Pooling Objects for Recognizing Scenes without Examples

Svetlana Kordumova, Thomas Mensink, Cees G.M.Snoek
University of Amsterdam, The Netherlands
{s.kordumova, tmensink, cgmsnoek}@uva.nl

ABSTRACT

In this paper we aim to recognize scenes in images without using any scene images as training data. Different from attribute based approaches, we do not carefully select the training classes to match the unseen scene classes. Instead, we propose a pooling over ten thousand of off-the-shelf object classifiers. To steer the knowledge transfer between objects and scenes we learn a semantic embedding with the aid of a large social multimedia corpus. Our key contributions are: we are the first to investigate pooling over ten thousand object classifiers to recognize scenes without examples; we explore the ontological hierarchy of objects and analyze the influence of object classifiers from different hierarchy levels; we exploit object positions in scene images and we demonstrate new scene retrieval scenarios with complex queries. Finally, we outperform attribute representations on two challenging scene datasets, SUNAttributes and Places2.

1. INTRODUCTION

This paper strives to recognize the scene of an image, such as *beach*, *supermarket* or *youth hostel*, without using any labeled scene example. Evidence from *supervised* scene recognition suggests that objects emerge in the representations learned from millions of labeled scene images [29], inspiring us to explore whether object classifiers can be leveraged to recognize scenes without examples.

The common approach to address classification without examples is to define scene-specific attributes, train attribute detectors and rely on an attribute-to-scene mapping for classification [12, 20, 10, 13]. In [12] for example, Lampert *et al.* use attribute annotations on images from the SUNAttributes [19] dataset, to learn attribute models and create attribute-to-scene mappings. Rather than relying on attributes and attribute-to-class mappings, recent works [9, 4] showed that events, actions and emoji's can be recognized without examples using a representation based on object classifier scores and object-class affinities estimated from a social media corpus [24]. Inspired by these efforts, we explore in this paper whether object classifier scores, together with a semantic embedding learned from social media, are as well a suitable representation for scene recognition without examples.

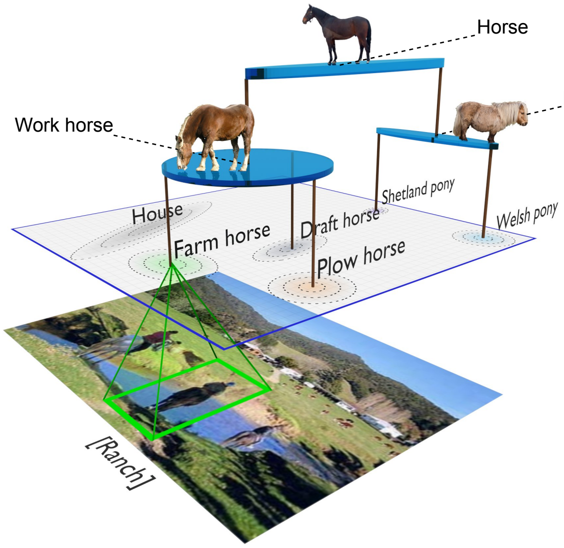

Figure 1: We investigate four approaches of object pooling for scene recognition without examples and rely on a freely available large-scale object corpus, social media text and an ontology.

Supervised image recognition is currently dominated by deep learning. Convolutional neural networks, the deeper the better [23, 8], learn the optimal image representation by tuning its millions of network parameters from huge amounts of labeled data, such as ImageNet [6] (for objects) or Places2 [30] (for scenes). In [29], Zhou *et al.* show evidence that objects emerge in deep nets learned from scene images. Motivating us to rely on a representation of objects to recognize scenes. By doing so we don't need any scene example. Our object classifiers correspond to the output of the last softmax layer of a very deep convolutional neural network trained on ImageNet [23]. These object classifiers have a lingual correspondence derived from nouns in WordNet [16], making them well suited for recognition without examples using a semantic word embedding, *e.g.* [15, 18, 22], as prior knowledge. Rather than simply using the response of all available object classifiers, we pose the question: *What objects to pool when representing scenes?*.

ICMR'16, June 06-09, 2016, New York, NY, USA
© 2016 ACM. ISBN 978-1-4503-4359-6/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2911996.2912007

In [9] Jain *et al.* evaluate two methods for pooling object classifiers to recognize unseen actions in video; one based on the semantics of the class and the other based on the appearance of the test imagery. We adopt and adapt their methods for scene recognition in images. In addition, we introduce a hierarchical pooling taking into account the ontological relations of objects as defined in WordNet, see Figure 1. Finally, we also introduce a pooling that considers the spatial extent of object classifiers by exploiting a recent box proposal [32]. This spatial pooling allows to answer complex search queries like "finding beach scenes with a human on the right of the image". We evaluate our proposals on two scene datasets SUNAttributes [19] and Places2 [30].

2. RELATED WORK

The goal of recognition without examples, as introduced by Lampert *et al.* [12], is to recognize a set of unseen classes Z using some set of training classes Y, which do not overlap with the test classes ($Z \cap Y = \emptyset$). To allow for knowledge transfer in between, usually a set of attributes A is defined, which semantically relates to both the training classes Y and the unseen test classes Z. The attributes come with annotated examples X, forming a training dataset $D = \{X, Y, A\}$, from which attribute models are learned, and a human defined class-to-attribute mapping is used to transfer knowledge. In this paper, instead of manually and carefully choosing training classes Y and attributes A to well fit the characteristics of the unseen scene classes, like scene attributes [19], we investigate pooling off-the-shelf object classifiers derived from ImageNet [21], with 15K diverse object categories, to recognize scenes without examples.

