Behavior Research Methods (2020) 52:2349–2371
https://doi.org/10.3758/s13428-020-01390-8

The Linguistic Analysis of Scene Semantics: LASS

Dylan Rose[1] · Peter Bex[1]

Published online: 28 April 2020

Abstract
In this paper, we define a new method for analyzing object-scene contextual relationships using computational linguistics: Linguistic Analysis of Scene Semantics, or LASS. LASS uses linguistic semantic similarity relationships between scene object and context labels embedded in a vector-space language model: Facebook Research's fastText. Importantly, the use of fastText permits semantic similarity score calculation between any set of strings and thus elements of any set of image data for which labels are available. Scene semantic similarity scores are then embedded in object segmentation mask locations in the image, creating a semantic similarity map. LASS can also be fully automated by generating context and object labels, as well as object segmentation masks, using deep learning. We compare semantic similarity maps between human- and neural network-generated annotations on a corpus of images taken from the LabelMe database. Semantic similarity maps produced by the fully automated LASS have a number of desirable properties, while maintaining a high degree of spatial and semantic similarity to them. Finally, we use LASS to evaluate the distribution of semantically consistent scene elements in space. Both show relatively uniform distributions of semantic relatedness to scene context, suggesting that contextually appropriate objects are likely to be found in all image regions. Taken together, these results suggest that LASS is accurate, automatic, flexible, and useful in a number of research contexts such as scene grammar and novelty detection.

Keywords Natural scenes · Scene semantics · Computational linguistics

Introduction

Stimuli for vision research are often simple, well-parameterized geometric objects such as lines, gratings, and polygons (Rust & Movshon, 2005). While these are easy to render and control experimentally, significant interest exists in examining visual function in more natural, lifelike contexts (Hayhoe & Ballard, 2005; Olshausen & Field, 2005). This, along with growing computing power and ease of file sharing and distribution, has led to significantly increased use of images of the natural and man-made worlds as experimental stimuli. An important problem associated with the use of natural scene content is the difficulty in parameterizing its content beyond analysis of image features (e.g. edges, contrast, color). This limitation is important, because visual interaction with a scene involves more than just such pictorial information: memory, language, and specific object and contextual knowledge all play a role in the behavior of the visual system under these constraints (Brockmole & Le-Hoa Vo, 2010; Hayhoe, Shrivastava, Mruczek, & Pelz, 2003; Henderson & Ferreira, 2004). The ability to measure such "top-down" content is therefore crucial for isolating experimentally relevant effects.

Biederman, Mezzanotte, and Rabinowitz (1982) first proposed a grammar of scene content, including scene syntactic and scene semantic components. Scene syntax refers to the appropriateness of an object's spatial properties in a scene, such as whether it was or needed to be supported by or interposed with other objects. Scene semantics refers to the need to retrieve the meaning of an object. For example, one understands that a mailbox does not belong in a kitchen based on e.g. knowledge that the probability of seeing such objects in that context is low or zero based on a history of interaction with such an object and context.

This conceptual system was modified by Võ and Wolfe (2013), whose definitions are now more commonly in use than those proposed originally by Biederman and colleagues. They define scene semantics as properties of objects identifying their "global meaning" of a scene. For example, these authors suggest that if one found a bed of grass in place of a carpet in an office, this would constitute a violation of the semantics of "office" scenes, as "office" means in part a place

✉ Dylan Rose
rose.dy@husky.neu.edu

[1] Psychology Department, Northeastern University, Boston, MA, USA

where carpet is expected and grass is not. Scene syntax refers to an object's placement aligning or failing to align with viewer expectations about its "typical location" in a scene, such as a bed of grass growing vertically on an outdoor wall instead of on the ground (Võ & Wolfe, 2013). See Fig. 1 for examples of scene syntactic and semantic violations taken from a data set of related images described in Öhlschläger and Võ (2017).

Initial efforts to study syntactic and semantic properties of scenes typically required direct manipulation of their content. To make subsequent analysis tractable, the authors used stimuli rendered as line drawings or 3D computer graphics (Hollingworth, 1998; Loftus & Mackworth, 1978; Võ & Henderson, 2011). A number of studies have also attempted to induce semantic or syntactic changes to image content in full color images of natural scenes (e.g. Coco, Araujo, & Petersson, 2017; Coco & Keller, 2014; Underwood & Foulsham, 2006). Though these latter are an improvement relative to synthesized scene images in terms of realism, both may still change global image statistics of a particular scene context in ways that confound scene grammatical pictorial information (Becker, Pashler, & Lubin, 2007).

Fig. 1 Images taken from the SCEGRAM database with scene semantic (**a**) and syntactical (**b**) violations. In (**a**), the toilet paper appears in the dishwasher rack, not where it would be expected (in a bathroom instead of a kitchen). In (**b**), the same toilet paper roll appears in the correct context but in a syntactically impossible location (hovering next to the toilet, suspended by fishing wire)

Two recent projects that theoretically avoid these issues provide stimulus sets of full color images of natural scenes for use in studying scene grammar. The first, the Berlin Object in Scene database (BOiS, Mohr et al., 2016), includes 130 color photographs of natural scenes. For each, a target object was selected, and versions of the same scene were photographed at an "expected" location, an "unexpected" location, and absent from the scene altogether. Expected vs. unexpected locations for each object were assessed by asking human observers to segment scenes into regions where an object was or was not likely to occur given a scene context label. The second is the SCEne GRAMmar database (SCEGRAM, Öhlschläger & Võ, 2017). This database was constructed for a set of 62 full color images of natural scenes in one of six scene grammatical conditions, fully crossing both scene syntax and scene semantic manipulations for each object and scene.

SCEGRAM and BOiS are unique, valuable tools for studying scene grammatical effects for a variety of research purposes. However, both are limited by their small size, degree of experimenter effort required for their creation, and the measurement techniques used to quantify the degree of scene grammatical manipulation actually induced in their images. First, the total number of images available between both sets across all the described conditions is only 1134. Though these databases no doubt took tremendous effort to create, they are small compared with other potentially relevant ones, such as LabelMe (Russell, Torralba, Murphy, & Freeman, 2008) or Microsoft's Common Objects in Context (COCO, Lin et al., 2014). A fraction of these images are also composed according to experimental conditions that may be irrelevant for a given experimental objective, further limiting their total size.

Second, though BOiS provides object position likelihood maps that could in principle be used to extend the set to other images and scene contexts for the same object set, SCEGRAM relied on direct physical manipulation of objects and thus could not be extended without using the authors' reported image composition methods. Third, BOiS focuses on scene *syntax*, and the authors did not attempt to isolate scene semantic effects from scene context effects. Finally, while both groups took obvious care in selecting objects and contexts that would ostensibly induce scene semantic or syntactic effects, they still relied entirely on subjective experimenter or participant judgments of whether a particular object in a particular image did or did not involve a violation of scene semantics or syntax.

Scene semantics through linguistic semantics

One possibility for addressing this last issue – effectively, how to produce an objective measurement of scene semantics – involves exploiting the strong link between visual perception and language. It has been shown that the linguistic properties of a stimulus can exert a strong influence on visual perception, particularly on eye movement behavior (e.g., Richardson, Dale, & Spivey, 2007; Anderson, Chiu, Huette, & Spivey, 2011; Draschkow, Wolfe, & Vo, 2014; Henderson & Ferreira, 2004). Scene syntactical and semantic violations have also been found to produce a similar electrophysiological response to those produced by the same violations in language (Võ & Wolfe, 2013).

Given these relationships, if one wishes to measure scene semantic relationships between objects in a particular context, it may be possible to do so by evaluating visual semantic relationships indirectly using linguistic relationships as a proxy. For example, if an experimenter says "An octopus doesn't belong in a farmyard", their judgment may depend as much on the linguistic use cases of "octopus" and "farmyard" as on perceptual interaction with octopuses and the typical occupants of barns. *Linguistic* semantic relationships between these terms could therefore potentially be used as a model for such relationships in the perceptual space of the natural world. Such a proxy or substitution is useful, as there exist a number of efficient computational linguistics tools for measuring semantic relationships between words. The most widely used of these are *vector-space models* (VSMs). Among them, latent semantic analysis (LSA, Dumais, Furnas, Landauer, Deerwester, & Harshman, 1988) is arguably the most straightforward, and will therefore serve as a useful introduction to the field.

Using VSMs depends on acceptance of the *distributional hypothesis*: words that mean similar things will appear in the same or similar contexts in written or spoken language (Sahlgren, 2008). LSA defines "appearing in the same or similar contexts" in terms of word frequency co-occurrence within a document. It does so by first constructing a table of terms by document frequency for each term in a corpus of text, across documents within the corpus. Because the resulting matrices are generally large and sparse, the table is transformed into a lower-rank feature space using singular value decomposition for the sake of computational efficiency. The semantic similarity between any two terms in the corpus can in these terms be expressed via an angular distance measure (typically the cosine) between vectors associated with a set of words in the resulting low-rank matrix. These values range in practice between zero and one, with zero indicating no semantic relationship between terms and one indicating word identity. Negative values are possible but rarely encountered in practice, and do not have a necessarily straightforward interpretation (i.e., they are not necessarily *antonyms*, see Landauer, McNamara, Dennis, & Kintsch, 2013; Thalenberg, 2008).

At least one study has already leveraged this perception/language connection using LSA to study top-down effects on eye movement behavior. In it, Hwang, Wang, and Pomplun (2011) began with a set of images taken from LabelMe. Each contained labels and segmentation masks for objects visible in the scene. The authors embedded these labels into a pre-trained LSA model and were thus able to calculate object-to-object semantic similarity scores for scene objects. These values were then embedded at scene locations defined by the object masks, creating a "semantic similarity map" for a particular object. A group of observers were shown the images and asked to perform either a free viewing or visual search task. The authors computed a semantic similarity map for each object observers fixated relative to all other non-fixated scene objects. Gaze transitions between points in these maps demonstrated the existence of a modest preference for sequentially fixating semantically similar items during free viewing, as well as a progressive degree of semantic guidance toward target objects across fixations during visual search (Hwang et al., 2011).

Though innovative, Hwang and colleagues' approach still has several technical limitations that restrict its usefulness for studying scene semantics "in the wild". The first and most obvious is that it does not consider relationships between the semantics in terms of scene objects and *scene context*, but only among scene objects themselves. This decision was appropriate given the stated goal of their research, and it may indeed be the case that object-to-object semantics *create* a form of scene context. However, outside of experimentally constructed arrays used for testing visual search, objects in the natural world do not appear without a surrounding visual environment, a known history of use or properties, and a sense of the contextual appropriateness of an object given these former. Any suitable technique must therefore be able to incorporate explicit contextual information to be useful in analyzing scene semantics, regardless of whether it is also able to capture potential "object-to-object" effects.

