITY

Comparing the style similarity of two objects or scenes is one of the challenges that have to be addressed when training a machine learning model for interior design or fashion 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 [22], [23] and measuring the differences between scales and orientations of bounding boxes. Other approaches propose the structure-transcending style similarity that accounts for element similarity [24]. In this work, we follow [25], and define style as *a distinctive manner which permits the grouping of works into related categories.* We enforce this definition by including context information that groups different objects together (in terms of clothing items in an outfit or furniture in a room picture in interior design catalog). This allows us to a take data-driven approach that measures style similarity without using hand-crafted features and predefined styles.

D. DEEP LEARNING IN FASHION

There has been a significant number of works published in the domain of fashion item retrieval or recommendation due to the potential of their application in highly profitable e-commerce business. Some of them focused on the notion of fashionability, e.g., [26] rated a user's photo in terms of how fashionable it is and provided fashion recommendations that would increase overall outfit score. Others focused on fashion items retrieval from online database when presented with user photos taken 'in the wild' usually with phone cameras [27]. Finally, there is ongoing research in terms of clothing cosegmentation [28], [29] that is an important preprocessing step for better item retrieval results.

Kiros *et al.* [8] present an encoder-decoder pipeline that learns a joint Visual-Semantic Embedding (VSE) from images and a text, which is later used to generate text captions for custom images. Their approach is inspired by successes in Neural Machine Translation (NMT) and perceives visual and textual modalities as the same concept described in different languages. The proposed architecture consists of LSTM, which is a type of recurrent neural network, for encoding sentences, convolutional neural network (CNN) for encoding images and structure-content neural language model (SC-NLM) for decoding. The authors show that their learned multimodal embedding space preserves semantic regularities in terms of vector space arithmetic e.g. image of a black car - ''black'' + ''red'' is near images of red cars. However, results of this task are only available in some example images. We would like to leverage their work and numerically evaluate multimodal query retrieval, specifically in the domain of fashion and interior design.

Ben-Younes *et al.* [30] introduced MUTAN, a method for multimodal fusion between visual and textual information using a bilinear framework. It uses a multimodal tensor-based Tucker decomposition in order to efficiently parametrize bilinear interactions between the two representations. Additional low-rank matrix constraint is designed to allow for controlling the full bilinear interaction complexity. While in the original paper, authors evaluate architecture primarily on the Visual Question Answering task, we would like to utilize it when learning a joint multimodal representation. In the similar manner, as with the previously mentioned VSE, we evaluate it on multimodal query retrieval in the domain of fashion and interior design.

Xintong Han *et al.* [31] train bi-LSTM model to predict next item in the outfit generation. Moreover, they learn a joint image-text embedding by regressing image features to their semantic representations aiming to inject attribute and category information as a regularization for training the LSTM. It should be noted, however, that their approach to stylistic compatibility is different from ours in a way that they optimize for generation of a complete outfit (e.g. it should not contain two pairs of shoes) whereas we would like to retrieve items of similar style regardless of the category they belong to. Also, they evaluate compatibility with ''fill-in-the-blanks'' test that does not incorporate retrieval from the full dataset of items. Only several example results are illustrated and no quantitative evaluation is presented.

Numerous works focus on the task of generating a compatible outfit from available clothing products [7], [31]. However, very few of the related works focus on the notion of product retrieval with multimodal query. Some attempts have been made to improve visual search with text information generated by running classification algorithm on the image [32]. Such methods however do not allow for explicit text input that is independent or different from visual information. Text information is only used as an alternative query and not as a complementary information to extend the information about the searched object. A similar line of research that works with multimodal representations for retrieval is dialog-based image retrieval [33]–[35]. However those methods focus primarily on conversational agents and sequential improvement of the results instead of one-shot search. Finally, research community has not yet paid much attention to define or evaluate style similarity.

III. FROM SINGLE TO MULTIMODAL SEARCH

In this section, we present a baseline style search engine model introduced in [9], which is the basis for our current research. It is built on top of two single-modal modules. More precisely, two searches are run independently for both image and text queries resulting in two initial sets of results. Then, the best matches are selected from initial pool of results according to blending methods - re-ranking based on visual features similarity to the query image as well as on contextual similarity (items that appear more often together in the same context).

For input, baseline style search engine takes two types of query information: an image containing object(-s), *e.g.* a picture of a dining room, and a textual query used to specify

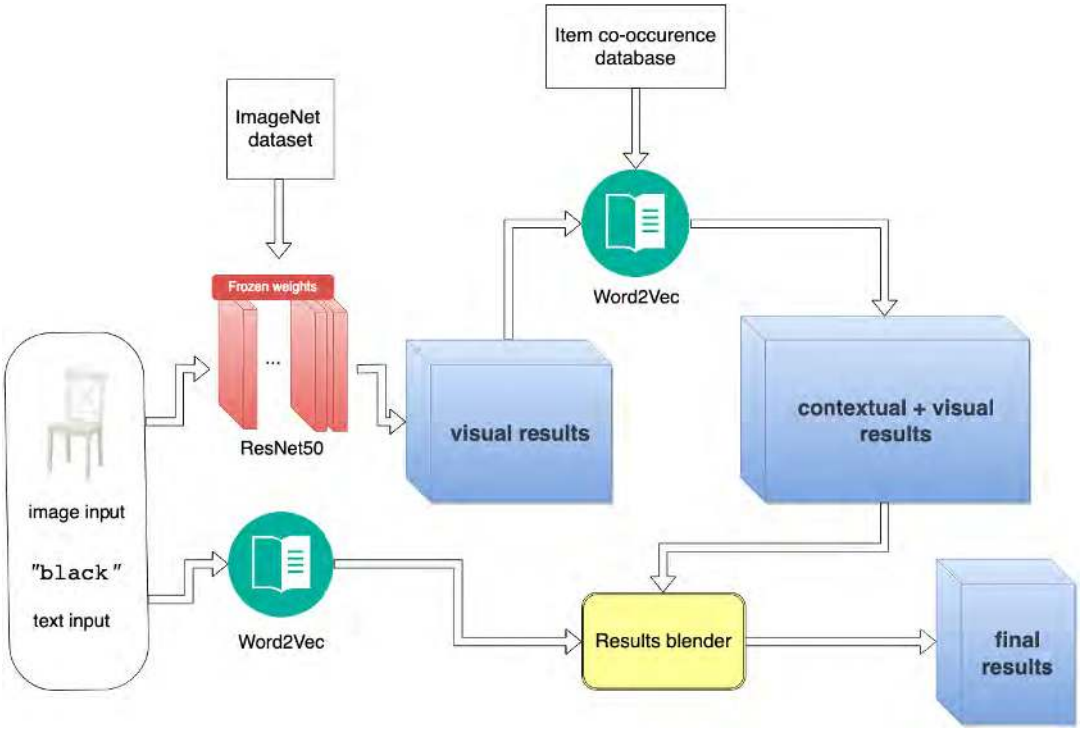


FIGURE 2. A high-level overview of our Early-fusion Blending architecture. Visual search finds closest neighbors of the image query in the space of extracted visual features that are the outputs of a pre-trained deep neural network. For each of the retrieved visually similar items, we search for the contextually similar, i.e. items that appeared together in the compatible sets from database, and extend the set of results with those items. The textual block allows to further specify search criteria with text in order to narrow down the set of stylistically and aesthetically similar items, to those that are relevant to the query.

search criteria, *e.g. cozy and fluffy*. If needed, 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 run through visual search method. For queries that already represent a single object, no object detection is required. 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 feature spaces and returns the resulting list of stylistically and aesthetically similar objects. 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. 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 [16], [36], where the object category is known at test time or inferred from textual tags provided by human labeling.

For object detection, we used YOLO 9000 [37], which is based on the DarkNet-19 model [37], [38] and is a variety of a neural network. The bounding boxes are then used to generate regions of interest in the pictures and search is performed on the extracted parts of the image.

Once the regions of interest are extracted, we feed them to a pretrained deep neural network to get a vector representation. More precisely, we use the outputs of fully connected layers of neural networks pretrained on ImageNet dataset [39]. We then normalize the extracted output vectors, so that their L_2 norm is equal to 1. We search for similar images within the dataset using this representation to retrieve a number of closest vectors (in terms of Euclidean distance).