Instead of relying on attributes, Norouzi [18] proposed to use a semantic embedding space to allow for knowledge transfer between the training classes Y and unseen classes Z. The semantic embedding space is trained to minimize the distance between words with similar semantics. We use the word2vec semantic embedding [15]. This is a single hidden layer neural network, trained to predict the surrounding words given an input word. The semantic embedding representation is a d-dimensional vector derived from the activation values of the hidden layer. This ensures that, the similarity between two words in the word2vec space will be close only if the words are found in similar semantic context. Cappallo *et al.* [4], show that a word2vec model trained using the textual data accompanied with images, works better than a model trained on Wikipedia, probably because of the more visual nature of these descriptions. We follow their approach and train on a large social media dataset [24], containing the surrounding text (tags, captions, *etc.*) of 100 million Flickr images. Although semantic embeddings have been used before for zero-shot learning [9, 18, 4] it has not yet been explored for objects and scenes.

To represent scene and object class names composed of multiple words, we aggregate them using Fisher vectors [5]. The Fisher vector characterizes each word by its deviation in the semantic embedding space *w.r.t.* a Gaussian mixture model. This has shown to perform better than simple word averaging [5], also in the context of recognizing actions and events in a zero-shot setting [9]. We use the Fisher vector for computing similarities between objects and scenes names.

The semantic structure defined by an ontology has been explored for many retrieval and classification problems. Fergus *et al.* [7] leverage the WordNet hierarchy to define semantic distance between any two categories, and use it to share labels for training classifiers. Zhu *et al.* [31] use the WordNet hierarchy to complete imprecise and incomplete social media tags by, transferring tag examples from child to parent nodes. When a tag occurs rarely in social media, Kordumova *et al.* use semantics from the WordNet ontology to enrich training data [11]. In [26] Vreeswijk *et al.*, compare training a classifier of a parent node with an union of their constituting child nodes. Different from these approaches, we investigate the WordNet hierarchy to steer the object representation in a semantic embedding for recognizing scenes without examples.

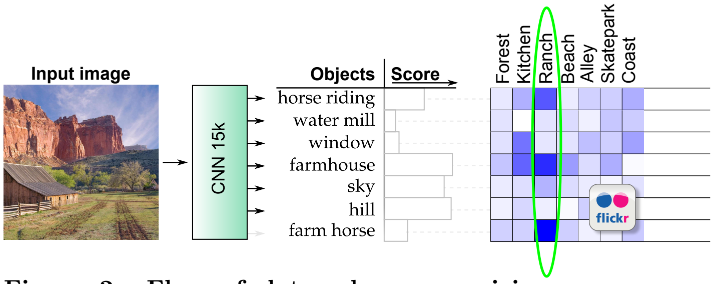

Figure 2: Flow of data when recognizing scenes without examples using object pooling. For an input test image we calculate its object prediction scores from a very deep convolutional neural network, learned to classify 15K objects from ImageNet. We calculate a knowledge transfer matrix between objects and scenes from a semantic similarity embedding learned using text of Flickr images. The scene with the highest score is assigned to the input test image.

Object proposals [2, 25, 32] suggest the most likely positions in an image to contain an object. Besides being successful in object detection, they have also proven to obtain better results for image retrieval and classification [27]. Inspired by [27], we use (unsupervised) object proposals with object classifiers trained on the whole image, for the task of recognizing scenes. We prefer object proposals over object detectors, for two reasons: detection increases the complexity of our approach, and limits the number of object categories to at most 1K, for which bounding box annotations are available. Encoding the spatial layout of objects allows us to answer complex queries, where instead of just querying with the scene name, we include object names and rough estimates of their position, like "finding castle scenes with grass on the bottom of the image". Different from other complex query approaches [28, 17, 14, 3], which all emphasize on word combinations, we propose a complex query with a scene, an object *and* its position. Next we describe our approach.

3. POOLING OBJECTS FOR SCENES

Our goal is to recognize unseen scenes Z, using a set of off-the-shelf object classifiers Y, which do not overlap with the scene classes, *i.e.* ($Z \cap Y = \emptyset$), and a semantic embedding space S. Similar to [18] we use a convex combination of object scores and semantic similarities to assign a class z to a test image v:

$$C(v) = \operatorname*{argmax}_{z \in Z} \sum_{y \in Y} s(z, y)\; p(y|v), \qquad (1)$$

where $p(y|v)$ is the object prediction score for class y given test example v and $s(z, y)$ denotes the semantic similarity between unseen class z and object class y.

The object prediction $p(y|v)$ is the last network layer (after softmax normalization) of a very deep convolutional neural network [23]. We use 15K object categories from the ImageNet dataset, for which at least 200 images are available for training, similar to [9, 4].

The semantic similarity $s(z, y)$, is the cosine similarity between a scene z and an object y:

$$s(z, y) = cos(s(y), s(z)) = s(y)^\top s(z), \tag{2}$$

where $s(\cdot)$ denotes the d-dimensional Fisher vector encoding of a word2vec vector. We visualize the flow of data for recognizing a scene in an input image in Figure 2.