Second, LSA cannot produce cosine similarity scores for terms that are not elements in the corpus on which it has been trained (Landauer et al., 2013). It is well documented that object labels generated by human observers using LabelMe often contain spelling errors, or unusual or compound constructions, or are otherwise simply irrelevant to the image content (see Fig. 2 for examples). Applying LSA to these data would be challenging without careful image curation and significant manual preprocessing. LabelMe and COCO continue to grow, and many other excellent resources are available for crowd-sourcing such tasks. Nevertheless, acquiring object position and label data is and will likely remain an expensive and time-consuming barrier to a wider implementation scope for this technique.

Fig. 2 Example of object segmentation and labeling taken from the LabelMe image database. This image contains numerous examples of labeling noise issues that would prevent the application of LSA to the data, including spelling errors ("sealing design", "emtest"), unusual and compound constructions ("person walking", "person dark", "a child"), and tags/masks irrelevant to the content of the scene ("dead body", "blood", "zero-point gravity")

The Linguistic Analysis of Scene Semantics (LASS)

Hwang and colleagues' fundamental approach is sound, but to be useful for the purposes of scene grammatical research, the issues raised must be addressed. Any proposed extension or adaptation of their method should therefore:

1. incorporate scene context information in the form of text or other linguistic descriptors;
2. evaluate semantic relationships between arbitrary object and scene context label data strings;
3. fully automate the process of creating object and context labels, as well as object segmentation masks.

Here we present a new, objective, completely automatic technique for measuring scene semantic information directly from arbitrary image content. Like Hwang and colleagues' technique, it has a computational linguistics algorithm at its heart. We therefore call it "Linguistic Analysis of Scene Semantics", or LASS, to distinguish it from this earlier work. We believe LASS has the potential to significantly expand the scope of study of scene semantics. LASS runs with the following three steps:

Generate scene context labels

The first set of information required for LASS is a set of scene context labels, such as "alley" or "restaurant". The specific method used to produce or obtain labels is unconstrained, though in order for the method to be fully automatic, an automatic approach for doing so is naturally preferred in this step. In the present study, we used a scene classification deep neural network, a Keras implementation of the VGG-16 convolutional neural network architecture (Krizhevsky, Sutskever, & Hinton, 2012) trained on MIT's Places365 data set (Zhou, Khosla, Lapedriza, Torralba, & Oliva, 2016).

This network assigns scene category labels to an image from a predefined set of 434 candidates. It does so by learning patterns of hierarchically organized image features associated with specific image classes. Once these relationships have been extracted from a training data set, new images can be passed into the network, and their activation of learned feature patterns returns a set of class membership probabilities for each of the learned scene classes. In order to avoid dependence on arbitrary experimenter or observer decisions as to what labels are correct, we took the top five most likely scene context labels the network produced instead of just the most likely.

Such a label or set of labels is certainly only a partial descriptor of what we might consider "scene context". However, if we consider a simple example of a set of statements such as "There is a carrot on the floor of a nuclear submarine" and "There is a carrot on the floor of the barn", we can see that it is at least a contextually useful window into it. We understand *a priori* that carrots rarely occur in nuclear submarines and frequently occur in barns, even if we have never spent much time inside either. We should further be able to make a consistent set of graded contextual appropriateness judgments if we

changed the context of our example from "barn" to "shed", from "shed" to "military storage facility", and from "military storage facility" to "nuclear submarine".

Generate scene object labels and masks

LASS's second step is to identify scene objects, segment their boundaries within the image, and provide them with a label. Again, as with scene context labels, either automatically or human observer-generated label and segmentation mask data can be used here. For this study, we used a deep-learning algorithm called Mask RCNN (He, Gkioxari, Dollár, & Girshick, 2017), implemented in Keras, to generate these data. Mask RCNN can be understood as first computing a set of object-level masks for one of 84 object categories within a number of network-identified rectangular ROIs. These are then refined to object-class-specific mask shapes, to which object labels are then applied. The algorithm has demonstrated excellent object segmentation and classification performance in Microsoft's COCO (see He et al., 2017, for a full description of the model's structure and behavior, and evaluations of its performance). An example of its output is presented in Fig. 3.

Calculate semantic similarity scores between objects and scene contexts and embed scores in object masks

Once a set of context labels, object labels, and object segmentation masks have been computed for an image, LASS's third step is to generate object-scene semantic similarity scores for each object. Although human-generated, crowd-sourced semantic similarity scores could be used by LASS, several computational linguistics models support the automation of this step. If a *set* of candidate scene context labels is being considered, the average of these scores between an object and each label is used. It is here that the technique's strongest constraint applies. Given that human observer- and even most automatically generated scene or object labels are unlikely to be exact matches for terms contained in a training corpus, a semantic similarity evaluation method that can accept arbitrary strings as input must be used for this computation. Otherwise, a significant portion of the label data will need manual preprocessing or be altogether unusable.

At the time of writing, only a single computational linguistics method, Facebook Research's fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017), has this feature. fastText is a direct extension of a vector-space language model derived from LSA, word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), and is thus algorithmically and conceptually related to it (Altszyler, Sigman, Ribeiro, & Slezak, 2016). Both word2vec and fastText create vector-space representations of text corpora similar to that of LSA, but model term "co-occurrence" as probabilities over fixed local window sizes, not as frequencies of co-occurrence across corpus documents.

FastText extends the behavior of word2vec by representing each model word vector as the sum of the latent dimension vector values for both a particular word and a set of sub-word *n*-grams. The most important advantage this confers over both

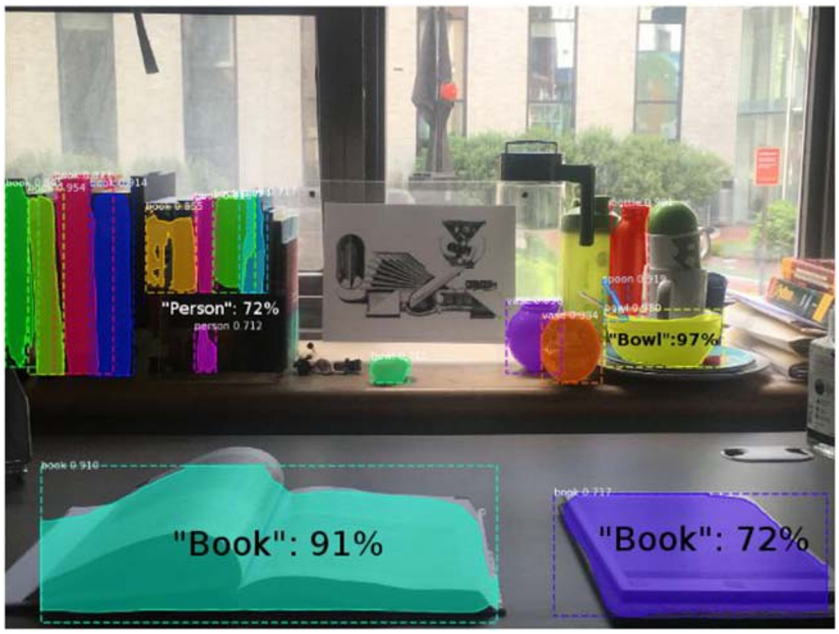

Fig. 3 Example Mask RCNN Output. Object labels and class probabilities are visible in white. The visible rectangular bounding boxes surrounding object masks are a by-product of the Mask RCNN algorithm and do not contribute to the final semantic similarity map. Colors for objects shown are randomly selected, and are presented only to enhance contrast between object masks and the image background

LSA and word2vec for our purposes is that it permits the calculation of similarity scores for terms that not only do not occur in the same document within the training corpus, but that were not included in the training set at all (Bojanowski et al., 2017). Similarity scores between objects and a context label are finally embedded into regions defined by each object mask, creating an object-contextual semantic similarity map for a given context label. An example of the output of this process for a randomly chosen image and three scene context labels generated in step 2 of LASS is provided in Fig. 4.

The final versions of the maps used in this study contain averaged values for each object across the set of five scene context labels used. This is done in part to avoid including steps dependent on experimenter fiat, to down-weight the possible contribution of unusual or unlikely context labels to the final map, and to potentially capture multiple distinct but semantically highly similar labels to be applied (e.g. "cafe" and "bar"). An example of this conversion and its effect on semantic similarity scores in the final similarity map is presented in Fig. 5.

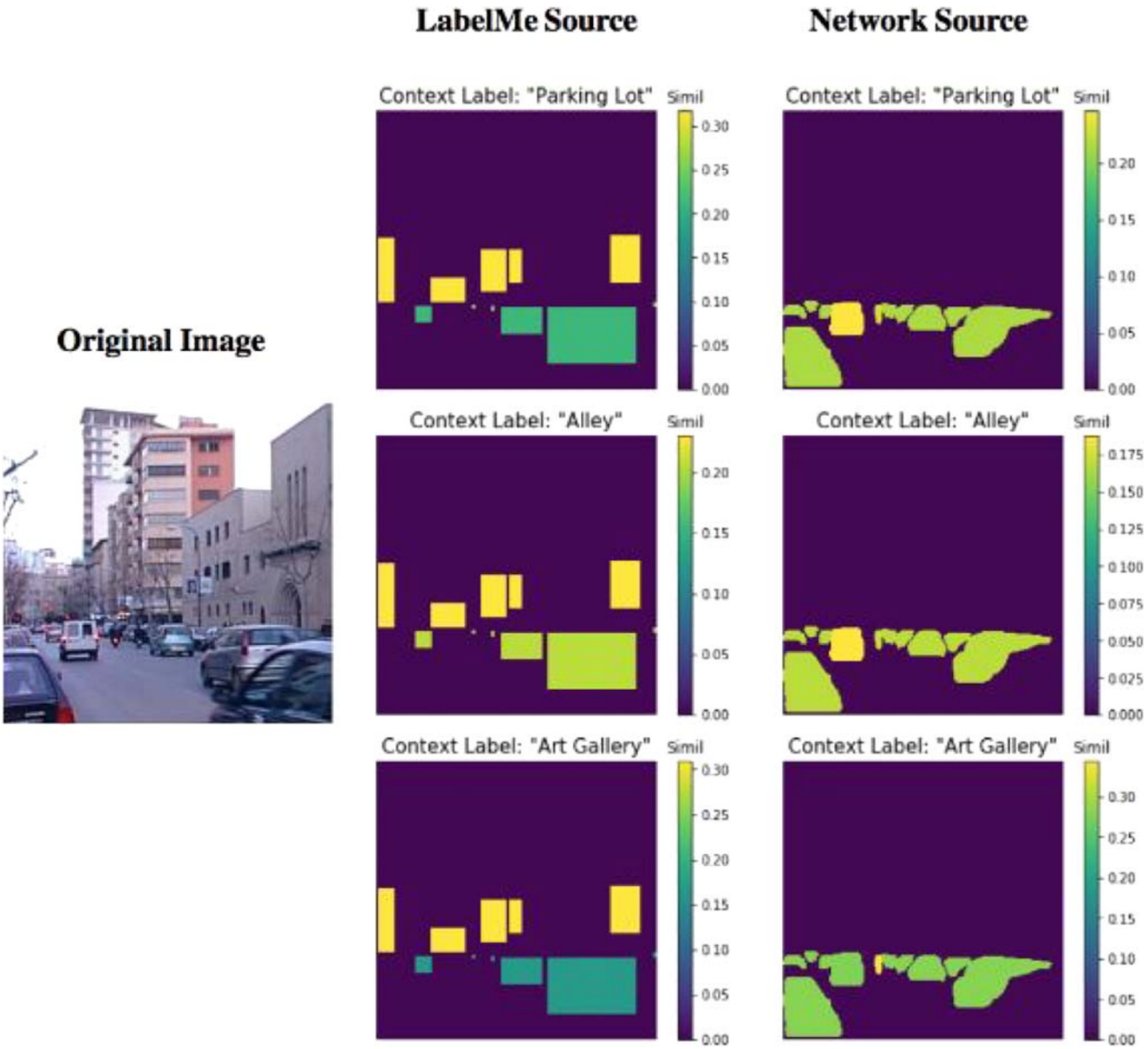

Fig. 4 Example semantic similarity maps for an image and three context labels for both neural network- and LabelMe-generated sources. Color in these images is scaled according to the semantic similarity values contained within a particular object mask, with yellow/red colors indicating high semantic similarity, and blues/purples indicating low semantic similarity. Regions where no object label is generated are assigned a similarity score of zero

