Structured Set Matching Networks for One-Shot Part Labeling

Jonghyun Choi[⋆*] Jayant Krishnamurthy[⋆†] Aniruddha Kembhavi[*] Ali Farhadi[*‡]

Allen Institute for Artificial Intelligence[*] Semantic Machines[†] University of Washington[‡]

jonghyunc@allenai.org jayant@semanticmachines.com anik@allenai.org ali@cs.uw.edu

Abstract

Diagrams often depict complex phenomena and serve as a good test bed for visual and textual reasoning. However, understanding diagrams using natural image understanding approaches requires large training datasets of diagrams, which are very hard to obtain. Instead, this can be addressed as a matching problem either between labeled diagrams, images or both. This problem is very challenging since the absence of significant color and texture renders local cues ambiguous and requires global reasoning. We consider the problem of one-shot part labeling: labeling multiple parts of an object in a target image given only a single source image of that category. For this set-to-set matching problem, we introduce the Structured Set Matching Network (SSMN), a structured prediction model that incorporates convolutional neural networks. The SSMN is trained using global normalization to maximize local match scores between corresponding elements and a global consistency score among all matched elements, while also enforcing a matching constraint between the two sets. The SSMN significantly outperforms several strong baselines on three label transfer scenarios: diagram-to-diagram, evaluated on a new diagram dataset of over 200 categories; image-to-image, evaluated on a dataset built on top of the Pascal Part Dataset; and image-to-diagram, evaluated on transferring labels across these datasets.

1. Introduction

A considerable portion of visual data consists of illustrations including diagrams, maps, sketches, paintings and infographics, which afford unique challenges from a computer vision perspective. While computer vision research has largely focused on understanding natural images, there has been a recent renewal of interest in understanding visual illustrations [24, 31, 51, 47, 52, 33, 55, 28]. Science and math diagrams are a particularly interesting subset of visual illustrations, because they often depict complex phenomena grounded in well defined curricula, and serve as a good test bed for visual and textual reasoning [24, 35, 36, 26, 18].

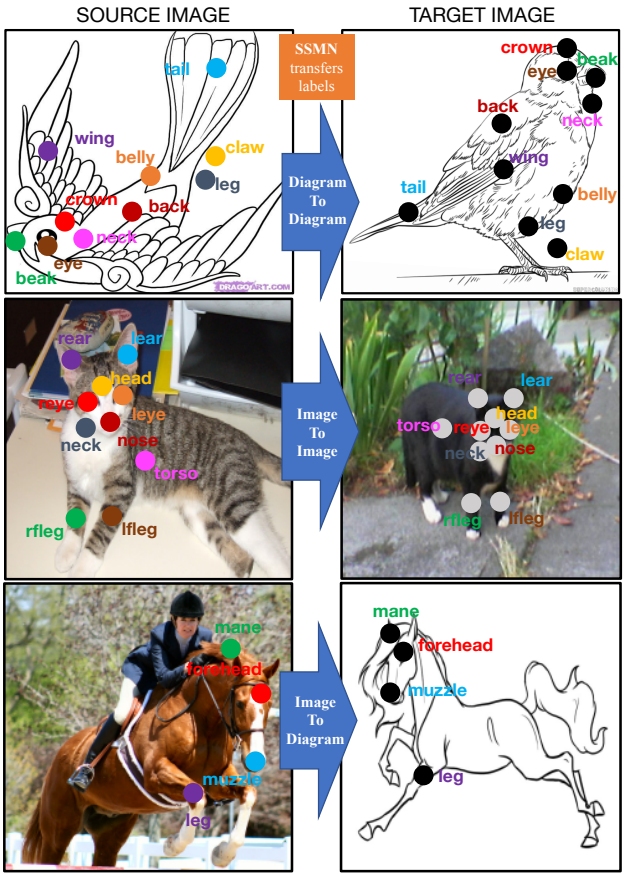


Figure 1. Matching results by our SSMN model. Given source images annotated by points with text labels, our model transfers labels to the points in the target images. Colors indicate each label. Black (Gray in row 2) dots indicate unlabeled points. SSMN is able to correctly label the target points in spite of significant geometric transformations and appearance differences between object parts in the source and target images of categories unobserved in training.

Understanding diagrams using natural image understanding approaches requires training models for diagram categories, object categories, part categories, etc. which requires large training corpora that are particularly hard to obtain for diagrams. Instead, this can be addressed by trans-

⋆ indicates equal contribution. Majority of the work has been done while JK is in AI2.

ferring labels from smaller labeled datasets of diagrams (within-domain) as well as from labeled datasets of natural images (cross-domain). Label transfer has previously shown impressive results in a within-domain natural image setting [29]. It is interesting to note that young children are able to correctly identify diagrams of objects and their parts, having seen just a few diagrammatic and natural image examples in story books and textbooks.

The task of label transfer is quite challenging, especially in diagrams. First, it requires a fine grained analysis of a diagram, but the absence of significant color or textural information renders local appearance cues inherently ambiguous. Second, overcoming these local ambiguities requires reasoning about the entire structure of the diagram, which is challenging. Finally, large datasets of object diagrams with fine grained part annotations, spanning the entire set of objects we are interested in, are expensive to acquire. Motivated by these challenges, we present the One-Shot Part Labeling task: labeling object parts in a diagram having seen only one labeled image from that category.