When considering off-the-shelf object classifiers, we expect that not all objects will contribute equally in describing scenes. ImageNet follows a hierarchical structure, derived from WordNet nouns, where objects like *animal, vehicle, plant, food* appear high in the hierarchy, followed by object categories like types of animals *cat, dog, bird, horse*, and fine-grained objects in the lowest level of the hierarchy, for example types of horses *tarpan, hackney, clydesdale*. However, it is not clear how specific the objects need to be in order to better retrieve scenes. We also expect that for different scenes, different objects should matter. To recognize a *beach* scene objects like *sand, ocean, sky* would be beneficial, whereas *bed, apple, church* would have a negligible or no influence. We describe and investigate four ways of pooling objects for scene recognition.

Semantic Pooling. This method considers the semantically closest objects for each scene class. Similarly as in [9], it does so by first ranking the objects based on their semantic similarity scores to a scene, calculated with Equation 2. For each scene $z \in Z$, we then select the closest objects, forming a subset of Y_z from the complete objects set Y:

$$Y_z = \{y \in Y \mid s_{yz} > t_s\}, \tag{3}$$

where t_s is a hyper parameter of selecting the objects with larger scene similarity, or the similarity value of the m-th closest object, so that we select the top m objects. In this case, for a test image, scene scores will be computed as in Equation 1, and only changing the summation over the set Y to the subset Y_z.

Appearance Pooling. A test image v is represented with probability values for each object category $p_{vy}, \forall y \in Y$. Each probability p_{vy} approximates the presence of an object of class y in the image. We follow [9, 18] and we sample only the prominent objects present in an image, resulting in a subset of objects:

$$Y_v = \{y \in Y \mid p_{vy} > t_p\}, \tag{4}$$

where t_p is a hyper parameter which regulates selecting objects with minimum presence probability in the image. To select the top m present objects, t_p takes the value of the $m - th$ ranked object. In this case, the scene scores of a test image will be computed as in Equation 1, with the only difference in the summation over the objects set, from Y to a subset Y_v. Doing so, we avoid summing over many unrelated object classes for an image with low probability scores, which do not appear in the image and will not contribute to the scene recognition.

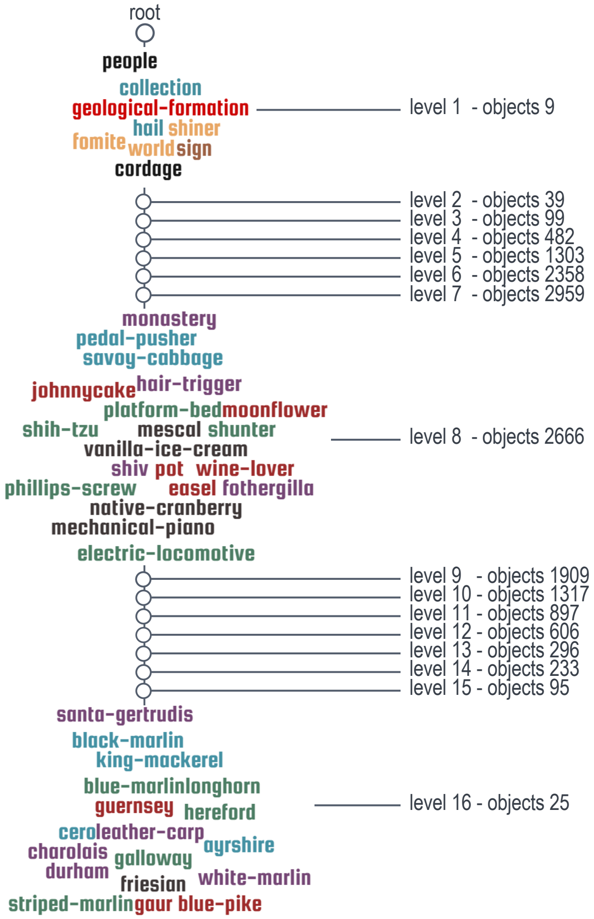

Figure 3: Flattened hierarchy tree of the ImageNet objects, indicating the number of objects per level. The first level contains very general objects, becoming more specific as the levels grow, narrowing down to fine-grained objects. We show word clouds of typical object names in the first, middle and last level.

Hierarchy Pooling. The hierarchical structure of objects portrays parent-child relationship of object categories, where the children objects are a subcategory of their parent class. The leaf nodes represent the most fine-grained object classes of their tree branch. We question how important are objects from different hierarchy levels for recognizing a scene. One would most naturally expect to find a *tree* in a *forest*, and a *sofa* in a *living room*, and to discriminate these two scenes the specificity of objects like *tree, sky, sofa, tv* would be sufficient, without going further into knowing the type of a tree. To discriminate a *bamboo forest* from a *broadleaf forests*, specific types of tree objects might matter.

The hierarchical structure of objects, as defined in the WordNet ontology, forms a tree. In Figure 3 we show a flattened tree version of a WordNet subspace, using only the nouns from ImageNet object categories. We denote as $L = \{L_1, ...L_k\}$ the set of all $k = 16$ tree levels from the flattened tree. Each level $L_i, i \in \{1, ..., k\}$ is a subset from all objects Y with depth i:

$$L_i = \{y \in Y \mid depth(y) = i\}. \tag{5}$$

For a new test example, we investigate how accurate will the objects from different levels predict its scene class. We use objects from each level separately by changing the summation over the set Y in Equation 1, to objects from only one level set L_i at a time.

