

ShapeLearner: Towards Shape-Based Visual Knowledge Harvesting

Huayong Xu^{†1}, Yafang Wang^{†1,2}, Kang Feng[†], Gerard de Melo[‡], Wei Wu[†], Andrei Sharf^{II}, Baoquan Chen[†]
[†]Shandong University, China; [‡]Rutgers University, USA; ^{II}Ben-Gurion University, Israel

Abstract. The deluge of images on the Web has led to a number of efforts to organize images semantically and mine visual knowledge. Despite enormous progress on categorizing entire images or bounding boxes, only few studies have targeted fine-grained image understanding at the level of specific shape contours. For instance, beyond recognizing that an image portrays a cat, we may wish to distinguish its legs, head, tail, and so on. To this end, we present ShapeLearner, a system that acquires such visual knowledge about object shapes and their parts in a semantic taxonomy, and then is able to exploit this hierarchy in order to analyze new kinds of objects that it has not observed before. ShapeLearner jointly learns this knowledge from sets of segmented images. The space of label and segmentation hypotheses is pruned and then evaluated using Integer Linear Programming. Experiments on a variety of shape classes show the accuracy and effectiveness of our method.

1 Introduction

Motivation. Over the last decade, we have observed an explosion in the number of images uploaded online. Sharing platforms like Flickr have long been driving forces in turning previously undistributed digital images into an abundant resource with billions of images online. This vast amount of data holds great potential to revolutionize the way computers organize and understand images. Deng et al. [9] introduced ImageNet, a hierarchical organization of images, enabling major advances in object recognition, to the point of current deep convolutional neural networks being able to outperform humans in certain respects [27].

Still, current object recognition systems mostly operate at the coarse-grained level of entire images or of rectangular bounding boxes, while segmentation algorithms tend to consider abstract distinctions (e.g., foreground/background).

In this work, we consider the next level of image understanding and knowledge mining, aiming at a more fine-grained understanding of images by automatically identifying specific shape contours and the parts of objects that they portray. One of the major challenges for this is that there is only limited relevant training data. While it is possible to collect millions of images with social media tags [34] and it is feasible to obtain bounding boxes via crowdsourcing [19], obtaining training data with fine-grained hierarchical image information is much more challenging. Analysis of objects with respect to their parts draws from cognitive research of the human vision systems. Shapes of parts play an important role in the lower stages of object recognition [23]. Given a relatively small object part, humans can recognize the object

when the part is sufficiently unique [4, 3]. Unlike deep convolutional neural networks, humans appear to be able to acquire new categories from very few training examples.

Thus, fine-grained image understanding has remained an open problem in AI, as it requires considerable background knowledge about the objects. Progress on this challenging task has the potential to benefit numerous applications in AI, e.g. in robotics and for self-driving cars to interpret their environment, or in photography and graphics for selective image manipulation (removing or replacing a part of an object).

Contribution. We introduce *ShapeLearner*³, a system that learns the shapes of families of objects, together with their parts and their geometric realization, making the following contributions.

1. ShapeLearner requires only a small number of manually annotated seed shapes for bootstrapping and then progressively learns from new images. It achieves this by jointly performing shape classification, segmentation, and annotation to transfer information from seen to unseen images.
2. ShapeLearner can automatically analyse entirely new kinds of shapes, relying on its inference mechanism based on soft constraints.
3. Rather than learning mere enumerations, the system acquires hierarchical knowledge about the objects and their parts (Figures 1c and d). This hierarchical organization is critical for jointly analysing families of objects.

2 Related Work

Image Knowledge Harvesting. In recent years, several new methods have appeared to organize the growing amount of images on the Web [10]. The most prominent of these is ImageNet [9], a hierarchically organized image knowledge base intended to serve as the visual counterpart to WordNet [11]. While ImageNet merely provides image-level labels, subsequent research attempted to localize individual objects within those images using bounding boxes [13]. The SUN Attribute dataset [24] provides coarse-grained crowd-sourced attributes of scenes (e.g. man-made, enclosed). LabelMe [28] crowd-sourced large amounts of polygon labels, but the system does not support any transfer learning. Moreover, the labels can be arbitrary words and thus require significant cleaning and organization. Our work differs from previous work by learning specific shape contours and subparts of objects and then being able to transfer this knowledge to new images and even new types of objects.

Other types of data have been organized as well. For videos, hierarchical taxonomies have been used to train classifiers [32]. For 3D

¹ The two authors contributed equally to the paper.

² The corresponding author: yafang.wang@sdu.edu.cn

³ <http://irc.cs.sdu.edu.cn/ShapeLearner/>

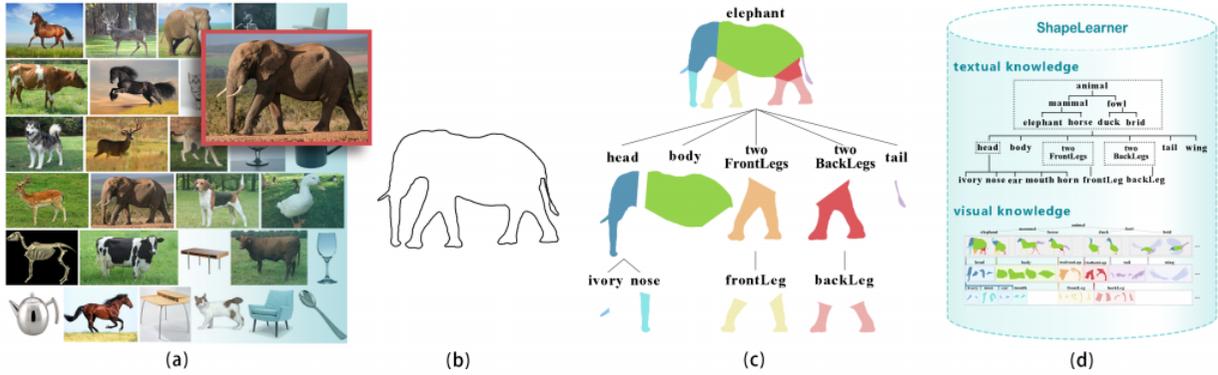


Figure 1: The proliferation of images on the Web (a) enables us to extract shapes to train ShapeLearner (b), a 2D shape learning system that acquires knowledge of shape families, geometrical instances of their inner parts and their inter-relations. Given an unknown shape (c), the system automatically determines a classification, segmentation, and hierarchical part annotation (d).

shapes, ShapeNet [5] and 3DNet [37] organize 3D (CAD) models following WordNet.

Stock graph based shape matching approaches [31] also build a hierarchy, but do not segment the shape into semantically meaningful parts. The goal is to model a complex shape using a hierarchical tree according to geometric features of the shape, which can then be used to compare the similarity of two different shapes. In our method, we use the more recent inner-distance method [20] to model the shape context for shape matching.

Segmentations and Semantic Relationships. Zhang et al. [40] observe that semantic relations of parts be shared among objects in a class and learn a set of classifiers for verb-object relationships within a class. Similarly, graph structures have been introduced for representing semantic relations of parts acquired from sets of images [22, 6, 39]. These methods focus on processing general images and scenes, while we believe to be the first to focus on the inner parts and geometries of individual shape classes.