To illustrate how the space of extracted visual features preserves the visual similarity of product items, we have visualized the visual features embedding (fig. 10) with common dimensionality reduction technique t-SNE [40]. It is clearly seen that products that share colour, shape or texture features appear close together.

To determine the pretrained neural network architecture providing the best performance, we conduct several experiments that are illustrated in Fig. 3. As a result, we choose ResNet-50 as our visual feature extraction architecture.

B. TEXT QUERY SEARCH

To extend the functionality of our Style Search Engine, we implement a text query search that allows to further

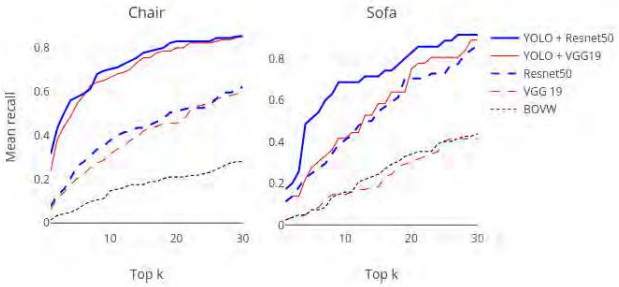


FIGURE 3. Architecture comparison for choosing the object detection model in Visual Search. The recall is plotted as a function of the number of returned items k. Best retrieval results are achieved for YOLO object detection and visual features exraction from Resnet-50.

specify the search criteria. This part of our engine is particularly useful when trying to search for product items that represent abstract concepts such as *minimalism*, *Scandinavian style*, *casual* and so on.

In order to perform such a search, we need to find a mapping from textual information to vector representation of the item, i.e, from the space of textual queries to the space of items in the database. The resulting representation should live in a multidimensional space, where stylistically similar objects reside close to each other.

To obtain the above-defined space embedding, we use a Continuous Bag-of-Words (CBOW) model that belongs to word2vec model family [20]. In order to train our model, we use the descriptions of items available as a metadata supplied with the catalog images. Such descriptions are available as part of both, the IKEA and the Polyvore datasets, which we describe in details in Sec. V. Textual description embedding is calculated as a mean vector of individual words embeddings.

In order to optimize hyper-parameters of CBOW for item 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.

Having found such a mapping, we can perform the search by returning k-nearest neighbors of the transformed query in the space of product descriptions from the database using cosine similarity as a distance measure.

C. CONTEXT SPACE SEARCH

In order to leverage the information about different item compatibility, which is available as a context data (outfit or room), we train an additional word2vec model (using the CBOW model), where different products are treated as words. Compatible sets of those products appearing in the same context are treated as sentences. It is worth noticing that our context embedding is trained without relying on any linguistic knowledge. The only information that the model sees during training is whether given objects appeared in the same set.

Fig. 4 shows the obtained feature embeddings using t-SNE dimensionality reduction algorithm [40] for IKEA dataset. One can see that some classes of objects, *e.g.* those that appear in a bathroom or a baby room, are clustered around the same region of the space.

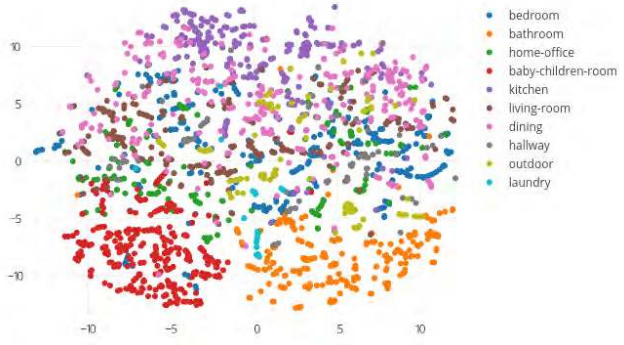


FIGURE 4. T-SNE visualization of interior items' embedding using context information only. Distinctive classes of objects, *e.g.*, those that appear in a bathroom or a baby room, are clustered around the same region of the space. No text descriptions nor information about image room categories was used during training.

D. BLENDING METHODS

Let us denote $p = (i, t)$ to be a representation of a product stored in the database $\mathbb{P}$. This representation consists of a catalog image $i \in \mathfrak{I}$ and the textual description $t \in \mathfrak{T}$. The multimodal query provided by the user is given by $Q = (i_q, t_q)$,where $i_q \in \mathfrak{I}$ is the visual query and $t_q \in \mathfrak{T}$ is the textual query.

We run a series of experiments with blending methods, aiming to combine the retrieval results from various modalities in the most effective way. To that end, we use the following approaches for blending.

1) LATE-FUSION BLENDING

In the simplest case, we retrieve top k items independently for each modality and take them to as a set of final results. We do not use the contextual information here.

2) EARLY-FUSION BLENDING

In order to use the full potential of our multimodal search engine, we combine the retrieval results of visual, textual as well as contextual search engines in the specific order. We optimize this order to present the most stylistically coherent sets to the user. To that end, we propose *Early-fusion Blending* (see Fig. 2) approach that uses features extracted from different modalities in a sequential manner.

More precisely, for a multimodal query (i_q, t_q), an initial set of results R_{vis} is returned for visual modality - closest images to i_q in terms of Euclidean distance d_{vis} between their visual representations. Then, we retrieve contextually similar products R_{cont} that are close to R_{vis} results in terms of d_{cont} distance in context embedding space (context space search described in section III-C). Finally, R_{vis} and R_{cont} form a list of candidate items R_{cand} from which we select the results R by extracting the textual features (word2vec vectors) from items descriptors and rank them using distance from the textual query d_{text}.

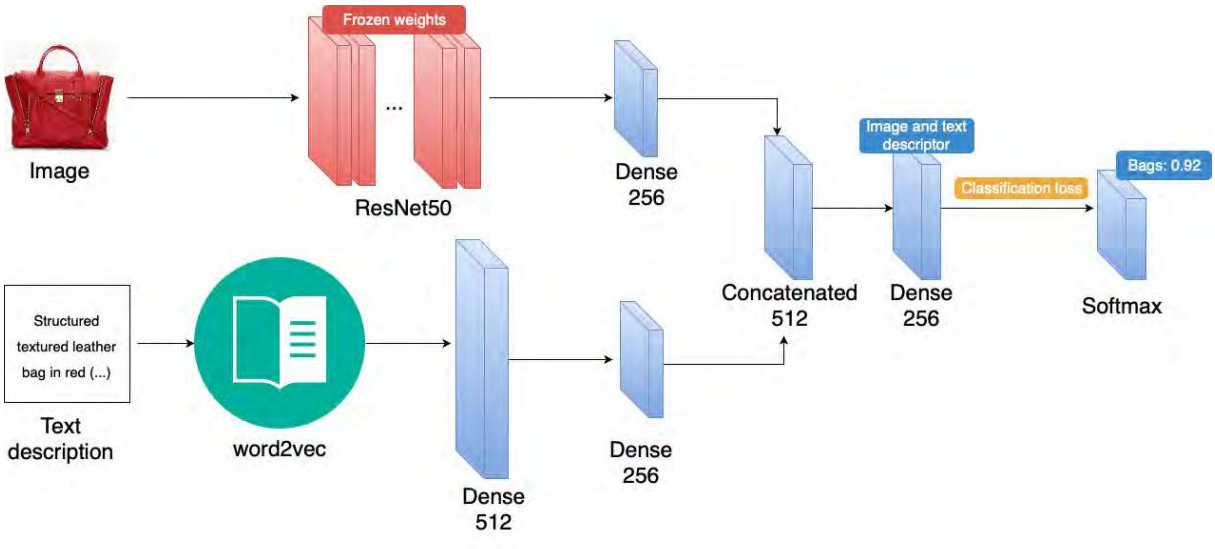


FIGURE 5. The proposed architecture of DeepStyle network. An image is first fed through the ResNet-50 network pretrained on ImageNet while the corresponding text description is transformed with Word2vec. Final layer predicts a clothing item category. Penultimate layer serves as a multimodal image and text representation of the product item.