Position Pooling. We employ [32] to generate bounding box coordinates over the image $[x_{min}, y_{min}, x_{max}, y_{max}]$, most likely to contain an object. For a test image, we exploit whether being more precise by object scoring the bounding box regions in an image, helps in recognizing the scene. Same as for the whole image, we use the last network layer with softmax normalization of [23], to compute object prediction scores on image regions. For m bounding boxes $\{b_i\}_{i=1}^{m}$, generated with an object proposal method, the object scores are $p(y|b_i), y \in Y$. There are multiple ways to pool object classifiers when objects scores of image regions are known. We employ two simple approaches, max pooling and average pooling. For max pooling, we consider object classifiers with the largest prediction scores from all generated positions in an image:

$$Y_b = \{y \in Y \mid max(p(y|b_1), ..., p(y|b_m)) > t_b\}, \qquad (6)$$

where t_b is a hyper parameter, selecting the top scored objects among all image regions. For average pooling we consider object classifiers with average prediction scores of all bounding box positions in an image, larger then a value parameter t_a:

$$Y_a = \{y \in Y \mid avg(p(y|b_1), ..., p(y|b_k)) > t_a\}. \qquad (7)$$

The position of objects in scene images allows for new scene retrieval scenarios. With a (textual) scene query, one can also specify what object(s) to be present, as well as where in the scene the object(s) should be. We choose three adjectives to approximate the *where* for horizontal position, *(left, center, right)*, and three for vertical position *(bottom, middle, top)*. Thus, an example query would be: *beach* with *human* on the *right.* We first find all images where the query scene was recognized, and then we rerank them based on the maximum prediction response for that object. If the query has specified also the object position, we consider object responses only from bounding boxes which center falls in that scene area. We compute the horizontal and vertical center of bounding boxes as $b_h = (x_{max} - x_{min})/2 + x_{min}$, and $b_v = (y_{max} - y_{min})/2 + y_{min}$. If I_h, I_w are the image width, height, and $t_w = I_w/3$, $t_h = I_h/3$, is one third of the image horizontal and vertical area, for (*left, center, right*) we consider the boxes which center falls in the intervals $(0, t_w), (t_w, 2t_w), (2t_w, I_h)$, and for (*bottom, middle, top*) we consider bounding boxes with center in the intervals $(0, t_h), (t_h, 2t_h), (2t_h, I_h)$ respectively.

We are now ready to evaluate our proposal quantitatively and qualitatively.

4. EXPERIMENTS

4.1 Datasets

SUNAttributes. The SUNAttributes dataset has 14,340 images hierarchically grouped in 3 levels [19]. Level 3 has 717 fine-grained scenes like *airplane cabin, athletic field outdoor, cybercafe.* Level 2 groups the fine-grained scenes in 16 categories like *shopping and dining* or *water, ice, snow*, and level 1 has three general categories of scenes *indoor, outdoor natural* and *outdoor man made.* Each of the 717 scenes in level 3 has 20 images per scene, which we use as test data. We scale all images to same dimension of 256x256 pixels.

The dataset also comes with 102 attribute annotations. When using attributes [12], the dataset is split into two parts of 10 random splits, where 90% of the images are used for attribute learning, and the rest for testing. We follow the same approach, and create 10 random splits. Because the data is split into disjoint train (90%), and test classes (10%), 71 classes of the dataset are present at test time for each random split. For our approach we only use the images from the test splits, since we do not need any scene images for learning. Same as in [12], we report results in accuracy. For scene level one and two, a prediction is considered correct if the ground truth class and the predicted class paths run through a common node in that level.

Places2. The Places2 dataset [30] contains more than 10 million images comprising 401 unique scene categories. The scene categories vary from indoor scenes like *bazaar indoor, cafeteria, toyshop*, and outdoor scenes like *bazaar outdoor, creek, desert sand.* The dataset features up to 30,000 training images and 50 validation images per class, scaled to 256x256 pixels. Since our goal is to retrieve scenes without any training example, we do not use the training set, and evaluate our approach on the validation set. This dataset was first released in the scene recognition subtask of the ImageNet 2015 challenge [21], and to the best of our knowledge, we are the first to report scene recognition results without examples on this dataset. As evaluation metric we report top-5 accuracy, as set by the challenge.

4.2 Implementation details

Object classifiers. We train an off-the-shelf deep convolutional network [23], with 22 layers, on all ImageNet categories having more than 200 images available for training. As object classifier scores we use the last network layer with softmax normalization. After removing 304 objects having an exact overlap in name with the scene classes, we end up with 14,989 object classifiers.

Object hierarchy. We use the parent-child noun relationships provided by WordNet [16] to construct a hierarchical tree[1]. The WordNet tree has 20 depth levels, from which we create a flattened subset version, where we take only the nouns that appear as object classes in ImageNet, and keep their depth position as in WordNet. Besides the *sun* object which appears in the second level, the first four levels do not contain objects from ImageNet. Therefore, we ignore them, and start counting from the fifth as a first level, resulting in a 16-level flattened tree, see again Figure 3, with ImageNet objects only.

Objects positions. We generate object positions with EdgeBoxes [32]. As we observe a stable recall, we use the top 25% of bounding boxes after ranking by their objectness score. The number of bounding boxes varies per image, between 2 and 1025 for SUNAttributes, and between 1 and 1077 for Places2. We also employ the common non-maximal suppression on the sorted boxes with an overlap threshold of 0.5 [32, 2, 25].