Grammar-like descriptors for visual words and visual phrases may be defined to enhance image processing and recognition [38]. Recently, Chen et al. [7] presented a method for harvesting large amounts object relationships from images based on their probabilistic structural patterns and geometric characteristics. While their analysis is at the level of object relationships, our method focuses on a fine-grained sub-part analysis. Multiple instances of objects and parts within a class provide important contextual information that can be utilized for joint learning and segmentation [1, 35, 18]. Huang et al. [17] recently presented a data-driven approach for simultaneous segmentation and annotation of free-hand sketches. Although this problem is quite different, we compare our algorithm with theirs later in Section 5.

Deep convolutional neural networks [36, 14] can be trained to produce segmentations, but they do not address our task setting, as they depend on the existence of very large numbers of training examples per label. Related work in this area [16, 36, 14] assumes a standard supervised setting: given a large training dataset for a given class, these methods learn new segmentations. Thus, existing approaches have been limited to very small numbers of object classes, often even just a single one such as human bodies. ShapeLearner, in contrast, is aimed at learning new part segmentations for many classes, given much more limited supervision and relying on knowledge transfer from related classes.

3 Overview and Knowledge Model

High-Level Perspective. ShapeLearner constructs a relational hierarchy that indexes 2D shapes by utilizing taxonomic knowledge of object shape classes and their inner parts. Our goal is to progressively acquire such knowledge by transferring information about indexed shapes onto new ones.

We bootstrap the system by providing labeled seed images in several categories (e.g., mammals, fowl, home appliances). This involves segmenting images collected via Google Images to separate the objects from their environment. Objects are then manually segmented further into meaningful parts and labeled following the WordNet taxonomy. ShapeLearner captures this information about parts and their relations in a tree-like hierarchy by connecting parts to their siblings and ancestors. This can be viewed as a knowledge base with *isA*, *isPartOf*, and *hasShape* relationships.

ShapeLearner includes a knowledge transfer algorithm for understanding unknown shapes. It accounts for both shape geometry and high-level semantic relations from its previously acquired knowledge to infer the correct classification and segmentation of the new object shape. This is illustrated in Figure 2: Given an unknown shape, we compute a raw set of segmentation candidates considering merely the shape’s geometry. We determine additional candidates by matching with geometrically similar shapes and transferring their segmentation. This yields a set of segmentation hypotheses about the unknown shape. ShapeLearner then transfers its knowledge onto the shape by relying on an inference step to remove false hypotheses and select a valid segmentation that complies with the shape’s hierarchical taxonomy. Finally, ShapeLearner transfers this knowledge back by indexing the new shape and progressively updating its store of visual knowledge.

In the final part of the paper, we highlight some applications based on ShapeLearner. We describe the ShapeExplorer system, which supports image retrieval based on partial shape queries and shape morphing, among other things. We also describe our system for keyword-based image retrieval with special support for attributes.

ShapeLearner’s Knowledge. ShapeLearner is directly linked to the WordNet [11] taxonomy, which provides a semantic organization of classes. Focusing on a subset of this taxonomy, we adopt its *isA* class hierarchy and additionally harvest knowledge for *isPartOf* and *hasShape* facts (e.g. *isPartOf*(leg, human), *hasShape*(baseball, round)). Thus, ShapeLearner acquires knowledge of an object’s *shape*, its *parts*, and *shapes of the parts* (see Figure 3).

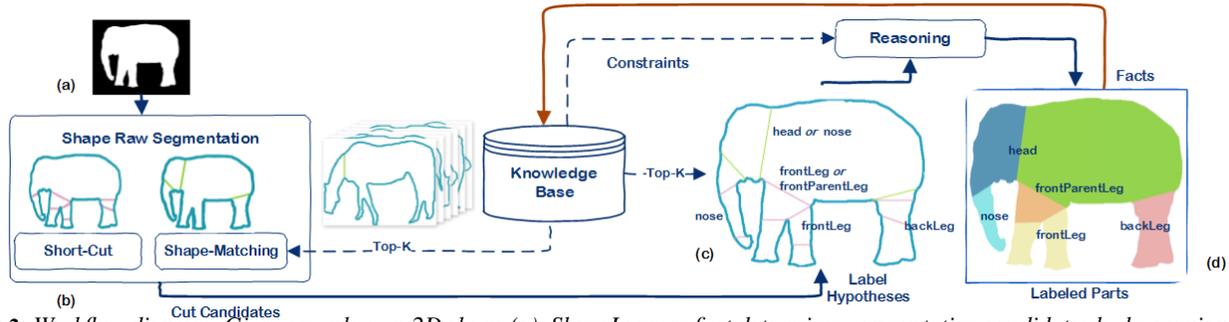


Figure 2: Workflow diagram. Given an unknown 2D shape (a), ShapeLearner first determines segmentation candidates by leveraging cut and shape matching information (b). The system uses its acquired knowledge to label candidates (c). Finally, it makes use of reasoning to prune false hypotheses and infer a classification and semantic segmentation of the shape (d).

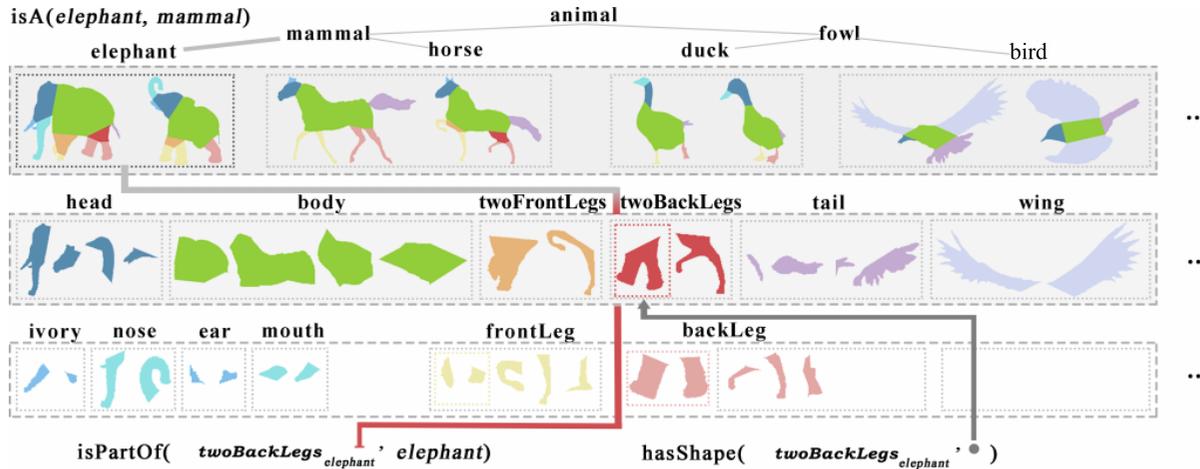


Figure 3: A snapshot of the knowledge in ShapeLearner's hierarchy, zooming in on mammals and fowl. We show also a subset of the relational facts *isA*, *isPartOf*, and *hasShape*.

