Deriving Visual Semantics from Spatial Context: An Adaptation of LSA and Word2Vec to generate Object and Scene Embeddings from Images

Matthias S. Treder[1,*], Juan Mayor-Torres[2], Christoph Teufel[2]

1 School of Computer Science, Cardiff University, United Kingdom
2 Cardiff University Brain Research Imaging Centre (CUBRIC), School of Psychology, Cardiff University, United Kingdom

*** trederm@cardiff.ac.uk**

Embeddings are an important tool for the representation of word meaning. Their effectiveness rests on the distributional hypothesis: words that occur in the same context carry similar semantic information. Here, we adapt this approach to index visual semantics in images of scenes. To this end, we formulate a distributional hypothesis for objects and scenes: Scenes that contain the same objects (object context) are semantically related. Similarly, objects that appear in the same spatial context (within a scene or subregions of a scene) are semantically related. We develop two approaches for learning object and scene embeddings from annotated images. In the first approach, we adapt LSA and Word2vec's Skipgram and CBOW models to generate two sets of embeddings from object co-occurrences in whole images, one for objects and one for scenes. The representational space spanned by these embeddings suggests that the distributional hypothesis holds for images. In an initial application of this approach, we show that our image-based embeddings improve scene classification models such as ResNet18 and VGG-11 (3.72% improvement on Top5 accuracy, 4.56% improvement on Top1 accuracy). In the second approach, rather than analyzing whole images of scenes, we focus on co-occurrences of objects within subregions of an image. We illustrate that this method yields a sensible hierarchical decomposition of a scene into collections of semantically related objects. Overall, these results suggest that object and scene embeddings from object co-occurrences and spatial context yield semantically meaningful representations as well as computational improvements for downstream applications such as scene classification.

1 Introduction

Categorizing visual scenes quickly and robustly is critical for navigating an environment, localizing targets, and deciding how to act in a given context, both for humans [15] and robots [13, 22]. Consequently, understanding scene perception is an important research topic in both biological and machine vision. The majority of computational approaches emphasize the importance of global summary statistics [19], or high-level features [24] in scene categorization. Human observers, however, experience a scene as being composed of multiple objects, and the deeper meaning of a scene is determined by the physical and semantic relationships between these objects as well as their relationship to the scene gist [15].

In general, visual scenes such as a room, a beach, or a parking lot are well-defined spatial locations that typically contain a large number of items arranged according to semantic (and syntactic) regularities. Importantly, scenes not only contain objects, which are defined as spatially distinct entities that can be moved or manipulated (e.g., a bar of soap, a stone, or a statue); they also incorporate "stuff", a term that refers to amorphous areas that do not have these properties (e.g., walls, floor, a river, or the sky) [33, 34]. In all of our analyses, objects and stuff are treated identically, and are assumed to be constituent parts of scenes. For brevity, we will therefore use the label *objects* as an umbrella term to refer to both proper objects and stuff.

The purpose of this paper is to introduce vector representations of objects and scene categories and show how they might be useful for scene analysis and understanding. Importantly, our approach is not dependent on a text corpus, and we do not use image captions or complex annotations. Instead, we only rely on scene labels, object labels and the spatial organization of objects within a scene to develop two approaches for learning object and scene embeddings from images. In the first approach, the embeddings are based on co-occurrences of objects within scenes. In the second approach, we zoom into images and define as local spatial context the co-occurrence of objects in a small spatial window.

In order to be useful, we aim for these representations to exhibit a number of properties critical for scene analysis and understanding: first, the embeddings should capture object-object relationships. For instance, we would like the embeddings to reflect the fact that a bar of soap and a towel are semantically closely related because they tend to co-occur in bathrooms and are used in the same functional contexts. Second, we hope to derive object-scene relationships such as a bar of soap being more closely related to a bathroom than to a living room. Third, we would like the embeddings to encode scene-scene relationships such that bathroom and bedroom are more closely related to each other than either of them to a football field. Finally, we aim for representations that are reliable at different hierarchical levels and can decompose scenes into sub-regions of semantically related objects.

In order to generate these image-based object and scene embeddings, we build on word embeddings, a technique that has been successfully employed to represent semantic relationships in natural language [11]. The motivation for word embeddings rests on the distributional hypothesis: words that occur in the same context, for instance, within the same sentence, tend to carry a similar meaning. Based on this assumption, the statistics of co-occurrences of words can be used as a proxy for semantic relatedness.

In analogy to this idea, one can formulate the *distributional hypothesis for objects and scenes*: scenes that contain the same objects (object context) are semantically related. Similarly, objects that appear in the same spatial context (other objects within the scene or within subregions of it) are semantically related. Note that, in analogy to the relationships between words in word embeddings, the relationships between objects and scenes are not functionally defined (e.g., toilet paper and toothbrush have very different functions) but purely governed by spatial proximity. Yet, empirical evidence supports the hypothesis that spatial relations mirror semantic relations. For instance, when Convolutional Neural Networks (CNNs) are trained to classify scenes, object detectors emerge as an intermediate representation of the network, suggesting that objects are informative with respect to scene category [31, 33].