Semantic embedding. We train a 500-dimensional skipgram word2vec model using the metadata of the YFCC100M dataset [24]. This dataset has 100 million Flickr images accompanied with titles, descriptions and tags. We encode the word2vec vectors with Fisher vectors, following the setting

[1] Visualisation: *isis-data.science.uva.nl/svetlana/WordNetTree*

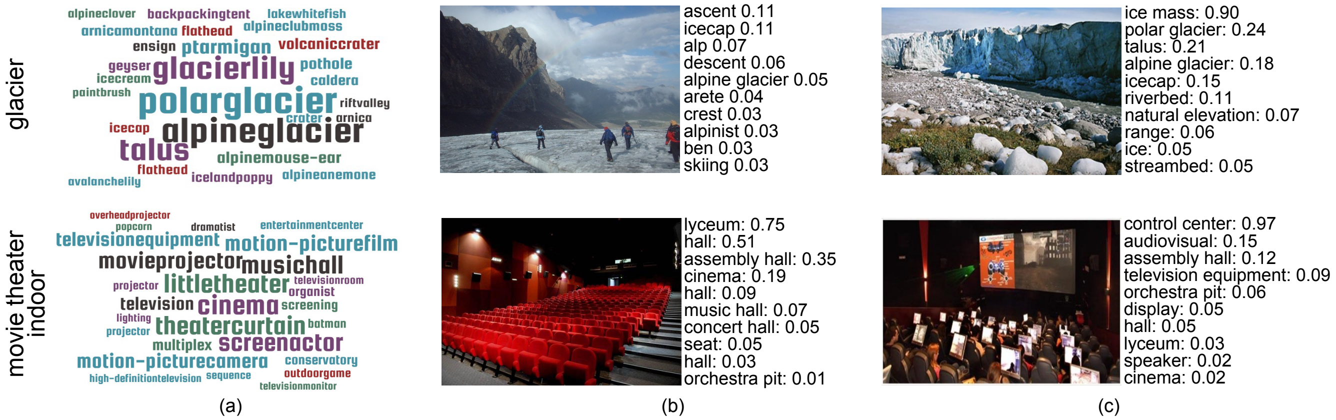

Figure 4: Pooled objects for two scenes, (a) word clouds of semantically pooled closest objects, (b) and (c) image examples with their top-scoring object predictions. With appearance pooling, the correct scene was assigned to the images in (b), whereas in (c) the *glacier* image was misclassified as *ice shelf* scene, and the *movie theater indoor* image as a *steel mill* scene. This happens when the top predicted objects in an image are closer in the semantic embedding space to the misclassified scenes. Overall, all pooled objects look reasonable.

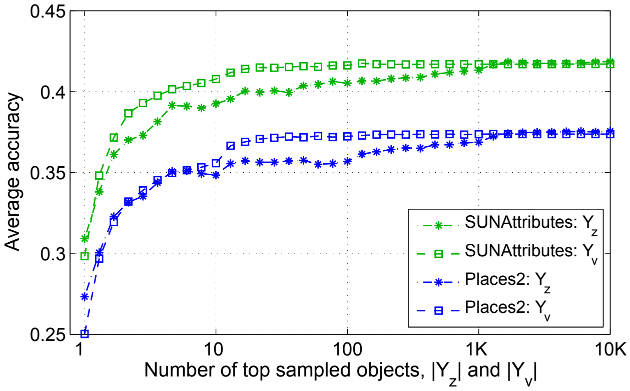

Figure 5: Impact of the number of semantically closest objects per scene $|Y_z|$, and most prominent objects per image $|Y_v|$, for scene recognition with no examples on the SUNAttributes and Places2 datasets.

of [9]. We learn k=2 Gaussian Mixture components, apply PCA to reduce the dimensionality by a factor of 2, and use only the partial derivatives *w.r.t* the mean. In this way, the encoded word2vec Fisher vector keeps the dimensionality, $d = 500$, as the original word2vec.

4.3 Semantic pooling and appearance pooling

We first investigate semantic pooling and appearance pooling, where the first selects the semantically closest objects to scenes Y_z, and the second the most prominent objects per image Y_v. We show pooled object examples for two scenes in Figure 4. We vary the threshold parameters, t_z and t_v, in a way that we select the top m ranked objects, where $m = |Y_z|$ or $m = |Y_v|$. In both cases, we see similar pattern in the results, see Figure 5.

When only selecting the single most semantically similar object per scene, $|Y_z| = 1$, the average accuracy for SUNAttributes is 30.91%, and for Places2 is 27.32%. When compared to random accuracy, 1.41% and 0.25% respectively, the results indicate that even one object per scene can help in discriminating scenes. As we sample more objects the accuracy improves. Up to 10 objects it grows rapidly, followed by a slow progress, which saturates after about 1K sampled objects. If we select the single most prominent object in an image, $|Y_v| = 1$, the accuracy for SUNAttributes is 29.82%, and for Places2 is 25.02%. Thus, when one object is selected, the most semantically similar one is more informative than the most prominent one. However, as we sample more prominent objects, the accuracy comes close to the maximum after only 100 objects are selected.

Object classifiers recognize scenes without examples reasonably. We expect that as object classifiers improve further [8], the more accurate the scene recognition without examples will become. From now on we use the 100 most prominent objects in an image to recognize its scene.