We begin by defining the four basic concepts that ShapeLearner relies on:

- Shapes $\mathcal{S} = \{s_0, s_1, \dots, s_{n_S}\}$ define the contour of independent 2D objects in an image.
- Classes $\mathcal{C} = \{c_0, c_1, \dots, c_{n_C}\}$ define a category (e.g., species) of objects in the data repository.
- Parts $\mathcal{P} = \{p_0, p_1, \dots, p_{n_P}\}$ define a decomposition of a shape into meaningful components.
- Labels $\mathcal{L} = \{l_0, l_1, \dots, l_{n_L}\}$ define the textual annotations for each part.

Initially, a seed set of parts is manually preprocessed and transferred into ShapeLearner. In this step, the user manually annotates parts in shapes with labels from WordNet (e.g., *head*, *tail*, etc.) as well as semantic relations such as `hasShape(elephant, elephantShape)`, `isA(elephant, mammal)`, and `isPartOf(tail, elephant)`. ShapeLearner stores this information in a hierarchical structure (see Figure 3).

Next, we use ShapeLearner to statistically infer the following knowledge based on available evidence:

- **Part number:** the number of parts per class may be fixed or bounded (e.g., a horse has 2 front legs, an elephant has 1 trunk).
- **Part distinctiveness:** Shape classes may have discriminate parts defined by the frequency of a part in all classes (e.g., the *elephant* class has trunks as a distinct part within the class of *mammals*).

Part distinctiveness is at the core of shape classification and disambiguation. The part distinctiveness score for a part p in class $c \in \mathcal{C}$ is calculated as the inverse fraction of classes containing this part: $\frac{|C|}{|p \in C|} \geq \epsilon$, where ϵ refers to the threshold for acknowledging a part as distinctive. In our experiments, we use $\epsilon = |C|$, which means that the part occurs in just a single class.

4 Shape Analysis

Classification and semantic segmentation of an unknown object shape typically pose a chicken-egg problem: we may require information about one in order to solve the other. Given an unknown 2D shape, ShapeLearner jointly solves for both classification and semantic segmentation by relying on an inference procedure to reason from its knowledge in accordance with statistical constraints and the shape geometry. In fact, it jointly optimizes classification, segmentation, as well as part annotation. We next provide the technical details of this process.

4.1 Shape Segmentation Hypotheses

Given an unknown shape of an object, we compute a set of possible part candidates specified by different cuts in the shape (see cuts in Figure 5(c)). Initially, we compute cuts accounting merely for the shape geometry, applying the short-cut rule of [21], which is motivated by the human vision system. This method yields somewhat consistent cuts tracking the geometric features of the shape contour. Nevertheless, our algorithm does not require an exact segmentation

into meaningful parts but only a loose approximation. A somewhat reasonable segmentation is sufficient at this step.

Next, ShapeLearner transfers additional segment hypotheses from its existing knowledge to further enrich the candidate set. Shape matching plays an important role in adding new cuts that further enrich segmentation and compensate when the short-cut geometry-based method is insufficient. For instance, in Figure 5(a), the smooth elephant head could not be segmented by the short-cut method.

To accomplish this, ShapeLearner finds the best matching shapes in its existing collection and transfers their segmentation onto the input shape. Shape matching is performed using the inner-distance similarity metric [20]. We found this method suitable as it is computationally efficient, rotation-invariant, and robust with respect to other state-of-the-art 2D contour matching techniques (e.g., [2]).

Following the inner distance metric [20], we define $C(\pi(A, B))$ as the matching cost value for two shapes A and B . In a nutshell, given two shapes A and B , described by their contour point sequences p_1, p_2, \dots, p_n and q_1, q_2, \dots, q_m , respectively, we define the cost value $c(p_i, q_j)$ as the χ^2 statistic assessing the similarity of the corresponding point histograms. We compute the optimal matching between A and B , denoted as $\pi : (p_i, q_{\pi(i)})$, using dynamic programming. According to the inner distance approach [20], the mapping from shape A to B should minimize the cost. This is based on dynamic programming to solve the sequence matching problem. We define the minimum cost value by $C(\pi) = \sum_{i=1}^n c(i, \pi(i))$ and the number of matching points is $M(\pi) = \sum_{i=1}^n \delta(i)$, where $\delta(i) = 1$ if $\pi(i) \neq 0$, and 0 if $\pi(i) = 0$.

Next, we define a cut, i.e. $\text{cut}_A(p_i, p_j)$, as the 2D line connecting contour points p_i, p_j in shape A . Thus, to transfer $\text{cut}_A(p_i, p_j)$ from shape A in ShapeLearner onto the input shape B , we simply use the computed shape matching π and transfer $\text{cut}_A(p_i, p_j)$ to $\text{cut}_B(q_{\pi(i)}, q_{\pi(j)})$ (Figure 5).

To reduce noise in the segmentation candidates, ShapeLearner considers only the top $k_1 = 5$ best matching shapes in its collection. Additionally, it relies on the following constraints to remove noisy cuts (Figure 4):

- Cuts should be located in the interior of the shape.
- When cuts intersect each other, only the one corresponding to the longest contour is kept.
- If two cuts are too close together, specifically $\|\text{cut}_B(d) - \text{cut}_B(e)\|_2 \leq \epsilon$, where $\epsilon = 0.01 \times |\text{shape_points}|$, they are merged together.

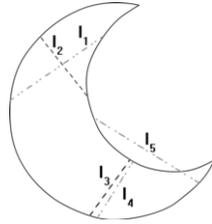


Figure 4: Cut constraints remove all dot dashed cuts (l_1, l_4, l_5).

4.2 Shape-Class and Part-Label Hypotheses

At this point, ShapeLearner has an unknown shape and a set of unlabeled segments, so the shape may belong to different classes and a cut may have different labels. Thus, ShapeLearner next annotates segments with possible label hypotheses from its knowledge and computes a valid segmentation that conforms to its acquired knowledge, by cleaning false segments and label hypotheses.

We assign a unique ID for each cut in the shape and denote an hypothesis as the pair $\text{label}(\text{cut}, \text{label}) [.]$. Additionally, we define class hypotheses as $\text{class}(\text{shape}, \text{class}) [.]$. A hypothesis may become a fact $\text{label}(\text{cut}, \text{label}) [1]$ or be evaluated as false, i.e. $\text{label}(\text{cut}, \text{label}) [0]$, following an inference

process (e.g., $\text{label}(\text{cut}@9, \text{nose}) [1], \text{class}(\text{shape}@1, \text{elephant}) [0]$).

Note that each cut corresponds to a part, so $\text{label}(\text{cut}@9, \text{nose}) [1]$ equals $\text{label}(\text{part}@9, \text{nose}) [1]$. Actually, each cut produces two parts (e.g., body and leg), but here we only consider the leg part. ShapeLearner matches the input shape against its knowledge and selects the top $k = 5$ best matching shapes using the inner distance metric. This yields multiple class and label assignments for the hypotheses.

