Local Matching Networks for Engineering Diagram Search

Zhuyun Dai *
Carnegie Mellon University
zhuyund@cs.cmu.edu

Zhen Fan *
Tsinghua University
fanz15@mails.tsinghua.edu.cn

Hafeezul Rahman
Carnegie Mellon University
hmohamma@andrew.cmu.edu

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

ABSTRACT

Finding diagrams that contain a specific part or a similar part is important in many engineering tasks. In this search task, the query part is expected to match only a small region in a complex image. This paper investigates several local matching networks that explicitly model local region-to-region similarities. Deep convolutional neural networks extract local features and model local matching patterns. Spatial convolution is employed to cross-match local regions at different scale levels, addressing cases where the target part appears at a different scale, position, and/or angle. A gating network automatically learns region importance, removing noise from sparse areas and visual metadata in engineering diagrams.

Experimental results show that local matching approaches are more effective for engineering diagram search than global matching approaches. Suppressing unimportant regions via the gating network enhances accuracy. Matching across different scales via spatial convolution substantially improves robustness to scale and rotation changes. A pipelined architecture efficiently searches a large collection of diagrams by using a simple local matching network to identify a small set of candidate images and a more sophisticated network with convolutional cross-scale matching to re-rank candidates.

KEYWORDS

Diagram Search, Neural IR, Image Retrieval

ACM Reference Format:
Zhuyun Dai, Zhen Fan *, Hafeezul Rahman, and Jamie Callan. 2019. Local Matching Networks for Engineering Diagram Search. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313500

1 INTRODUCTION

In large engineering organizations that manufacture complex products, many engineers at different locations may collaborate to design a product. Searching a corpus of engineering diagrams for similar parts helps staff estimate production costs for new parts based on the costs of similar existing parts; and reduce production costs by using existing parts or replacing several slightly different parts with a new, more widely applicable part.

Engineering diagrams use line drawings and metadata such as part reference numbers to represent the elements of a part or an assembly of parts. Often they are created by computer-aided design (CAD) systems. However, the detailed descriptions stored in CAD systems are unwieldy, often unnecessary, and sometimes too sensitive to be shared widely within an organization. Thus, engineering diagrams – *images* – are the main method of describing parts and assemblies for many tasks. Finding engineering diagrams that contain a specific part or assembly is a type of *image retrieval*.

Image retrieval is a long-studied research area. Most prior research focuses on pictures, and specifically how to represent pictures so that similar pictures are easy to recognize. The recent advances achieved by convolutional neural networks (CNN) create opportunities for generating competitive image representations. Several works investigate taking the output of the network, or aggregating intermediate convolution features, to create an image feature vector that can be used for retrieval via k-nearest neighbour search [1, 2, 7, 11, 25]. Most existing CNN-based image search approaches use a single representation of the entire image; we call this a *global* representation. Global representations have many advantages: they are fast, suppress or generalize unnecessary detail, and describe the main elements of an image. However, they also lose detail, which can be a problem for engineering diagram search.

Engineering diagram search has specific characteristics. Often, the query diagram is expected to match only a small region of an engineering diagram, for example, to find every diagram that uses a specific part. Thus, it is a more *local* matching problem. Engineering diagrams differ from most pictures. Typical engineering diagrams are gray-scale line drawings that have little texture. They have a visual syntax and contain visual metadata elements, for example, locator elements that place the diagram within a larger context; callouts that provide other perspectives; bubbles that provide additional detail; and lines that illustrate relationships among selected parts. It is an open question whether existing image retrieval algorithms based on global, general visual features are effective for engineering diagram search. This paper investigates several global and local approaches for engineering diagram search.

One of our new approaches is an unsupervised ranker called **DI**agram **S**earc**H** with Local Histogram Pooling Matching Network (DISH-HP). DISH-HP models the relevance of a diagram to a query based on the similarity of local regions, using histogram pooling to combine evidence [8]. It first extracts local convolution features from the middle layers of a pretrained network, such as VGG-16

*The first two authors contributed equally.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313500

[26]. Each local feature represents a small region in the original diagram. DISH-HP calculates the similarity scores between every pair of query and document regions. It then groups the region pairs into bins according to their similarity scores. Two diagrams are considered similar if many of the region pairs fall into the high similarity bins. The histogram-based method is efficient so that DISH-HP can scan over all diagrams in a collection to find the most similar candidates.

The second of our new approaches is called Diagram Search with Local Convolutional Matching Network (DISH-Conv). It was inspired by Conv-KNRM, a state-of-the-art neural text ranking model, that uses convolutional neural networks to match text n-grams [3]. The new DISH-Conv model employs spatial convolutional neural networks to build feature vectors for larger regions from small regions, generating multi-scale region representations. A cross-scale matching layer calculates the similarity between regions of different scales, enabling a smaller region in the query to match to a larger region in the candidate diagram. A *gating* network is proposed that automatically detects unimportant regions of a diagram, reducing noisy signals from low-density regions and visual metadata such as callouts. DISH-Conv can be trained end-to-end with relevance signals, so that the convolution, matching, and gating are optimized towards the specific dataset and task.

DISH-HP and DISH-Conv can be used in a pipelined re-ranking architecture. DISH-HP is the first stage that quickly ranks images. The top N candidates are passed to DISH-Conv to refine the ranking. N is limited to a few hundred or a few thousand, so that the second stage is able to use a more accurate ranking model that would be too computationally expensive for the first stage brute-force search.

Evaluation was done on a new diagram search dataset developed by crawling furniture assembly manuals from the Ikea™ website. Queries were generated automatically by extracting query-like regions from diagrams and transforming them in various ways, for example, by changing position, scale, and/or rotation. Experiments on large-scale auto-generated relevance labels show that DISH-HP is more accurate than several unsupervised rankers that use global image representations. Using DISH-Conv to re-rank DISH-HP results improves accuracy substantially. Larger improvements over baseline models are shown on more difficult queries that have rotation and scale changes. Evaluation with a small set of manual relevance assessments further confirms these results. Analysis reveals that the advantage of DISH-Conv is a combination of its abilities to transform rotation changes through the convolutions, match at multiple scale level through the cross-matching layer, and suppress unimportant regions through the gating network.

The rest of this paper is organized as follows. Sections 2 gives a brief overview of related work. Section 3 and 4 introduce the proposed models DISH-HP and DISH-Conv. Section 5 discusses the Ikea diagram search dataset and our experimental methodology. Section 6 reports and discusses experimental results. Section 7 summarizes and concludes the paper.