One-Shot Part Labeling is the task of matching elements of two sets: the fully-labeled parts of a source image and the unlabeled parts of a target image. Although previous work has considered matching a single target to a set of sources [25, 46], there is little prior work on set-to-set matching, which poses additional challenges as the model must predict a one-to-one matching. For this setting, we propose the Structure Set Matching Network (SSMN), a model that leverages the matching structure to improve accuracy. Our key observation is that a matching implies a transformation between the source and target objects and not all transformations are equally likely. For example, in Figure 1 (top), the matching would be highly implausible if we swapped the labels of "wing" and "tail," as this would imply a strange deformation of the depicted bird. However, transformations such as rotations and perspective shifts are common. The SSMN *learns* which transformations are likely and uses this information to improve its predictions.

The Structured Set Matching Network (SSMN) is an end-to-end learning model for matching the elements in two sets. The model combines convolutional neural networks (CNNs) into a structured prediction model. The CNNs extract local appearance features of parts from the source and target images. The structured prediction model maximises local matching scores (derived from the CNNs) between corresponding elements along with a global consistency score amongst all matched elements that represents whether the source-to-target transformation is reasonable. Crucially, the model is trained with global normalization to reduce errors from *label bias* [27] – roughly, model scores for points later in a sequence of predictions matter less – which we show is guaranteed to occur for RNNs and other locally-normalized models in set-to-set matching (Sec.4).

Off-the-shelf CNNs perform poorly on extracting features from diagrams [24, 52], owing to the fact that diagrams are very sparse and have little to no texture. Our key insight to overcoming this is to convert diagrams to distance transform images. The distance transform introduces *meaningful* textures into the images that capture the location and orientation of nearby edges. Our experiments show that this introduced texture improves performance and enables the use of model architectures built for natural images.

We compile three datasets: (1) a new diagram dataset named Diagram Part Labeling (DiPART), which consists of 4,921 diagram images across 200 objects categories, each annotated with 10 parts. (2) a natural image part labeling dataset named Pascal Part Matching (PPM) built on top of the popular Pascal Part dataset [6]. (3) a combination of the above two datasets (Cross-DiPART-PPM) to evaluate the task of cross-domain label transfer. The SSMN significantly outperforms several strong baselines on all three datasets.

In summary, our contributions include: (a) presenting the task of One-Shot Diagram Part Labeling (b) proposing Structured Set Matching Networks, an end-to-end combination of CNNs and structured prediction for matching elements in two sets (c) proposing converting diagrams into distance transforms, prior to passing them through a CNN (d) presenting a new diagram dataset DiPART towards the task of one-shot part labeling (e) obtaining state-of-the-art results on 3 challenging setups: diagram-to-diagram, image-to-image and image-to-diagram.

2. Related Work

One-Shot Learning. Early work on one-shot learning includes Fei-Fei *et al*. [15, 16] who showed that one can take advantage of knowledge coming from previously learned categories, regardless of how different these categories might be. Koch *et al*. [25] proposed using a Siamese network for one-shot learning and demonstrated their model on the Omniglot dataset for character recognition. More recently Vinyals *et al*. [46] proposed a matching network for one-shot learning, which incorporates additional context into the representations of each element and the similarity function using LSTMs. The SSMN model builds on matching networks by incorporating a global consistency model that improves accuracy in the set-to-set case.

Visual Illustrations. There is a large body of work in sketch based image retrieval (SBIR) [51, 33, 47, 55, 14]. SBIR has several applications including online product searches [51]. The key challenge in SBIR is embedding sketches and natural images into a common space, and is often solved with variants of Siamese networks. In SSMN, each pair of source and target encoders with the corresponding similarity network (Section 3.1) can be thought of as a Siamese network. There also has been work in sketch classification [13]. More recently [52] proposed a CNN architecture to produce state-of-the-art results on this set. They noted that off-the-shelf CNN architectures do not work well for sketches, and instead proposed a few modifications. Our analysis shows that converting diagrams to distance trans-

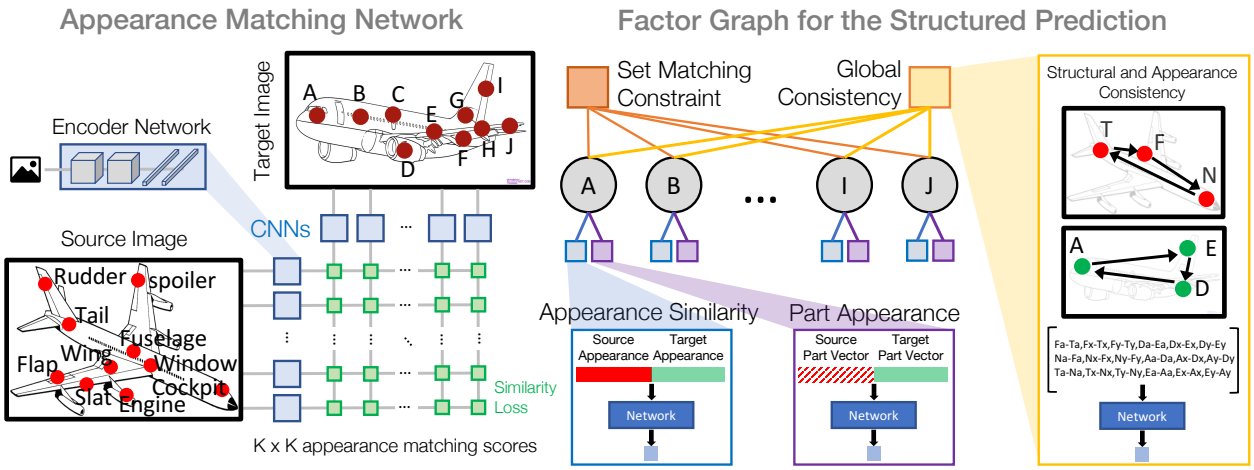


Figure 2. Overview of the Structured Set Matching Network (SSMN) model.

form images allows us to use architectures resembling ones designed for natural images. Work in understanding diagrams for question answering includes domains of science [24, 26], geometry [35, 36] and reasoning [18]. Abstract scenes have also been analyzed to learn semantics [58] and common sense [44].