We define the cut confidence weight with respect to the top k resulting set as follows. Given a cut c_j , label l_i , and hypotheses: $\text{label}(\text{cut}@j, l_i) [.]$, the confidence weight of cut c_j with label l_i is calculated as $w_{c_j, l_i} = \alpha \times p_1 + (1 - \alpha) \times p_2$, ($\alpha = 0.6$ in our experiments), based on two factors:

- p_1 : the confidence of assigning label l_i to cut c_j is $\frac{h_l}{k}$, where h_l is the frequency of label l_i in the top k result set.
- p_2 : A cut may match to more than one similar class. If a cut has many possible label hypotheses (say l_1, l_i, \dots, l_m), the confidence for each part is defined by the part shape matching $w'_{c_j, l_i} = M_{l_i}(\pi) / C_{l_i}(\pi)$. Then $p_2 = \frac{w'_{c_j, l_i}}{\sum_l w'_{c_j, l}}$.

Similarly, we define the class confidence weight with respect to the top k result set as follows. Given the unknown part-shape s_j , class c_i and hypothesis $\text{class}(\text{shape}@j, c_i) [.]$, the confidence of class c_i with respect to the top k result set is calculated as $w_{s_j, c_i} = \frac{h_c}{k}$, where h_c is the number of hits for class c_i .

4.3 Shape Inference

ShapeLearner jointly solves for a consistent classification and labeling by pruning noisy hypotheses and searching for the optimum class and label assignment with respect to its knowledge constraints. We formulate this problem as an Integer Linear Programming (ILP) that considers both cut labels and shape classes to yield a consistent set of truth value hypotheses.

We formulate the ILP variables as follows:

- $x_{p, l} \in \{0, 1\}$ denotes $\text{label}(\text{part}, \text{label})$ hypothesis $l \in \mathcal{L}$ for part $p \in \mathcal{P}$.
- $y_{s, c} \in \{0, 1\}$ denotes $\text{class}(\text{shape}, \text{class})$ hypothesis $c \in \mathcal{C}$ for shape $s \in \mathcal{S}$.

For each shape s , the objective function maximizes the overall confidence of hypotheses (where $w_{x_{p, l}}$ and $w_{y_{s, c}}$ are the confidence weights for cut and class hypotheses respectively, $w_{x_{p, l}} = w_{x_{c, l}}$ in the previous step):

$$\max \sum_{p \in \mathcal{P}, l \in \mathcal{L}} w_{x_{p, l}} x_{p, l} + \sum_{c \in \mathcal{C}} w_{y_{s, c}} y_{s, c}$$

subject to the following constraints derived statistically from the knowledge collection.

Class Constraints.

- A shape s can be assigned to one class at most:

$$\sum_{c \in \mathcal{C}} y_{s, c} \leq 1$$

- **Part-distinctiveness-1:** A shape class assignment should conform to its distinctive parts (if any). Denoting $(l, c) \in D_P$ as the pair set (distinctive part, class), then:

$$\forall p \in \mathcal{P} \wedge c \in \mathcal{C} \wedge (l, c) \in D_P, x_{p, l} - y_{s, c} \leq 0$$

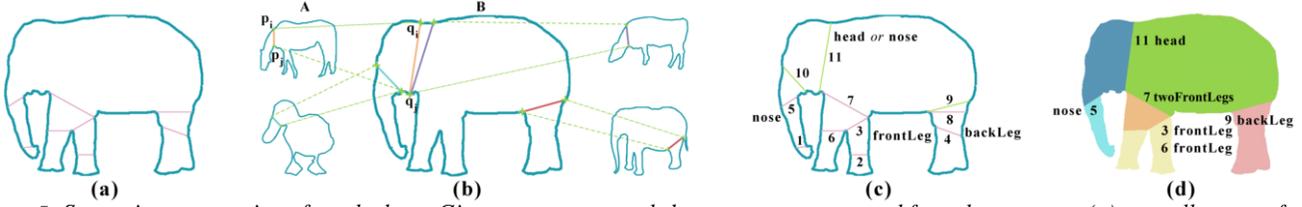


Figure 5: Semantic segmentation of an elephant. Given an unsegmented shape, cuts are computed from the geometry (a), as well as transferred from similar shapes (b). This yields multiple class hypotheses (c) which are pruned, yielding a correct semantic segmentation and annotation of the shape (d).

- **Part-distinctiveness-2:** A shape class should not consist of parts that do not belong to the class (according to `isPartOf`).

$$\forall p \in \mathcal{P} \wedge c \in \mathcal{C} \wedge (l, c) \notin \text{isPartOf}(p, c), x_{p,l} + y_{s,c} \leq 1$$

Label Constraints.

- **Part-inclusion** should conform to ShapeLearner’s part hierarchy. Denoting H_P as the set of inclusion part pairs (i.e., $(l, l') \in H_P$ if and only if l' includes l and $\text{isPartOf}(p, p')$), and using \subset for “included by”, we require

$$\forall p, p' \in \mathcal{P}, l, l' \in \mathcal{L} \wedge (l, l') \in H_P \wedge p \not\subset p', x_{p,l} + x_{p',l'} \leq 1,$$

$$\forall p, p' \in \mathcal{P}, l, l' \in \mathcal{L} \wedge (l, l') \notin H_P \wedge p \subset p', x_{p,l} + x_{p',l'} \leq 1.$$

- **Part-number:** The number of parts in a shape class should conform to the class. Denoting the number of parts as n_P , we add the constraint that

$$\sum_{p \in \mathcal{P}} x_{p,l} \leq n_P c, l.$$

Note that we require the number of parts to be less than or equal to n_P due to possible occlusions of the shape in the image (cf. the back leg in Figure 5).

After this inference step, the accepted clean facts (i.e., those of the form `label(part, label)` [1] or `class(shape, class)` [1]) are integrated into ShapeLearner’s knowledge base. The shape of each part is added as `hasShape(part, part-shape)`. Given a shape of a new class not yet in ShapeLearner, parts of the new class are identified via knowledge transfer. If the new class name is X , new facts are added as `isPartOf(part, X)` and `hasShape(part, part-shape)`.

5 Results

We now present a thorough set of experiments to evaluate ShapeLearner.

Dataset. To compile a dataset for seeding and evaluating ShapeLearner, we collected images from Google Images, Flickr, as well as public domain data used by Ren et al. [25]. We manually collect and sort these images, removing noise, frontal views, and heavily occluded shapes. We then segment the shape from its background with the aid of the open-source tool GrabCut [26]. This segmentation does not need to be precise. Instead, we account for the multiplicity of parts instances to average out the results and remove outliers. The ground truth data was labeled by 3 people. We only keep cuts or draw new cuts agreed by the majority. We extract the shape’s contour and segment it into meaningful parts simply by drawing straight lines inside the contour.

Shape and subparts are classified and annotated before being provided to ShapeLearner. Taxonomy relations (`isA`, `isPartOf`) are taken

from WordNet and textual sources [15, 33] and are used to create the hierarchy.

In total, our dataset consists of 2,020 images in 50 shape families in 7 broad classes (as shown in Table 1). Examples include humans, vases, kangaroos, mammal skeletons, handbags, umbrellas, goblets, and mushrooms. Based on these diverse seeds, our system can classify a wide range of objects if they are somewhat similar to seed images.