1.1 Related work

Embeddings have been useful in domains other than text data. For instance, they are used to represent genes in gene-expression data [5], computer network log data [35], and graph-based data such as molecules [14].

In computer vision, several studies attempt to use image information in order to enhance word embeddings learnt from text corpora. For instance, [16] add a feature vector derived from a VGG-16 CNN to a neural network that learns word embeddings from the respective image captions. Instead of using fixed image vectors, [1] use a model based on a Variational Autoencoder that learns a latent visual representation along with the word embedding. [8] learn multi-modal embeddings by combining images with spoken rather than textual image captions. [10] use embeddings initialized with Word2vec to predict annotations of abstract scenes, yielding modified embeddings that might better encode visual semantic relationships. Using a GloVe-based embedding model, [7] train scene embeddings from co-occurrences of objects within images.

A number of papers use features derived from objects in a scene to enhance scene classification. [12] propose an object filter bank to derive a set of image descriptors that serves as input for simple off-the-shelf classifiers such as SVM. [13] use semantic information from an image segmentation model as a regularizer for the first layers of an AlexNet CNN. A study by [3] is closest to ours. The authors propose to train object and scene embeddings from co-ocurrences within images. They then refine the prediction of a scene classifier using the predicted scene embedding.

Our approach, too, relies on co-occurrences of objects. However, it differs from [3] in several important ways and offers the following additional contributions: (i) whereas previous papers focused on downstream applications such as scene classification only, we also investigate whether embeddings are semantically meaningful. Second, (ii) we adapt and compare a variety of models (LSA, Skipgram, CBOW). And third, (iii) we extend a state-of-the-art scene classification architecture with LSA embeddings. Unlike previous approaches, we directly fuse image features with object-based embeddings in the last CNN layer. Finally, (iv) we go beyond a 'bag-of-objects' approach and model local spatial context by considering the spatial proximity of objects within images.

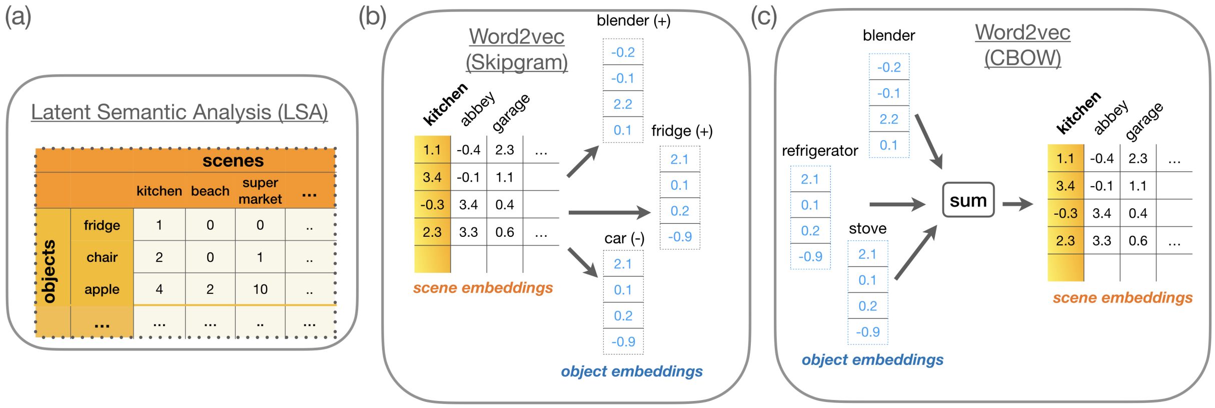

Figure 1. Object and scene embeddings using LSA and Word2vec. (a) LSA operates on an object-scene occurrence matrix. (b) Single forward pass through a Skipgram model. A scene embedding (kitchen) is randomly selected. It is used to predict positive (blender, fridge) and negative (car) examples. (c) Forward pass through a CBOW model. Embeddings from a set of objects from a scene category are summed and used to predict the scene category.

1.2 Dataset

We use the ADE20K dataset [34] to train the object and scene embeddings (version ADE20K_2016_07_26). Unlike many other visual datasets, ADE20K contains a dense image annotation with every pixel being labeled. Additionally, each image is assigned to a scene category (e.g., abbey, bathroom). In total, the dataset contains 3,148 objects labels, 872 scene labels, and 22,210 individual images. We remove unlabeled objects and scenes, and only consider object and scene categories that occur in at least five images. An image has to contain at least two different objects to be selected. After applying these selection criteria, 1,140 object categories and 19,290 images remain in the dataset. The least common scene 'stone circle' appears in five images whereas the most common scene 'street' appears 2,241 times. The least common object 'carriage' appears in five images whereas the most common object 'wall' appears in 11,559 images. Sampling strategies, similar to those used for text data, are used to deal with this imbalance.