4.4 Hierarchy pooling

Here we investigate what is the discriminative power of objects from different hierarchy levels when recognizing scenes without examples. In Figure 6 (a), we show results when we pool objects from each level individually. Interestingly, the top- and low-level objects show lowest accuracy. The top-levels contain mostly general object categories, and the low-levels have fine-grained object categories, see Figure 3. From the results we conclude that general and fine-grained object categories from the top- and low-levels of the ImageNet hierarchy, are not discriminative for recognizing scenes. The objects from level 6 and 7 show best performance. If we use objects from only these two levels, in total 5317, or one third of all objects, we already achieve an accuracy of 32.5% for Places2, and 34.5% for SUNAttributes.

The number of objects in each level varies. In our experiments we use the top 100 most prominent objects in an image. Level 1,2,3,15,16 contain less than 100 objects, thus, we believe that the number of objects present might also influence the scene recognition. Therefore, we perform two additional experiments where we investigate what happens

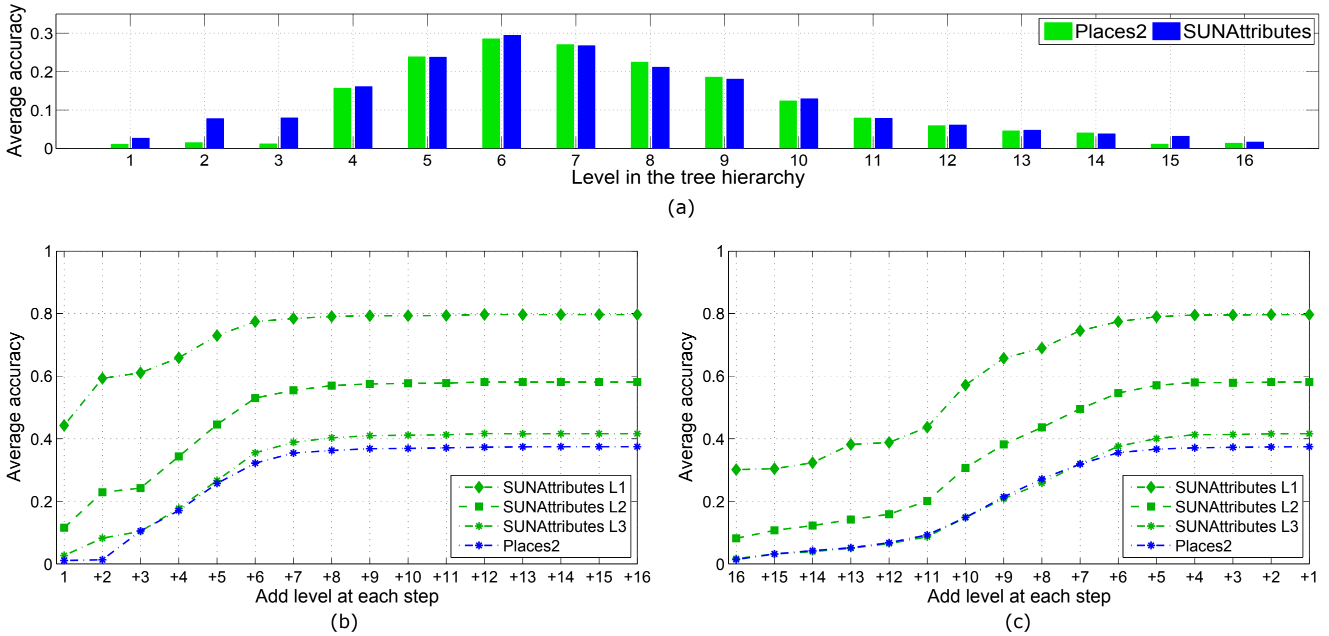

Figure 6: Scene recognition without examples using a representation containing classification scores from ImageNet objects, where we vary the objects in the representation using hierarchy levels. In (a), each hierarchy level forms one object representation, and we show results on Places2, and Level3 SUNAttribute scenes. Object classifiers from the middle levels are most discriminative, whereas the first levels (general objects), and the last levels (fine-grained objects), are not discriminative for recognizing scenes. In (b), in each step objects from the next level of the hierarchy are added, and in (c) in each step objects from the previous level of the hierarchy are added. Again we conclude that the objects from the middle level contribute the most for discriminating scenes.

if in each step we add objects from the next level, and see how much each level contributes. We first start from level one, and continue adding up objects from the next levels until all objects are used. We show the results in Figure 6 (b), for Places2 scene categories, as well as for all three scene levels of SUNAttributes. We see a similar pattern for all scene categories. The average recognition accuracy grows substantially up to level eight. From the ninth level on, there is minor growth in the accuracy. This again confirms that the lower levels do not add extra information which contributes for the recognition of scenes. As we go the other way around, we start from the lowest level and add up objects from the previous levels, see Figure 6 (c), the results confirm the previous findings. The fine-grained object classifiers have a small contribution for discriminating scenes, middle level objects help the most, and the general objects from the top levels do not contribute in recognizing scenes.