2 RELATED WORK

Image Search. Until recently, the vast majority of image search approaches were variants of the bag-of-words model [24] based on hand-crafted features, typically SIFT [13]. In the recent decade, out-of-the-box features from pre-trained Convolutional Neural Networks (CNNs) were shown to give state-of-the-art results in many computer vision tasks, including image search. The early CNN-based image retrieval work [2, 7] takes the activation of fully connected layers as *global* descriptors. Later works shift the focus to convolution layers because lower layers preserve more detailed visual information and are more suitable for the search task. The convolution features are extracted from sliding windows at different image regions, hence are usually considered as *local* features. Most prior work used max/sum pooling over the convolution features to produce the single image representation Babenko and Lempitsky [1], Jimenez et al. [11], Kalantidis et al. [12], Tolias et al. [25]. Although local convolution features are used, they are pooled into a single representation, hence the image matching is still performed *globally*. It may raise problems for diagram search where tasks include matching *part* of one diagram to *part* of another diagram.

Patch-based approaches aim to represent a image with multiple image patches to address the problem when the item of interest has appeared on a different scale at an arbitrary position in the relevant image [7, 21]. Patch-based methods are computationally expensive because each individual patch is resized to the same size as the original image and fed forward into a deep convolutional neural network to extract features.

Most CNN-based image retrieval work focused on real-world photographs [1, 2, 11, 25]. They were tested against benchmark datasets consisting of landmark photographs, such as the Oxford Buildings dataset [18], the Paris Dataset [19], and the INRIA Holidays dataset [10]. The effectiveness of CNN based features on engineering diagrams remains to be explored,

Engineering Diagram Search. A lot of work on image-based engineering diagram search were developed before the bag-of-word model [24] was widely applied in image search. These systems often take several steps to first identify lines and shapes and then to calculate similarities between two diagrams. Task specific features and matching criterias were designed, leading to complex systems and extensive feature engineering [6, 15, 20]. A more recent work by Eitz et al. [5] published in 2011 demonstrates the effectiveness of bag-of-words models with SIFT-style features on line-drawings and sketch-based image retrieval. To the best of our knowledge, little exploration has been made on using deep convolution networks for engineering diagram search.

Neural Ranking Models for Text Search. Large progress has been made recently on using deep neural networks for text search. Recent neural ranking models can be categorized as *representation-based* and *interaction-based* [8]. Representation-based models use global representations of the query and document and match in the representation space [9, 22]. They are similar to most image search approaches based on global image representations. Interaction-based models use local interactions between the query and document words and neural networks that learn matching patterns [3, 8, 27]. The DRMM model [8] uses histogram pooling to summarize word-word interactions. It builds a word-word similarity matrix from word embeddings, groups the word-word soft matches into a histogram based on the similarity score, and learns to combine the bins. It allows the model to take into consideration both strong word matches and weaker word matches. The paper also proposed two methods to suppress unimportant query words; one method is

based on term frequency signals and the other is based on the word embedding. The K-NRM model [27] integrates DRMM with the ability to learn customized embeddings. It uses kernel-pooling, an approximation of the histogram with RBF kernels. Kernel-pooling shares the advantage of histogram pooling while also being differentiable so that word embeddings can be learned end-to-end and align with the search task. The Conv-KNRM model [3] extends K-NRM by using convolutional neural networks to build n-gram representations. It uses different sizes of convolutions to build unigram, bigram and trigram embeddings from word embeddings. A *cross-matching* layer compares across n-grams of various lengths in the unified embedding space. It allows the query bi-gram "deep learning" to be matched to the document tri-gram "convolutional neural network". The n-gram matching signals improve ranking accuracy over the unigram models.

3 DISH-HP: LOCAL HISTOGRAM POOLING MATCHING NETWORK

Interaction-based text retrieval models, which match a query to localized regions of a document, are more accurate than representation-based models because i) text queries are not expected to match all, or even most, of a text document, ii) local interactions provide multiple and detailed match signals, and iii) irrelevant regions have no effect. Interaction-based models could be beneficial to diagram search by modeling the interactions of a local region from the query diagram and a local region from the document diagram.

The section describes a diagram search model based on *Local Histogram Pooling Matching Network* (**DISH-HP**). DISH-HP is inspired by the histogram pooling technique proposed by Guo et al. [8]. It models the relevance between two diagrams based on how similar their local regions are.

The input to DISH-HP is a list of local convolution feature vectors extracted from the query diagram Q and the document diagram D. In our experiments, features are extracted from the last convolution layer of the VGG-16 network [26]. This layer divides the image into $R = W \times H = 14 \times 14 = 196$ grids, each corresponding to a 16 pixel $\times$ 16 pixel rectangular region in the diagram. For each of the R regions, a $d = 512$ dimension feature vector is generated. Hence, Q and D are represented by matrices of size $[R, d] = [196, 512]$.

Given the feature matrices of Q and D, DISH-HP collects the similarity scores between every pair of query region and document region. It then groups the region pairs into bins based on the similarity scores and builds a histogram. The histogram is pooled to produce a relevance score, where two diagrams are considered similar if many of the region pairs fall into the high similarity bins. More specifically, DISH-HP builds a similarity matrix M of size $[R, R]$ where the cell $M_{i,j}$ is the cosine similarity between the i-th query feature and the j-th document feature. A histogram is built for each row of the similarity matrix by assigning the similarity values into 10 bins: $B_1 = [0.8, 1]$, $B_2 = [0.6, 0.8)$, ..., $B_{10} = [-1, -0.8)]$. DISH-HP then counts the points in each bin. The counts indicate how many document regions are very similar to the query region (e.g., bin $[0.8, 1]$), how many are somewhat similar (e.g., bin $[0.6, 0.8)$), and how many are dissimilar. The number of points in the first K high-similarity bins $\{B_1, ..., B_K\}$ are taken as the score for the document diagram to match the i-th query region:

$$s(Q_i, D) = \sum_{j=1...K} |B_j|. \quad (1)$$

The final relevance score is the log-sum of counts from every query region:

$$s(Q, D) = \sum_{i=1...R} \log s(Q_i, D). \quad (2)$$

Log-sum squashes very high values into smaller values. It helps to avoid the matching being dominated by a few query regions that are repeatedly matched in the document diagram.

One issue with the above method is that the relevance score is often overwhelmed by noisy match signals from low-density regions. Engineering diagrams are often very sparse; blank or near-blank regions will dominate the high-similarity bins. To address this issue, features from low-density regions are filtered out before computing the similarity matrix. A feature vector is removed if its L2-norm is smaller than a threshold T, with the assumption that lower L2-norm indicates lower density of that region. The L2-norm threshold T is a hyper-parameter to be tuned.

In summary, DISH-HP explicitly models the similarity between local regions from the query and the document. DISH-HP is expected to retrieve more-relevant documents than global matching methods because the local, region-level matching provides more evidence than matching at the global image level. In terms of efficiency, DISH-HP is slightly slower than global methods. DISH-HP needs to compute R^2 cosine similarity values for a document with R local regions; global methods only need to compute a single global cosine similarity. The bottleneck for both global and local method lies in extraction features from the query using a deep convolutions neural network.