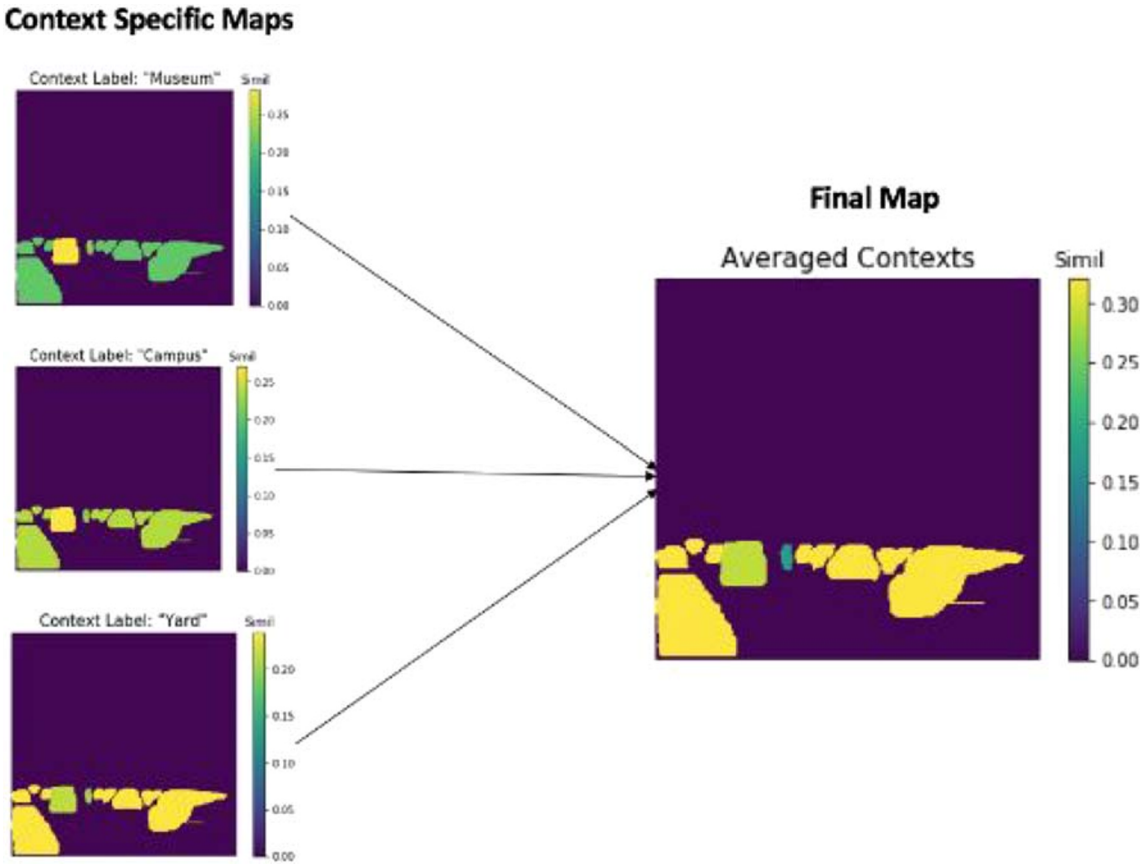

Fig. 5 Context-specific to final semantic similarity map conversion. Scores in the final maps described in this paper are averages of each object's semantic similarity score for all of the context labels evaluated. Semantic similarity is color-coded, with yellow/red colors indicating high semantic similarity, and blues/purples indicating low semantic similarity. Note that this calculation takes place at the level of the text strings, not by averaging the maps themselves, though the results are similar

Objectives

Object labels and masks: property distribution comparisons

Our first objective in trying to validate LASS is to determine whether the behavior of its fully automated form differed significantly from the behavior when human observer data was used instead. If the object detection and segmentation network used produced object sets and masks that differed substantially from those derived from human observer data, then the former is unlikely to capture scene semantic effects meaningful to human observers (note that it is assumed that human observer data is meaningful to human observers). The most important parameters for ensuring consistency between human observer and automated maps are the shape, quantity, and positional properties of the object masks, and the semantic similarity of their object and context labels. While human observer data has obvious deficits in terms of mask and label accuracy (Fig. 2), it is still driven in part by human scene semantic perception and decision-making and thus effectively remains a form of scene semantic information ground truth.

Of these two feature types, the semantic similarity of the object label sets is the more important for our purposes. While object mask placement and properties are crucial to constructing an accurate semantic similarity map, human observers frequently make overly general, inaccurate, or inconsistent segmentation masks for otherwise perceptually identical objects. Mask property noise in terms of inconsistencies between human- and automatically generated masks is relatively tolerable, provided the objects so identified are closely semantically related.

Semantic similarity map comparisons

If can be shown that human- and machine vision-identified scene objects and their properties are consistent, then our second objective is to demonstrate that the semantic similarity maps produced from these object sets are also consistent. This comparison addresses a more complex set of relationships between maps from different data sources, such as their sparsity and relative spatial distributions of semantic content. These features are crucial for some potential uses cases of semantic similarity maps, such as gaze prediction or anomaly

detection. Even if a machine vision system is capable of correctly identifying an object and defining its spatial boundaries in an image, if that object is not likely to be identified by a human observer, then this information is unlikely to be informative for predicting gaze position.

Scene semantic statistics distributions

Our third and final objective, closely related to the second, is to provide a set of descriptive statistics on scene semantic properties of images for both human- and automatically generated semantic similarity maps. If identified object properties and the semantic similarity maps derived from these are consistent across data sources, these distributions should also be similar. Any observed differences, however, may help identify specific biases inherent to either source in terms, for example, of their estimation of the scene semantic "center" of specific image contexts.

Of particular interest are the positional distributions of scene semantic information relative to the image center. This information may provide a preliminary window into understanding joint scene semantic-syntactic structure, as it could potentially be used to identify scene object content that is semantically consistent with the context but otherwise at an unusual location relative to a distribution of previous measurements of such relationships. It is also of broader theoretical value to consider differences in these distributions between *specific* image contexts, such as whether the placement of "knives" differs between the otherwise closely semantically related contexts of "kitchens" and "shops".

Methods

Image corpus

The initial data set for this study comprised randomly selected images from the LabelMe image database. The only selection criterion was a minimum of two segmented objects per image. The labels generated by human observers in this database were not corrected or modified in any way. Seven images in the selected set were corrupt and excluded from further analysis. A further 841 images did not yield object labels or a scene context label using either Mask RCNN or one of the scene context label-generating networks. These images were thus also excluded from further analysis, bringing the size of the image set to 9159.

Scene context label generation

Scene context labels for each image were generated using a VGG16 model convolutional neural network trained on the Places365 image database (Zhou et al., 2016). Places365 contains more than ten million images tagged as belonging to one of 434 scene category labels. For this study, we used a model pre-trained on this set, implemented in the Python deep learning library Keras (Chollet, 2015). The model and implementation were taken from a public repository[1]. This code provides a mechanism for retraining or "fine-tuning" the model, but for the sake of simplicity and reproducibility, we used the default model configuration and weights provided. We applied the network to each of the images in the final data set and extracted the top five most likely scene context classes from the output. The same set of labels for each image was later used to calculate scene semantic similarity for both the LabelMe- and network-generated object sets. In order to control for the possibility that our results might differ based on the scene labeling network used, we also generated five scene labels for each image using a PyTorch implementation of ResNet-50 taken from a public repository[2].

Object label and mask generation

Object labels and masks were either taken from segmentation and label information provided by LabelMe, or generated directly from image content using Mask RCNN (He et al., 2017). We used a version of this network trained on Microsoft's COCO database. COCO contains high-quality object segmentation masks and labels for objects in one of 91 object categories "easily recognizable by a four year old child" on proximately 328,000 images (Lin et al., 2014, p. 1). The specific implementation of Mask RCNN we used was also written and trained using Keras[3]. Note that it classifies objects into a reduced subset of only 81 categories relative to the 91 provided to COCO annotators. Note also that this implementation exposes a large number of model parameters for user "tweaking". Except for specific manipulations of the object classification confidence threshold as described in the Results section, however, default values for these parameters as defined in the original Mask RCNN paper were used here.

Semantic similarity score and map generation

Semantic similarity scores were computed using a Python implementation of the fastText algorithm (Bojanowski et al., 2017) provided in the *Gensim* vector-space modeling package (Rehurek & Sojka, 2011). We used a pre-trained vector-space model provided by the authors of the original fastText paper. The training corpus contained approximately one million words taken from English Wikipedia articles[4]. Model training

[1] https://github.com/GKalliatakis/Keras-VGG16-places365
[2] https://github.com/CSAILVision/places365
[3] https://github.com/matterport/Mask_RCNN
[4] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

parameters were the "defaults" used in Bojanowski et al. (2017) (i.e. a range of *n*-gram sizes from three to six characters are used to compose a particular word vector). After loading the pre-trained word-vector set, semantic similarity scores were generated using the vector object's bound "n_similarity" method. This function averages cosine similarity scores for each pair of words between two provided word lists. Each object label in the available list for a particular image and data source was used as the first of these two sets.

Semantic similarity *maps* were created from semantic similarity scores for an image by first initializing an equal-sized zero matrix. Semantic similarity scores for a specific object were then embedded in the coordinates defined by the object mask within it, and the embedding was repeated for each object in sequence. The result was a binary matrix the size of the original image with scene semantic similarity scores for each object in regions defined by their masks. Data in image regions containing overlapping or occluded objects were overwritten by that of the foremost object.