This process can be formulated as:

$$
\begin{aligned}
R_{vis} &= \left\{ p : \underset{p_1,\ldots,p_{n_1} \in \mathbb{P}}{\operatorname{argmin}} \sum_{j=1}^{n_1} d_{vis}(i_q, i_j) \right\} \\
\Rightarrow R_{cont} &= \bigcup_{r \in R_{vis}} \left\{ p : \underset{p_1,\ldots,p_{n_2} \in \mathbb{P}}{\operatorname{argmin}} \sum_{j=1}^{n_2} d_{cont}(c_r, c_j) \right\} \\
\Rightarrow R_{cand} &= R_{cont} \cup R_{vis} \\
R &= \left\{ p : \underset{p_1,\ldots,p_{n3} \in R_{cand}}{\operatorname{argmin}} \sum_{j=1}^{n_3} d_{text}(t_q, t_j) \right\} \qquad (1)
\end{aligned}
$$

where n_1, n_2 and n_3 are parameters to be chosen.

IV. DEEPSTYLE: MULTIMODAL STYLE SEARCH ENGINE WITH DEEP LEARNING

Inspired by recent advancements in deep learning for computer vision, we experiment with end-to-end approaches that learn the embedding space jointly. In this section, we describe experiments with artificial neural networks that we did to create a joint image-text model. Our goal is to have one model that takes image and text and returns product items satisfying both modalities. First, we start with a simple approach and experiment with a single neural network that is fed with multiple inputs and learns a multimodal embedding space. Such embedding can later be used to retrieve results using a multimodal query. The first proposed architecture is a multimodal DeepStyle network that learns common image-text embedding through classification task. Then, we go further and improve over the first network with the information we have about products' context (outfit). The most straightforward way to make neural network learn the distances between similar and non-similar items is by introducing a Siamese architecture with shared weights and contrastive loss. The resulting architecture that learns to map pairs from the same outfit close in the multi-modal embedding space is called DeepStyle-Siamese network.

A. DEEPSTYLE

Our proposed neural network learns common embedding through classification task. Our architecture, dubbed *DeepStyle*, is inspired by [7], where they use a multimodal joint embedding for fashion product retrieval. In contrast to their work, our goal is not to retrieve images with text query (or vice versa) but to retrieve items where a text query compliments the image and provides additional query requirements.

Similarly to [7], our network has two inputs - image features (output of penultimate layer of pretrained CNN) and text features (processed with the same *word2vec* model trained on descriptions). Each input vector is followed by the *fully-connected* (Dense) layer in order to bring them to the common dimensionality. This way, we avoid our network to be biased towards particular modality. The resulting vectors are concatenated into the single embedding vector and the parameters of the resulting network are optimized for classification loss to enforce the concept of semantic regularities. For this purpose, product category labels (with arbitrary number of classes) should be present in the dataset. Unlike [7], we do not consider the image and the text branches separately for predictions but add a fully connected layer on top of the concatenated image and text embeddings that is used to predict a single class. Illustration of network architecture is presented in fig. 5. For more detailed explanation of neural network components see the Appendix A.

B. DEEPSTYLE-SIAMESE

We want to also include context information (whether or not two items appeared in the same context) to our network. For this purpose, we design a Siamese network [41] where each

branch has a dual input consisting of image and text features. Positive pairs are generated as image-text pairs from the same outfit while unrelated pairs are obtained by randomly sampling an item (image and description) from a different outfit.

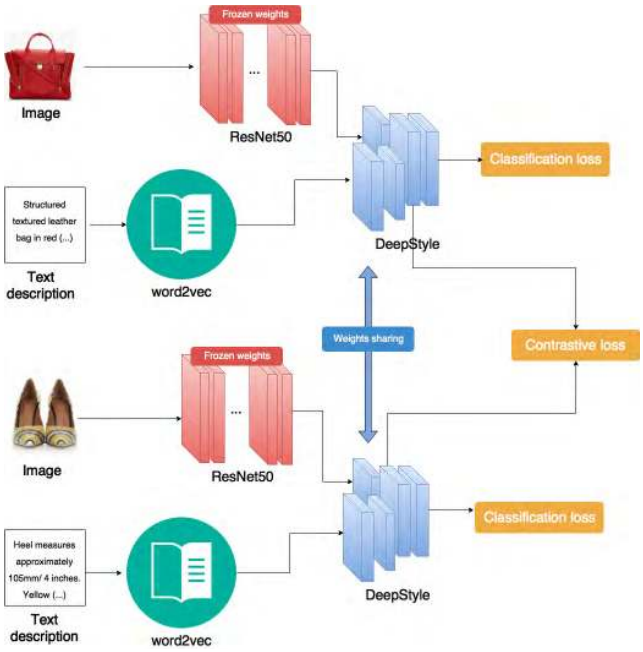


FIGURE 6. The architecture of DeepStyle-Siamese network. DeepStyle block is the block of dense and concatenation layers from Fig. 5 that has shared weights between the image-text pairs. Three kinds of losses are optimised - the classification loss for each image-text branch and the contrastive loss for image-text pairs. Contrastive loss is computed on joint image and text descriptors.

As seen in fig. 6, two types of losses are optimized. Classification loss is used as before to help network learn semantic regularities. Also, minimizing contrastive loss encourages image-text pairs from the same outfit to have a small distance between embedding vectors while different outfit items to have distance larger than a predefined margin.

Formally, contrastive loss is defined in the following manner [41]:

$$L_C(d, y) = (1 - y)\frac{1}{2}d^2 + y\frac{1}{2}\{max(0, m - d)\}^2, \qquad (2)$$

where d is the Euclidean distance between two different embedded image-text vectors (i, t) and (i', t'), y is a binary label indicating whether two vectors are from the same outfit ($y = 0$) or from different outfits ($y = 1$) and m is a predefined margin for the minimal distance between items from different outfits.

Full training loss L consists of weighted sum of contrastive loss and cross entropy classification losses:

$$L = \alpha L_C(d, y) + \beta L_X(Cl_1(i, t), \tilde{y}(i, t)) + \gamma L_X(Cl_2(i', t'), \tilde{y}(i', t')), \qquad (3)$$

where L_X is the cross entropy loss, $Cl_1(i, t)$ and $Cl_2(i, t)$ are outputs of the first and second classification branches respectively and $\tilde{y}(i, t)$ is the category label for product with image i and text description t. Parameters α, β, γ are treated as hyperparameters for tuning.

V. DATASETS

Although several datasets for standard visual search methods exist, *e.g.* Oxford 5K [13] or Paris 6K [42], they are not suitable for our experiments, as our multimodal approach requires an additional type of information to be evaluated. More precisely, dataset that can be used with a multimodal search engine should fulfill the following conditions:

- It should contain both images of individual objects as well as scene images (room/outfit image) with those objects present.
- It should have a ground truth defining which objects are present in scene photo.
- It should also have textual descriptions.

We specifically focus on datasets containing pictures of interior design and fashion as both domains are highly dependant on style and would benefit from style search engine applications. In addition, we analyze datasets with varying degrees of context information, as in real life applications it might differ from dataset to dataset. For example, in some cases (specifically when the database is not very extensive), items can co-occur very often together (in context of the same design, look or outfit). Whereas in other cases, when database of available items is much bigger, the majority of items will not have many co-ocurrences with other items. We apply our Multimodal Search Engine for both types of datasets and perform quantitative evaluation to find the best model.

A. INTERIOR DESIGN

To our knowledge, there is no publicly available dataset that contains the interior design items and fulfill previously mentioned criteria. Hence, we collect our own dataset by scraping the website of one of the most popular interior design distributors - IKEA.[1] We collect 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. 7. We also group together products from some of the most frequent object classes (*e.g.* chair, table, sofa) for more detailed analysis. In addition, we divide room scene photos into 10 categories based on the room class (kitchen, living room, bedroom, children room, office). The vast majority of furniture items in the dataset (especially from the frequent classes above) have rich context as they appear in more than one room.