Labeling Accuracy. To quantify ShapeLearner’s output quality, we rely on a pixel-based metric to evaluate the part segmentation [17]. Given a segmented part, we measure its overlap with the ground-truth part as the number of pixels that are correctly labeled in the overlap vs. the incorrect ones. A part is considered adequately labeled if a reasonable percentage (precision $> 75\%$) of pixels are in the overlap. The terminology is as follows.

- **True Positive (TP):** correct cut/pixel label
- **True Negative (TN):** correct removed cut/pixel label
- **False Positive (FP):** a cut/pixel label supposed to be removed but not removed
- **False Negative (FN):** a cut/pixel supposed to be labeled, but removed.

Given these, we can use the standard definition of precision as $\frac{TP}{TP+FP}$, recall as $\frac{TP}{TP+FN}$, and $F_1 = \frac{2TP}{2TP+FP+FN}$. The class precision scores given in Table 1 and Table 2 refer to the precision of inferring class labels for the shape.

Baselines. Given all part hypotheses, we evaluate our method (both class constraints and label constraints) against two simpler baselines. However, we experiment with baselines that omit the Part-inclusion constraint and optionally the Part-distinctiveness constraint to highlight the importance of our algorithm’s advanced inference:

- **N:** the inference includes Part-number constraints and class constraints, but not Part-distinctiveness and Part-inclusion constraints.
- **N+D:** the inference includes Part-number, Part-distinctiveness constraints and class constraints, but not Part-inclusion constraints.

Comparison. For an experimental comparison, we used 20 images per family as seed data. The remaining ones in each of the 50 families were manually segmented and used as ground-truth for our evaluation. Table 1 provides an evaluation of the segmentation and classification for these baselines with respect to precision, recall, and the F_1 measure. Our method outperforms these baselines in almost all cases (except for a few cases with lower recall). Figure 6 illustrates a subset of this evaluation, providing F_1 results of baselines and of our method.

In Figure 8(a), we investigate the scalability of our method with respect to the number of initial seeds for classes with size larger than 50. Note that precision, recall, and F_1 of the segmentation increase as

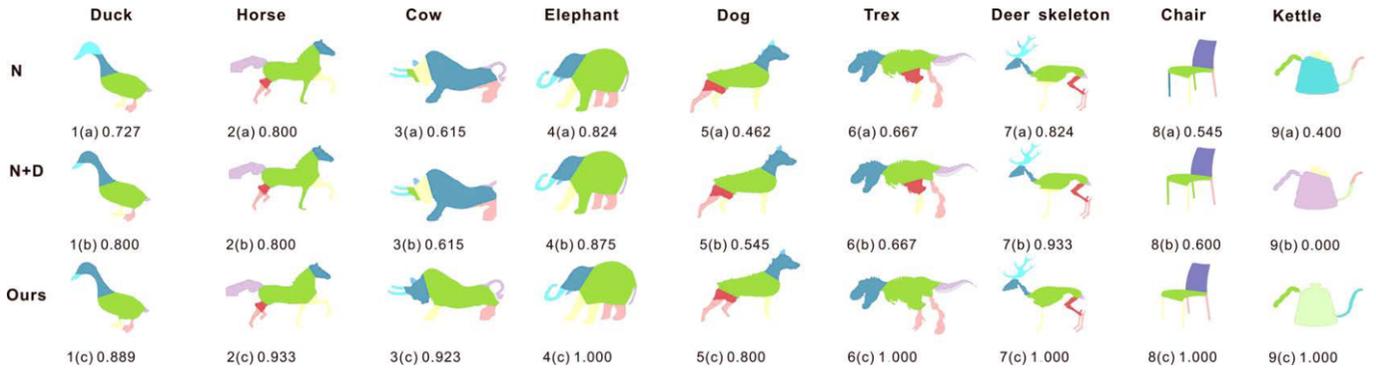


Figure 6: Representative results by our method and baseline solutions. The F_1 measure is shown below each result.

the number of seeds gets larger. After 20 seeds, the results appear to converge and the improvement becomes marginal. Thus, 20 seeds are a reasonable threshold in our experiments. This shows that a small number of seeds can suffice to represent a shape-space sufficiently well and adding more seeds can be redundant.

Figures 8(b) and 8(c) graphically plot a comparison between the baselines and ShapeLearner’s full inference mechanism according to the values in Table 1. We see that even for a small number of seeds, our method outperforms other baselines and has very good precision, recall, and F_1 .

Our classification (Table 1, bottom part) also outperforms the baselines on average. For a few classes, we did not improve over the baselines, since their contours were quite similar and lacked distinctive parts. For example, the small horn of the deer is similar to the ear of the horse. A cat’s tail may be recognized as a back leg in unique situations when the tail hangs down and the cat’s hind legs are occluded. Similarly, skeleton classes can be quite challenging. They are similar in appearance both with other kinds of skeletons and with the respective full living animal. Ribs in the skeleton are similar to legs in size and orientation. Nevertheless, the segmentation of skeletons is often successful in part precisely due to their similarity with living mammals, enabling ShapeLearner to transfer the corresponding knowledge.

Our method can infer a semantically correct segmentation even for classes that are not currently indexed in ShapeLearner. The experimental results in Table 3 show that even without any human-labeled seeds from the target class, ShapeLearner is able to exploit seeds of classes from related categories to transfer segmentation and annotation information. When the seeds from different classes are rather similar with the test shape, the outcome can be even better than the direct segmentation and annotation (see the tiger and bedroom lamp examples in Table 3). Morphological differences between tortoises and other reptiles are quite profound. Thus in this case, the transfer segmentation is less successful.

Figure 7 provides examples of three entirely new classes (a lion, ostrich, and alpaca) that were properly segmented and annotated by ShapeLearner without any prior knowledge about these classes.

We also compared the running time of our method with all constraints (“Ours”) to the method without constraints (“N”). The results show that including all constraints does not increase the complexity, increasing the runtime by just 0.5 seconds on average.

Evaluation and Comparison. Although our paper has a different target, we compare our method with segmentation algorithms for hand-drawn sketches ([17], direct retrieval (DR), and [29]). One major difference is that that work is aimed at analysing the brush strokes, which may contain significant information on the shape’s interior,

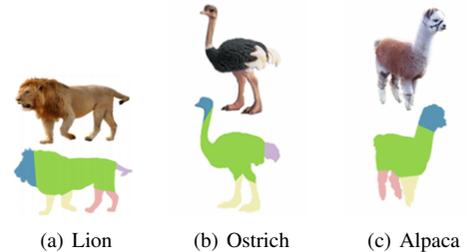


Figure 7: Semantic segmentation of three new shapes (without prior indexing of these classes by ShapeLearner).

while ours considers only the contour. From their dataset, we select all object classes with meaningful contours (omitting three classes consisting of many thin lines rather than clear contours) and compare the average segmentation and annotation precision (see Table 4). While the algorithm of Huang et al. can resolve many ambiguities due to occlusions based on the interior brush strokes, our method nevertheless gives superior results on a majority of classes, demonstrating the effective power of ShapeLearner’s knowledge. For the airplane and vase classes, our method was inferior due to the large variety (airplanes) and non-distinctiveness of parts (vases). Unfortunately, we could not perform a more in-depth comparison (e.g., w.r.t. occlusions and a larger variety of classes) since their code is not publicly available.