Object label and mask property comparisons between data sources

We evaluated the semantic relatedness of the object label sets in three related ways. First, we generated semantic similarity scores between the label sets using the same method described for computing scene semantic similarity scores. Because these values only have a meaningfully interpretable range between zero and one, we consider it contextually appropriate to treat them as an interval measure. Statistics computed on a distribution of paired label sets may therefore be interpreted as percentage values above the "no similarity" point at zero. Second, we performed a permutation test on the labels using randomly selected pairs of images between the human observer- and automatically generated label data sources. Finally, for both sets of labels available for a specific image, we compared each set to an equal-sized list of words selected at random from a free dictionary English dictionary file provided by the Spell Checker Oriented Word Lists (SCOWL) database[5]. Figure 6 provides examples of each of these comparison types. Distributions of these scores for each image were compared using a Kruskal–Wallis nonparametric analysis of variance (ANOVA). Pairwise post hoc comparisons were made between the different sets using Bonferroni-corrected Wilcoxon rank-sum tests.

Three features of the object masks are of particular interest given their important roles in creating consistent semantic similarity maps between human observer- and machine vision-identified objects. First, we compared counts of objects found across images between the mask sets. We then examined two additional mask properties: size and "specificity". These data were generated using the Python package *SymPy*'s "Geometry" module. For each object mask in each source set we created a "Polygon" object using the list of object mask vertex coordinate pairs. We then calculated the size of the mask in terms of the internal area of the polygon using the bound "area" method. This function may return either positive or negative values depending on the orientation of its vertex points. Polygons which have sides that cross, for example, may produce negative values. For this reason, we report this feature of the object masks in absolute pixel units.

The meaning of an object mask's area is straightforward. In order to capture the less straightforward but equally important property of mask "specificity" with respect to an object's actual boundaries, we report the number of vertex pairs associated with it. This inferred relationship is reasonable given the necessary relationship between numbers of sides and numbers of vertices. Natural and even man-made objects are rarely, if ever, perfect simple geometric shapes such as triangles or rectangles. A more "specific" mask for a particular object should therefore be composed of fewer straight lines and more curves, and thus should have fewer sets of vertices.

Semantic similarity map comparisons

We compared semantic similarity maps for each image using a pixel-wise Pearson correlation between maps produced from each object data source. This was done using a custom Python function written for this purpose[6]. The code in this repository provides Python versions of MATLAB code associated with the MIT Saliency Benchmark website for developing and evaluating saliency maps, and has been demonstrated to produce output identical to that produced by its MATLAB counterparts.

Semantic similarity map value spatial distributions

We provide a set of descriptive results documenting the spatial and angular distributions of semantic similarity with respect to the photographic center of the images. To do this, we computed the *average radial profile* of semantic similarity maps across images for both the LabelMe- and network-generated label sets. Average radial profiles are commonly used in image processing to describe changes in binary intensity maps as a function of distance or rotation relative to their centers (see the papers cited in Mamassian, Knill, & Kersten, 1998). Intensity values within each of the rings either across the range of distances or angles relative to the image center can then be averaged, and intensity over distance or angular rotation functions computed over the resulting values.

As input to this method we created aggregated semantic similarity maps across images using the horizontally and vertically

[5] http://wordlist.aspell.net/ , 'english-words.95'

[6] https://github.com/tarunsharma1/saliency_metrics

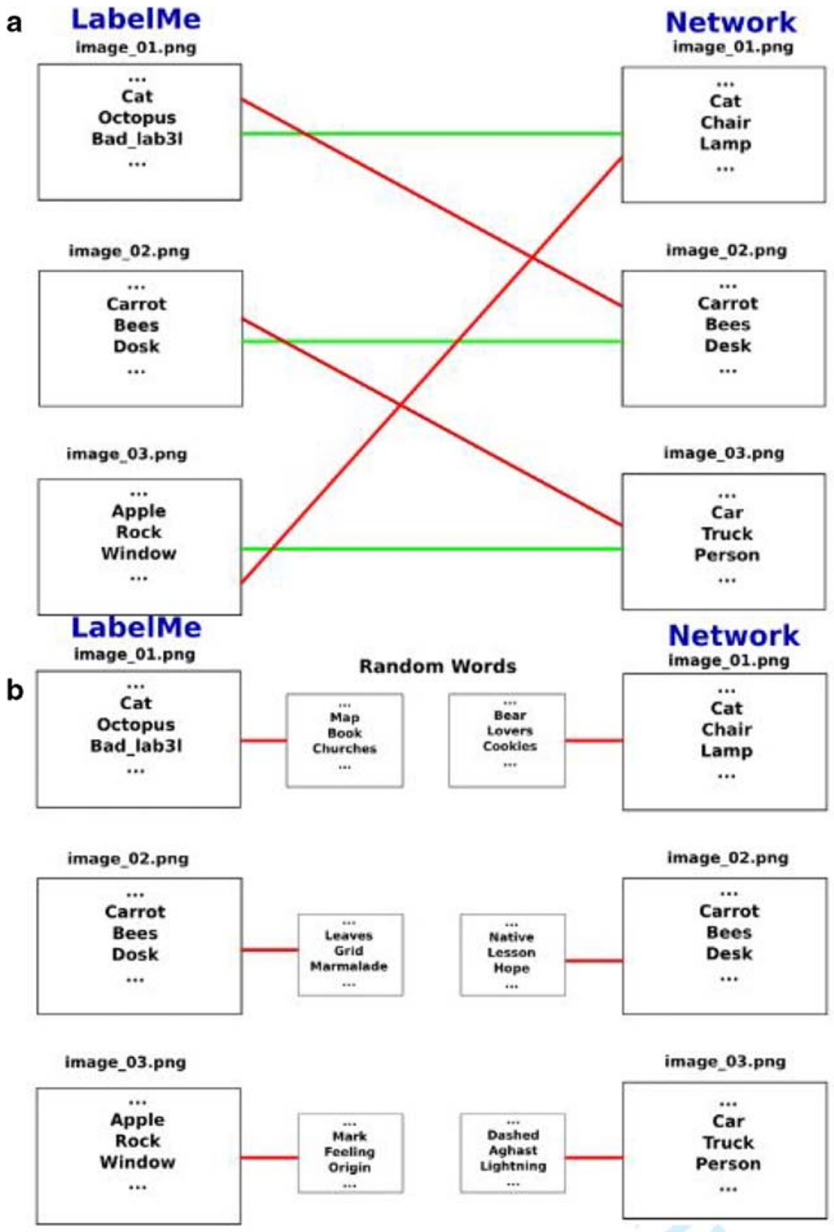

Fig. 6 Examples of object label comparison methods. In (**a**), object labels between sets are compared for the same image (green lines) and across images (red lines). In (**b**), each label set provided by both LabelMe and the network for all images is compared with an equal-sized but randomly selected word list from a free dictionary

normalized centroid locations for each object mask in both maps, along with the semantic similarity score at that location. These values were mapped to a 10 × 10 spatial grid using the MATLAB "meshgrid" function. The resulting maps were then averaged across images within each of the map data source sets. Radial average profile data were extracted from these gridded data using a heavily modified version of a publicly available MATLAB script[7]. Each grid was divided into a set of eight distance bands in each of eight angle sets. The slope of functions fitted to the resulting data can be understood as measuring the "steepness" of semantic similarity "falloff" as one moves away from the center of the semantic similarity maps.

Mask RCNN detection confidence threshold effect

An important consideration for deploying LASS in a fully automated fashion is the confidence threshold for object detection in Mask RCNN. This parameter sets the level of certainty that the network is required to have that it has correctly identified an object before returning a label and mask. Too high a threshold may cause the algorithm to fail to detect

[7] https://www.mathworks.com/matlabcentral/answers/uploaded_files/48809/average_radial_profile_2.m

any objects, making it impossible to use for a given image. Setting too low a threshold will lead to false positives. We here present data on the behavior of LASS for a randomly selected subset of 100 images from our primary corpus across a set of ten threshold values (5% to 95% confidence in 10% increments) under the following testing conditions:

Proportion of sample images with no detected objects as a function of threshold

High confidence threshold values can cause Mask RCNN to fail to detect any objects in an image, making it impossible to use with LASS if other label data sources are not available. It is therefore crucial to strike a balance between the accuracy of the label and mask data and its data dependence. This relationship can be clarified by examining the proportion of images in the sample that return no object classifications as a function of threshold. We also fit a beta regression with a double-log-link function to this data using the R package "betareg" (Cribari-Neto & Zeileis, 2010). Beta regression is commonly used for modeling data with proportional response variables, and the use of the double-log-link function helps mitigate the effects of the obvious nonlinearity in the data on the model fit.

Semantic similarity between LabelMe- and Mask RCNN-generated labels as a function of threshold

Lower threshold values may allow Mask RCNN to detect more scene objects, but this increase could result from an increase in the number of spurious or unlikely scene objects. Such a reduction in label quality could be seen in a reduction of object label similarity to the labels available through LabelMe as a function of decreased confidence thresholds. To evaluate the significance of this effect, we again fit a double-log-link function beta regression to the raw object-object semantic similarity score data across threshold values between the two object data sources.

Semantic similarity between LabelMe- and Mask RCNN-generated labels as a function of threshold

Finally, it is possible that the observed nonlinearities in the relationship between confidence threshold and semantic similarity scores may impact the spatial arrangement of these scores as well. This can be tested by examining the correlation between semantic similarity maps from the network and LabelMe data sources across threshold values. To test for the existence of such an effect, we fit a simple generalized linear model with an identity link function (using the R package "glm") with correlation coefficients between LASS maps generated using different object label sources (LabelMe or Mask RCNN), context label sources (VGG-16 or ResNet-50), different numbers of context labels (first label or all labels), and different confidence threshold values for object detection using Mask RCNN.

Results

Object label and mask property comparisons between data sources

Object label distributions

The LabelMe label set for the 9159 images contained a total of 227,838 labels across 10,666 unique object label classes. The label set generated by the network contained 93,783 labels with 80 unique object labels. Of the unique labels contained in the LabelMe set, 2146 or 20% were contained in our dictionary. Of those labels in the set generated by the network, 63 or 79% were contained in the same dictionary. Bar charts of the top ten most commonly occurring labels in both sets are presented in Fig. 7. For these data, label frequencies from both sources appear to follow a classic "Zipf-like" exponential distribution (Piantadosi, 2014), though the slope of the distribution appears to be significantly steeper for the LabelMe set than the network-generated set. The specific labels in each set differ, with only "car" occurring in the top ten for both.

We used two different networks to generate scene context labels: one based on VGG-16, and another based on the ResNet-50 architecture. Both were trained on the Places365 image data set, which includes 365 possible scene context labels. The VGG-16 network produced 359 unique scene context labels, or 98% of the possible label set. The ResNet-50-based network produced 362 unique scene context labels, 99% of the possible label set.