B. FASHION

Several datasets for fashion related tasks are already publicly available. *DeepFashion* [43] contains 800 000 images divided into several subsets for different computer vision tasks. However, it lacks the context (outfit) information as well as the detailed text description. *Fashion Icon* [28] dataset

[1] *https://ikea.com/*

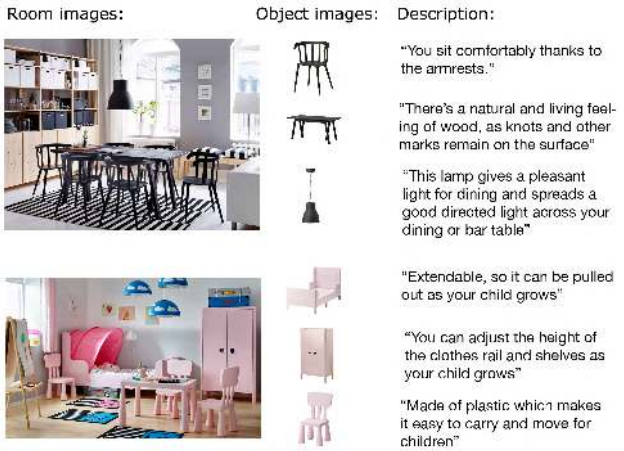


FIGURE 7. Example entries from IKEA dataset. It contains room images, object images, and their respective text descriptions.

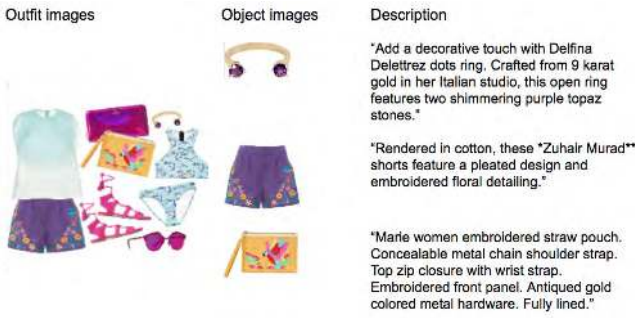


FIGURE 8. Example entries from Polyvore dataset. It contains outfit images, item images, and their respective text descriptions.

contains video frames for human parsing but no individual product images. In contrast, *Polyvore* [31] dataset has satisfied our dataset conditions mentioned before (see Fig. 8).

Polyvore dataset contains 111 589 clothing items that are grouped into compatible outfits (of 5-10 items per outfit). We perform additional dataset cleaning - remove non-clothing items such as electronic gadgets, furniture, cosmetics, designer logos, plants, furniture. In addition, we perform additional scraping of Polyvore[2] website for product items in the cleaned dataset to obtain longer product descriptions and add the descriptions where they are missing. As a result, we have 82 229 items from 85 categories with text descriptions and context information. Context information is much weaker when compared to IKEA dataset. Only 30% of clothing items appear in more than one outfit. Sample images from Polyvore dataset together with illustration of cleaning procedure are shown in fig. 9.

Item (query) images are already object photos. Therefore, for fashion dataset object detection step from style search engine is omitted for evaluation.

VI. EVALUATION

In this section we want to evaluate our method and check how well it performs in the task of finding similar items when compared to baselines by querying on a subset of images and a set of popular text queries.

[2] *http://polyvore.com*

FIGURE 9. Illustration of Polyvore dataset. Cleaned items column shows items removed from the original dataset [31] after cleaning procedures.

A. EVALUATION METRICS

1) SIMILARITY SCORE

As mentioned in Sec. II-C, defining a similarity metric that allows quantifying the stylistic similarity between products is a challenging task and an active area of research. In this work, we propose the following similarity measure that is inspired by [25] and based on the probabilistic data-driven approach.

Let us remind that $\mathcal{P}$ is a set of all possible product items available in the catalog. Let us then denote $\mathcal{C}$ to be a set of all sets that contain stylistically compatible items (such as outfits or interior design rooms). Then we search for a similarity function between two items $p_1, p_2 \in \mathcal{P}$ which determines if they fit well together. We propose the empirical similarity function $s_c : \mathcal{P} \times \mathcal{P} \rightarrow [0, 1]$ which is computed in the following way:

$$s_c(p_1, p_2) = \frac{|\{C_i \in \mathcal{C} : p_1 \in C_i \wedge p_2 \in C_i\}|}{\max_{p \in \{p_1, p_2\}} |\{C_j \in \mathcal{C} : p \in C_j\}|}. \quad (4)$$

In fact, it is the number of compatible sets $\mathcal{C}_i$ that are empirically found from $\mathcal{C}$, in which both p_1 and p_2 appear, normalized by the maximum number of compatible sets in which any of those items occur. This metric can be interpreted as an empirical probability for the two objects p_1 and p_2 to appear in the same compatible set and it is expressed by the similarity score lying in the interval [0, 1]

In order to account for datasets that have weak context information (where two items rarely co-occur in the same compatible set), we add an additional similarity measure s_n that is directly derived from their name overlap. It counts for overlap of some of the most frequent descriptive words such as *elegant, denim, casual, etc*. It should be mentioned, however, that product name information should be independent

FIGURE 10. T-SNE visualization of clothing items' visual features embedding. Distinctive classes of objects, *e.g.* those that share visual similarities are clustered around the same region of the space.

from the text description (that is used during training). As a result, name-derived similarity is non-zero only on datasets that have this kind of additional name information.

$$s_n(p_1, p_2) = \mathbb{1}\{W_{p_1} \cap W_{p_2} \neq \emptyset\}, \tag{5}$$

where W_f is a set of frequent descriptive words appearing in the name of item f.

To summarize, an evaluated pair is considered to be similar if either of the two conditions is satisfied:

- items co-occurred in the same outfit before
- names of the two items are overlapping

Formally,

$$s(p_1, p_2) = \max\left(s_c(p_1, p_2), s_n(p_1, p_2)\right). \tag{6}$$

We also experiment with alternative similarity function. The main motivation to use this method, is an attempt to capture the transitive nature of the style compatibility. In other words, if an item 1 appeared in the set A but not in B, while item 2 appeared in the set B but not in A, we can still treat them as somehow compatible if there is an item 3 which appeared in both A and B.

Firstly, in order to leverage the information about different item compatibility, which is available as an empirical compatible sets data, we train a word2vec model were different products are treated as words and compatible set's of those products, appearing in the same outfits, as sentences. The model has a context window of 3 items and does not ignore any item appearing at least once, and it is set to produce 100 dimensional vectors using CBOW.

Secondly, for similar purpose, we train word2vec model on the item names data. This way, we evaluate the extent to which our system capture the semantic information contained in the textual part of the multimodal query. The model has a context window of 4 words and similarly builds vocabulary on all words appearing at least once, producing 100 dimensional vectors using CBOW.

Finally, to compute style similarity score between the two items we use an average of the two cosine distances in the mentioned embeddings. This way we achieve continuous method for evaluation.

2) INTRA-LIST SIMILARITY

Given that our multimodal query search engine provides a non-ranked list of stylistically similar items, the definition of the evaluation problem differs significantly from other information retrieval domains. For this reason, instead of using some of the usual metrics for performance evaluation like mAP [44] or nDCG [45], which use a ranked list of items as an input, we apply a modified version of the established metric for non-ranked list retrieval. Inspired by the [46], we define the average intra-list similarity for a generated results list R of length k to be:

$$AILS(R) = \binom{k}{2}^{-1} \sum_{p_i \in R} \sum_{p_j \in R, p_i \neq p_j} s(p_1, p_2), \tag{7}$$

that is an average similarity score computed across all possible pairs in the list of generated items. By doing so, we are aiming to assess the overall compatibility of the generated set. As mentioned in [46], this metric is also permutation-insensitive, hence the order of retrieved results does not matter, making it suitable for not ranked results.

B. BASELINE METHODS

In experiments, we compare our approach with several baselines.

One area of research that uses multimodal representations is Visual Question Answering (VQA). We take several recent methods and use their intermediate multimodal embedding as feature extractor for products database. Then comparison is made with our method in multimodal products retrieval.