Scene embeddings from object co-occurrences

We harness two word embedding algorithms, Latent semantic analysis (LSA) [4] and Word2vec [17], to learn two sets of embeddings, one for objects and one for scenes. For LSA, embeddings are obtained via matrix factorization of a scene-object co-occurrence matrix. For Word2vec, we either use a scene category to predict the objects it contains (Skipgram model) or use a set of objects to predict the category of the scene they appear in (CBOW). The resultant embeddings are used to test the distributional hypothesis for objects and scenes, and as feature vectors for a scene classification model.

1.3 Latent semantic analysis (LSA)

LSA models the relationship between words and a collection of documents they appear in [4]. Its applications include document retrieval for search queries and topic modelling. It is based on the singular value decomposition (SVD) of a term-document matrix, wherein rows represent terms or words, columns represent documents, and each entry corresponds to the frequency of occurrence. In order to derive image-based embeddings using LSA, we replace the document-word matrix by a scene-object matrix $\mathbf{X} \in \mathbb{R}^{n,m}$, where each row represents an object, each column represents a scene category, and the (i, j)-th entry is nonzero only if object i appears in scene j. This is illustrated in Fig. 1a. Using SVD, we can perform a low rank approximation of $\mathbf{X}$ as

$$\mathbf{X} \approx \mathbf{O}\mathbf{\Lambda}\mathbf{S}^{\top}. \tag{1}$$

Here, the rows of $\mathbf{O} \in \mathbb{R}^{n \times d}$ act as object embeddings, $\mathbf{S} \in \mathbb{R}^{m \times d}$ as scene embeddings, $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ is a diagonal matrix of singular values, and d is the embedding dimension which is selected by the user. To use this approach with

ADE20k data, we first binarize the $1,140$-dimensional vector of object counts for each image. Subsequently, object vectors from images representing the same category are added together, yielding a $1,140 \times 682$ matrix. Three different normalization approaches for the object counts are explored: *LSA-norm* (divide each column by its total count), *LSA-log* (log-transform the counts), and *LSA-tfidf* (TF-IDF transform). TF-IDF is a popular normalization scheme [2] and is given by the product

$$\mathrm{tfidf}(o,s) = \mathrm{tf}(o,s) \cdot \mathrm{idf}(o)$$

where $\mathrm{tf}(o,s)$ is the term frequency, i.e., the number of times an object o occurs in a scene category s, and $\mathrm{idf}(o) = m/|\sum_{i=1}^{m} \mathbf{X}(o,i) > 0|$ is the inverse document frequency that down-weighs objects that occur in many scene categories. An advantage of LSA over methods such as Word2vec is that in addition to embeddings for items in the training vocabulary, it also provides a linear transform that can be applied to unseen images. For a vector of object counts $\mathbf{x} \in \mathbb{R}^n$ representing a test image, the respective embedding is given by

$$\mathrm{embed}(\mathbf{x}) = \mathbf{O}^\top \mathbf{x}. \tag{2}$$

We will exploit this property in section 1.6 when we perform scene classification on test images.

1.4 Word2vec

Word2vec constructs word embeddings by relating target words to words with a context window, e.g., other words from the same sentence [17]. It comes in two flavors: In the Skipgram model, the conditional probability of a context word given the target word is maximized, whereas the CBOW model maximizes the probability of the target word given the sum of the embeddings of its context words. In typical text applications, input and output words come from the same domain, and two sets of embeddings (for inputs and outputs) are obtained. After training, the two sets are either averaged or one of them is selected to represent the embeddings.

Here, we consider an asymmetrical approach in which inputs and outputs stem from different domains, one representing objects and the other scenes. Therefore, one matrix contains the object embeddings, the other matrix contains the scene embeddings. For the Skipgram model (depicted in Fig. 1b), the objective it to maximize the average log probability for an object o given the scene s it appears in,

$$\sum_{s \in S} \sum_{o \in \mathrm{context}(s)} \log p(o|s) \tag{3}$$

where S is the set of scene images and context(s) is defined as the set of objects present in the respective image. Although our dataset is comparably small, we consider the subsampling and negative sampling strategies from [18] for the Skipgram model. To subsample scenes categories (inputs), the relative number of images from a given category is defined as its frequency $f(s)$. An image is rejected from the training data with a probability of $p(s) = 1 - \sqrt{t/f(s)}$. Since the image corpus is relatively small, we set $t = 0.005$ and the subsampling is repeated in every epoch. For an input image, five objects from the scene are chosen as positive outputs (objects are repeated for images with less than five objects). For negative sampling of objects, denote the frequency of an object as $f(o)$. We thus sample 20 negative objects with a probability of $p(o) = f(o)^{3/4}/\sum_{o' \in O} f(o)^{3/4}$, where O is the set of all objects.