Distributions of the top ten most frequent labels generated by each network are shown in Fig. 8. Like the object labels, both networks appear to generate "Zipf-like" distributions of scene context labels. There was significant overlap in the top ten labels for both sources, with the only major differences being the inclusion of hospital, apartment building/outdoor, and promenade in the top ten generated by the VGG-16 network, but not in the top ten generated by the ResNet-50 network.

Comparisons between object mask count, shape, and specificity property distributions between object label data sources

Distributions of the average number of masks, object mask area, and number of sides for the object masks produced by LabelMe observers and the network are shown in Fig. 9. Across images, the network generated significantly fewer masks on average than LabelMe observers ($W = 7.0231196 \times 10^7, p < 0.001$). Those it

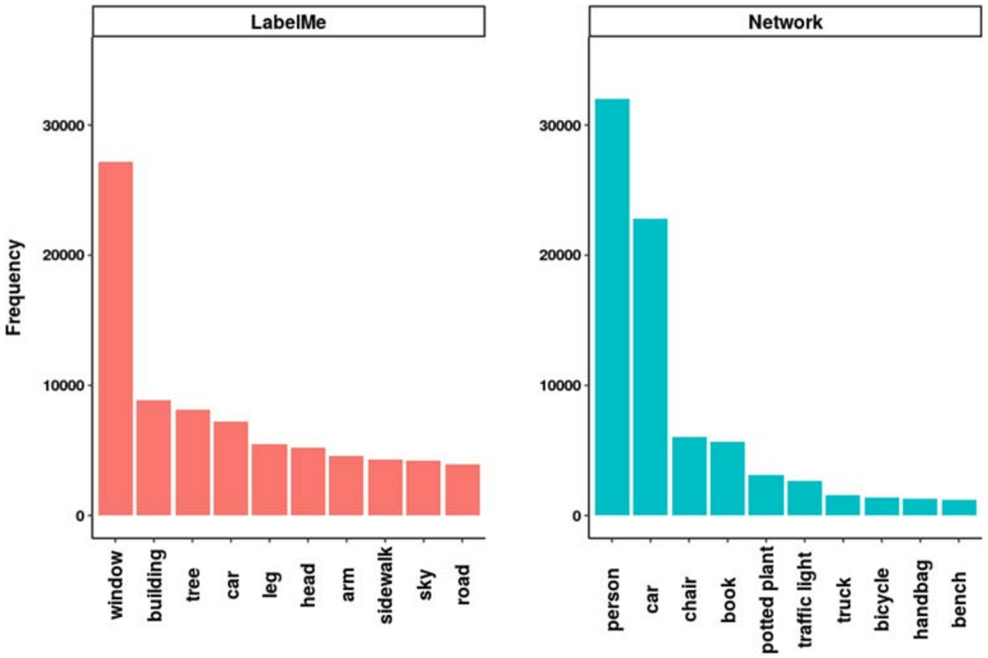

Fig. 7 Frequency histograms of the top ten most commonly occurring labels from LabelMe and Mask RCNN

did were also significantly smaller ($W = 2.0958008 \times 10^{10}$, $p < 0.001$) and had significantly fewer sides ($W = 1.5033984 \times 10^{10}$, $p < 0.001$) than those produced by human observers. We interpret these results to mean that the network-generated masks are more likely than human observer-generated masks to conform to a smaller (and thus presumably more accurate) set of scene objects, and that the masks produced for them adhere more tightly to the boundaries of the object they identify.

Finally, given the "Zipf-like" distribution of object classes for each object data source, it is likely that the relevant summary statistics are biased toward the mask properties of the two or three most common classes for each data source. As none of these overlap between the two sources, differences between their mask properties are therefore likely to be derived not from differences in the average mask shape and form for the same object, but from differences in the frequency of occurrence for those classes.

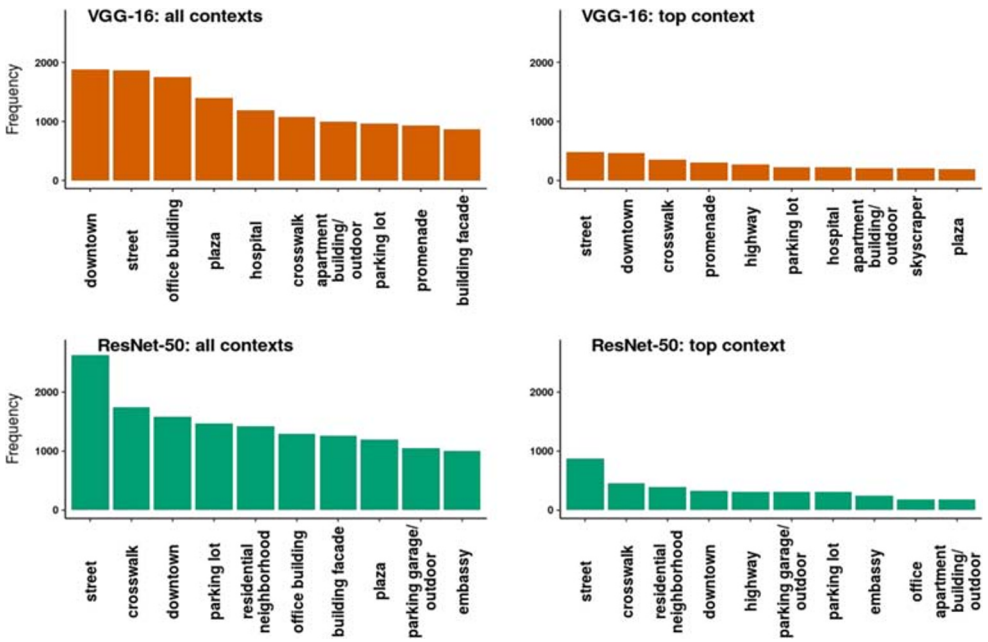

Fig. 8 Frequency histograms of the top ten most commonly occurring scene context labels across images. In (**a**), all five labels for each image are considered; in (**b**), only the first (most likely label) is considered

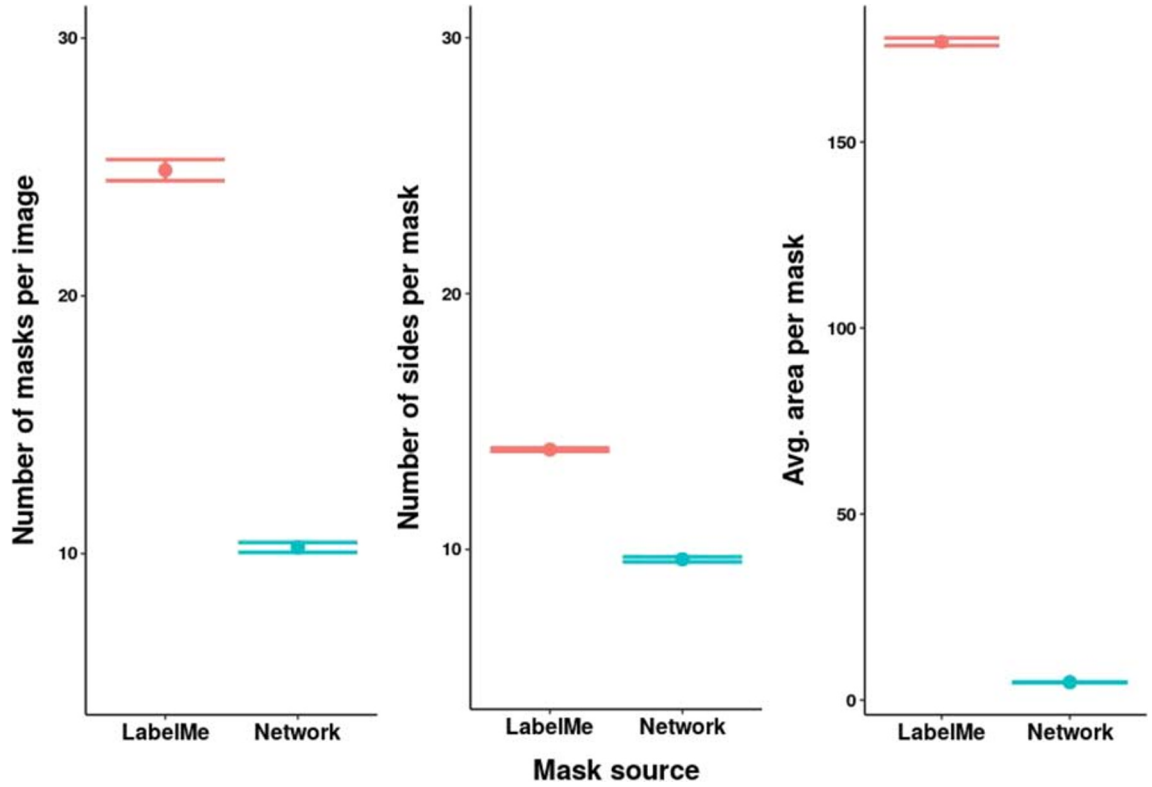

Fig. 9 Distributional results for mask properties between human observer and neural network-generated object masks

Semantic similarity between object labels from different sources

Distributions of semantic similarity scores between correctly and randomly paired label sets across data sources for a particular image, as well as for those labels compared with randomly selected word sets, are shown in Fig. 10. The Kruskal–Wallis test of differences between the different distributions of comparison scores was highly significant ($\chi^2(2) = 1759.14, p < 0.001$). Follow-up pairwise Bonferroni-corrected Wilcoxon rank-sum tests for each pair of comparison types (random images to paired images, random words to paired images, random images to random words) were all statistically significant ($p < 0.001$). Despite the significance of the tests, the differences between the distributions are perhaps smaller than would be desirable. This could potentially be addressed in future work by increasing the *n*-gram size of the language model, forcing comparisons between a smaller number of longer sub-word elements which are less likely to overlap.