We fine-tune the weights of Visual Semantic Embedding (VSE) [8] model made publicly available by authors on our datasets. The model was pretrained on MS COCO dataset that has 80 categories with broad semantic context, hence it's applicable to our datasets. Original VSE implementation uses VGG 19 [47] architecture for feature extraction. In order to allow fair comparison, we train an additional baseline model with VSE that uses Resnet-50 as a feature extractor.

Another recent VQA approach for multimodal representation learning from text and image is MUTAN [30]. It is a multimodal tensor-based Tucker decomposition that efficiently parametrizes bilinear interactions between visual and textual representations.

Furthermore, we compare our approach with one-shot multimodal search [32] that has been recently applied for item retrieval from multimodal database. The proposed method

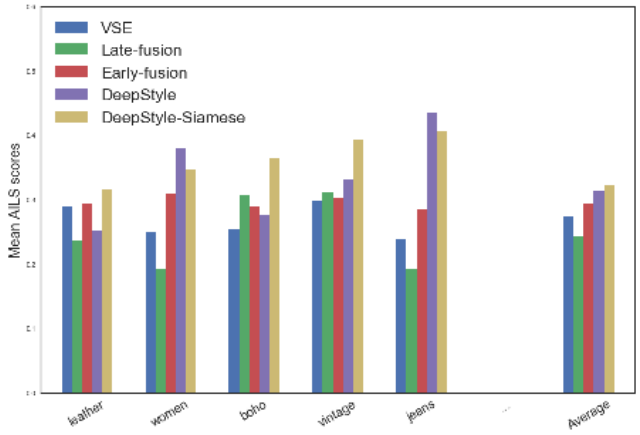


FIGURE 11. Mean AILS metric scores for selected textual queries and the average of the mean scores for other methods and strongest baseline (VSE-VGG). We can see that our DeepStyle-Siamese architecture significantly outperforms other architectures on multiple text queries.

generates augmented feature vectors from concatenated text and image vectors.

We also compare our method with Late and Early-fusion Blending strategies from our previous work [9].

C. RESULTS

1) TRAINING DETAILS

We run the training for 50 epochs with batch size set to 128 until validation loss stops improving. We experiment with several optimizers and achieve the best results with Adam [48] and learning rate set to be 0.0001. More detailed experiments with different metrics and architecture structure are presented in section B (Appendix B). Training time of one epoch takes on average 230 seconds on GeForce GTX 1080 Ti graphics card.

2) EVALUATION PROTOCOL

In order to test the ability of our method to generalize, we evaluate it using a dataset different from the training dataset. For both datasets, we set aside 10% of the initial number of items for that purpose. All results shown in this section come from the following evaluation procedure:

1) For each item/text query from the test set we extract visual and textual features.
2) We run engine and retrieve a set of k most compatible items from the trained embedding space.
3) We evaluate the query results by computing an Average Intra-List Similarity metric for all possible pairs between the retrieved items and the query, which gives $\binom{k}{2}$ pairs for k retrieved items.
4) The final results are computed as the mean of AILS scores for all of the tested queries.

It should be noted that for the IKEA dataset, object detection is performed on room images and similar items are returned for the most confident item in the picture. On the other hand, for Polyvore dataset, the test set images are already catalog items of clothes on white background, hence the object detection is not necessary and this step is omitted.

3) QUANTITATIVE RESULTS

Tab. 1 shows the results of the blending methods for the IKEA dataset in terms of the mean value of our similarity metric.

When analyzing the results of blending approaches, we experiment with several textual queries in order to evaluate system robustness towards changes in the text search. We observe that DeepStyle approach outperforms all baselines for almost all text queries achieving the highest average similarity score. DeepStyle-Siamese approach gives the best results, outperforming the strongest baseline (VSE-VGG19) by 21% for IKEA dataset. It should also be noted that network complexity is not directly correlated with its ability to learn style similarity that is illustrated by worse similarity results on VSE baseline that extracts Resnet-50 features instead of VGG-19. For coherence, we include an additional experiment of training DeepStyle-Siamese network with VGG-19 feature extraction as input. Similarity values on test set for this DeepStyle version are slightly worse than trained with Resnet features, however the difference is not significant.

Tab. 2 shows the results of all of the tested methods for the Polyvore dataset in terms of the mean value of our similarity metric. Here, we also evaluate two joint architectures, namely DeepStyle and DeepStyle-Siamese. Fig.11 shows that DeepStyle architecture yields better results in terms of an average performance over different textual queries, when compared to our previous blending approaches, as well as other baselines. In this case, DeepStyle-Siamese also yields the best average similarity results. In terms of an average performance, it scores by 32% higher, when compared to the strongest baseline model, and more than 4% higher, when compared to DeepStyle.

TABLE 1. Mean AILS results averaged for IKEA dataset and sample text queries from the set of most frequent words in text descriptions.

Text query	MUTAN [30]	One-shot MM search [32]	VSE [8]		Blending [9]		DeepStyle	DeepStyle-Siamese
			ResNet	VGG19	Late-fusion	Early-fusion		
decorative	0.1589	0.1443	0.1526	0.1475	0.2742	0.2332	0.2453	**0.2840**
black	**0.3271**	0.1332	0.1928	0.3217	0.2361	0.2354	0.1967	0.2237
white	0.1588	0.1219	0.1693	0.1476	0.2534	0.2048	0.1730	**0.2742**
smooth	0.0011	0.1436	0.1158	0.1648	0.2667	0.2472	**0.3022**	0.2642
cosy	0.1121	0.1454	0.2116	0.2918	0.1073	0.2283	**0.3591**	0.2730
fabric	0.0280	0.1485	0.2437	0.1038	0.1352	0.2225	0.0817	**0.2487**
colourful	0.1121	0.1367	0.1839	0.3163	0.2698	0.2327	**0.3568**	0.2623
Average	0.1295	0.1391	0.1814	0.2134	0.2164	0.2287	0.2449	**0.2589**

TABLE 2. Mean AILS results for fashion Search on Polyvore dataset. Sample text queries are selected from the set of most frequent words in text descriptions.

Text query	MUTAN [30]	One-shot MM search [32]	VSE [8]		Blending [9]		DeepStyle	DeepStyle-Siamese
			ResNet	VGG19	Late-fusion	Early-fusion		
black	0.1520	0.1439	0.2580	0.2932	0.2038	0.3038	0.334	**0.421**
white	0.1682	0.1439	0.2610	0.2524	0.2047	0.2898	0.295	**0.360**
leather	0.1510	0.1439	0.2607	0.2885	0.2355	0.2946	0.267	**0.432**
jeans	0.1668	0.1417	0.2565	0.2381	0.1925	0.2843	0.609	**0.619**
wool	0.1603	0.1428	0.2578	0.3025	0.1836	0.2657	**0.501**	0.310
women	0.1491	0.1395	0.2566	0.2488	0.1931	0.3088	0.380	**0.414**
men	0.1910	0.1417	0.2662	0.2836	0.1944	**0.2900**	0.270	0.236
floral	0.1642	0.1406	0.2635	0.2729	0.3212	0.2954	0.371	**0.442**
vintage	0.1782	0.1329	0.2567	0.2986	0.3104	0.3035	0.428	**0.540**
boho	0.1577	0.1395	0.2597	0.2543	0.3074	0.2893	**0.325**	0.246
casual	0.1663	0.1377	0.2626	0.2808	0.3361	0.3030	0.337	**0.4**
Average	0.1504	0.1418	0.2383	0.2740	0.2439	0.2935	0.374	**0.402**

TABLE 3. Mean similarity per text query category with and without context information available during the training stage.