4.5 Position pooling

When we sample the top 100 objects by average pooling over image positions, as generated with EdgeBoxes, we get slightly better results than max pooling, 32.57% vs 32.31% for SUNAttributes, and 31.87% vs 31.35% for Places2. When we merge pooled objects from image positions, for $t_a = 0.3$, with objects pooled from the whole image, the scores do not change much. For SUNAttributes it changes from 41.62% to 41.46%, and for Places2 from 37.23% to 37.63%, for appearance and apperance+position pooling respectively. When we look into individual scene accuracy, in some cases position pooling helps, whereas in others, appearance pooling from the whole image is enough. For example the objects *milldam* and *waterspout* pooled from positions, helped to correctly classify a *dam* scene which was misclassified as *river*. In another example, a *field cultivated* image was misclassified into a *pasture* scene, when the objects *geyser, waterspout, parhelion, cloud, azure,...* were pooled from image positions. We did not find a general rule on when position pooling helps over the whole image, and since it requires more processing, we recommend to use appearance pooling from the whole image only.

A more interesting benefit of knowing the object positions is that they allow retrieval with complex queries. In Figure 7 we show qualitative results for four different queries, two with scene+object and two with scene+object+position. We show the top and bottom retrieved images for each query. For the top retrieved images we also draw the bounding box which contributes the most for the object. We do not show bounding boxes on the lowest ranked images, since the object is not present, the object scores are low and do not make sense. Interestingly, the top retrieved scenes contain the object we query with. When an object position is specified, like *right* or *bottom* as in the last two queries in Figure 7, the most contributing bounding box of the object in the top retrieved images, comes from the *right* or *bottom* area in the scene. The bottom ranked images, in all cases are missing the object we query with from the scene. Thus, we observe that knowing the object positions in scene images can indeed be used for retrieving images with complex queries of scenes, objects and their spatial extent.

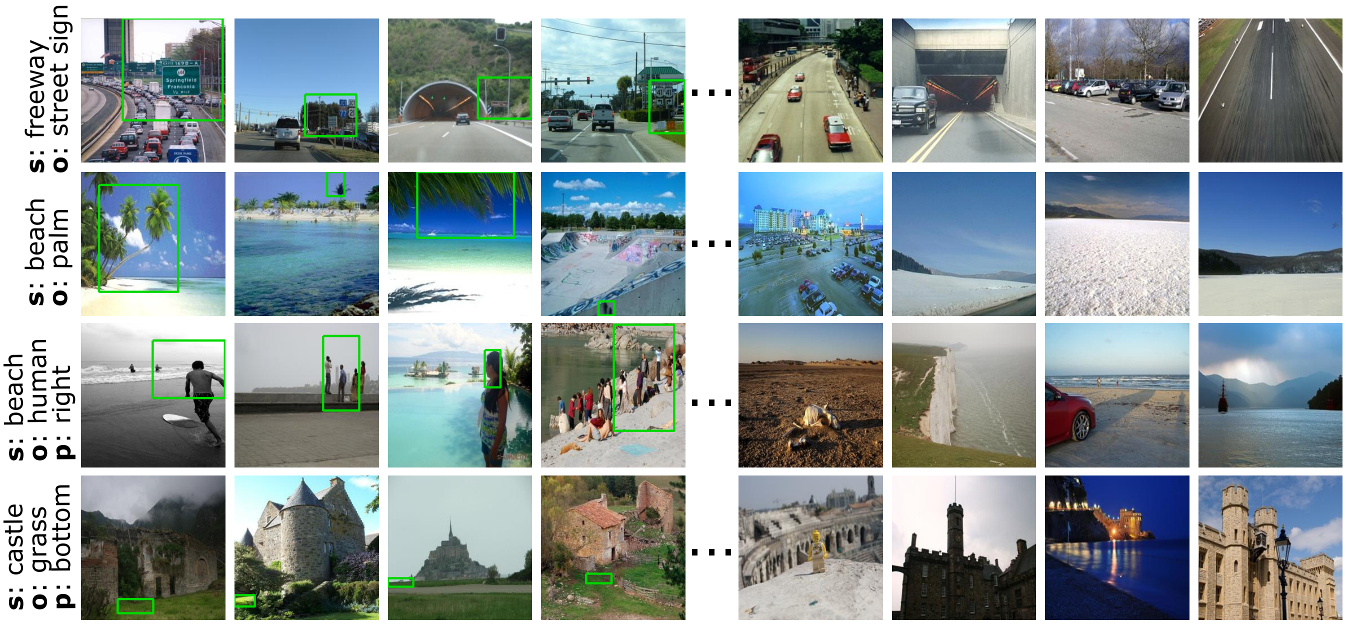

Figure 7: Top and bottom ranked images when as query we use scenes+object (first two rows), and scenes+object+position (last two rows). Over the top retrieved images we draw the bounding box which contributes the most for finding the object. We do not show bounding boxes on the lowest ranked images, since the objects are not present and the object scores are low, so it does not make sense. Interestingly, the top ranked images are quite relevant to the query, and when the object position is specified, the filtered bounding boxes from only that area in the image indeed help. The bottom ranked images, in all cases are missing the object we query with from the scene.

4.6 Comparison with attributes

Finally we compare pooling off-the-shelf objects with attribute representations for recognizing scenes without examples. We report all results in Table 1.

Our object pooling performs better than the direct attribute prediction (DAP) reported by Lampert *et al.* [12], 41.63% vs 22.20%, on the SUNAttributes dataset. Since their results were calculated using GIST and HOG features, we repeat their experiment with our convolutional neural network features, for fair comparison. We take features from the layer before the last one, of the same network we use throughout all our experiments. We L_2 normalize the features, and for each attribute we learn a linear binary classifier with libsvm and probability outputs. In this setting, the DAP approach improves from 22.20% to 34.78% on SUNAttributes. Still the result does not go over our accuracy with object pooling.