Semantic similarity map correlations

Distributions of image-wise correlation coefficients by the number (one or five) and source (VGG-16 vs. ResNet-50) of the scene context labels used to generate the semantic similarity map for each image between the object label sources are shown in Fig. 11. Means and medians of each distribution are also shown in each plot. Across context label sources and the number of labels, distribution of correlation coefficients between maps generated using LabelMe data and Mask RCNN data is highly positively skewed, with most values greater than or equal to zero. Negative correlations likely indicate differences in object mask

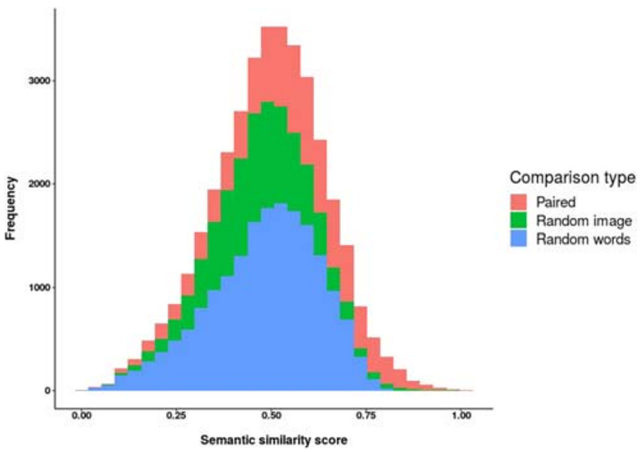

Fig. 10 Distributions of semantic similarity scores between object label data sources for the same image, a randomly paired image, and a randomly selected list of words. The paired and randomly selected image-to-image comparisons used only the image object label data provided through LabelMe or by the Mask RCNN network. Comparisons made against randomly selected words were made for both label data sources for each image, meaning this distribution contains twice as many points as the other two

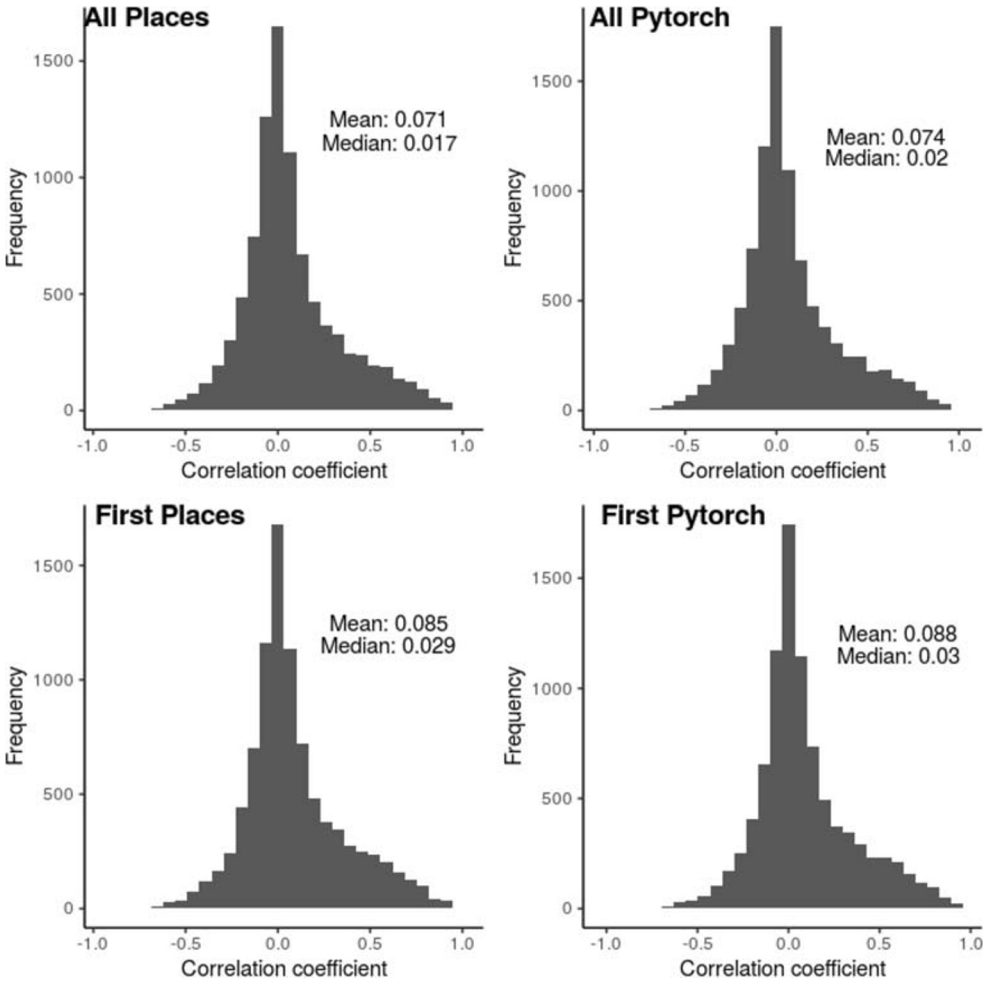

Fig. 11 Distribution of image-wise correlation coefficients for LabelMe- and Network-generated semantic similarity maps. Correlation coefficients are calculated pixel-wise between pixels for a given image between map data sources

placement in areas that are empty in one map but contain an identified object in another.

Means and 95% confidence intervals for distributions of correlation coefficients for each combination of context label source and number of labels are shown in Fig. 12. Results from a set of one-sample *t* tests comparing each distribution with the expected value $\mu = 0$ are shown in Table 1. In each test, the mean of the distribution was significantly greater than zero, though the differences were small. The low value of these correlation coefficients is likely attributable to differences in object placement between the LabelMe- and the network-derived object masks and not to differences in the semantic similarity scores of those objects per se.

A Kruskal–Wallis test indicated that the differences between these groups were also significant ($\chi^2(3) = 40.84, p < 0.0001$). Follow-up pairwise Bonferroni-corrected Wilcoxon rank-sum tests for each pair of context label source and label number groups found that distributions of correlation coefficients between maps using all of the VGG-16- and all of the ResNet-50-generated labels, as well as between the first VGG-16 and the first ResNet-50 label, were not significant. All other pairwise comparisons were statistically significant ($p < 0.0001$). Together, these results suggest that there are small but nontrivial differences in agreement between the semantic similarity maps as a result of using different numbers of context labels from different sources. The highest level of agreement was between maps generated using only the first label generated by ResNet-50; the lowest was between those generated using all of the context labels produced by VGG-16.

Gridded semantic saliency score data and their radial distribution functions for maps generated using object labels taken from LabelMe are shown in Fig. 13; the same set of results for the Mask RCNN-generated object label data are shown in Fig. 14. The vertical axis of the grids in both sets of plots is flipped, meaning that values in the lower-left-hand corner of

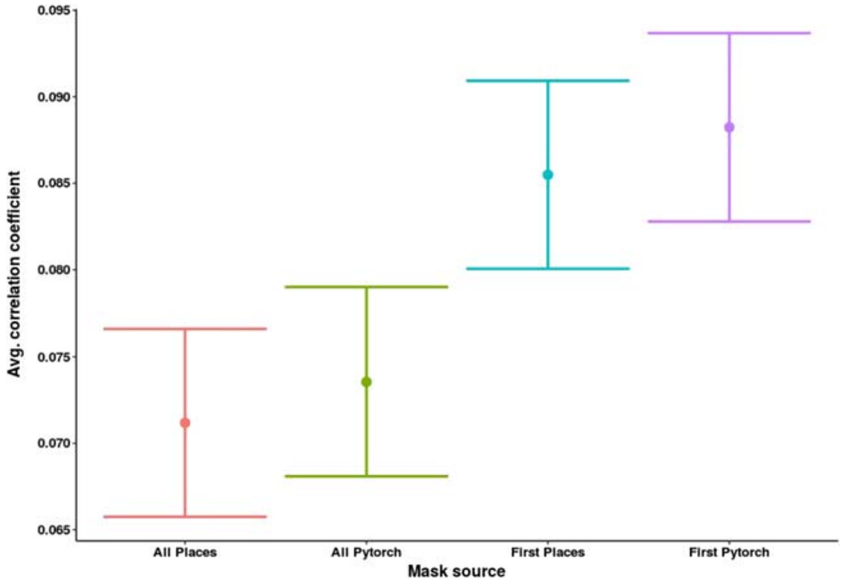

Fig. 12 Means and 95% confidence intervals for correlation coefficient distributions by scene context label number and source. The dashed line represents the value of the null distribution means for a set of one-sample *t* tests

each matrix represent semantic similarity scores in the region near the screen origin. Qualitative inspection of the plots suggests a slight concentration of semantic similarity in the center of images, but the pattern is diffuse. Of note are the values running from the upper left to lower left, and from lower left to lower right, in the grid data for the Mask RCNN object data source. No scores were generated in these regions across all maps, and the values shown were therefore imputed using the mean grid cell value. This suggests that the network has a strong bias toward the identification of objects away from the edges of images and toward their center.

In order to test the radial average profile of the aggregated maps quantitatively, we fit a simple linear regression model with the average score as the dependent variable, and distance, object label source (LabelMe or Mask RCNN), context label source (VGG-16 or ResNet-50), and the number of context labels (first label only or all five labels) included as independent variables. The fitted model parameters and the results of tests of significance applied to them are shown in Table 2. Both map data object source (estimate: −0.04, standard error (s.e.): 0.004, $t = -9.98$, $p < 0.0001$) and the number of context labels used (estimate: −0.09, s.e.: 0.004, $t = -20.18$, $p < 0.0001$) had statistically significant effects on radial average grid values. Both distance from the center (estimate: −0.001 s.e.: 0.001, $t = -0.93$, $p = 0.35$) and the source of the scene context labels (estimate: 0.005, s.e.: 0.004, $t = 1.07$, $p = 0.289$) had small but non-statistically significant effects on radial average values.

Taken together, these results suggest that the most significant effects on the radial average distribution of semantic similarity in a map are whether the object positions and labels have been generated using a neural network, and whether a single or all five context labels were used. By examining the estimates of the effects, we found that the sharpest "drop-off" in semantic similarity scores moving out from the center occurs for maps created using object labels and masks generated by Mask RCNN compared with only the first context label produced by either network. These effects make sense, first because Mask RCNN never detected any objects in a set of rectangular regions at the top and left edges of the images, which would naturally cause scores for these maps to be lower in those regions. The observed reduction in semantic similarity scores when using only a single context label also makes sense, because having any single object match a single context label well is less likely than having a number of partial matches across a set of labels.