Text query category	Avg similarity (Wikipedia) *no context*	Avg similarity (product descriptions) *with context*
Color	0.2888	0.2898
Human body	0.2911	0.2934
Fabrics	0.2909	0.2935
Style	0.2893	0.2941

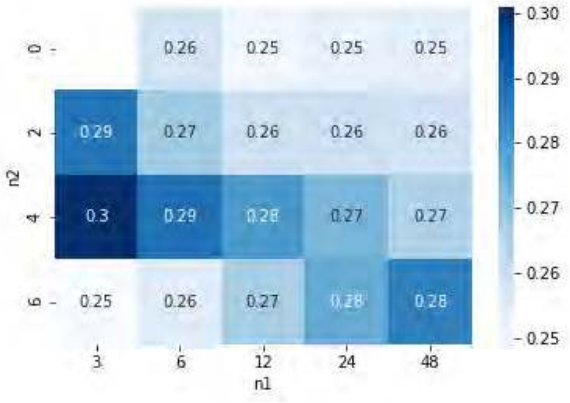


FIGURE 12. Hyperparameter analysis for early-fusion blending and $n_3 = 4$ number of final results. The choice of $n_1 = 3$ and $n_2 = 4$ gives optimal similarity results.

It can be observed by the reader, that adding contextual information helps both systems to achieve better results. For blending approaches, Early-fusion, where the contextual embedding was used as a part of the retrieval process, outperforms Late-fusion, where this embedding was not used. Similarly, DeepStyle-Siamese architecture which was learned using matching pairs of furniture, hence implicitly using contextual information, outperforms plain DeepStyle architecture which was not using it.

4) TEXT QUERY ANALYSIS

The choice of text queries for input is completely arbitrary as they provide additional description that does not have to be related to image content. Hence we analyze if any types of text queries work better with our model. We group the set of most common descriptive words in Polyvore descriptions by separate categories, such as fabrics (leather, suede, denim),

TABLE 4. Mean number of distinct categories present in the results list for different fashion search methods.

Method	Avg number of categories
VSE-Resnet [8]	2.47
VSE-VGG [8]	2.87
DeepStyle-Siamese	3.01
Late-fusion Blending	3.08
MUTAN [30]	3.47
Early-fusion Blending	3.89

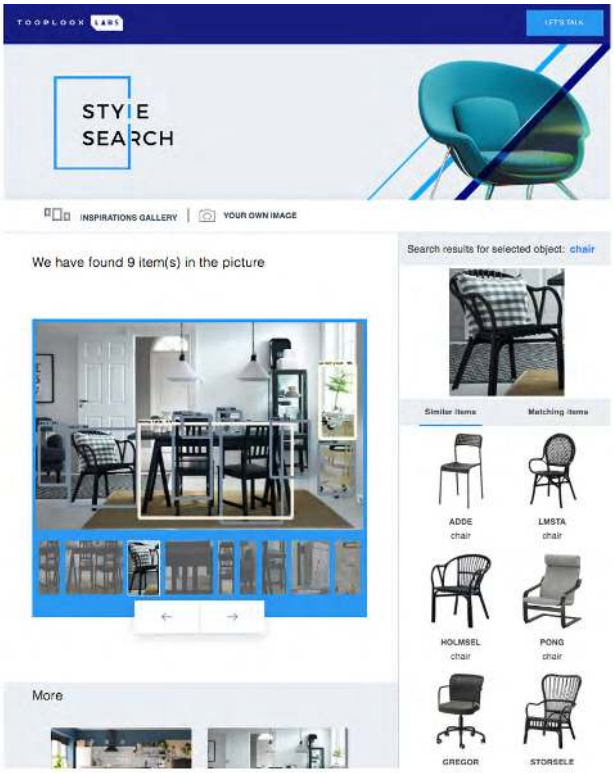


FIGURE 13. Sample screenshot of our Style Search Engine for interior design applied in web application showing product detection and retrieval of visually similar products.

FIGURE 14. Sample multimodal results compared with baselines. We observe how results differ for typical fashion text queries ("wool", "men", "leather", "boho") on two test images. Our method (DeepStyle-Siamese) returns similar items corresponding to text query as well as extends visual search with items from different categories that share style similarities. We can see how our approach is different from other methods that focus mostly on shapes and colours similarity (VSE-VGG, VSE-Resnet) or methods that rely stronger on text retrieval (MUTAN).

color, style (floral, vintage, classic) and human body (ankle, skinny, average). The comprehensive analysis is presented in table 3. One may observe that text queries related to color give slightly better similarity results. This might seem intuitive as the concept of color seems easy to define and learn. On the other hand, the average similarity difference is not significant between various text groups, implying that all types of queries can be used with our method.

5) CONTEXTUAL ANALYSIS

Moreover, we investigated influence of using embedding trained on data coming from the domain source and general source. For this reason, we trained our word2vec embeddings using two datasets separately, firstly using the data dump from English Wikipedia [49], hence not including contextual information and secondly on the dataset that was built using all product descriptions, hence taking contextual approach.

As it can be observed in the table 3, the differences in the performance are not significantly large. From the practical perspective however, the Wikipedia dataset is substantially larger, which influences both the training time and the size of the embedding. For this reason, we decided to use the embedding trained on the dataset of product descriptions in the final model.

6) HYPERPARAMETER ANALYSIS

We analyze influence of hyper-parameters on blending methods. The number of final results presented to the user is set to 4 for all methods and baselines, hence we set $n_3 = 4$. Figure 12. displays how different values of n_1 and n_2 impact similarity. In the range of considered values we observe that the right balance between parameters gives similarity values higher by 0.05 and the optimal parameters are $n_1 = 3$ and $n_2 = 4$.

7) CATEGORY DIVERSITY ANALYSIS

We analyze mean number of distinct object categories present in the results set (for each query and $k = 4$ result items). Mean similarity as it is defined in Section VI depends on both name similarity as well as item co-occurence in outfit. Hence, method that would only return similar objects of the same class would not maximize the similarity metric. We see from Table 4 that VSE-Resnet has the lowest average number of distinct categories, which suggests that results from this method mostly focus on visual similarity. On the other side of the spectrum, MUTAN [30] and Early-Fusion Blending results have the most intra-results category diversity which means lower similarity in terms of object categories.

8) QUALITATIVE RESULTS

Fig. 1 and 16 display sample results for user queries in both fashion and interior design domains. Fig. 1 illustrates that semantics are preserved with multimodal query and user is presented with results that combine both visual and textual queries. In the Fig. 14 detailed qualitative analysis is presented. Multimodal search results are shown for sample images with typical fashion text queries. We can see that our method is capable of retrieving visually similar results that correspond to text query but can also extend to objects from different categories that fit the semantics and have higher outfit compatibility.

VII. WEB APPLICATION

To apply our method in real-life application, we implemented a Web-based application of our Style Search Engine with application to Interior Design. 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 Flask[3] - a lightweight server library. It is currently released to public.[4] Fig. 16 shows a screenshot from the working Web application with Style Search Engine.

[3]http://flask.pocoo.org/

[4]http://stylesearch.tooploox.com/

VIII. CONCLUSIONS

In this paper we propose a new method for multimodal query item retrieval. The proposed method is a Siamese neural network architecture that learns style similarity by leveraging on empirical context information - how often given items appear in the same stylistic context. Our method surpasses baseline methods and achieves state-of-the-art results for the generation of stylistically compatible item sets using multimodal queries.

The biggest advantage of our method is two-fold. First, it allows to extend the visual query with arbitrary text input and convey information that is not included in visual input, thus allowing the user to find better suited products. Second, it retrieves results that are compatible stylistically.

The main disadvantage of the method is the need for vast labeled data in terms of scene images (context information where items appear together). Semi-supervised learning approaches that could reduce the need for such data are subject to our future work.

We successfully apply our methodology for several commercial domain applications - fashion and interior design, by exploiting the product images and their associated metadata. Finally, we deploy a publicly available web implementation of our solution and release the new data set with the IKEA furniture items.