Part Recognition. There is a large body of work in detecting parts of objects as a step towards detecting the entire object including [6, 17, 57, 37, 2, 40] to name a few. In contrast to these efforts (which learn part classifiers from many training images), we focus on one-shot labeling.

Learning Correspondences and Similarity Metrics. Labeling parts from a source image can be translated into a correspondence problem, which have received a lot of attention over the years. Recently, deep learning models have been employed for finding dense correspondences [8, 20, 54, 21] and patch correspondences [53, 22]. The SSMN differs from the majority of them due to its ability to jointly reason over the set of elements in the source and target. There has also been a fair amount of work on learning a metric for similarity [4, 7, 39]. The appearance similarity factor in the SSMN model builds on past work in this area. Recently, Han *et al*. [21] have proposed incorporating geometric plausibility into a model for semantic correspondence, a notion that is also strongly leveraged by the SSMN.

Global Normalization with Neural Networks. Most work on structured prediction with neural networks uses locally-normalized models, *e.g*., for caption generation [23]. Such models are less expressive than globally-normalized models (*e.g*., CRFs) [1] and suffer from *label bias* [27], which, as we show in Sec 4, is a significant problem in set-to-set matching. A few recent works have explored global normalization with neural networks for pose estimation [42] and semantic image segmentation [34, 56]. Models that permit inference via a dynamic program, such as linear chain CRFs, can be trained with log-likelihood by implementing the inference algorithm (which is just sums and products) as part of the neural network's computation graph, then performing backpropagation [19, 12, 50, 11, 32]. Some work has also considered using approximate inference during training [5, 42, 48]. Search-based learning objectives, such as early-update Perceptron [9] and LaSO [10], are other training schemes for globally-normalized models that have an advantage over log-likelihood: they do not require the computation of marginals during training. This approach has recently been applied to syntactic parsing [1] and machine translation [49], and we also use it to train SSMN.

3. Structured Set Matching Network

The structured set matching network (SSMN) is a model for matching elements in two sets that aims to maximise local match scores between corresponding elements and a global consistency score amongst all matched elements, while also enforcing a matching constraint between the two sets. We describe SSMN in the context of the problem of one-shot part labeling, though the model is applicable to any instance of set-to-set matching.

The one-shot part labeling problem is to label the parts of an object having seen only one example image from that category. We formulate this problem as a label transfer problem from a source to a target image. Both images are labeled with K parts, each of which is a single point marked within the part as shown in Figure 2. Each part of the source image is further labeled with its name, *e.g.*, "tail." The model's output is an assignment of part names to the points marked in the target image, each of which much be uniquely drawn from the source image.

There are several modeling challenges in the one-shot part labeling task. First, the model must generalize to images of unseen categories, with parts that were never encountered during training. Thus, the model cannot simply learn a classifier for each part name. Second, spatially close part locations and the absence of any color or textual information in diagrams renders local appearance cues inherently ambiguous. Thus, part labeling cannot be decom-

posed into independent labeling decisions for each target part without losing valuable information. Third, pose variations between pairs of images renders absolute positions ambiguous. To overcome the ambiguities, the model must jointly reason about the relative positions of all the matched parts to estimate whether the pose variation is globally consistent. Finally, the model must enforce a 1:1 matching.

The SSMN addresses the above challenges by using a convolutional neural network to extract local appearance cues about the parts from the source and target images, and using a structured model to reason about the entire joint assignment of target parts to source parts without making per-part independence assumptions. It is a non-probabilistic model; thus, it is similar to a conditional random field [27] with neural network factors, except its scores are not probabilities. Figure 2 shows an overview of the SSMN applied to the problem of one shot part labeling. The factor graph representation shows the four factors is the SSMN:

1. Appearance similarity (f_a) – Captures the local appearance similarity between a part in the source image and a part in the target image. (Sec. 3.1)
2. Part appearance (f_p) – Captures the appearance similarity between a part in the target image and a name assigned to it. In the one-shot setting, this is only valuable for parts seen a priori amongst object categories in the training data. (Sec. 3.2)
3. Global consistency (f_{gc}) – Scores whether the relationships between target parts are globally consistent with those of the matched source parts, *i.e.* whether the source-to-target transformation is reasonable (Sec. 3.3)
4. Matching constraint (f_m) – Enforces that target labels are matched to unique source labels.

The first three of these factors represent neural networks whose parameters are learned during training. The fourth factor is a hard constraint. Let m denote a matching where $m(i) = j$ if target part i is matched to source j. SSMN assigns a score to m using these factors as:

$$f(m) = f_{gc}(m) + f_m(m) + \textstyle\sum_i f_a(m(i), i) + f_p(m(i), i). \quad (1)$$

3.1. Appearance Similarity

The appearance of each object part is encoded by an *encoder network*, a CNN whose architecture is akin to the early layers of VGG16 [38]. The input to the CNN is an image patch extracted around the annotated part and resized to a canonical size. The output of the CNN is an embedding of the local appearance cues for the corresponding object part. The $2K$ object parts (from both the source and target images) are each fed to $2K$ copies of the encoder with shared weights, producing $2K$ appearance embeddings. The model creates contextualized versions of these embeddings by running a *context network*, a bidirectional LSTM, over the source embeddings and, as an independent sequence, the target embeddings. Note that the source and target points are given as sets, so we shuffle them arbitrarily before running the LSTMs. The similarity score between a source and target point is the dot product of their contextualized embeddings. This produces K^2 appearance similarity scores (f_a) as depicted by green boxes in Fig 2.