4 DISH-CONV: LOCAL CONVOLUTIONAL MATCHING NETWORK

DISH-HP is designed to be simple and efficient in order to be used for initial ranking. However, the simple architecture has several drawbacks that may limit its search effectiveness. First, DISH-HP uses several heuristics, such as which bins are used and how low-density areas are detected. These heuristics may not be optimal. More importantly, the local features are only for a fixed size of image regions (e.g. 16 pixel $\times$ 16 pixel), while in many cases the same part may be drawn at different scales in different diagrams.

This section proposes DISH-Conv, a supervised neural diagram search model based on *Local Convolutional Matching Network* (**DISH-Conv**). The drawbacks of DISH-HP are addressed by using machine-learned weights to replace heuristics and using better network architectures to generate richer features.

DISH-Conv is inspired by Conv-KNRM, a state-of-the-art neural ranking model proposed by Dai et al. [3]. It adapts the use of convolutions and cross-matching from text search to diagram search. The architecture of DISH-Conv is shown in Figure 1. Given local convolution features from a query and a document diagram, it builds multi-scale region representations through spatial convolutions, mutes unimportant region features through a gating network,

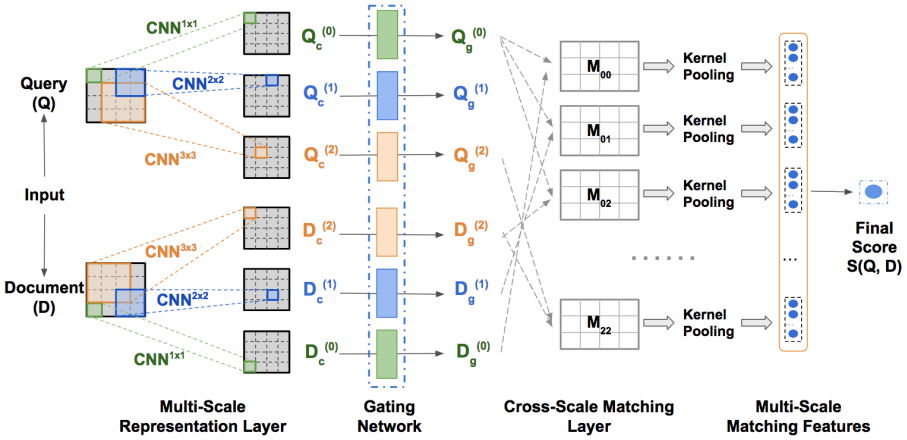


Figure 1: The DISH-Conv architecture. Query and document diagrams are input. The multi-scale representation layer uses spatial convolutions of different kernel sizes to generate features at multiple scales. The gating network computes the importance of each feature, and suppresses features from unimportant areas. The cross-scale matching layer compares the similarity between query regions and document diagram regions of different sizes, and forms the similarity matrices for different scales. The kernel pooling aggregates the similarity matrices into cross-scale matching features. A *learning-to-rank* layer combines all features and generate a final score.

and generates cross-scale local match signals through the cross-matching layer. It then ranks diagrams with the cross-scale match signals using kernel-pooling and learning-to-rank.

4.1 Generating Multi-Scale Representations

DISH-Conv starts with the same process as DISH-HP. Local feature vectors are extracted from the query diagram Q and the document diagram D. The feature vectors are organized in a 3D array of size $[H, W, d]$: $H \times W$ local regions each with a d dimension feature vector.

Intuitively, features extracted from regions of different sizes could improve the robustness on diagrams of different scales. Following this intuition, DISH-Conv uses 2D spatial convolutions to generate features for larger regions. A 2×2 kernel builds representations from 4 adjacent regions, generating a new feature vector whose receptive fields is 4 times larger than the original feature vector. Similar, 3×3 kernels generate representations for even larger regions. We also employ a 1×1 convolution. It has the same receptive fields as the input features, but can transform the features to better fit the ranking task.

The convolutions form a pyramid of region representations at multiple scale levels. In addition to scale, the convolution can also learn other types of transformation. It is able to model rotation because rotation changes are essentially a linear transformation of the pixel coordinates. It can also learn transformation in texture and structure, optimizing the input features toward the specific task and dataset.

4.2 Gating Network

During the development of DISH-HP, we identified that the model should ignore unimportant regions of a diagram. DISH-HP employed a heuristic that compares the L2-norm of local feature vectors to a fixed threshold. However, it is not guaranteed that the features of

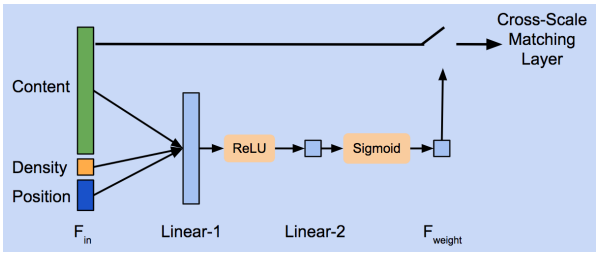


Figure 2: The gating network architecture.

blank areas are always close to zero; and, some high-density areas are also unimportant, for example, callouts and bubbles.

A gating network was developed to automatically learn the importance of local features and to down-weight unimportant features during the matching process (Figure 2). The gating network predicts the importance of a region based on three types of signals: content, position, and density. The content is represented by the region emebedding from the multi-scale representation layer (C features, C is the number of convolution channels). The position features are the coordinates of the center of the perceptive field (2 features). The density feature is the L2-norm as used in DISH-HP (1 feature). These signals are concatenated, generating a $C + 3$ dimension input vector F_{in}, and passed through a 2-layer Perceptron [16]:

$$\vec{h} = ReLU(W_1 \vec{F}_{in} + B_1) \tag{3}$$

$$F_{weight} = \sigma(W_2 \vec{h} + B_2), \tag{4}$$

where W_1, B_1, W_2, and B_2 are the trainable parameters of the Perceptron. σ is the Sigmoid function that converts the free-range network output into the range of $[0, 1]$. The output, $F_{weight} \in [0, 1]$, is multiplied to the original feature vector. F_{weight} close to 1 means that the local feature vector is important and is allowed to pass

through the gate; F_{weight} close to 0 means that the feature vector is unimportant and its effect on the ranker should be muted.

The gating network is trained jointly with the other parts of DISH-Conv, so that it automatically learns to recognize regions that are important for the search task. The importance of a local feature vector is based on not only the density, but more importantly, its content. It allows dense but meaningless regions, such as bubbles, to be detected.

4.3 Ranking with Cross-Scale Match Signals

The multi-scale local region representations, with unimportant regions suppressed by the gating network, are sent to the next cross-scale matching layer. This layer calculates the similarity between each pair of query-document regions. The regions can be taken from different scales. For example, a 16×16 pixel area in the query diagram is compared to 16×16, 32×32, and 48×48 areas in the candidate diagrams (from 1×1, 2×2, and 3×3 convolution). In total, the match is performed at 9 different scales.