APPENDIX A

The basic model of *Fully-connected network*, also known as the *Multi-layered Perceptron*, can be described as a series of functional transformations. Given the n-dimensional input vector $x = x_1, x_2, ...x_n$ we construct M linear combinations of the input variables as follows:

$$a_j(\mathbf{x}) = w_{j0}^{(1)} + \sum_{i=1}^{n} w_{ij}^{(1)} x_i, \tag{8}$$

where $j = 1, \ldots, M$ and M is a parameter to be selected. Superscript (1) indicates that the corresponding weight parameters (represented by the links in the network) are in the first layer of the network, and $w_{j0}^{(1)}$ is called the bias term. Such combinations are often called *activations*. Each activation is transformed using the *activation function*, which is chosen depending on the network layer type. For multilabel classification layers, the softmax is typically used. With activation function applied the transformation is the following:

$$h_j(\mathbf{x}) = \sigma_{hidden}\left(a_j\right) = \sigma_{hidden}\left(w_{j0}^{(1)} + \sum_{i=1}^{n} w_{ij}^{(1)} x_i\right), \tag{9}$$

where σ_{hidden} is the activation function for hidden units. It can be observed that for σ_{hidden} equal to identity function neural network model becomes a linear regression model. Similarly, if the sigmoid function is being used, resemblance to logistic regression can be immediately observed.

TABLE 5. Experiments with architecture structure of deepStyle model on Polyvore dataset.

Text model	Text embedding size	dense_1	dense_2	Multimodal embedding size	Multimodal dense layers	dense_mm	Loss	Accuracy
GloVe	200	512	256	512	1	-	0.846	0.742
GloVe	200	512	512	1024	1	-	**0.823**	**0.748**
GloVe	200	1024	64	128	1	-	0.915	0.723
GloVe	200	1024	128	256	1	-	0.929	0.715
GloVe	200	1024	256	512	1	-	0.841	0.744
GloVe	200	1024	512	1024	1	-	0.843	0.746
GloVe	200	512	256	512	2	512	0.849	0.738
GloVe	200	512	512	1024	2	1024	0.859	0.74
GloVe	200	1024	64	128	2	128	0.862	0.733
GloVe	200	1024	128	256	2	256	0.874	0.734
GloVe	200	1024	256	512	2	512	0.854	0.742
GloVe	200	1024	512	1024	2	1024	0.861	0.739
GloVe	300	512	256	512	1	-	**0.697**	0.782
GloVe	300	512	512	1024	1	-	0.702	0.785
GloVe	300	1024	64	128	1	-	0.746	0.771
GloVe	300	1024	128	256	1	-	0.713	0.778
GloVe	300	1024	256	512	1	-	0.705	0.785
GloVe	300	1024	512	1024	1	-	0.736	**0.787**
GloVe	300	512	256	512	2	512	0.721	0.777
GloVe	300	512	512	1024	2	1024	0.744	0.780
GloVe	300	1024	64	128	2	128	0.749	0.770
GloVe	300	1024	128	256	2	256	0.734	0.777
GloVe	300	1024	256	512	2	512	0.748	0.778
GloVe	300	1024	512	1024	2	1024	0.827	0.777
Word2vec	**260**	**512**	**256**	**512**	**1**	**-**	**0.695**	**0.788**
Word2vec	260	512	512	1024	1	-	0.702	0.784
Word2vec	260	1024	64	128	1	-	0.737	0.774
Word2vec	260	1024	128	256	1	-	0.708	0.782
Word2vec	260	1024	256	512	1	-	0.707	0.786
Word2vec	260	1024	512	1024	1	-	0.742	0.782
Word2vec	260	512	256	512	2	512	0.720	0.778
Word2vec	260	512	512	1024	2	1024	0.755	0.779
Word2vec	260	1024	64	128	2	128	0.749	0.767
Word2vec	260	1024	128	256	2	256	0.726	0.775
Word2vec	260	1024	256	512	2	512	0.730	0.784
Word2vec	260	1024	512	1024	2	1024	0.821	0.780

In the context of our paper, we use fully-connected layers in two ways. First, in order to reduce the dimensionality of the input vectors, we use a single layer of hidden units with the number of activations that matches desired output dimensionality. As an activation function in this layer, we use ReLu (rectified linear unit), defined as:

$$f(x) = \max\{0, x\},$$

which is widely recommended activation functions in the deep learning community [50]. In the case of classification layer, the number of activation units corresponds to the number of classes, which are followed by the sigmoid activation.

The training of neural network, defined as obtaining optimal weight and bias terms, is done via minimization of the loss function, typically by (stochastic) gradient descent (*back-propagation*) [51]. To optimize our network we use *Adam*, which belongs to the family of the so called *adaptive learning* optimizers [48]. It is one of the most widely used algorithms for optimization of deep networks, known for its fast convergence property [50].

TABLE 6. Comparison of distance metrics in training DeepStyle-Siamese model. Architecture, learning rate, batch size, and other hyper-parameters were kept the same during all experiments. The best results were achieved with Euclidean and L1 metric for both data sets.

Dataset	Metric	Validation Loss	Contrastive Accuracy	Categorical Accuracy
Polyvore	Euclidean	0.833	**0.620**	0.553
Polyvore	L1	**0.615**	0.586	**0.787**
Polyvore	Cosine	0.794	0.510	0.781
Polyvore	Chi Squared	0.653	0.593	0.784
IKEA	Euclidean	0.530	**0.560**	0.953
IKEA	L1	**0.505**	0.521	**0.967**
IKEA	Cosine	0.652	0.507	0.952
IKEA	Chi Squared	0.547	0.553	0.958

APPENDIX B

We provide additional experimental analysis of hyper-parameters related to network structure such as number of layers, number of neurons in fully-connected layers, embedding size, text embedding model and choice of distance metrics. Results in Table 6 show that Euclidean and L1 distance

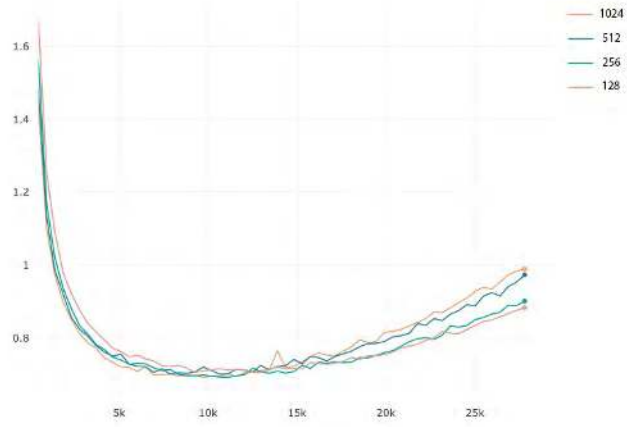


FIGURE 15. Validation loss per step for different multimodal embedding sizes and Polyvore data set. Similar loss values are achieved at earlier training stages but for larger number of iterations choosing higher embedding size is more prone to overfitting.

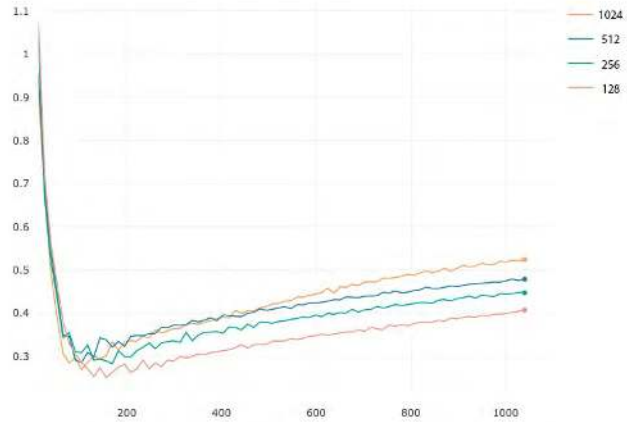


FIGURE 16. Validation loss per step for different multimodal embedding sizes and IKEA data set. For larger multimodal embedding size network quickly stops improving, which suggests that for smaller data sets a smaller multimodal embedding size should be used.

metrics are best suited for the defined task. The findings in Table 5 suggest that the best performance is achieved with Word2vec text model for text vectorizing and single multimodal dense layer. Some of the parameters depend on data features such as data set size. As illustrated in Figures 15. and 16, higher values of embedding size (number of neurons in Dense layer) might lead to overfitting on smaller datasets (e.g IKEA data set). Hence, the number of neurons in fully connected layers might be adjusted empirically for other applications.