For the CBOW model, uniform sampling is used throughout. First, an object is randomly sampled. Then an image containing this object is randomly selected, and the remaining objects are sampled from this image. A context size of five objects is used, and the scene category corresponding to the image serves as target. Both models are implemented in Pytorch, using 100 epochs with an Adam optimizer and a learning rate of 0.01.

1.5 Testing the distributional hypothesis for scenes

If the distributional hypothesis for scenes is correct and relevant semantic information is preserved in the embeddings, the embedding vectors should cluster according to semantic properties such as membership in supercategories. For instance, in the Places dataset [33] scene categories can be indoor (e.g. bathroom), natural outdoor (e.g. beach), or urban outdoor (e.g. football field). In line with this, we find that scenes within these supercategories appear more similar to each other than to scenes from other supercategories, using a cosine distance metric. This is exemplarily

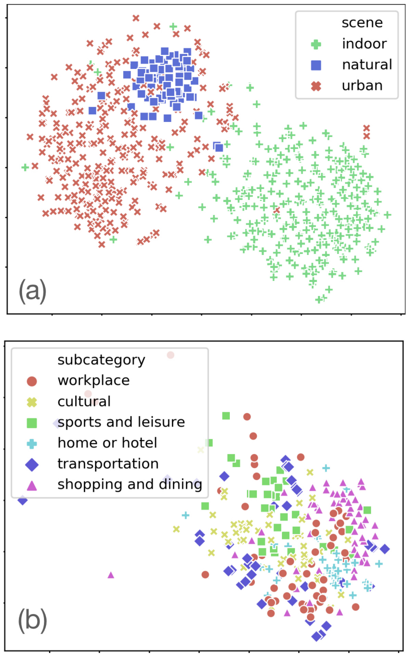

Figure 2. t-SNE visualization of scene embeddings for the Skipgram model (d=100), using a cosine metric. (a) Supercategories. (b) Subcategories of 'indoor'.

Method	Wilcoxon z	p-value
LSA-norm	-16.33	< 0.0001
LSA-tfidf	-14.72	< 0.0001
LSA-log	-15.62	< 0.0001
Skipgram	-261.18	< 0.0001
CBOW	-98.69	< 0.0001

Table 1. Within vs between category Wilcoxon rank-sum test results (z-statistic and p-value) for an embedding dimension of d=100.

depicted for the Skipgram model in Fig. 2a. Indoor, urban and natural scenes form well-defined clusters. Urban and natural scenes are partially overlapping, which could be explained by the fact that they are both outdoor scenes and hence share a number of objects such as sky, sun, or mountain.

As a more quantitative evaluation, the results for a Wilcoxon rank-sum test are reported in Table 1 for all models and d=100. The comparison of the within-category vs between-categories cosine distances is highly significant and illustrates that scenes within a category are much more similar to each other than to scenes from other categories. This relationship holds for subcategories, too (Fig. 2b). Results and visualizations for other embedding dimensions and all subcategories are provided in the supplementary material.

1.6 Scene classification

To evaluate the efficacy and relevance of the LSA embeddings, we compare scene classification performance of Residual Networks [9] (ResNets) and VGG networks [24] *without LSA* features to the performance when the corresponding LSA features are appended to the second last fully connected layer. In a second type of evaluation, we add an *object presence* vector rather than the LSA features. It is a 1200-dimensional binary array from the ADE20K Matlab API with an entry of 1 if an object is present in a particular scene and 0 otherwise.

1.6.1 ResNet18 and VGG11 training:

ResNet18 and VGG11 are trained in two classification analyses: first, (i) the training/validation split proposed by the ADE20K filenames, and second, (ii) a 5-fold cross-validation based on the combined training/validation data. For both analyses we use batches of 100 composed of randomly cropped 224×224 px images. Each random crop is normalized using a uint8 normalization transform in Pytorch [20].

On the training/validation split we select the 506 scene categories contained in both sets. For the 5-fold cross-validation the number of classes depends on each fold (499-509). For ResNet18 we use an initial global learning rate of 0.1 with a linear weight decay of 0.0001 after each epoch. For VGG11, to prevent a divergence of the weight amplitudes we use an initial learning rate of 0.01 with a weight decay of 10^{-5} after each epoch. The models' weights and biases are initialized using the Glorot's initializer [26] and optimized using Adam. Models are trained for 100 epochs in all analyses.

1.6.2 Adding LSA features:

We add the LSA embeddings to two extra fully-connected (FC) ReLU layers added to the end of each model architecture as an additional classification subnetwork [30]. The first FC layer has 4096 neurons. It receives inputs from both the 512 units of the last layer of ResNet18/VGG11 and the 300-dimensional vector of LSA embedding features. This feeds into a second FC ReLU layer with 4096 units which is followed by a final softmax layer. We use a fan-out/fan-in ratio greater than one on our proposed FC layers connection in order to optimize the layer-to-layer activation [6].