For each of the 9 scale combination, the match signals are aggregated through *kernel-pooling* [27]. Kernel-pooling is a soft version of the histogram pooling used in DISH-HP that approximates the histogram function with RBF kernels. It makes the histogram pooling differentiable so that gradients can be back-propagated to the previous convolution layer. The resulting kernel features are similar to the histogram features in DISH-HP. The kernel features from all 9 scale combinations are then combined by a learning-to-rank layer to generate the final score. The learning-to-rank layer assigns different importance to local match signals with different strength (e.g. 'strongly similar', 'weakly similar', 'dissimilar') and at different scales (e.g. 'small query region to small document region' or 'large query region to medium document region'). The output is the predicted relevance score between the query and the document $s(Q, D)$.

DISH-Conv is trained to minimize a standard pairwise loss function:

$$L(Q, P, N) = max(0, \alpha - s(Q, P) + s(Q, N)). \quad (5)$$

The loss function keeps relevant diagrams P closer than any negative diagrams N for each query Q, with a margin α. A query's relevant documents are taken as positive examples. Negative examples are sampled uniformly from the dataset.

4.4 Summary

DISH-Conv is motivated by the interaction-based neural ranking models designed originally to model word-word interaction in text search [3, 8, 27]. DISH-Conv treats local regions as words and models region-region interactions between two diagrams. Its architecture was inspired by the convolution and cross-matching technique proposed by Dai et al. [3]. However, our intuition of using the convolutions and cross-matching is very different from Dai et al. [3]. Our focus is on *scales* and *rotations* while the goal of the original work was to match phrases. We employ spatial convolutions to learn scale-invariant and rotation-invariant transformations, and use cross-matching to generate match signals across multiple scales. A novel gating network is proposed to learn local region importance from position, density, and more importantly, content. Collectively, these elements address several issues identified in DISH-HP, and should enable DISH-Conv to produce more accurate rankings.

5 EXPERIMENTAL METHODOLOGY

This section introduces a new dataset used for experimental evaluation, and the baseline algorithms that were used for comparison.

5.1 Datasets

Although our work is motivated by common tasks in large engineering organizations, there are no standard datasets to use for evaluation because engineering diagrams tend to be proprietary data. A new dataset based on Ikea™ furniture assembly manuals was developed to support experimental evaluation. Although we are unable to redistribute the dataset, others can create a substantially similar dataset based on the descriptions below and additional information posted on our website[1].

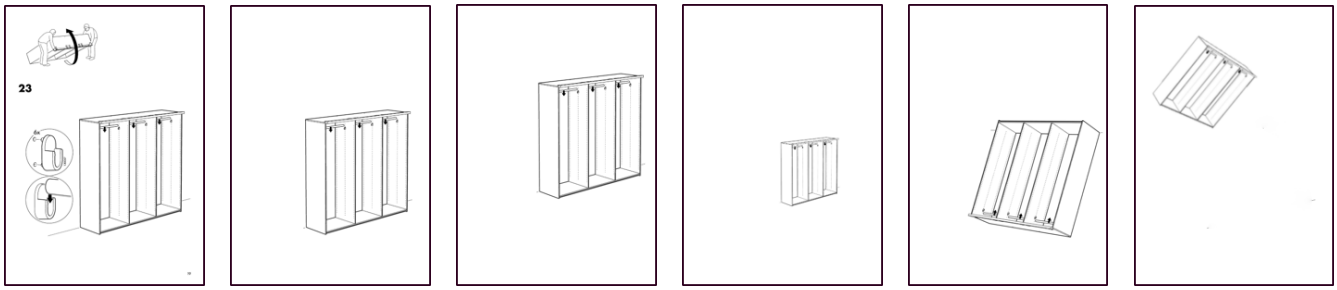
Images: We downloaded product assembly manuals from Ikea™ [2]. Each manual was split into page images. Images that were entirely blank, primarily text, or duplicates of other images were discarded. The result was a corpus of 13,464 furniture assembly diagrams (the Ikea dataset). Each diagram was a black-and-white pdf document. Resolutions of the diagrams vary, with most diagrams around 1000*600 or 600*1000. See Figure 3(a) for an example.

Queries and Relevance Assessments: An automated query generator was developed to create a large set of queries. It tries to locate parts in a diagram that would make reasonable queries. It first selects the zone with relatively high black pixel density as an initial query region, and then gradually expands it by merging the current region with adjacent zones. This process is repeated until the black pixel density of the query region is lower than a threshold. Next, the query region is validated to make sure that it is not too small, too large, entirely black or filled with text. Finally, a query is defined by inserting the validated region into a white background, keeping the image size and the position of the region exactly the same as the original query.

In typical use, queries would not have the same position, scale, and rotation as a region of a relevant diagram. The initial set of queries is named *psr* queries, with lower case letters indicating that the position (p), scale (s), and rotation (r) of the query image are unchanged. Four additional sets of queries were generated to have specific characteristics and increasing difficulty for use in diagnosing weaknesses in retrieval algorithms. The names of these query sets use capitalized letters to mark relative changes. *Psr* queries change the position of the query image, but keep scale and rotation invariant. *pSr* queries change the scale of the query image within the range $[0.5, 2]$. *psR* queries change the rotation 0 to 360 degrees. *PSR* queries change position, scale, and rotation simultaneously. Figures 3b-3f show examples of all five types of queries.

When generating queries, a position, scale, or rotation change is generated randomly within an appropriate range. The transformed query is required to fit entirely within the image boundary so that it does not lose information during the transformation. For position and scale changes, simple rules ensure that the transformed query

[1] http://boston.lti.cs.cmu.edu/appendices/TheWebConf2019-Zhuyun-Dai/
[2] https://www.ikea.com/ms/en_US/customer_service/assembly_instructions.html



(a) A relevant document. (b) Query with no change (psr). (c) Query with changed position (Psr). (d) Query with changed scale (pSr). (e) Query with changed rotation (psR). (f) Query with changed position, scale, and rotation (PSR).

Figure 3: Five types of automatically-generated queries for the Ikea dataset.

Table 1: The Ikea Dataset Characteristics

	Description	Number
Corpus	Diagrams from Ikea manuals	13,464
psr	No change	3,649
Psr	Random position change	3,649
pSr	Random scale change (0.5 to 2)	3,649
psR	Random rotation change (0 to 360)	2,988
PSR	Position, scale, & rotation change	3,005

lies within the image boundary. For rotation changes, it is more complex to predict the part boundaries. Therefore, if the rotated query was not entirely within the image boundaries, the change was discarded, and a new random rotation angle generated. If the rotation change failed five times, the image was discarded from the set of queries. Thus, the *psr*, *Psr*, and *pSr* query sets all have the same number of queries (3,649), but the *psR* and *PSR* query sets are somewhat smaller (about 3,000). Table 1 shows the characteristics of the Ikea collection and queries.