Due to the sparse nature of diagrams and the absence of much color and texture information, CNN models pretrained on natural image datasets perform very poorly as encoder networks. Furthermore, off the shelf CNN architectures also perform poorly on diagrams, even when no pretraining is used [52], and require custom modifications such as larger filter sizes. Our key insight to overcoming this problem is to convert diagrams to distance transform images, which introduces meaningful textures into the images that capture the location and orientation of nearby edges. This noticeably improves performance for diagrams, whilst using the CNN architectures designed for natural images.

3.2. Part Appearance

In the one-shot setting, at test time, the model observes a single fully labeled image from an object category, that it has not seen before. However, some common part names are likely to recur. For example, if various animal categories appear across training, validation and test categories, parts such as "leg" will recur. Thus, a model can benefit from learning typical appearances of these common parts across all types of images. The part appearance factor enables the model to learn this kind of information.

Let p_i be a parameter vector for the i^{th} source part's name, and t_j be the output of the *encoder network* for the j^{th} target part (Sec. 3.1). The part appearance model assigns a match score $f_p(i, j)$ between source i and target j: $f_p(i, j) = w_2^T \texttt{relu}(W_1[p_i\ t_j]^T + b)$. Along with the layer parameters, p_i is also learned at training time. The model has a separate parameter vector p_i for each part name that appears at least twice in the training data; all other parts are mapped to a special "unknown" parameter vector.

3.3. Global Consistency

In addition to local appearance, consistency of the relations between matched source and target parts provides a valuable signal for part set matching. However, these relations may be transformed in an unknown but systematic way when moving from the source to the target. For example, if the target is left-right flipped relative to the source, all parts to the left of part x in the source should be on the right of x in the target. Alternatively, the target may be drawn in a different style that affects the appearance of each part. Given a matching, the global consistency factor learns whether the implied source-to-target transformation is likely.

We factor the global consistency (f_{gc}) into the sum of two terms: structural consistency (f_{sc}) for pose variations and appearance consistency (f_{ac}) for style variations. Both terms are neural networks that score entire matchings m using the same architecture, but different inputs and parameters. The score for a matching is computed from a set of *relation vectors* $\Delta(m)_{ij}$ for each part pair i, j in the matching

m, then applying fully connected layers and sum-pooling:

$$h_{ij}(m) = \texttt{relu}(W_2\,\texttt{relu}(W_1\Delta(m)_{ij} + b_1) + b_2),$$
$$f_*(m) = \sum_i^{|m|}\sum_j^{|m|} w^T h_{ij}(m), \quad (2)$$

where $*$ could be sc or ac. For structural consistency (f_{sc}), $\Delta(m)_{ij}$ encode the relative positions of pairs of matched parts. Recall that $m(i)$ denotes the source part matched to target part i. Let $loc^s_{m(i)}$ and loc^t_i denote the x/y positions of source part $m(i)$ and target part i respectively. The relative positions of a pair of parts i, j are then encoded as a 4-dim vector, $\Delta(m)_{ij} = [loc^s_{m(j)} - loc^s_{m(i)}, loc^t_j - loc^t_i]$. For appearance consistency (f_{ac}), the Δ vectors replaced by $[app^s_{m(j)} - app^s_{m(i)}, app^t_j - app^t_i]$, where $app^s_{m(i)}$ and app^t_i represent the appearance embeddings output by the encoder network in Sec. 3.1.

4. Training and Inference

Training. We train the SSMN by optimizing a structured loss on the set of part-matched images using stochastic gradient descent (SGD). Each iteration of SGD evaluates the model on a single pair of images to compute a per-example loss. Gradients are then backpropagated through the neural networks that define the model's factors.

Crucially, we train SSMN with global normalization. We found locally-normalized models performed poorly on set-to-set matching because they progressively begin to ignore scoring information as the sequence continues. A locally-normalized model, such as an RNN, would order the target points and then learn a probability distribution $P(m(i) = j|m(i-1), ..., m(1))$. After each prediction, the space of possible source points for the remaining points decreases by 1 in order to guarantee a matching. This process is problematic: note the probability for the final point is always 1, as there is only 1 source point remaining to choose from. Thus, even if the model is confident that the final pair does not match based on a pairwise similarity score, that information will be *ignored entirely* in its probabilities. This problem is an instance of *label bias* [27], known to reduce the accuracy of locally-normalized models. This observation is also consistent with that of Vinyals *et al*. [45, 46], who observed that treating unordered sets as ordered sequences enables the use of RNN models, which provide improvements to matching performance; however the ordering of elements passed to the RNNs matters.

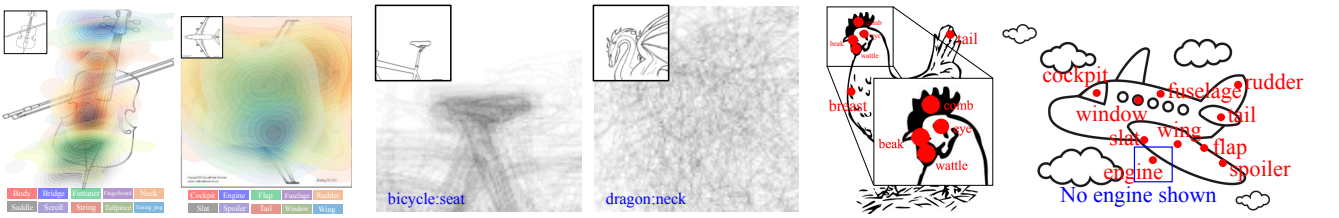
Our training uses Learning as Search Optimization (LaSO) framework [10], an objective function that is well-suited to training globally-normalized models with intractable exact inference. These models often rely on beam search to perform approximate inference, as does SSMN. During training, the LaSO objective penalizes the model each time the correct labeling falls off the beam, thereby training the model parameters to work well with the beam search. Also, unlike other objectives for globally-normalized models (*e.g*., log-likelihood of the matching), LaSO's gradient can be calculated without computing the marginal distribution over matchings or the highest-scoring matching. This is important as, in SSMN, both quantities are intractable to compute exactly due to the global consistency factor.