LSA transforms are generated using the training exemplars in each analysis. The transform is then applied to the object annotations of the respective test images. The weights and biases of the extra FC layers are updated using the same training parameters as in the baseline model. Therefore, we do not use ResNet18 and VGG11 with pre-trained weights but rather train the full model including the additional FC layers from scratch.

For consistency with the image batch generation we use Pytorch's *SubsetRandomSampler* synchronized with the corresponding image batches to generate corresponding LSA training batches. The LSA embeddings are not randomly cropped. Rather, batch indices are permuted on each training iteration. This approach guarantees consistent and reliable Class-Activation maps as reported in the supplementary material [32].

1.6.3 Results:

Table 2 shows the scene classification results of ResNet18 and VGG11 pipelines on the training/validation split. ResNet18 shows an average improvement of 3.72% on Top1 and 4.56% on Top5 accuracies when the LSA embeddings are included in the FC layer in comparison with the baseline *without LSA* features.

ResNet18 shows an average improvement of 3.72% on Top1 and 4.56% on Top5 accuracies when LSA embeddings are included as compared to the baseline without additional features. Compared to the baseline with the object presence vector, ResNet18 with LSA features shows an improvement of 2.99% on Top1 and 3.77% on Top5. VGG11 does not show a significant improvement on any of the baselines. We hypothesize that the information from LSA embeddings is not properly absorbed by the VGG11 network due to the large amount of FC-layers connected at the end of the layer and a poor segmentation associated with a relatively large size average-pool layer compared to ResNet18 [23].

Training /valida- tion	**Without LSA**		**Object presence**		**LSA**		**LSA categories**		**LSA tf-idf**		**LSA log**	
	Top1	**Top5**	**Top1**	**Top5**	**Top1**	**Top5**	**Top1**	**Top5**	**Top1**	**Top5**	**Top1**	**Top5**
VGG11	44.56	65.34	42.88	65.88	44.65	64.41	45.34	65.11	45.07	66.12	44.79	64.33
ResNet18	49.01	71.06	50.77	71.65	**53.47**	**75.45**	51.06	**73.77**	**52.37**	**75.88**	50.35	**73.94**

Table 2. Training/validation results showing the Top1 and Top5 accuracies for the two baselines: (1) evaluating scene classification with the image data only or *Without LSA*, and (2) using the object presence obtained from a binary *Object Vector* included on the ADE20K API. The evaluations related to the LSA features are (1) using the *LSA* normalized embeddings, (2) using the LSA embeddings obtained from *combining the scene categories*, (3) using the LSA embeddings from the *tf-idf* transform, and (4) the LSA embeddings after the *log* transform. Values in bold are $p < 0.001$ comparing the scores with the *without LSA* baseline.

Table 3 shows the results for the 5-fold cross-validation modality in percentages. No significant improvements are found comparing any performance including LSA information with any performance baseline such as *without LSA*

5-fold cross-val	Without LSA		Object presence		LSA		LSA categories		LSA tf-idf		LSA log	
	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
VGG11	40.225± 10.553	61.346± 8.914	41.445± 9.573	61.047± 9.657	41.112± 9.141	61.914± 9.775	41.134± 9.256	61.331± 9.671	41.916± 9.661	62.061± 9.814	41.618± 9.467	61.390± 9.551
ResNet18	42.556± 11.237	62.277± 10.333	42.959± 10.111	62.691± 9.989	44.112± 9.991	63.491± 10.143	43.562± 9.877	62.952± 10.225	44.067± 10.023	63.001± 10.120	43.067± 9.741	62.114± 10.342

Table 3. 5-fold cross-validation results showing the Top1 and Top5 accuracies. Standard deviations are calculated across the 5 folds.