Metrics: In the automated query generation process, the source diagram is treated as the query's (only) relevant document. A good retrieval system should rank that document highly. Mean Reciprocal Rank (MRR) is a common metric when there is one relevant document. We also want to know the number of queries for which the relevant document appears at rank 1 and in the top 10 results, thus we also use Recall at cutoff at 1 and 10 (R@1 and R@10). A small set of *manual* labeled queries with multiple relevant documents were studied at Section 6.5.

5.2 Proposed Models and Baselines

Proposed models include two local matching networks, DISH-HP and DISH-Conv. DISH-HP is the unsupervised ranking algorithm proposed in Section 3. DISH-Conv is the supervised local convolutional matching network proposed in Section 4. The proposed models are compared to several baselines, including a bag-of-words image retrieval model based on SIFT features, a neural image retrieval model based on CNN features, and an encoder-decoder network designed specifically for line drawings.

LIRE [14] is an open source image retrieval system based on Lucene. It uses bag-of-words image retrieval models with SIFT features. The indexing and retrieval are unsupervised. We used the open source software [3] to index the images in our corpus and perform the retrieval.

FG-SBIR [17] is a neural retrieval system designed for sketches and line drawings. It is based on an encoder-decoder architecture. The encoder is based on the VGG-16 network [23]; the decoder consists of a chain of upsampling blocks to reconstruct the sketch image from the VGG-16encoder's fully connected layer, helping the encoder to learn a richer representation. The output from the encoder is taken as the image descriptor and used for retrieval. We used the open source implementation [4]. FG-SBIR is a *global* matching approach as the entire image is encoded into a single embedding; local region features are not preserved.

CAM [11] is a state-of-the-art instance retrieval approach based on deep convolutional neural networks. It uses CNN features extracted from an off-the-shelf VGG convolution network trained on ImageNet [4]. For each pre-defined class in ImageNet, CAM generate a class-focused image vector by weighted-summing the CNN features based on their activation to this class. The vectors from all classes are then normalized, whitened, and sum-pooled into a single image vector. Image retrieval is based on the Euclidean distance between the final image vectors. We took the pre-trained CAM model released by the authors [5]. We did not re-train CAM on the Ikea dataset due to lack of class labels. CAM is also a *global* matching approach as local region features are not preserved

The above three baselines used off-the-shelf software. We do not have control over how they process data and extract features. To gain better understanding of the proposed models, we created a set of DISH variants as additional controlled baselines. They differ from DISH-HP and DISH-Conv only in the architectures of the matching networks. The implementation framework, data processing, and feature extraction are controlled to be exactly the same.

DISH-SPoC and DISH-MPoC are two global variants of DISH-HP. They were created to test the effectiveness of local matching against global matching. DISH-SPoC is based on the SPoC descriptor proposed by Babenko and Lempitsky [1], but uses the DISH framework. It takes the same CNN features as DISH-HP, and applies the same filtering on white areas. The filtered CNN features are then aggregated through sum-pooling to generate a single, global image feature vector. The similarity between the query and a document

[3] http://www.lire-project.net/
[4] https://github.com/zoeyangdw/FG-SBIR
[5] https://github.com/imatge-upc/retrieval-2017-cam

diagram is calculated from the cosine similarity between the corresponding vectors. DISH-MPoC is a similar method that replaces the sum-pooling with max-pooling. DISH-SPoC and DISH-MPoC are unsupervised methods.

DISH-Conv-N is DISH-Conv without the spatial convolution; the letter N stands for 'No convolution'. In DISH-Conv-N, the input local CNN features are directly used to calculate region similarities. The model lacks the multi-scale region representation and cross-scale matching in DISH-Conv. DISH-Conv-N can also be viewed as a supervised version of DISH-HP with the hard histogram replaced by soft RBF kernels. DISH-Conv-N is a supervised model.

DISH-Conv-S is a simplified version of DISH-Conv that replaces the spatial convolution with spatial pooling when generating multi-scale region representations. The spatial pooling takes the average of features from smaller regions to generate features for larger regions. Pooling does not have additional parameters, so this model is faster and easier to tune compared to DISH-Conv. DISH-Conv-S separates the effects of convolution from the effects of aggregating small regions into larger regions. It is also a supervised model

Initial Ranking and Re-ranking: We employed a two-stage search pipeline. In the first stage, an initial ranker takes the query and compares it against all diagrams in the corpus, and returns a list of similar diagrams as candidates. In the second stage, a different ranker re-ranks the top 2,000 candidates. DISH-HP, DISH-SPoC and DISH-MPoC are simple models designed for efficient ranking; they were tested on the initial ranking task. DISH-Conv, DISH-Conv-N, and DISH-Conv-S were designed to improve DISH-HP with more complex models; they were tested on the re-ranking task. LIRE, FG-SBIR, and CAM were tested in both stages, providing baselines for both tasks.

5.3 Model Configuration and Implementation

Feature extraction: Off-the-shelf baselines (LIRE, FG-SBIR, and CAM) used their own implementations for feature extraction. All DISH models took the features from an off-the-shelf VGG-16 network trained on ImageNet[6]. Following prior research [21], we extracted features from the last convolution layer. This layer divides the image into $R = W \times H = 14 \times 14 = 196$ grids, each corresponding to a 16 pixel $\times$ 16 pixel rectangular region in the diagram. For each of the 196 regions, a $d = 512$ dimension feature vector is generated. We did not fine-tune the VGG-16 network on engineering diagrams, as the focus of this work is on *matching* local features.

Model Training: Supervised models (DISH-Conv, DISH-Conv-N, and DISH-Conv-S) were trained and tested via 5 fold cross-validation. For each query we sample 10 negative examples. We used the Adam optimizer with batch size 16, learning rate 0.001, and a learning rate decay of 0.5 when validation loss does not decrease. The training stops after 30 epochs or when learning rate drops below $3.12e-5$. The models were implemented in PyTorch.

Hyper Parameters: For DISH-HP, the top $K = 2$ bins were used and low-density area filter threshold T was set as 30. The parameters were chosen through a simple parameter sweep. The kernel pooling layers in DISH-Conv, DISH-Conv-N, and DISH-Conv-S all followed the configurations suggested by prior research [27] and used 11 kernels. The first kernel is the exact match kernel where $\mu = 1, \sigma = 10^{-3}$, or bin $[1, 1]$. The other 10 kernels equally split the cosine range $[-1, 1]$: the μ of bin centers were: $\mu_1 = 0.9, \mu_2 = 0.7, ..., \mu_{10} = -0.9$. The σ of the soft match bins were set to be 0.1. The multi-scale presentation layer in DISH-Conv uses three types of spatial convolutions with different kernel sizes ($1 \times 1, 2 \times 2, 3 \times 3$), 128 output channels, and a stride of 1.

[6]https://download.pytorch.org/models/vgg16-397923af.pth

6 EXPERIMENTAL RESULTS

Two experiments on the Ikea dataset tested the accuracy of the different methods of retrieving diagrams in ranking and re-ranking configurations on large-scale auto-generated relevance labels. Two analyses investigate the behavior and effectiveness of the convolutions and the gating network. An evaluation on a small set of manually-judged queries were conducted to confirm the results from experiments with auto-generated relevance labels.