We can not report results with DAP on Places2, since this approach requires attribute-to-scene mapping, which is not available on this dataset. Instead, we investigate how attributes work with a semantic embedding [18], when attribute annotations are missing. For all 102 attribute names of SUNAttributes we find their match among the 15K objects. 22 attributes have an exact name match, and for 80 we use the closest object, like *sailing/boating* with *sailboat*, or *sunbathing* with *sunbather*. As expected, the attributes with semantic embedding perform lower then a DAP setting, with an accuracy of 10.07% on SUNAttributes (71 classes), and 1.27% on the more challenging Places2 dataset (401 classes). The results are comprehensible, since there are only 102 attributes, and in our setting we use 100 objects pooled from a set of 15K object classifiers.

We also consider an upper limit comparison with a supervised alternative. The best reported number on the 2015 Places2 challenge [1] with supervised deep learning is 83.12%, and with our method, where no scene examples are used, we achieve an accuracy of 42.72%. While encouraging, there is a lot of space left for further improvement.

We conclude, rather than using attributes scenes are better recognized without examples by pooling relevant objects from the image representation, when a large corpus of object classifiers is available, and with a semantic embedding learned from a large social media corpus to guide the knowledge transfer between objects and scenes.

5. CONCLUSIONS

In this paper we recognize scenes without examples by using off-the-shelf object classifiers from a large collection of 15K object classes. We pose the question *what objects to pool when representing scenes?*. Throughout extensive analysis by experiments on two large-scale scene datasets we conclude that object classifiers from a large objects collection are suited for scene recognition without examples. From the four pooling methods considered we recommend object pooling by appearance, as it requires only 100 objects per image. General and fine-grained object categories, from the top and bottom level of the WordNet hierarchy, do not contribute much for scene recognition and can be avoided in the image representation. When the object classifiers are run on object proposals instead of the full image, our proposal allows for complex queries including position

	SUN Attributes	**Places2**
Random	1.41	0.25
Attributes DAP [12]	22.20 ± 1.60	n.a.
Using deep learning:*		
Attributes DAP [12]	34.78 ± 2.54	n.a.
Attributes-Embedding [18]	10.07 ± 2.22	1.27
Object pooling	**41.63 ± 3.16**	**37.46**

Table 1: Comparison of object pooling by appearance vs attributes for recognizing scenes without examples. Between the lines marked with deep learning*, we show results which are not available in the cited papers, but computed using their approach with our deep learning setting. Object pooling recognizes scenes better then attribute representations on both datasets.

quantifiers, like “finding beach scenes with a human on the right of the image”. Finally, object pooling outperform attribute representation for scene recognition without examples, with the additional benefit of learning the knowledge transfer from social multimedia, rather than manually specifying the scene-to-attribute mappings.

Acknowledgments. This research is supported by the STW STORY project, the NWO VENI What&Where project and the Dutch national program COMMIT. Thanks to Morris Franken for helping with the visualisations.

6. REFERENCES

[1] http://places2.csail.mit.edu/results2015.html.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012.

[3] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *MM*, 2013.

[4] S. Cappallo, T. Mensink, and C. Snoek. Image2emoji: Zero-shot emoji prediction for visual media. In *MM*, 2015.

[5] S. Clinchant and F. Perronnin. Textual similarity with a bag-of-embedded-words model. In *ICTIR*, 2013.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[7] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*, 2010.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016.

[9] M. Jain, J. van Gemert, T. Mensink, and C. Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015.

[10] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014.

[11] S. Kordumova, J. van Gemert, and C. Snoek. Exploring the long tail of social media tags. In *MMM*, 2016.

[12] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 36(3):453–465, 2013.

[13] H. Li, D. Li, and X. Luo. Bap: Bimodal attribute prediction for zero-shot image categorization. In *MM*, 2014.

[14] X. Li, C. Snoek, M. Worring, and A. Smeulders. Harvesting social images for bi-concept search. *TMM*, 14(4):1091–1104, 2012.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[16] G. A. Miller. Wordnet: A lexical database for english. In *CACM*, 1995.

[17] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua. Harvesting visual concepts for image search with complex queries. In *MM*, 2012.

[18] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.

[19] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014.

[20] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[22] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for unsupervised word embeddings. In *EMNLP*, 2015.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[24] B. Thomee, B. Elizalde, D. Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *CACM*, 59(2):64–73, 2016.

[25] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[26] D. Vreeswijk, K. van de Sande, C. Snoek, and A. Smeulders. All vehicles are cars: Subclass preferences in container concepts. In *ICMR*, 2012.

[27] L. Xie, Q. Tian, R. Hong, and B. Zhang. Image classification and retrieval are one. In *ICMR*, 2015.

[28] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. Zhou, and T.-S. Chua. Learning concept bundles for video search with complex queries. In *MM*, 2011.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places2: A large-scale database for scene understanding. In *arXiv*, 2015.

[31] S. Zhu, C.-W. Ngo, and Y.-G. Jiang. Sampling and ontologically pooling web images for visual concept learning. *TMM*, 14(4):1068–1078, 2012.

[32] C. L. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, 2014.