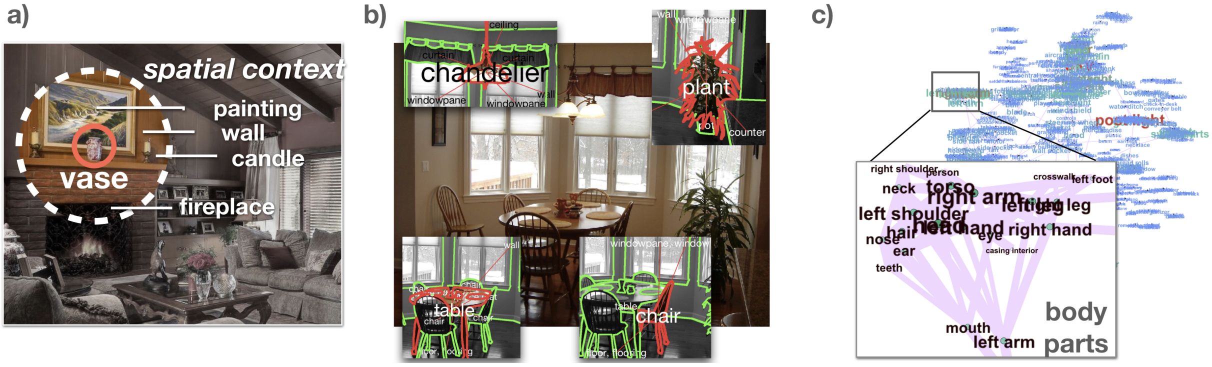

Figure 3. (a) Spatial context is defined as the objects in close proximity to a target object (vase). (b) Automated analysis of spatial context for a dining room image containing 29 objects. The insets show 4 different target objects (red outline) together with their spatial neighbors (green outline). (c) Graph analysis of the object embeddings from the Skipgram model (d=300). Spatially closely related objects such as body parts appear as subclusters within the graph.

and *Object presence* information. The lack of significant improvement observed on the 5-fold cross-validation might be attributable to a smaller size of the training set in comparison to the training/validation split. We hypothesize that the size of the training set is important for obtaining a significant performance improvement on deep feed-forward networks. However, the information of the LSA embeddings can be properly transferred using the extra FC layers as we report on the Class Activation Map (CAM) analysis. CAM and additional analyses are presented in the supplementary material.

2 Object embeddings from spatial context

In the Skipgram model in Section 1.4 the context of scene is defined as all objects occurring within it. While this 'bag-of-objects' approach is successful in creating meaningful scene embeddings, it is unclear whether this is an adequate approach for object embeddings. After all, the arrangement of objects within a scene is not random but follows semantic and syntactic rules, giving rise to hierarchies of objects within images [28]. Often, these relationships are indexed by spatial proximity. Proximity can be due to functional relatedness (a towel hanging from a towel ring or towel radiator) or object/part relationships (a body consists of different body parts). Our previous quantification of spatial relationships using co-occurrence within images probably lacks the granularity to model such tight-knit relationships.

To alleviate this issue, we zoom in on sub-regions within images, and focus on objects and their local neighborhood. In particular, we define as *spatial context* the objects in an image in close proximity to a target object (Fig. 3a). Importantly, this allows us to transfer the notion of window size used by context-based methods such as Word2vec to image data. There are different ways to operationalize spatial context (e.g., radius around center, distance from object border). Furthermore, frames of reference can be proximal (image pixel coordinates) or distal (inferred 3D scene coordinates).

Here, we perform an image-based (proximal) analysis of spatial context. Since center coordinates are not always

Target object	Closest neighboring objects	
	Skipgram with Spatial Context	Skipgram
towel	towel rack 0.182 towel ring 0.211 towel radiator 0.226	countertop 0.238 screen door 0.259 shower 0.268
bicycle	bicycle rack 0.351 parking meter 0.418 sidewalk 0.418	entrance 0.495 street sign 0.501 stall 0.506
pen	notepad 0.324 stapler 0.427 paper 0.430	paintbrush 0.417 sandpit 0.431 scotch tape 0.435

Table 4. Closest neighbors and cosine distances for different probe objects using Skipgram models (d=100) with spatial context.

meaningful for stuff (e.g. wall) and objects with holes, we consider *distance to object boundary* as a more meaningful distance metric. We parse a total of 604,355 object instances across the ADE20K dataset. For each of the images, a sparse object x object distance matrix is created with up to 345 instances per image. A nonzero entry at position i, j means that objects i and j are in the spatial context of each other. The calculation of the spatial context is based on the segmentation maps, where we use Matlab's *imdilate* function with a 7x7 square-shaped structured element to expand a target object's boundary by 3 px. The dilated target now touches its immediate neighbors (spatial context) whose indices are obtained by intersecting all objects with the target. The distance between target and neighbor is given by the inverse proportion of pixels the dilated target has in common with the neighbor, which implies that distances are not symmetric. For objects that share object-part relationships (e.g. hand and arm) we set the distance to a small nonzero value of 10^{-10}. Fig. 3b shows the effect of such parsing on four objects in a dining room image.

To train word embeddings we use these distance matrices with the Word2vec Skipgram model. In contrast to Section 1.4, the input to the model is now the target object and the outputs are uniformly sampled objects in the spatial context (positive examples) or outside the context (negative examples). Consequently, we maximize the probabilities

$$\sum_{o \in O} \sum_{o' \in \text{context}(o)} \log p(o'|o) \tag{4}$$

where context(o) now refers to the local spatial context. In each iteration, 5 positive and 20 negative examples are randomly selected. For objects with less than 5 neighbors, positive examples are repeated. Hyperparameters are the same as before. The two resultant sets of embeddings are averaged. Fig. 3c shows a graph analysis of the distance matrix between the embeddings thresholded at a cosine distance of 0.6. There is emergent subclusters with collections of objects that represent close spatial or object/part relationships. Another qualitative analysis is presented in Table 4. The Skipgram model developed in this section (second column) is contrasted with the Skipgram model in Section 1.4 based on object co-occurrences in scenes (third column). For the Skipgram model using spatial context, the closest objects for a given target are now indeed objects that are expected in close vicinity to the target.