6.1 Initial Ranking Performance of DISH-HP

DISH-HP is simple, fast, and does not require training. It was tested on the first stage ranking task where the query is compared to all diagrams in the collection. As shown in Table 2, DISH-HP outperforms all other methods by large margins.

Baselines. Among the baselines, only LIRE uses SIFT features; the rest use deep CNN features. LIRE is the least accurate, demonstrating the effectiveness of deep CNN features on engineering diagrams (Table 2). FG-SBIR, which is designed for sketches and line drawings, has higher precision (MRR and Recall@1), but CAM retrieves more relevant diagrams (Recall@10). These results indicate that CNN feature extractors learned from photographs can also extract useful patterns from engineering diagrams.

DISH-SPoC and DISH-MPoC outperforms FB-SBIR and CAM. These four methods are all global matching methods based on CNN features. The main differences of DISH-SPoC and DISH-MPoC with FG-SBIR and CAM are i) they have low-density region filtering, and ii) they use simple sum/max to aggregate local CNN features while FG-SBIR and CAM learn feature aggregation from a dataset. Low-density region filtering is important because these regions bring in noisy signals. The simple pooling functions are task-neutral, while FG-SBIR and CAM are tuned for their training datasets. DISH-MPoC is the strongest baseline.

Local vs. Global. DISH-HP outperforms all other methods by large margins. Like DISH-MPoC, DISH-HP uses deep CNN features and filters blank regions. However, DISH-HP is a local matching method that leverages local region matching evidence. In global methods such as DISH-MPoC, matching is at the entire image level; no local, regional similarity signals are used. They squeeze the image into a 512 dimensional vector. The vector is effective at capturing high-level visual features, such as the shape of a part. However, it loses information at lower granularity levels. As a result, irrelevant diagrams are retrieved, for example, a wheel is retrieved for a round washer because they have similar shapes. Global match methods also missed documents that are partially matched to the query because the other parts in the diagram make the feature vector less similar to the query's feature vector. On the other hand, DISH-HP leverages local similarity signals that tell if the sub-parts between two diagrams are similar. It helps to filter out irrelevant documents and to retrieve partially matched documents.

Table 2: Ranking accuracy on several types of queries. The query is compared to all document diagrams in the Ikea datset.

	Invariant (psr)			Position (Psr)			Scale (pSr)			Rotation (psR)			ALL (PSR)		
Method	MRR	R@1	@10	MRR	R@1	R@10	MRR	R@1	R@10	MRR	R@1	R@10	MRR	R@1	R@10
LIRE	0.19	14%	29%	0.17	9%	19%	0.03	1%	3%	0.00	0	1%	0.00	0%	0%
CAM	0.38	6%	48%	0.31	6%	41%	0.23	5%	32%	0.08	2%	13%	0.04	2%	7%
FG-SBIR	0.44	23%	36%	0.41	19%	35%	0.39	19%	32%	0.10	5%	14%	0.06	3%	8%
DISH-SPoC	0.52	45%	63%	0.48	41%	59%	0.37	30%	48%	0.10	7%	16%	0.07	4%	11%
DISH-MPoC	0.66	58%	76%	0.57	50%	70%	0.44	37%	57%	0.12	9%	18%	0.08	5%	12%
DISH-HP	**0.68**	**59%**	**84%**	**0.65**	**56%**	**82%**	**0.54**	**44%**	**70%**	**0.15**	**11%**	**21%**	**0.10**	**7%**	**14%**

Table 3: Re-ranking accuracy on several types of queries. The top 2,000 documents retrieved from the Ikea dataset by DISH-HP were re-ranked by each method.

	Invariant (psr)			Position (Psr)			Scale (pSr)			Rotation (psR)			ALL (PSR)		
Method	MRR	R@1	R@10	MRR	R@1	R@10	MRR	R@1	R@10	MRR	R@1	R@10	MRR	R@1	R@10
DISH-HP	0.68	59%	84%	0.65	56%	82%	0.54	44%	70%	0.15	11%	21%	0.10	7%	14%
LIRE	0.55	23%	87%	0.66	27%	91%	0.13	2%	46%	0.01	0%	1%	0.01	0	0%
CAM	0.76	60%	88%	0.68	58%	83%	0.60	47%	71%	0.18	12%	22%	0.11	8%	15%
FG-SBIR	0.82	72%	89%	0.76	70%	84%	0.64	58%	72%	0.20	14%	23%	0.11	7%	14%
DISH-Conv-N	0.93	89%	99%	0.83	83%	98%	0.73	65%	85%	0.21	15%	30%	0.09	6%	14%
DISH-Conv-S	**0.94**	**89%**	**99%**	**0.90**	**84%**	**98%**	0.74	65%	86%	0.22	17%	31%	0.10	6%	16%
DISH-Conv	0.91	87%	97%	0.82	74%	94%	**0.78**	**70%**	**90%**	**0.38**	**28%**	**57%**	**0.17**	**10%**	**30%**

Sensitivity to position, scale and rotation changes. DISH-HP is robust to position changes. It had almost the same performance on the position changed queries (Psr) compared to the invariant queries (psr), whereas DISH-MPoC suffered a significant drop when the query is shifted. DISH-HP was relatively sensitive to scale changes: the Recall@10 decreased from 84% to 70% when the query scale was changed (pSr). Rotation changes were more difficult: the Recall@10 was only 21% on rotation changed queries (psR). Queries that simultaneously changed position, scale and rotation (PSR) were the most difficult.

In summary, the experimental results on the baselines show the effectiveness of deep CNN features on engineering diagrams compared to hand-crafted SIFT features. A comparison between models learned on ImageNet and models learned on line drawings indicate that off-the-shelf CNN feature extractors can achieve good Recall, although tuning them for line drawings may increase Precision. Experimental results on DISH-HP demonstrate the power of *local* matching. The local region match signals provide evidence to filter out irrelevant diagrams that share similar shapes with the query but differ in detail. It also enables retrieving relevant diagrams that partially match the query. DISH-HP largely outperforms commonly-used kNN-style global retrieval approaches, including state-of-the-art instance retrieval models. The Recall of DISH-HP was not perfect, especially on scale and rotation changes. But it may serve as good starting point for a retrieval pipeline and to generate training data for more advanced ranking models.

6.2 Re-ranking Performance of DISH-Conv

The second experiment investigated DISH-Conv and its variants. DISH-Conv uses learned weights, multi-scale matching, and convolutional features to improve ranking, especially for queries at different scales and rotations. Due to its greater computational cost, DISH-Conv was only tested as a second stage that re-ranks the top N results returned by a first-stage ranker such as DISH-HP. As a reference, we also tested the baseline methods on the re-ranking task.