The LaSO objective function for a single training example is as follows. Each training example inputs to the model a pair of annotated images, and a label m^* that represents the correct part matching for the pair. The LaSO objective is defined in terms of the intermediate results of a beam search with beam size B in the model given the input. Let $\hat{m}^i_t$ denote the i^{th} highest-scoring matching on the beam after the t^{th} search step. Let m^*_t denote the correct partial matching after t time steps. The LaSO objective function encourages the score of m^*_t to be higher than that of the lowest-scoring element on the beam at each time step of the search:

$$\mathcal{L}(f) = \sum_{t=1}^{T} \max(0, \Delta(m^*_t, \hat{m}^i_t) + f(\hat{m}^B_t) - f(m^*_t)). \quad (3)$$

This loss function is a margin-based objective, similar to that of a structured SVM [43] or max-margin Markov network [41]. The loss is 0 whenever the score of the correct partial matching $f(m^*_t)$ is greater than that of the lowest-scoring beam element $f(\hat{m}^B_t)$ by the margin $\Delta(m^*_t, \hat{m}^i_t)$, and nonzero otherwise. We set $\Delta(m^*_t, \hat{m}^i_t)$ to be the number of incorrectly matched points in $\hat{m}^i_t$ (We have omitted the dependence of f on the input and model parameters for brevity). At the last time step, B is set to 1 to encourage the correct matching to have the highest score. If at any point during the search the correct partial matching falls off the beam, the search is restarted by clearing the search queue and enqueuing only the correct partial matching.

Calculating the gradient of the neural network parameters with respect to this loss function has two steps. The first step is the forward computation, which runs beam search inference in the SSMN on the input and the corresponding forward passes of its constituent neural networks. After each step of the beam search, the gradient computation checks for a margin violation. If a margin violation is found, it is recorded and the search is restarted from the correct partial matching. If not, the beam search continues normally. The output of the forward computation is a collection of margin violations and a value for the loss function. The second step is the backward computation, which backpropagates the loss through neural networks to compute the gradient. The loss $\mathcal{L}$ is a sum of terms of the form $f(m)$, and $f(m)$ is a sum of scores output by f's constituent neural networks (Equation 1). Thus, the gradient $\frac{\partial \mathcal{L}}{\partial f}$ is simply a weighted sum of the gradients of the constituent neural networks, each of which can be calculated using standard backpropagation. The inputs with respect to which the gradients are calculated, as well as each gradient's weight in the sum, depend on the the particular margin violations encountered in



(a) Spatial distribution of part locations. (Left) Coherently (Right) Incoherently distributed parts

(b) Average images of a part patch of an object category. (Left) Scale and shape coherence (Right) Incoherence

(c) Annotation examples. (Left) Spatially close parts (Right) Occluded by other parts or missing.

Figure 3. Challenges in the DiPART dataset. Local position and appearance cues are often insufficient to provide good matching.

the forward computation. We refer the reader to [49] for a detailed description of the gradient computation for LaSO in neural sequence-to-sequence modeling.[1]

Encoder Network Initialization. The encoder networks (Section 3.1) are pre-trained by optimizing a surrogate objective. A bank of CNNs encode image patches of parts and a bank of similarity networks compute the similarities between the appearance encodings (represented as a $K \times K$ matrix in Figure 2, where K is number of parts). Each row and each column of this matrix undergo a softmax operation followed by a K category cross-entropy objective. The surrogate objective is the sum of these $2K$ cross-entropy objectives. This surrogate objective encodes local appearances, but is faster to train than the SSMN objective, and is hence suitable for pre-training the appearance encoder networks. We refer to this surrogate objective as the appearance matching network (AMN) objective.

Inference. Exact inference in SSMN is intractable due to the global consistency factor, which defines a global score for the entire matching.[2] Thus, exact inference would require enumerating and scoring all $K!$ permutations of K parts. Instead, we use approximate inference via beam search. As outlined above, SSMN is trained to ensure that the beam search is a good approximate inference strategy. The beam search starts by ordering target parts arbitrarily. The search maintains a queue of partial matchings, which at time step $i-1$ consists of B partial matchings between the first $i-1$ target parts and the source parts. The i^{th} search step generates several new matchings for each partial matching on the queue by matching the i^{th} target part with each unmatched source part. The search computes a score for each expanded matching and enqueues it for the next step. The search queue is then pruned to the B highest-scoring partial matchings. This is repeated until each target part has been assigned to a source part label. The global consistency factor is used to score partial matchings by generating the *relation vectors* (in Eq.2) for the points matched thus far.

[1] We implement SSMN using Probabilistic Neural Programs (PNP) [30], a library for structured prediction with neural networks that provides a generic implementation of LaSO.

[2] Without global consistency, exact inference in SSMN can be performed with the Hungarian algorithm for maximum-weighted matching.

5. Datasets

DiPART Dataset. We present the Diagram Part Labeling (DiPART) dataset, consisting of 4,921 images across 200 object categories. Categories span rigid objects (*e.g.*, cars) and non-rigid objects (*e.g.*, animals). Images are obtained from Google Image Search and parts are labelled by annotators. DiPART is split into train, val and test sets, with no categories overlapped. Since each pair of images within a category can be chosen as a data point, the number of data points is large (101,670 train, 21,262 val, and 20,110 test).