6 Use Cases

Finally, we present a set of use cases utilizing ShapeLearner to solve a set of challenging shape related problems.

6.1 ShapeExplorer

We have developed a system called ShapeExplorer, an interactive software tool based on a detailed analysis of images in terms of object shapes and parts. For instance, given an image of a donkey, the system may draw on ShapeLearner and its previously acquired knowledge about zebras and dogs to automatically locate and label the head, legs, tail, and so on. Based on such semantic models, ShapeExplorer can then generate morphing animations, synthesize new shape contours, and support object part-based queries (see Figure 9), as well as clipart-based image retrieval. Details were published in Ge et al. [12]. Please also refer to the URL <https://youtu.be/JTQcQkBhvyk> for an online video of this system.

6.2 Keyword Queries

Our system enables novel forms of image queries referring to specific parts of objects, e.g. for “pans with long handles”.

Table 1: Experimental results for segmentation and annotation (top) and classification (bottom).

System	Mammals	Home Appliances	Misc. Artifacts	Foods	Reptiles	Fowl	Skeletons	All Avg.	
<i>N</i>	68.3%	86.0%	88.5%	100.0%	74.5%	65.8%	57.3%	77.2%	<i>Prec.</i>
<i>N+D</i>	69.2%	90.4%	92.3%	100.0%	74.5%	66.6%	59.8%	79.0%	
<i>Ours</i>	79.4%	92.5%	92.6%	100.0%	84.1%	75.6%	71.8%	85.1%	
<i>N</i>	85.5%	93.3%	93.7%	100.0%	85.4%	90.4%	76.4%	89.2%	<i>Recall</i>
<i>N+D</i>	85.1%	92.4%	94.0%	100.0%	85.4%	90.3%	77.1%	89.2%	
<i>Ours</i>	86.9%	93.6%	94.4%	100.0%	83.0%	91.4%	80.8%	90.0%	
<i>N</i>	74.8%	88.1%	90.2%	100.0%	78.9%	74.3%	64.5%	81.6%	<i>F1</i>
<i>N+D</i>	75.3%	90.9%	93.0%	100.0%	78.9%	74.9%	66.4%	82.8%	
<i>Ours</i>	82.2%	92.5%	93.3%	100.0%	82.9%	81.1%	74.6%	86.7%	
<i>N</i>	65.3%	91.4%	87.8%	93.3%	92.5%	87.8%	61.9%	82.9%	<i>Class</i>
<i>N+D</i>	72.0%	93.2%	92.6%	93.3%	95.0%	88.9%	58.8%	84.8%	
<i>Ours</i>	71.9%	93.7%	92.6%	93.3%	95.0%	89.3%	59.3%	85.0%	

Table 2: Excerpts for segmentation and annotation (top) and classification (bottom).

System	Mammals					Home Appliances			Misc. Artifacts		Foods	Reptiles		Fowl		Skeletons		
	Elephant	Cow	Deer	Horse	Cat	Vase	Hairdryer	Broom	Rifle	Axe	Mushroom	Tortoise	Crocodile	Duck	Bird	Mammals	Dinosaur	
<i>N</i>	74.6%	62.4%	80.6%	64.5%	63.3%	67.8%	96.7%	96.7%	59.1%	93.3%	100.0%	65.6%	68.8%	63.2%	67.4%	62.2%	52.5%	<i>Prec.</i>
<i>N+D</i>	75.8%	63.2%	81.7%	64.6%	65.5%	73.3%	96.7%	96.7%	78.2%	93.3%	100.0%	65.6%	68.8%	63.8%	69.2%	64.3%	55.3%	
<i>Ours</i>	86.0%	71.4%	87.0%	77.9%	79.6%	73.3%	96.7%	96.7%	79.9%	93.3%	100.0%	78.2%	81.4%	74.4%	79.2%	75.1%	68.4%	
<i>N</i>	90.5%	81.3%	87.7%	83.9%	80.3%	80.0%	96.7%	96.7%	85.3%	93.3%	100.0%	80.9%	84.8%	89.5%	90.6%	78.4%	74.5%	<i>Recall</i>
<i>N+D</i>	88.9%	79.6%	87.7%	83.1%	80.6%	78.3%	96.7%	96.7%	86.8%	93.3%	100.0%	80.9%	84.8%	87.5%	92.2%	78.4%	75.9%	
<i>Ours</i>	91.1%	84.2%	90.0%	86.2%	84.8%	78.3%	96.7%	96.7%	88.5%	93.3%	100.0%	81.1%	89.9%	86.7%	93.1%	82.8%	78.8%	
<i>N</i>	81.2%	69.7%	83.0%	72.0%	70.1%	72.1%	96.7%	96.7%	67.8%	93.3%	100.0%	71.4%	75.4%	72.6%	75.6%	68.3%	60.6%	<i>F1</i>
<i>N+D</i>	81.3%	69.5%	83.7%	71.8%	71.5%	75.0%	96.7%	96.7%	81.6%	93.3%	100.0%	71.4%	75.4%	72.4%	77.5%	69.6%	63.1%	
<i>Ours</i>	88.0%	76.5%	87.8%	81.2%	81.4%	75.0%	96.7%	96.7%	83.3%	93.3%	100.0%	79.2%	84.9%	78.6%	84.1%	77.9%	71.4%	
<i>N</i>	94.4%	53.1%	90.4%	37.8%	86.0%	90.0%	96.7%	76.7%	69.0%	70.0%	93.3%	96.6%	90.0%	83.5%	83.3%	34.0%	89.9%	<i>Class</i>
<i>N+D</i>	94.4%	60.5%	80.9%	76.7%	76.0%	86.7%	100.0%	76.7%	93.1%	70.0%	93.3%	96.6%	100.0%	89.9%	76.7%	37.7%	79.8%	
<i>Ours</i>	94.4%	59.3%	80.9%	75.6%	76.0%	86.7%	100.0%	76.7%	93.1%	70.0%	93.3%	96.6%	100.0%	91.1%	76.7%	38.9%	79.8%	

Table 3: Experimental results for with seeds (top) and with only transfer (bottom).