Table 1 One-sample *t* test results comparing distributions of correlation coefficients to a zero-mean null distribution by number of context labels and context label source

Source/No. of Labels	*Df*	*t*	*p*<
All Places	9158	25.687	0.0001
All PyTorch	9158	26.435	0.0001
First Places	9158	30.816	0.0001
First PyTorch	9158	31.741	0.0001

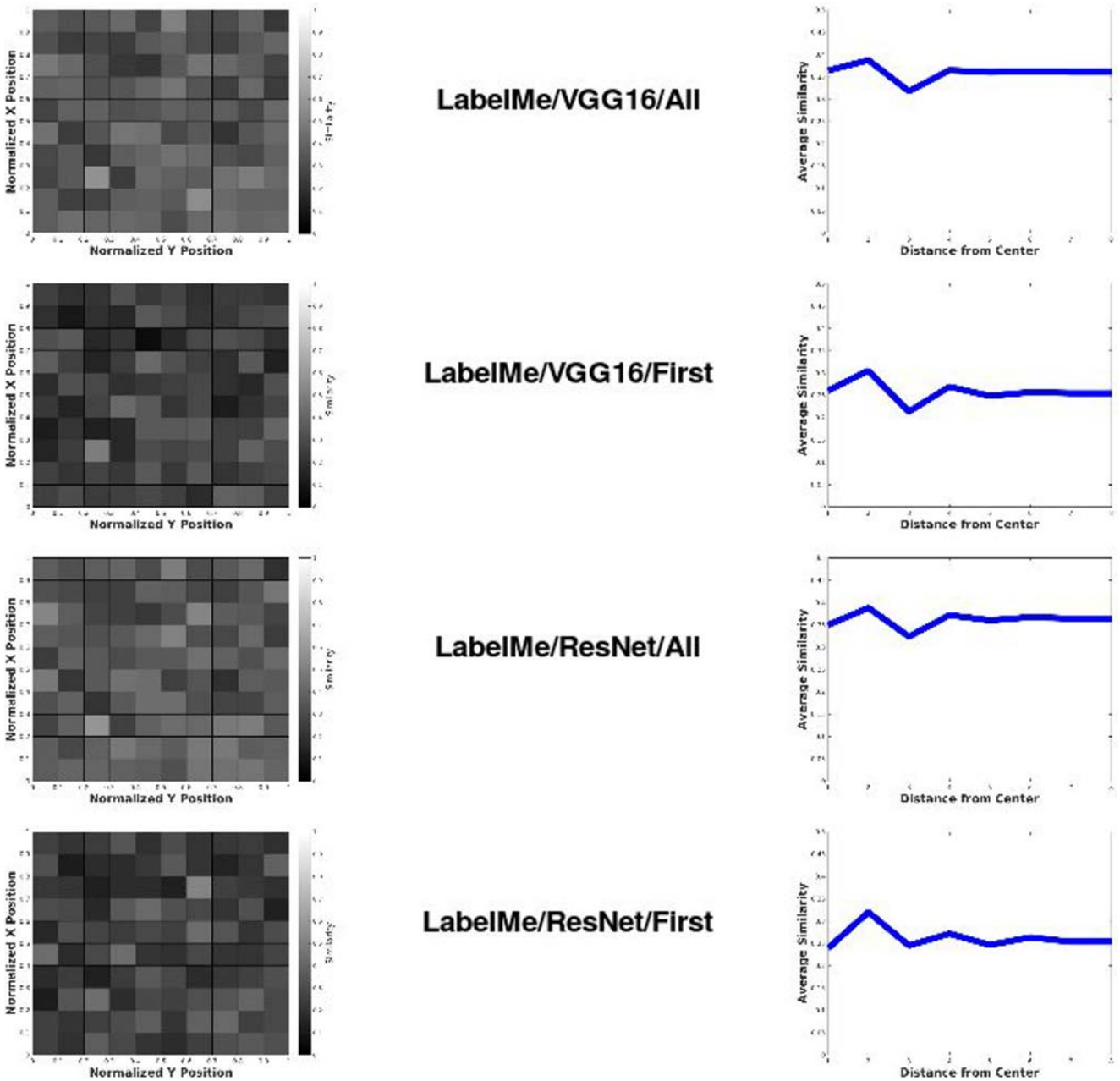

Fig. 13 Semantic similarity score grid data and radial average profile functions for both data sources across images with object labels taken from LabelMe. Left column: gridded semantic similarity score data. Values in each grid element represent averaged semantic similarity scores for objects whose centroids fell into that grid location. Right column: radial average profile data. These data are extracted from the grid by dividing it into a set of concentric rings, with arcs created by dividing all of the rings into quadrants. Data presented here are averaged across quadrants equidistant from the center of the grid but across angles.

Proportion of sample images with no detected objects as a function of threshold

Figure 15 shows that increased detection thresholds lead to significant increases in the proportion of images in the sample that yield no detections. However, this relationship is clearly nonlinear, with a sharp spike in the proportion without detections evident after the 55% threshold. Note also that the proportion never reaches zero. This is significant, as it suggests that some human observer data may be required even if label

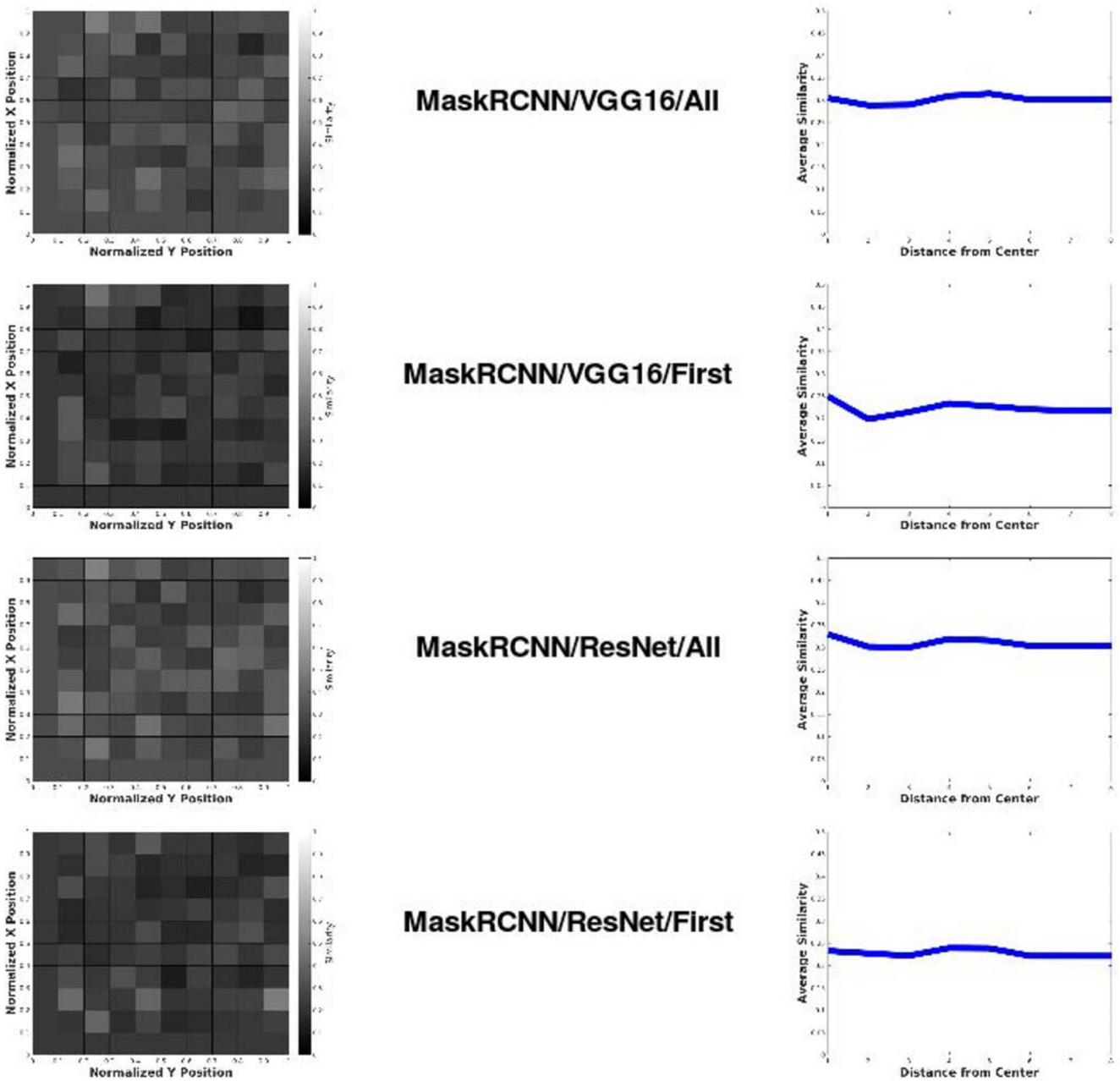

Fig. 14 Semantic similarity score grid data and radial average profile functions for both data sources across images with object labels taken from Mask RCNN. Left column: gridded semantic similarity score data. Values in each grid element represent averaged semantic similarity scores for objects whose centroids fell into that grid location. Right column: radial average profile data. These data are extracted from the grid by dividing it into a set of concentric rings, with arcs created by dividing all of the rings into quadrants. Data presented here are averaged across quadrants equidistant from the center of the grid but across angles

and mask data are generated primarily by Mask RCNN. Nevertheless, the fraction of images in a data set where this additional step will be necessary is likely to be fairly small.

Table 3 shows the results for the fitted model between the proportion of images without objects detected in the sample as the independent variable and the confidence threshold as the dependent variable. There was a significant effect of log-threshold on the log-proportion of images without detected objects (estimate: 1.09, s.e.: 0.12, $p < 0.01$), corresponding to an approximate 2.98% increase in detection failures for every 5% increase in confidence threshold.

Table 2 Fitted radial average plot data model. *$p < 0.1$; **$p < 0.05$; ***$p < 0.01$

	Dependent variable: Average similarity value
Object label source	−0.04*** (0.004)
Context label source	0.005 (0.004)
Number of contexts	−0.09*** (0.004)
Distance	−0.001 (0.001)
Observations	64
R^2	0.90
Adjusted R^2	0.89
Residual std. error	0.02 ($Df = 59$)
F Statistic	127.20***($Df = 4; 59$)

Semantic similarity between LabelMe- and Mask RCNN-generated object labels as a function of threshold

Figure 16 presents means and 95% confidence intervals for semantic similarity scores computed between Mask RCNN-generated object labels and those taken from LabelMe for the same image as a function of Mask RCNN object detection confidence thresholds. Increasing the object detection confidence threshold leads to a small reduction in the similarity between network- and LabelMe-generated labels. Again, there is evident nonlinearity in this effect, with similarity scores relatively consistent across low threshold values until an inflection point near the 55% confidence threshold, after which the semantic similarity of labels from the different sources falls off somewhat more sharply.