DiPART is challenging for several reasons. First, the absence of color and dense texture cues in diagrams renders local appearance cues ambiguous. Second, having access to only point supervision [3] at training time is challenging compared to having detailed segmentation annotations for parts as in previous natural image datasets (*e.g.*, Pascal Part [6]). Third, parts for several categories are located very close by, requiring very fine grained analysis of the texture-sparse diagrams (Fig. 3-(c)). Fourth, the appearances and locations of parts are generally not coherent across samples within a category. Finally, the one-shot setting renders this even more challenging.

Pascal Part Matching (PPM) Dataset. To evaluate SSMN on labeling parts in natural images, we use images from the Pascal Part dataset [6] with more than 10 parts and convert part segments to point annotations using the centers of mass. We called it Pascal Part Matching (PPM), which consists of 74,660 train and 18,120 test pairs in 8 categories with 10 parts.

Cross-DiPART-PPM Dataset. For cross domain matching experiments, we find all overlapping categories and part names between DiPART and Pascal Part Matching to make Cross-DiPART-PPM. It consists of 5 categories with 4 parts and 22,969 image-to-diagram pairs (18,489 train and 4,480 test).

More details about the datasets including download links as well as more results can be found in the project page.

6. Experiments

Set-up. Training neural networks with global normalization typically requires pre-training with log-likelihood to obtain good results [1, 49]. In training the SSMN, we pretrained it

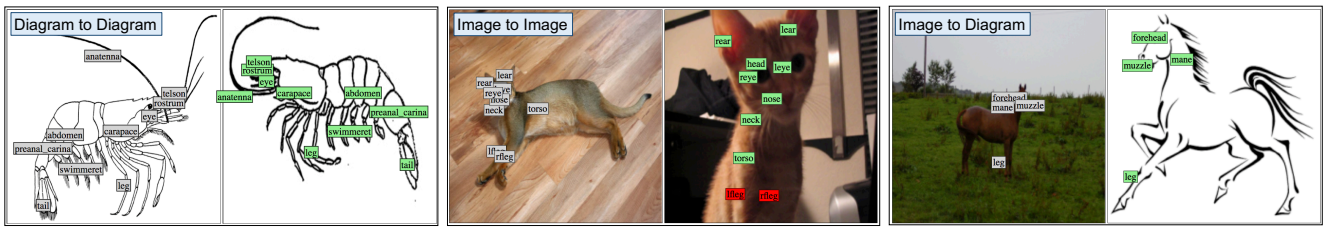


Figure 4. Qualitative results from our SSMN model. In each pair of images, the labeled source images is on the left and the target is on the right. A green box indicates a correct match and a red box indicates an incorrect match.

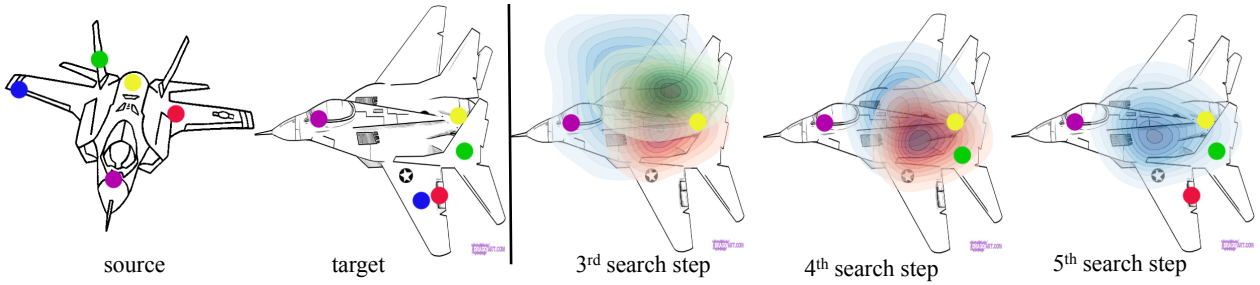


Figure 5. Visualization of expected part locations during the search by the structural consistency factor (f_{sc}). The example has both the varying pose and a non-trivial transformation of source to target part locations. The matched parts of each diagram are represented by the color-coded points. Each color-coded heatmap shows the score assigned by the structural factor to every location in the diagram for the unmatched part with the same color. For visual clarity, we display 5 of the 10 parts.

using the appearance matching network (AMN) surrogate objective (Sec 4), then fixed the weights of the convolutional layers while retraining the remainder of the network with 1 epoch of LaSO with a beam size parameter of 5 in most experiments unless mentioned (as described in Sec 4). We found that additional training epochs did not improve accuracy, presumably because pre-training brought the parameters close to a good solution. At test time, we ran the SSMN with a beam size of 100 in all experiments, except the ones that measured accuracy at different beam sizes. We chose this number based on experiments on the validation set which found that accuracy plateaued beyond 100.

The *encoder network* uses two convolutional layers (64 filters of 3×3 and 96 filters of 3×3), each followed by 2×2 max pooling with stride of 2. This is followed by two fully connected layers, with 128 and 64 hidden units respectively. The *context network* is a single layer bidirectional LSTM with 50 hidden units. We train the network using SGD with momentum and decay and 10^{-4} initial learning rate.

Note that the datasets we used in all the evaluations have no categories overlaps in train and test splits. It is a challenging set-up that matching the appearances of source and target that are never seen in training phase.

Baselines. We compare SSMN to the following baselines.

Nearest Neighbor (RGB) computes matches using local appearance cues only by comparing raw image patches centered on the part's point using a euclidean metric.

Affine Transform baseline selects the matching of points that minimizes the error of a least-squares fit of the target part locations given the source part locations. This is not scalable to compute exactly, as it requires running a least-squares fit for every matching (3.7 million for 10 part matchings). We ran this approximately using beam search with width equal to 100.