Method	Feline			Reptiles				Lamp			Canine			
	Cat	Leopard	Tiger	Tortoise	Crocodile	Lizard	Gecko	Desk Lamp	Floor Lamp	Bedroom Lamp	Dog	Wolf	Fox	
<i>Precision</i>	79.6%	73.7%	75.1%	78.2%	81.4%	88.2%	88.7%	92.6%	94.6%	85.0%	79.8%	71.5%	77.3%	<i>Direct</i>
<i>Recall</i>	84.8%	91.1%	78.1%	81.1%	89.9%	78.5%	82.4%	90.7%	96.4%	85.6%	92%	84.6%	84.9%	
<i>F1</i>	81.4%	80.3%	76.2%	79.2%	84.9%	82.6%	84.9%	91.4%	95.2%	83.2%	84.5%	76.1%	80.2%	
<i>Precision</i>	66.7%	70.1%	86.7%	46.2%	71.3%	78.4%	81.2%	88.9%	92.9%	91.7%	78.0%	69.8%	74.3%	<i>Trans.</i>
<i>Recall</i>	60.0%	72.6%	70.2%	52.9%	69.5%	71.9%	77.5%	87.0%	100.0%	78.9%	69.2%	77.1%	66.1%	
<i>F1</i>	62.0%	69.6%	76.7%	48.6%	69.8%	74.8%	78.3%	87.7%	95.2%	82.1%	71.6%	71.2%	68.6%	

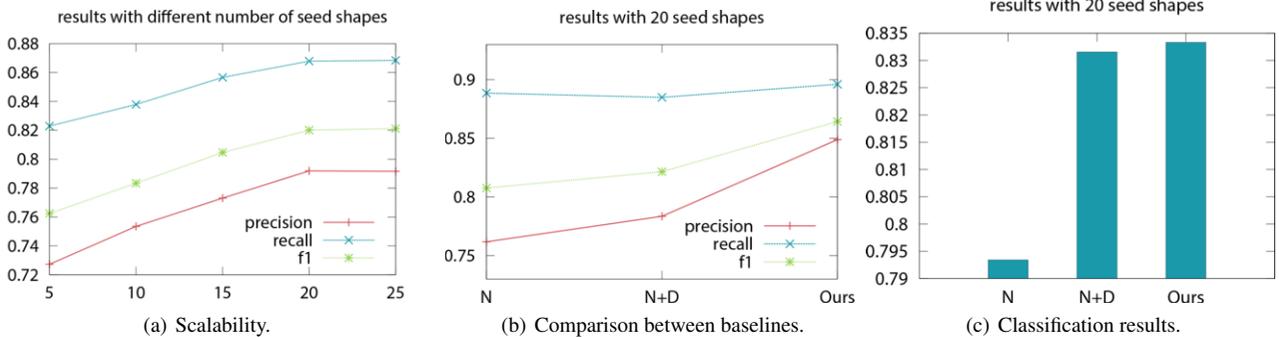


Figure 8: Experimental results graphs. In (a) we show the scalability of the average precision, recall and F1, and in (b) the comparison with other baselines. In (c) we show classification precision comparison with other baselines.

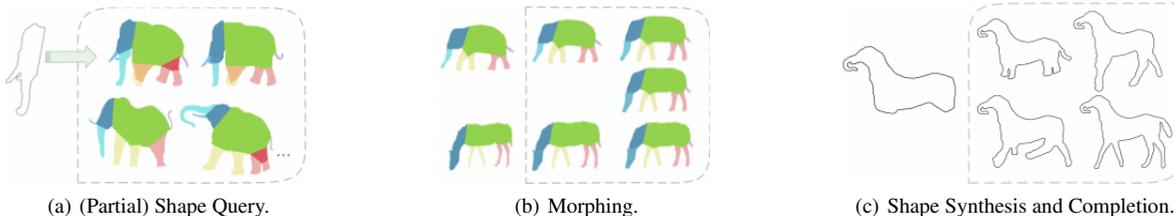


Figure 9: ShapeLearner use cases, demonstrating partial shape querying of an elephant head and trunk (a), part-based morphing between a horse and an elephant (b) and synthesis of a new creature (c).

Table 4: Comparison with Huang et al. (2014), DR, and Shen et al. (2012) in precision.

Class	DR	Shen	Huang	Ours
airplane	40.2%	56.1%	66.2%	65.8%
candelabra	39.8%	56.1%	56.7%	68.5%
rifle	49.6%	48.5%	62.2%	67.2%
fourleg	52.3%	50.0%	67.2%	80.9%
vase	51.7%	54.1%	63.1%	51.0%
human	49.2%	47.7%	64.0%	94.1%
lamp	67.8%	76.9%	89.3%	94.9%

Nouns are matched with object and part names in the database, while adjectives are matched with attributes as described below. Stop words and other unmatched words are ignored.

Attributes that can be matched include colors, angles, size, and length. All the objects are normalized according to their bounding boxes. Given the segmented parts of an object, the key line of a part is defined as the line connecting the middle point of the cut and the midpoint of its contour. The angle of a part is defined as the angular offset from a vertical line, i.e., the angle between the key line and a vertical line. The length of a component is defined as the length of the skeleton of its shape. To obtain the skeleton, we relied on an existing method [30]. Table 5 provides 7 example queries. The corresponding query results are shown in Figure 10.

Table 5: Example keyword queries.

Query	
Q1	horse with head down
Q2	horse with long tail
Q3	long tail of horse
Q4	cup with small handle
Q5	small cup handle
Q6	cup with black handle
Q7	pot with handle on the top

We observe that if the segmentation is correct, we obtain meaningful results, e.g. for Q1 and Q6. The quality of the cut of a part affects the results. In Q2 and Q3, the tail of the fifth horse is shortened due to inaccuracies in the cut of the tail. In Q4 and Q5, the system has located a handle at the top of the first cup, rather than on the right side.

7 Conclusion

We have introduced ShapeLearner, a novel system for organizing 2D shapes and their parts in a hierarchical structure that learns to



Figure 10: Keyword query results.

process new images and even new categories of objects. Our system starts with annotated seed data but then augments its knowledge by automatically processing new images and shapes. We derive a set of statistical constraints that we apply to correctly classify and segment an unknown input shape. ShapeLearner is able to transfer hypotheses based on visual similarity and relies on integer linear programming for joint inference. Our experiments show that, after seeding, ShapeLearner is able to collect valuable knowledge about shapes from uncategorized images. We additionally present several applications as use-cases of ShapeLearner, showcasing enhanced shape processing and manipulation.

In future work, we would like to extend ShapeLearner to focus not only on 2D shapes represented by their contours, but also to analyse the interior textures, for which we are exploring the use of deep convolutional neural networks. While a reduction to 2D shape contours reduces some of the noise, it results in a minimalist geometric representation. By going beyond it, ShapeLearner could thus also be extended to handle object shapes with severe shape occlusions.

We also plan to extend ShapeLearner to cover a wider range of semantic relationships and integrate it more tightly with the growing ecosystem of large-scale resources centered around the WordNet taxonomy, including ImageNet [9], YAGO [15], and UWN [8].

Finally, we are in the process of extending the seed data to cover many new categories, including medical data on bones and organs. We are also investigating crowdsourcing techniques to harvest a very broad range of categories. Initial experiments indicate that novices can fairly quickly learn how to mark parts of an object's shape. Thus, crowdsourcing techniques could enable us to quickly grow ShapeLearner's knowledge to cover thousands of categories of objects.

Acknowledgments

This project was sponsored by National 973 Program (No. 2015CB352500), National Natural Science Foundation of China (No. 61503217), Shandong Provincial Natural Science Foundation of China (No. ZR2014FP002), and The Fundamental Research Funds of Shandong University (No. 2014TB005, 2014JC001). Gerard de Melo's research is supported by China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61550110504.