3 Discussion

Object and scene embeddings derived from object co-occurrences in images yield semantically relevant representations. The embeddings cluster significantly along scene supercategories (indoor, urban, and natural) and subcategories (e.g., workplace, transportation, shopping and dining). This nicely dovetails with the distributional hypothesis which states that scenes are to some extent determined by the collection of objects they contain. The results are robust for a variety of models (LSA, Skipgram, CBOW) and embedding dimensions (d=50, 100, 300). Moreover, when using ADE20K's training/validation split, incorporation of LSA embeddings into a CNN for scene classification yields an improvement of 4.62% Top5 classification accuracy over a model without object information and 3.77% over a model that includes a vector of objects present in the scene. This suggests that the low-dimensional representation of object vectors provided by LSA is relevant for scene classification.

In Section 2, we abandon the 'bag-of-objects' approach in order to generate object embeddings that take spatial context into account. We define spatial context as the set of objects whose boundaries touch the boundary of a target object. Embeddings based on this spatially more sophisticated approach encode meaningful and hierarchical relationships of objects that cannot be achieved with the more coarse-grained analyses.

More broadly speaking, there are two principal applications for object/scene embeddings generated with the proposed approaches. First, scene segmentation and analysis is a central problem in robotics [22] which includes finding compact representations of images. In our experiment, embeddings appended to the last layer can improve the performance of a scene classification CNN. In line with our findings, [3] show that embeddings can refine predictions of a scene classifier. These findings are corroborated by the fact that object detectors emerge in CNNs for scene classification [31].

Second, in cognitive psychology, vision science, and sensory neuroscience scene perception, analysis, and interpretation is a model case for understanding how the brain represents complex visual information. Similar to embeddings, the human brain builds representation and expectations using image statistics and co-occurrences (both temporal and spatial) [25]. Since embeddings quantify object-object, scene-scene, and object-scene distances, they might provide an invaluable resource for studying the brain processes related to high-level image analysis and memory processes. For instance, object-scene pairs have been used as stimulus material in cued memory recall experiments (e.g. [27]), and embeddings might both explain some variability observed for different object-scene combinations and guide the selection of stimulus material.

Another area, where image-based embeddings could provide an important tool, are the brain mechanisms underlying information sampling via eye-movements. For instance, it is well-established that semantic object-scene relationships have an important influence on oculomotor control in humans. Distances derived from image-based embeddings might provide a means to characterize these effects in detail, a possibility that current state-of-the-art models of oculomotor control fail to provide [21].

There are several limitations that warrant consideration. A possible critique of our first approach is that the information relating scenes to objects is already contained in the object-scene co-occurrence matrix. This is, of course, trivially true, since the embeddings are directly built from the co-occurrences. However, we show that embeddings are able to represent this information in a lower dimensional space which is not only spatially more efficient but can also have a regularizing or denoising function [4]. A related limitation is the fact that our scene classifier relies on object annotations which are not available in many datasets. However, [3] show that such annotations can be generated on the fly using CNNs for object detection and segmentation. Consequently, a viable approach could be to train the embeddings on annotated data and then deploy them on a system that self-generates annotations.

Lastly, our analysis of spatial context is a proof-of-concept case, rather than a rigorous quantitative investigation. Yet, the initial findings point towards several avenues for future research. First, a relevant question is whether spatial context is better defined based on proximal (image-based) or distal (3D-scene based) coordinates. The latter requires inferred 3D coordinates which can be derived from image reconstruction techniques that infer depth metrics [29]. Second, it is conceivable that some scene categories differ not so much by the collection of objects they contain but rather by the spatial relationships between the objects. In such a case, incorporating local spatial context might be critical for good performance. Merging our approaches in sections 1.2 and 2 might be a way to address this.

In conclusion, we show that object and scene embeddings can be created from object co-occurrences and modeling of local spatial context. These embeddings are both semantically meaningful and computationally useful for downstream applications such as scene classification.

References

1. M. Ailem, B. Zhang, A. Bellet, P. Denis, and F. Sha. A probabilistic model for joint learning of word embeddings from texts and images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 1478–1487. Association for Computational Linguistics, 2020.

2. J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitinger, and A. Nürnberger. Research paper recommender system evaluation: A quantitative literature survey. In *ACM International Conference Proceeding Series*, pages 15–22, 2013.

3. B. X. Chen, R. Sahdev, D. Wu, X. Zhao, M. Papagelis, and J. K. Tsotsos. Scene Classification in Indoor Environments for Robots using Context Based Word Embeddings. In *2018 IEEE International Conference of Robotics and Automation (ICRA) Workshop*, 8 2019.

4. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 9 1990.

5. J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi. Gene2vec: Distributed representation of genes based on co-expression. *BMC Genomics*, 20(S1):82, 2 2019.