Methods \ Dataset	DiPART	PPM
Random	10.0%	10.0%
Nearest Neighbor (RGB)	29.4%	11.1%
Affine Transform	32.1%	26.9%
UCN [8]	38.9%	20.2%
Matching Network (MN) [46]	41.3%	40.2%
⌞ MN+Hungarian	45.6%	42.7%
Appearance Matching Network+NN	35.7%	42.3%
SSMN-f_{gc}	44.7%	40.6%
SSMN (Ours)	**58.1%**	**46.6%**

Table 1. Accuracies of SSMN and other methods on both datasets.

Matching Network (MN) [46] independently predicts a source point for each target point. This network runs the appearance matching network described in Section 3.1, *i.e.* the encoder network with bidirectional LSTMs, to score each source given a target. The network is trained by feeding these scores into a K-way softmax then maximizing log-likelihood. A limitation of MN is that it does not enforce a 1:1 matching, hence may yield an invalid solution.

MN + Hungarian solves this problem by finding the maximum weighted matching given the matching network's scores. In contrast to the SSMN, this baseline uses the Hungarian algorithm as a post-processing step and is not aware of the matching constraint during training.

Appearance Matching Network + NN computes nearest neighbor matches using only appearance cues by the Appearance Matching network. Source and target points are fed into the encoders and matched using cosine similarity.

Universal Correspondence Network (UCN) [8] originates

from the the semantic correspondence (SC) matching literature. Minimal post processing was required to adapt it to our task and compute an accuracy metric comparable to the SSMN and other baselines. Best results were obtained when fine tuning their pre-trained network on our datasets.

Table 1 compares the accuracy of SSMN with the above baselines on the test sets for the DiPART and PPM datasets. The nearest neighbor baselines (both RGB and Appearance Matching Network) perform poorly, since they only use appearance cues with no contextual information and no matching constraint. The MN models outperforms all other baselines in both datasets. It clearly demonstrates that using sequential context, even in a set environment yields good results, consistent with the findings in [46]. Enforcing a 1:1 matching constraint via the Hungarian algorithm further improves this model. SSMN also outperforms UCN on both datasets. The SSMN outperforms other baselines because of its ability to model global consistency among the source and target sets. Training with global normalization is crucial for this improvement: if we train SSMN with local normalization, accuracy drops significantly (SSMN-f_{gc}).

Effect of Beam Size. Fig. 6 shows the test accuracies as a function of inference beam size. SSMN outperforms baselines even for beam sizes as low as 10, and saturates beyond 100. Note that even a beam size of 100 represents a tiny fraction (0.0027%) of the search space of matchings (10!).

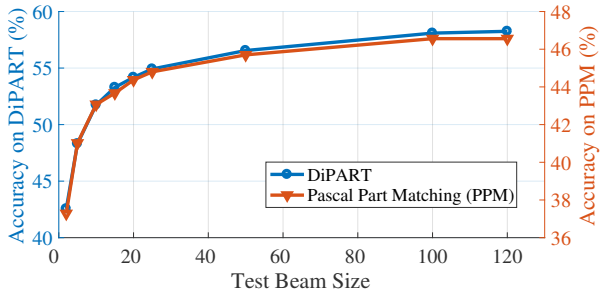


Figure 6. Accuracy as a function of inference beam size.

Distance Transform (DT). We propose using DT images as inputs to our encoder networks, as opposed to the original diagrams. We compared the two approaches using just the appearance matching network, in order to isolate appearance cues from structural cues. Using DT images provides an accuracy of 38.4% where as the original image produces 33.5%, a noticeable improvement. An interesting observation was that when we swept the space of filter sizes to find the best performing one for each configuration, the best filters for the original image were 15×15 as reported in [52] but the best filters for the DT image was 3×3, which is consistent with CNN architectures built for natural images.

Does General Part Appearance Help? 51% of part names in the validation and 54% in the test set appear in the training set of DiPART. Hence one might expect the part appearance factor (f_p) in SSMN to help significantly. An ablation study found that removing it caused very little drop in validation accuracy (within 0.1%). This shows that, even though part names overlap significantly, part appearance cues do not always transfer between categories; *e.g.*, a *head* of an elephant and a giraffe look significantly different.

Qualitative Analysis. Figure 4 shows qualitative examples and Figure 5 visualizes SSMN's search procedure.

Matching Variable Numbers of Parts. DiPART and PPM are setup to contain a fixed number of parts across images and a complete 1:1 matching between source and target sets. However, SSMN makes no such strict assumptions and can also be used in a relaxed setups. We modified DiPART to contain 9 parts in the source and 8 parts in the target image. 7 of these have a 1:1 matching and 1 part in each set has no correspondence in the counterpart. Table 2 compares SSMN to the strongest baseline (MN+Hungarian). As 1:1 matching is not guaranteed, the Hungarian algorithm only marginally improves accuracy over MN while SSMN still provides large improvements.

	MN	MN+Hungarian	SSMN
Test Accuracy	31.1%	31.5%	38.2%

Table 2. Accuracies in DiPART with varying part setup.

Cross Domain Matching. We evaluate cross domain matching on Cross-DiPART-PPM. This is the most challenging among the three setups. By using different *encoder network* architecture for source and target, we demonstrate that SSMN is able to transfer labels across domains (images-to-diagrams) reasonably well. Since global geometric consistencies are preserved regardless of visual signatures, the SSMN outperforms the strongest baseline (Table 3). In this setup, the part classification term (f_p, Sec. 3.2) drops performance since the part classifiers do not generalize across domains. Thus, SSMN without f_p (SSMN-f_p) provides further improvements.