REFERENCES

- [1] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen, 'icoseg: Interactive co-segmentation with intelligent scribble guidance', in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pp. 3169–3176, (2010).
- [2] Serge Belongie, Jitendra Malik, and Jan Puzicha, 'Shape matching and object recognition using shape contexts', *IEEE Trans. Pat. Ana. & Mach. Int.*, **24**, 509–522, (2001).
- [3] Irving Biederman, 'Recognition-by-components: A theory of human image understanding', *Psychological Review*, **94**, 115–147, (1987).
- [4] Thomas O. Binford, 'Visual perception by computer', in *Proc. IEEE Conf. on Systems and Control*, (1971).
- [5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, 'ShapeNet: An Information-Rich 3D Model Repository', Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, (2015).
- [6] Na Chen, Qian-Yi Zhou, and Viktor Prasanna, 'Understanding web images by object relation network', in *Proc. WWW*, pp. 291–300, (2012).
- [7] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta, 'NEIL: extracting visual knowledge from web data', in *Proc. ICCB*, pp. 1409–1416, (2013).
- [8] Gerard de Melo and Gerhard Weikum, 'Towards a universal wordnet by learning from combined evidence', in *Proc. CIKM 2009*, (2009).
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, 'Imagenet: A large-scale hierarchical image database', in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, (2009).
- [10] Santosh Divvala, Ali Farhadi, and Carlos Guestrin, 'Learning everything about anything: Webly-supervised visual concept learning', in *CVPR*, (2014).
- [11] Christiane Fellbaum, *WordNet: An Electronic Lexical Database*, Bradford Books, 1998.
- [12] Tong Ge, Yafang Wang, Gerard de Melo, Zengguang Hao, Andrei Sharf, and Baoquan Chen, 'Shapeexplorer: Querying and exploring shapes using visual knowledge', in *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*, pp. 648–651, (2016).
- [13] Matthieu Guillaumin and Vittorio Ferrari, 'Large-scale knowledge transfer for object localization in imagenet.', in *Proc. CVPR*, pp. 3202–3209, (2012).
- [14] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik, 'Hypercolumns for object segmentation and fine-grained localization', in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 447–456, (2015).
- [15] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum, 'YAGO2: Exploring and querying world knowledge in time, space, context, and many languages', in *Proc. WWW 2011*, (2011).
- [16] Qi-Xing Huang, Vladlen Koltun, and Leonidas J. Guibas, 'Joint shape segmentation with linear programming', *ACM Trans. Graph.*, **30**(6), 125, (2011).
- [17] Zhe Huang, Hongbo Fu, and Rynson W. H. Lau, 'Data-driven segmentation and labeling of freehand sketches', *ACM Trans. on Graphics*, (2014).
- [18] Hongwen Kang, Martial Hebert, and Takeo Kanade, 'Discovering object instances from scenes of daily living', in *Proc. ICCB*, (2011).
- [19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei, 'Visual genome: Connecting language and vision using crowdsourced dense image annotations', (2016).
- [20] Haibin Ling and David W. Jacobs, 'Shape classification using the inner-distance', *IEEE Trans. Pat. Ana. & Mach. Int.*, (2007).
- [21] Lei Luo, Chunhua Shen, Xinwang Liu, and Chunyuan Zhang, 'A computational model of the short-cut rule for 2d shape decomposition', *CoRR*, **abs/1409.2104**, (2014).
- [22] Tomasz Malisiewicz and Alexei A. Efros, 'Beyond categories: The visual memex model for reasoning about object relationships', in *NIPS*, (2009).
- [23] David Marr, 'Early processing of visual information', *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, **275**(942), 483–519, (1976).
- [24] Genevieve Patterson, Chen Xu, Hang Su, and James Hays, 'The sun attribute database: Beyond categories for deeper scene understanding', *Int. J. Comp. Vis.*, 59–81, (2014).
- [25] Zhou Ren, Junsong Yuan, Chunyuan Li, and Wenyu Liu, 'Minimum near-convex decomposition for robust shape representation.', in *Proc. ICCB*, pp. 303–310, (2011).
- [26] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "'grabcut": Interactive foreground extraction using iterated graph cuts', in *Proc. SIGGRAPH*, pp. 309–314, (2004).
- [27] Olga Russakovsky, Jia Deng, et al., 'ImageNet Large Scale Visual Recognition Challenge', *Int. J. Comp. Vis.*, (2015).
- [28] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman, 'LabelMe: A database and web-based tool for image annotation', *Int. J. Comp. Vis.*, (1-3), 157–173, (2008).
- [29] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu, 'Structure recovery by part assembly', *ACM Trans. Graph.*, **31**(6), 180, (2012).
- [30] Wei Shen, Yan Wang, Xiang Bai, Hongyuan Wang, and Longin Jan Latecki, 'Shape clustering: Common structure discovery', *Pattern Recognition*, **46**(2), 539–550, (2013).
- [31] K. Siddiqi, A. Shokoufandeh, S. J. Dickenson, and S. W. Zucker, 'Shock graphs and shape matching', in *Sixth International Conference on Computer Vision*, pp. 222–229, (1998).
- [32] Yang Song, Ming Zhao, Jay Yagnik, and Xiaoyun Wu, 'Taxonomic classification for web-based videos.', in *Proc. CVPR*, pp. 871–878, (2010).
- [33] Niket Tandon, Gerard de Melo, Fabian Suchanek, and Gerhard Weikum, 'WebChild: Harvesting and organizing commonsense knowledge from the web', in *Proc. WSDM 2014*, (2014).
- [34] Bart Thomee, Benjamin Elizalde, David A. Shamma, Karl Ni, Gerald Friedland, Douglas Poland, Damian Borth, and Li-Jia Li, 'YFCC100M: The new data in multimedia research', *Commun. ACM*, **59**(2), 64–73, (January 2016).
- [35] S. Vicente, C. Rother, and V. Kolmogorov, 'Object cosegmentation', in *Proc. CVPR*, (2011).
- [36] Peng Wang, Xiaohui Shen, Zhe L. Lin, Scott Cohen, Brian L. Price, and Alan L. Yuille, 'Joint object and part segmentation using deep learned potentials', in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1573–1581, (2015).
- [37] Walter Wohlkinger, Aitor Aldoma, Radu Bogdan Rusu, and Markus Vincze, '3dnet: Large-scale object class recognition from CAD models', in *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pp. 5384–5391, (2012).
- [38] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Wen Gao, 'Generating Descriptive Visual Words and Visual Phrases for Large-Scale Image Applications', *IEEE Trans. on Image Processing*, (2011).
- [39] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Wen Gao, 'ObjectPatchNet: Towards scalable and semantic image annotation and retrieval', *Comput. Vis. Image Underst.*, 16–29, (2014).
- [40] Xinming Zhang, Zheng-Jun Zha, and Changsheng Xu, 'Learning "verb-object" concepts for semantic image annotation.', in *ACM Multimedia*, pp. 1077–1080, (2011).