6. X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and D. M. Titterington, editors, *JMLR Proceedings*, pages 249–256. JMLR.org, 3 2010.

7. T. Gupta, A. Schwing, and D. Hoiem. ViCo: Word Embeddings from Visual Co-occurrences. *Proceedings of the IEEE International Conference on Computer Vision*, 1:7424–7433, 8 2019.

8. D. Harwath and J. Glass. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015 - Proceedings*, pages 237–244. Institute of Electrical and Electronics Engineers Inc., 2 2016.

9. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 770–778. IEEE Computer Society, 12 2016.

10. S. Kottur, R. Vedantam, J. M. F. Moura, and D. Parikh. Visual Word2Vec (vis-w2v): Learning Visually Grounded Word Embeddings Using Abstract Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 4985–4994, 11 2016.

11. S. Lai, K. Liu, L. Xu, and J. Zhao. How to Generate a Good Word Embedding? *IEEE Intelligent Systems*, 31(6):5–14, 7 2016.

12. L. J. Li, H. Su, Y. Lim, and L. Fei-Fei. Objects as attributes for scene classification. In K. K.N., editor, *Trends and Topics in Computer Vision. ECCV 2010. Lecture Notes in Computer Science*, volume 6553, pages 57–69. Springer, Berlin, Heidelberg, 2012.

13. Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu. Understand scene categories by objects: A semantic regularized scene classifier using Convolutional Neural Networks. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2318–2325. Institute of Electrical and Electronics Engineers Inc., 6 2016.

14. S. Liu, M. F. Demirel, and Y. Liang. N-Gram Graph: Simple Unsupervised Representation for Graphs, with Applications to Molecules. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8466–8478. Curran Associates, Inc., 2019.

15. G. L. Malcolm, I. I. Groen, and C. I. Baker. Making Sense of Real-World Scenes. *Trends in Cognitive Sciences*, 20(11):843–856, 11 2016.

16. J. Mao, J. Xu, K. Jing, and A. L. Yuille. Training and Evaluating Multimodal Word Embeddings with Large-scale Web Annotated Images. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 442–450. Curran Associates, Inc., 2016.

17. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings.* International Conference on Learning Representations, ICLR, 1 2013.

18. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

19. A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 5 2001.

20. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury Google, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. K. Xamla, E. Yang, Z. Devito, M. Raison Nabla, A. Tejani, S. Chilamkurthy, Q. Ai, B. Steiner, L. F. Facebook, J. B. Facebook, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, pages 8026–8037, 2019.

21. M. Pedziwiatr, M. Kümmerer, T. Wallis, M. Bethge, and C. Teufel. Meaning maps and saliency models based on deep convolutional neural networks are insensitive to image meaning when predicting human fixations. *bioRxiv*, page 840256, 11 2019.

22. J. C. Rangel, M. Cazorla, I. García-Varea, J. Martínez-Gómez, E. Fromont, and M. Sebban. Scene classification based on semantic labeling. *Advanced Robotics*, 30(11-12):758–769, 6 2016.

23. D. Shen, G. Wu, and H.-I. Suk. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248, 6 2017.

24. K. Simonyan. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2015.

25. C. Teufel and P. C. Fletcher. Forms of prediction in the nervous system. *Nature Reviews Neuroscience*, 21(4):231–242, 4 2020.

26. J. M. M. Torres and E. A. Stepanov. Enhanced face/audio emotion recognition: video and instance level classification using ConvNets and restricted Boltzmann Machines. In *Proceedings of the International Conference on Web Intelligence*, pages 939–946, 2017.

27. M. Treder, I. Charest, S. Michelmann, M. C. Martin-Buro, F. Roux, F. Carceller-Benito, A. Ugalde-Canitrot, D. Rollings, V. Sawlani, R. Chelvarajah, M. Wimber, S. Hanslmayr, and B. P. Staresina. The hippocampus as the switchboard between perception and memory. *bioRxiv*, page 2020.05.20.104539, 5 2020.

28. M. L. H. Võ, S. E. Boettcher, and D. Draschkow. Reading scenes: how scene grammar guides attention and aids perception in real-world environments. *Current Opinion in Psychology*, 29:205–210, 10 2019.

29. J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Advances in Neural Information Processing Systems*, pages 82–90, 10 2016.

30. H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1143–1152. IEEE Computer Society, 12 2016.

31. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object Detectors Emerge in Deep Scene CNNs. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014.

32. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 2921–2929. IEEE Computer Society, 12 2016.

33. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 6 2018.

34. B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 1, pages 5122–5130. Institute of Electrical and Electronics Engineers Inc., 11 2017.

35. X. Zhuo, J. Zhang, and S. W. Son. Network intrusion detection using word embeddings. In *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, volume 1, pages 4686–4695. Institute of Electrical and Electronics Engineers Inc., 7 2017.