As shown in Table 3, LIRE failed to improve the initial ranking of DISH-HP, but both CAM and FG-SBIR were able to improve it. DISH-Conv greatly improved ranking accuracy. In invariant (psr) and position changed queries, over 80% of the relevant documents were ranked first. On queries with scale (pSr) changes, DISH-Conv is able to find 90% of the relevant documents by rank 10, 28% more than DISH-HP. On queries with scale (pSr) changes, DISH-Conv almost tripled Recall@10 of DISH-HP. Together, they lead to higher performance on the most difficult *PSR* queries.

Sources of effectivenss. The two variants, DISH-Conv-N and DISH-Conv-S, help us to understand the sources of effectiveness.

DISH-Conv-N is DISH-Conv without convolution. It can be viewed as a supervised version of DISH-HP , where the hard histogram is replaced by soft RBF kernels, the weight of the kernels is learned, and the heuristic-based region filter is replaced by an end-to-end trained gating network. DISH-Conv-N outperforms DISH-HP, demonstrating that machine-learned kernel weights and machine-learned gating network are more effective than the heuristics.

DISH-Conv-S uses spatial pooling to generate multi-scale representations and matches two diagrams at multiple scale levels. Compared to DISH-Conv-N that only models single-scale match, the pooling-based multi-scale signals in DISH-Conv-S slightly improved the search accuracy for scale changed queries (pSr). Unexpectedly, the model has very high accuracy on position changes (Psr). When a query changes position, the perceptive field receives different content, and the convolution features changes. Pooling features from adjacent regions produces a descriptor for a larger region and compensates for the position change. The results shows that matching at multiple scales is not only effective for scale invariance, but also improves position invariance.

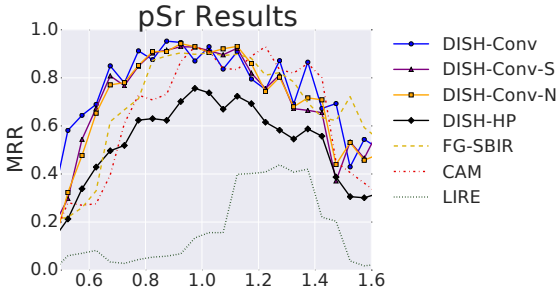
DISH-Conv is the most effective model on queries with scale and rotation changes (pSr, psR and PSR). In contrast to DISH-Conv-S which uses simple average pooling to generate features, DISH-Conv learns a matrix transformation of small-region features to larger-region features. The transformation is more powerful in handling complex changes in scale and rotation. Besides, the parameters are *learned* from the training data, so that they are tuned to fit the characteristics of the diagram dataset and the ranking task. The convolution, the cross-scale matching, together with machine-learned weights, make DISH-Conv more accurate and more robust.

6.3 Scale and Rotation Invariance Analysis

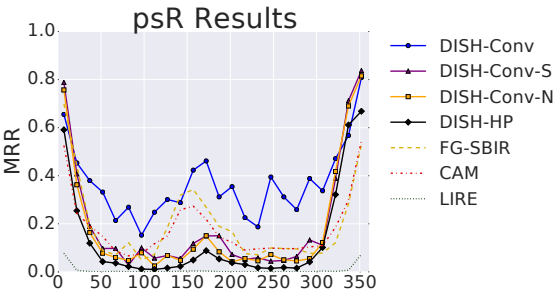
DISH-Conv uses convolutions and cross-matching to handle scale and rotation differences between queries and diagrams. An analysis investigates the effectiveness of these components (Figure 4).

Figure 4a shows each model's sensitivity to scale changes in pSr queries. Most models are best when the scale is unchanged (1.0). Ranking accuracy gradually decreases when the query is scaled down or up. DISH-Conv and DISH-Conv-S are best for all scale changes, due to their multi-scale representations and cross-scale matching. DISH-Conv is more robust to extreme scale changes, demonstrating that convolution is more powerful than pooling.

Figure 4b shows each model's sensitivity to rotation changes in psR queries. All models except DISH-Conv have a bow-like curve. Their MRR scores are high when there is no rotation, but drop to near 0 when the query is rotated ± 50 degrees. MRR improves somewhat near 180 degrees rotation because some queries are symmetric (e.g., rectangles); the symmetry effect is more obvious on FG-SBIR and CAM. Among all models, DISH-HP is the least able to handle rotation, and DISH-Conv is the most robust, although there is still a significant impact. Convolutions in DISH-Conv learn feature transformations that compensate somewhat for the rotation changes.



(a) Scale change



(b) Rotation change

Figure 4: Re-ranking accuracy (MRR) for different amounts of change in query scale and rotation.

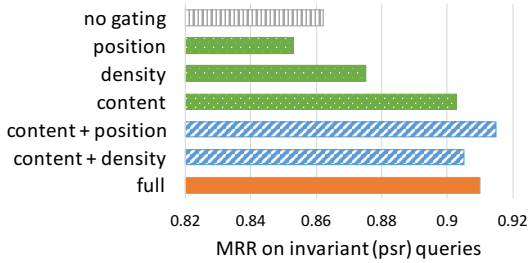


Figure 5: Re-ranking accuracy (MRR) of DISH-Conv on invarinat (psr) queries when gating uses different features.

6.4 Gating Network Analysis

DISH-Conv uses a gating network that estimates the importance of each region based on three types of features: the region's pixel density, estimated by the L2 norm of the feature vector; the region's position, represented by the coordinates of the region center; and the region's content, represented by its feature vector. An ablation study examined the effectiveness of the gating network. Figure 5 compares the accuracy of DISH-Conv with and without the gating network. It also compares gating networks using different types of features.

As shown in Figure 5, the MRR of DISH-Conv drops when the gating network is disabled. Without the gating network, the matches from meaningless areas dominated DISH-Conv's match signals, which produces poor results.

Comparison of gates using position, density, and/or content features indicates that content is the most powerful feature. The content gate learns the importance of a region from its CNN feature vectors, essentially capturing the *meaning* of the region. The position gate predicts a region's importance based on its location in the diagram, assuming that the center of the image is more likely to contain content of interest, which is not necessarily true. Using the position feature alone leads to worse results. The density gate used the L2 norm of the region feature vector, which is similar to the low-density region filter used by DISH-HP. The density gate is moderately better than having no gate. Combining all three types of features (full) provides further improvement, but the best combination omits the density feature, which is subsumed by the content feature.

In summary, identifying important regions and ignoring noisy regions is crucial for matching diagrams. A novel gating network was proposed to detect regions that are unimportant for ranking. Experimental results show that the gating network boosts the ranking accuracy of Local-Conv. The feature ablation study shows that learning to recognize unimportant contents delivers most of the gain, followed by the position of the region within the image.

6.5 Manual Relevance Assessments

The experiments above measure search accuracy using large-scale, auto-generated relevance labels, where a query's source diagram is

Table 4: Manual Relevance Assessments statistics

	Relevance Judgment Criteria	Avg Doc/Query
3	Same furniture	5.7
2	Likely to be the same furniture	1.3
1	Different, but similar assembly	5.8
0	Not related	63.6

Table 5: Re-ranking accuracy on 50 rotation queries (psR) with manual relevance assessments.