	Random	MN	MN+Hungarian	SSMN	SSMN-f_p
Test Accuracy	25%	28.0%	26.4%	30.8%	33.1%

Table 3. Cross domain accuracies (Cross-DiPART-PPM data)

More results can be found in the supplementary material.

7. Conclusion

We consider the challenging task of one-shot part labeling, or labeling object parts given a single example image from the category. We formulate this as set-to-set matching, and propose the Structured Set Matching Network (SSMN), a combined structured prediction and neural network model that leverages local appearance information and global consistency of the entire matching. SSMN outperforms strong baselines on three challenging setups: diagram-to-diagram, image-to-image and image-to-diagram.

Acknowledgement. This work is in part supported by ONR N00014-13-1-0720, NSF IIS-1338054, NSF-1652052, NRI-1637479, Allen Distinguished Investigator Award, and the Allen Institute for Artificial Intelligence. JC would like to thank Christopher B. Choy (for the help in comparing with the UCN), Kai Han, Rafael S. de Rezende and Minsu Cho (for the discussion about SCNet) and Seunghoon Hong (for an initial discussion).

References

[1] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins. Globally normalized transition-based neural networks. In *ACL*, 2016. 3, 6

[2] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 3

[3] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 6

[4] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature Verification Using A "Siamese" Time Delay Neural Network. In *NIPS*, 1993. 3

[5] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1785–1794. JMLR Workshop and Conference Proceedings, 2015. 3

[6] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect What You Can: Detecting and Representing Objects using Holistic Models and Body Parts. In *CVPR*, 2014. 2, 3, 6

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. 1:539–546, 2005. 3

[8] C. B. Choy, J. Gwak, S. Savarese, and M. K. Chandraker. Universal correspondence network. In *NIPS*, 2016. 3, 7

[9] M. Collins and B. Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 111–118, Barcelona, Spain, July 2004. 3

[10] H. Daumé, III and D. Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 169–176, New York, NY, USA, 2005. ACM. 3, 5

[11] T.-M.-T. Do and T. Artieres. Neural conditional random fields. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, number EPFL-CONF-150585. JMLR: W&CP, 2010. 3

[12] J. Eisner. Inside-outside and forward-backward algorithms are just backprop. In *Proceedings of the EMNLP Workshop on Structured Prediction for NLP*, 2016. 3

[13] M. Eitz, J. Hays, and M. Alexa. How Do Humans Sketch Objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4), 2012. 2

[14] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31:31:1–31:10, 2012. 2

[15] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003. 2

[16] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:594–611, 2006. 2

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 3

[18] A. Ghosh, V. Kulharia, A. Mukerjee, V. P. Namboodiri, and M. Bansal. Contextual rnn-gans for abstract reasoning diagram generation. *CoRR*, abs/1609.09444, 2017. 1, 3

[19] M. R. Gormley, M. Dredze, and J. Eisner. Approximation-aware dependency parsing by belief propagation. *Transactions of the Association for Computational Linguistics (TACL)*, 2015. 3

[20] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[21] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce. SCNet: Learning Semantic Correspondence. In *ICCV*, 2017. 3

[22] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 3

[23] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 3

[24] A. Kembhavi, M. Salvato, E. Kolve, M. J. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 1, 2, 3

[25] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML*, 2015. 2

[26] J. Krishnamurthy, O. Tafjord, and A. Kembhavi. Semantic parsing to probabilistic programs for situated question answering. In *EMNLP*, 2016. 1, 3

[27] J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001. 2, 3, 4, 5

[28] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler. Rent3d: Floor-plan priors for monocular layout estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[29] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:2368–2382, 2011. 2

[30] K. W. Murray and J. Krishnamurthy. Probabilistic neural programs. *CoRR*, abs/1612.00712, 2016. 6

[31] S. Ouyang, T. M. Hospedales, Y.-Z. Song, and X. Li. Forgetmenot: Memory-aware forensic facial sketch matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[32] J. Peng, L. Bo, and J. Xu. Conditional neural fields. In *Advances in neural information processing systems*, pages 1419–1427, 2009. 3

[33] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016. 1, 2

[34] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 3

[35] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *AAAI*, 2014. 1, 3

[36] M. J. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, and C. Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 2015. 1, 3

[37] A. Shrivastava and A. Gupta. Building part-based object detectors via 3d geometry. In *2013 IEEE International Conference on Computer Vision*, 2013. 3

[38] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014. 4

[39] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *CVPR*, 2016. 3

[40] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *2011 International Conference on Computer Vision*, 2011. 3

[41] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003. 5

[42] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014. 3

[43] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 5

[44] R. Vedantam, X. Lin, T. Batra, C. L. Zitnick, and D. Parikh. Learning common sense through visual abstraction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[45] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. In *ICLR*, 2016. 5

[46] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching Networks for One Shot Learning. In *NIPS*, 2016. 2, 5, 7, 8

[47] F. Wang, L. Kang, and Y. Li. Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR*, abs/1504.03504, 2015. 1, 2

[48] S. Wang, S. Fidler, and R. Urtasun. Proximal deep structured models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 865–873. Curran Associates, Inc., 2016. 3

[49] S. Wiseman and A. M. Rush. Sequence-to-sequence learning as beam-search optimization. In *EMNLP*, 2016. 3, 6

[50] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[51] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy. Sketch me that shoe. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[52] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Sketch-a-net: A deep neural network that beats humans. In *International Journal of Computer Vision*, 2016. 1, 2, 4, 8

[53] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *CVPR*, 2015. 3

[54] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. *CVPR*, 2015. 3

[55] W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *CVPR*, 2011. 1, 2

[56] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1529–1537, Washington, DC, USA, 2015. IEEE Computer Society. 3

[57] L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 3

[58] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 3