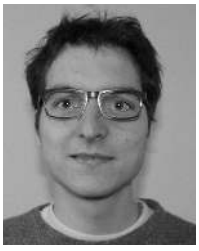
REFERENCES

[1] G. Bradski. (2014). *OpenCV*. [Online]. Available: https://opencv.org/
[2] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," *Toward Category-Level Object Recognition*. Berlin, Germany: Springer, 2006.
[3] B. Davis. (2017). *Image Recognition in Ecommerce: Visual Search, Product Tagging and Content Curation*. [Online]. Available: https://econsultancy.com/blog
[4] H. Li, Y. Wang, T. Mei, J. Wang, and S. Li, "Interactive multimodal visual search on mobile device," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 594–607, Apr. 2013.
[5] J. Sang, T. Mei, Y.-Q. Xu, C. Zhao, C. Xu, and S. Li, "Interaction design for mobile visual search," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1665–1676, Nov. 2013.
[6] D. M. Chen and B. Girod, "A hybrid mobile visual search system with compact global signatures," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 1019–1030, Jul. 2015.
[7] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1946–1955, Aug. 2017.
[8] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *CoRR*, vol. abs/1411.2539, Nov. 2014.
[9] I. Tautkute, A. Mo ejko, W. Stokowiec, T. Trzci ski, Ł. Brocki, and K. Marasek, "What looks good with my sofa: Multimodal search engine for interior design," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, vol. 11, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., 2017, pp. 1275–1282.
[10] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. CVPR*, 2006, pp. 2161–2168.
[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[12] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, nos. 2–3, pp. 146–162, 1954. doi: 10.1080/00437956.1954.11659520.
[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. CVPR*, 2007, pp. 1–8.
[14] L. Xie, J. Wang, B. Zhang, and Q. Tian, "Fine-grained image search," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 636–647, May 2015.
[15] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. ICLR*, 2016.
[16] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 98:1–98:10, 2015.
[17] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin, "Visual search at alibaba," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 993–1001.
[18] H. Hu, Y. Wang, L. Yang, P. Komlev, L. Huang, X. S. Chen, J. Huang, Y. Wu, M. Merchant, and A. Sacheti, "Web-scale responsive visual search at bing," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 359–367.
[19] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.
[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, Jan. 2013.
[21] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1–12.
[22] M. E. Yumer and L. B. Kara, "Co-constrained handles for deformation in shape collections," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 187.
[23] O. van Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or, "Co-hierarchical analysis of shape structures," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 69.
[24] Z. Lun, E. Kalogerakis, and A. Sheffer, "Elements of style: Learning perceptual shape style similarity," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 84:1–84:14, 2015.
[25] *Art History and Its Methods: A Critical Anthology*, vol. 33, no. 6. London, U.K.: Phaidon Press, 1996.
[26] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, "Neuroaesthetics in fashion: Modeling the perception of fashionability," in *Proc. CVPR*, Jun. 2015, pp. 869–877.
[27] 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 *Proc. CVPR*, 2012, pp. 3330–3337.
[28] S. Liu, X. Liang, L. Liu, K. Lu, L. Lin, X. Cao, and S. Yan, "Fashion parsing with video context," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1347–1358, Aug. 2015.
[29] B. Zhao, X. Wu, Q. Peng, and S. Yan, "Clothing cosegmentation for shopping images with cluttered background," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1111–1123, Jun. 2016.
[30] H. Ben-younes, R. Cadène, M. Cord, and N. Thome, "MUTAN: Multimodal tucker fusion for visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2631–2639.

[31] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional LSTMs," in *Proc. ACM Multimedia Conf.*, Mountain View, CA, USA, Oct. 2017, pp. 1078–1086.
[32] J. Yim, J. J. Kim, and D. Shin, "One-shot item search with multimodal data," 2018, *arXiv:1811.10969*. [Online]. Available: https://arxiv.org/abs/1811.10969
[33] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. S. Feris, "Dialog-based interactive image retrieval," in *Proc. NIPS*, 2018, pp. 678–688.
[34] S. Agarwal, O. Dušek, I. Konstas, and V. Rieser, "A knowledge-grounded multimodal search-based conversational agent," in *Proc. EMNLP Workshop SCAI, 2nd Int. Workshop Search-Oriented Conversational AI*, 2018, pp. 1–8.
[35] S. Agarwal, O. Dušek, I. Konstas, and V. Rieser, "Improving context modelling in multimodal dialogue generation," in *Proc. 11th Int. Conf. Natural Lang. Gener.*, 2018, pp. 1–6.
[36] Y. Jing, D. C. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel, "Visual search at pinterest," *CoRR*, vol. abs/1505.07647, 2015.
[37] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7263–7271.
[38] J. Redmon. (2016). *Darknet: Open Source Neural Networks in C*. [Online]. Available: http://pjreddie.com/darknet/
[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[40] G. Hinton and L. van der Maaten, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[41] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, 2006, pp. 1735–1742.
[42] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. CVPR*, 2008, pp. 1–8.
[43] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1096–1104.
[44] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[45] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.
[46] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. ICWWW*, 2007, pp. 22–32.
[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, pp. 730–734, Sep. 2014. doi: 10.1109/ACPR.2015.7486599.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.
[49] (2017). *English Wikimedia Downloads*. [Online]. Available: https://dumps.wikimedia.org/2
[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[51] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. (Springer Series in Statistics). New York, NY, USA: Springer, 2001.

IVONA TAUTKUTE received the B.S and M.S. degrees in mathematics from the University of Warsaw, Poland, in 2015. She is currently pursuing the Ph.D. degree in computer science with the Institute of Multimedia, Polish-Japanese Academy of Information Technology. She is currently an AI/ML Researcher with Tooploox Sp. z o.o., Poland. She is the author or the coauthor of several scientific papers at international conferences. Her research interests include computer vision, artificial neural networks, and artificial intelligence.

TOMASZ TRZCI SKI received the M.Sc. degree in research on information and communication technologies from the Universitat Politècnica de Catalunya, the M.Sc. degree in electronics engineering from the Politecnico di Torino, in 2010, and the Ph.D. degree in computer vision from the École Polytechnique Fédérale de Lausanne, in 2014. He is currently an Assistant Professor with the Division of Computer Graphics, Institute of Computer Science, Warsaw University of Technology. His professional appointments include Telefónica R&D, in 2010, Qualcomm Corporate R&D, in 2012, and Google, in 2013. In 2017, he was appointed as a Visiting Scholar with Stanford University. He is a member of the Computer Vision Foundation and the Scientific Board of the PLinML Conference. He is a Chief Scientist and a Partner with Tooploox Sp. z o.o., a software services company with more than hundred people on board, where he leads a team of machine learning researchers and engineers. He is currently an Associate Editor of IEEE Access and frequently serves as a Reviewer of major computer vision conferences including CVPR, ICCV, ECCV, ACCV, BMVC, ICML, and MICCAI, and international journals such as TPAMI, IJCV, CVIU, TIP, and TMM.



ALEKSANDER P. SKORUPA received the B.Sc. degree in mathematics and economics from the University of Glasgow and the M.Sc. degree *(summa cum laude)* in statistics from Edinburgh University. He joined Tooploox Sp. z o.o., as a Machine Learning Researcher, in 2017. His research interests include mainly on recommendation systems and data mining in social media.

ŁUKASZ BROCKI defended his doctoral thesis entitled connexional language model in speech ecognition systems, in 2011. He has participated in many international research projects, including Luna, EU-Bridge, or Clarin. His current research interests include technologies related to deep learning, evolutionary optimization, multilayered, recurrent neural networks, and extreme learning machines. Since 2016, he has been a member of the Scientific Advisory Board of Complexica.

KRZYSZTOF MARASEK received the Ph.D. degree from the Warsaw University of Technology, and the Postdoctoral degree from the University of Stuttgart, in 2004, where he was a Senior Scientist with the Stuttgart Sony Tech Center. He is the Head of the Multimedia Department, Polish-Japanese Academy of Information Technology. Since 2006, he has been a member of the Faculty's Scientific Council, a Visiting Professor with the University of North Carolina, Charlotte, USA, and a member of the IPPT Scientific Council and the Institute of Information Processing in Warsaw. He is a Reviewer of the Fifth and Sixth Edition of the EU Framework Program.