	Auto Labels			Manual Labels		
Method	MRR	R@1	R@10	MRR	R@1	R@10
DISH-HP	0.11	6%	18%	0.17	3%	10%
FG-SBIR	0.13	7%	21%	0.16	4%	10%
DISH-Conv-N	0.15	8%	28%	0.23	6%	14%
DISH-Conv-S	0.15	10%	22%	0.19	5%	11%
DISH-Conv	**0.31**	**22%**	**46%**	**0.37**	**8%**	**22%**

treated as its only relevant document. The last experiment measures accuracy using a small set of manual relevance judgments.

For each of 50 queries, the top 30 diagrams retrieved by Psr, pSr, and psR query variants were judged. 76 diagrams per query were assessed, on average. Three assessors judged the relevance of each diagram on a scale of $0-3$. Table 4 shows the relevance criteria and distribution. Diagrams with labels of 2 and 3 were considered relevant. On average, a query had 7 relevant diagrams.

These assessments were used to evaluate DISH methods and the best baseline FG-SBIR in the re-ranking setting. We report results for rotation (psR) queries because they differentiate the ranking methods the most. As shown in Table 5, the relative order of the ranking methods using manual labels is consistent with that on auto-generated labels: all DISH-Conv methods improved the initial ranking from DISH-HP; DISH-Conv methods were better than FG-SBIR; and DISH-Conv was most accurate. R@1 and R@10 values are lower for manual labels than for auto labels because there are 7× more relevant documents in this condition. MRR values are higher for manual labels, meaning that some top-ranked documents that are considered non-relevant by the auto labels are actually relevant.

In summary, the relative order of the ranking methods does not change when switching from auto labels to manual labels, validating the conclusions drawn from previous experiments.

7 CONCLUSION

Engineering diagram search has specific characteristics in that diagrams are sparse line drawings that contain visual metadata; the query is matched to a small region of a diagram; and the target part can appear at arbitrary position, scale, and rotation. This paper proposes local matching networks that model the interactions between local regions of the query diagram and the document diagrams, providing multiple and detailed match signals. Novel techniques were developed to address the specific issues in engineering diagrams.

This paper first presents an unsupervised model, DISH-HP. It leverages local region features extracted from the middle layer of a deep convolutional neural network. It calculates the similarity scores between query regions and document regions, and uses histogram pooling to combine evidence. The local match signals allow small differences to be recognized. Experimental results showed that matching local, region-level signals leads to superior performance over several retrieval algorithms based on global, image-level features.

The paper then presents a supervised model, DISH-Conv, to improve the robustness to scale and rotation changes. It is inspired by a state-of-the-art neural text ranking model, Conv-KNRM, which uses convolutional networks to generate n-gram embeddings from word embeddings. DISH-Conv adapts that idea and employs spatial convolutions to generate larger-scale representations from small-scale representations, enabling the query and the document diagrams to match at multiple scales. A novel gating network is proposed to improve the search accuracy by automatically detecting unimportant regions of an image. Experimental results show that DISH-Conv improves the ranking of DISH-HP substantially, especially when the target part appears at a different scale and/or is rotated to a different angle in the document diagram.

Our analysis reveals that the power of DISH-Conv is a combination of its ability to transform features through the convolutions, match at multiple scale levels, and suppress unimportant regions through the gating network. The spatial convolution in DISH-Conv generates multi-scale representations that are able to model changes in scale. The convolution is also able to learn a feature transformation that compensates for moderate changes of rotation. The gating network is able to capture the meaning of a region from its feature vector, and suppresses noisy match signals from unimportant regions. The convolution and the gating network are trained end-to-end so that they automatically learn how to transform features and assign importance for the specific task and dataset.

A pipelined system leverages the advantages of both models. DISH-HP efficiently scans the collection to identify a small set of candidate images. DISH-Conv re-ranks the candidates with less noise and richer match signals, delivering more accurate rankings. Together, DISH-HP and DISH-Conv are able to search a large corpus of engineering diagrams efficiently and effectively.

REFERENCES

[1] Artem Babenko and Victor S. Lempitsky. 2015. Aggregating Local Deep Features for Image Retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*. 1269–1277.

[2] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. 2014. Neural Codes for Image Retrieval. In *Proceedings of the 13th European Conference on Computer Vision*. 584–599.

[3] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 126–134.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 248–255.

[5] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. 2011. Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors. *IEEE Transactions on Visualization and Computer Graphics* (2011), 1624–1636.

[6] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. 2005. Content-based retrieval of technical drawings. *International Journal of Computer Applications in Technology* (2005), 86–100.

[7] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. 2014. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. In *Proceedings of the 13th European Conference on Computer Vision*. 392–407.

[8] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. 55–64.

[9] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and*

Knowledge Management. 2333–2338.

[10] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Proceedings of the 10th European Conference on Computer Vision*. 304–317.

[11] Albert Jimenez, Jose M. Alvarez, and Xavier Giró i Nieto. 2017. Class Weighted Convolutional Features for Visual Instance Search. In *Proceedings of the British Machine Vision Conference 2017*.

[12] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *Proceedings of the 15th European Conference on Computer Vision*. 685–701.

[13] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the 5th IEEE International Conference on Computer Vision*. 1150–1157.

[14] Mathias Lux and Oge Marques. 2013. *Visual Information Retrieval Using Java and LIRE*. Morgan & Claypool Publishers.

[15] Stefan Müller and Gerhard Rigoll. 1999. Engineering Drawing Database Retrieval Using Statistical Pattern Spotting Techniques. In *Proceedings of the 3rd International Workshop on Graphics Recognition Recent Advances,*. 246–255.

[16] Nils Nilsson. 1965. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. McGraw-Hill.

[17] Kaiyue Pang, Yi-Zhe Song, Tony Xiang, and Timothy M. Hospedales. 2017. Cross-domain Generative Learning for Fine-Grained Sketch-Based Image Retrieval. In *Proceedings of the British Machine Vision Conference*.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2007. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2008. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *Proceedings of the 14th IEEE Conference on Computer Vision and Pattern Recognition*.

[20] Jiantao Pu and Karthik Ramani. 2006. On visual similarity based 2D drawing retrieval. *Computer-Aided Design* 38, 3 (2006), 249–259.

[21] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 806–813.

[22] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 101–110.

[23] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[24] Josef Sivic and Andrew Zisserman. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision*. 1470–1477.

[25] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv Preprint arXiv:1511.05879* (2015).

[26] Visual Geometry Group, Department of Engineering Science, University of Oxford. 2018. Very Deep Convolutional Networks for Large-Scale Visual Recognition. http://www.robots.ox.ac.uk/~vgg/research/very_deep/, Last accessed on 2018-10-15.

[27] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 55–64.