Table 3 Fitted double-log-link function beta-regression model for the proportion of images with no identified objects as a function of Mask RCNN object detection confidence threshold. *$p < 0.1$; **$p < 0.05$; ***$p < 0.01$

	Dependent variable: Proportion images w/o detection
Threshold	1.09** (0.13)
Observations	10
R^2	0.85
Log likelihood	27.48

Table 4 presents the model fitting results for a double-log-link beta-regression model where the semantic similarity between Mask RCNN- and LabelMe-derived object labels was the dependent variable and the Mask RCNN object detection confidence threshold was the dependent variable. The effect of threshold confidence level in this model was not statistically significant (estimate: −0.05, s.e.: 0.06, $z = -1.0$, $p = 0.316$). This suggests a desirable degree of consistency between the

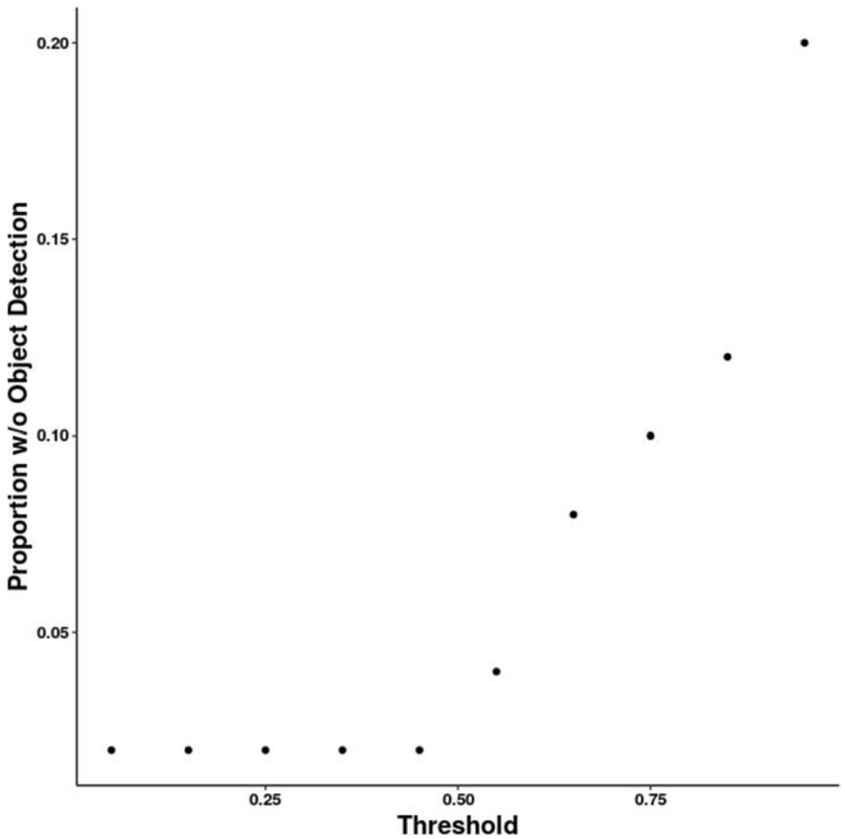

Fig. 15 Proportion of images in the sample with no object detections across threshold values. Black dots represent the proportion of images without any object detections at a given threshold value

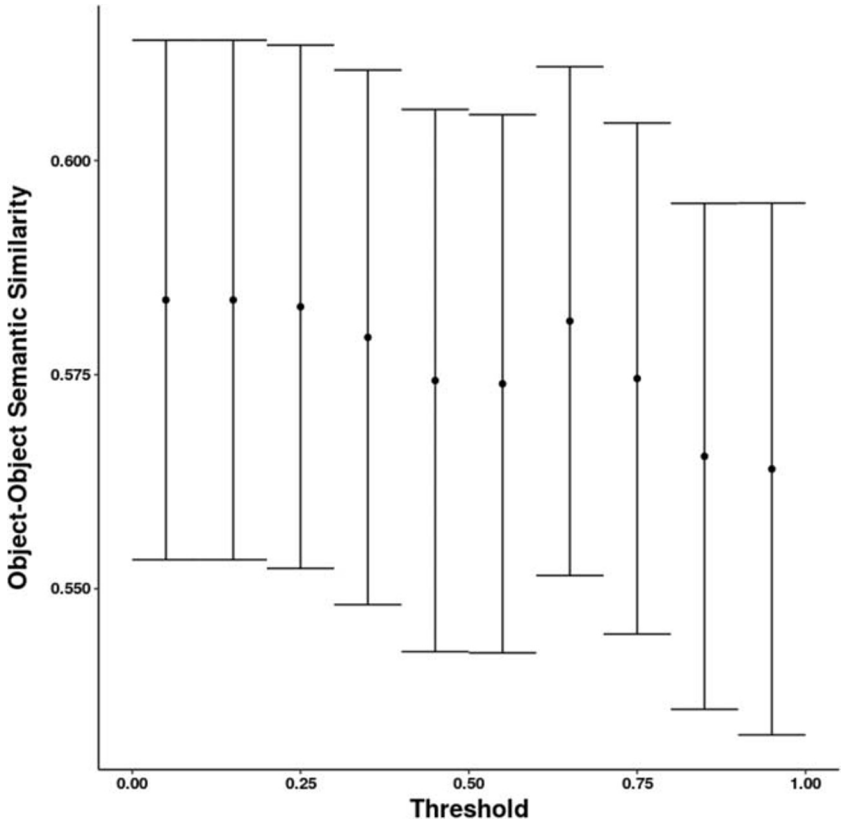

Fig. 16 Means and 95% confidence intervals for Mask RCNN to LabelMe object label similarities across Mask RCNN object detection confidence thresholds

network and LabelMe object label data sources across threshold values, freeing experimenters to select threshold values on the basis of other map properties, such as confidence threshold effects on the correlation between Mask RCNN- and LabelMe-derived LASS maps, as considered in the next section.

Correlation of LabelMe- and network-generated maps as a function of threshold

Figure 17 shows means and 95% confidence intervals for correlation coefficients computed between LabelMe and Mask RCNN object data-derived LASS maps between context label data sources, the number of context labels used, and across threshold values. There is a slight increase in map-to-map correlations between the data sources as the threshold increases. This is likely attributable to a reduction in the number of false-positive object detections or incorrect object class identifications evident at higher confidence threshold values. However, there is also an apparent nonlinearity in this trend above the 75% confidence threshold level, with correlation coefficients rising sharply above that threshold.

Table 5 shows parameters from a simple linear model fit to this same data. The correlation coefficient between maps for the same image generated using LabelMe- and Mask RCNN-sourced object data was the dependent variable; threshold values, the source of the context labels, and the number of context labels used were the independent variables. Of the three independent variables, only the value of the detection confidence threshold had a statistically significant effect on map correlations. Taken together, the data in Fig. 17 and the model fitting results suggest that while the number and source of the scene context labels have only a modest effect on the agreement between Mask RCNN and LabelMe data-derived LASS maps, increasing the Mask RCNN detection confidence threshold slightly above the default level (70%) significantly improves this agreement.

Table 4 Fitted beta-regression model for Mask RCNN/LabelMe object label similarity as a function of Mask RCNN object detection confidence threshold. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$

	Dependent variable: Mask RCNN to LabelMe object similarity
Threshold	−0.06 (0.06)
Observations	936
R^2	0.001
Log likelihood	431.79

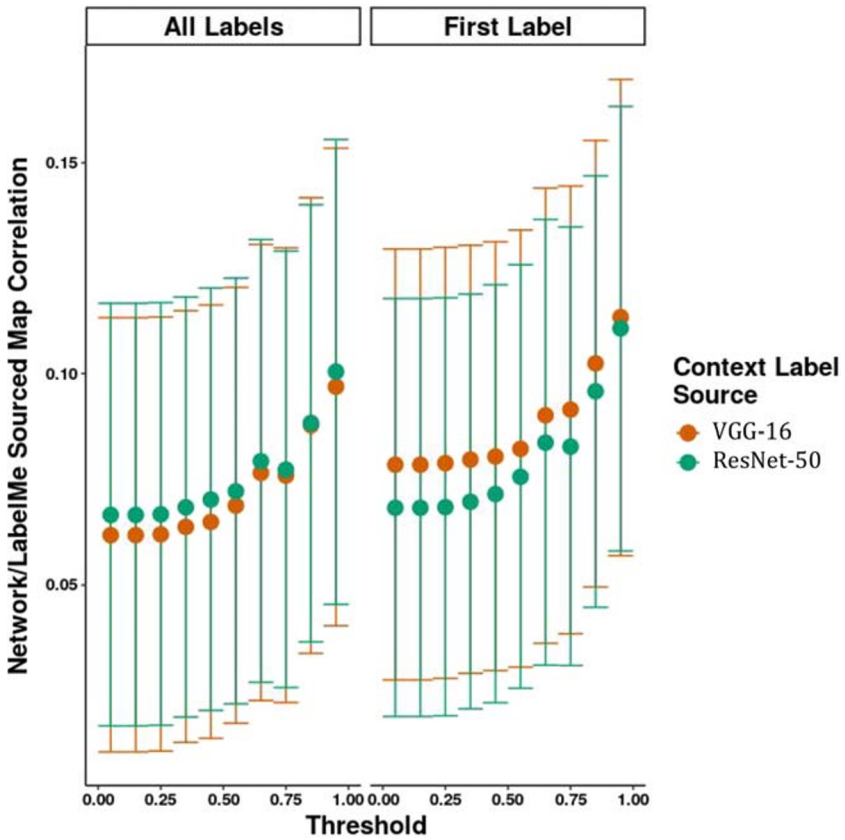

Fig. 17 Means and 95% confidence intervals for network- to LabelMe-derived semantic similarity map correlations across threshold values

Discussion

In this paper, we documented the steps necessary to use a new method – the "Linguistic Analysis of Scene Semantics" or LASS – and provided descriptive results as a form of preliminary use case for it. LASS was created to reduce the time and cost investment necessary to collect human observer data required for the study of scene semantic effects in natural scenes. It extends an existing technique (Hwang et al., 2011) for studying object-to-object semantic relationships in unmodified natural images to the object-to-context case, while simultaneously gaining several desirable properties.

Table 5 Fitted generalized linear model for correlation between Mask RCNN- and LabelMe-derived LASS maps across Mask RCNN object detection confidence threshold values, source of scene context label, and the number of context labels used. $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$

	Dependent variable: Correlation
Threshold	0.04** (0.01)
Context label source (ResNet-50)	−0.002 (0.01)
No. context labels (First only)	0.01 (0.01)
Observations	3,744
Log likelihood	−123.57
Akaike inf. criterion	255.13

First, both LASS and Hwang, Wang, and Pomplun's method depend on an assumption of a first-order relationship between linguistic and visual semantics. While language plays an active role in visual semantic processing, it is likely to be only a *partial* role. However, by accepting as a simplifying assumption that it is the only analytically relevant information, *visual* semantics become amenable to study indirectly using powerful computational *linguistic* semantic tools. LASS's semantic measurement approach given this constraint is significantly more powerful and flexible than that used by Hwang et al. Specifically, LASS uses a related but much newer algorithm, Facebook Research's fastText (Bojanowski et al., 2017), instead of LSA (Landauer et al., 2013). fastText measures semantic similarity between words in terms of nested sets of *n*-gram size sub-word units instead of between entire words.

This permits fastText to evaluate term-to-term relationships between terms that may not have been included in the original training corpus of the model through comparisons between term *parts*. In the context of a noisy and error-filled object annotation corpus such as the one used in this experiment (LabelMe, Russell et al., 2008), this property allows LASS to be applied to arbitrary images without first requiring

subsequent laborious label correction or substitution by the experimenter. Indeed, for the 10,000 images considered in this study, only 20% of the object label classes generated by human observers were contained in the English language dictionary we selected for this experiment[8]. In contrast, 79% of those created by the network were in the same dictionary.

Second, unlike previous works that rely on careful photographic construction and crowd-sourced human labor for generating object segmentation masks and labels (e.g. SCEGRAM, Öhlschläger & Võ, 2017; BOiS, Mohr et al., 2016), LASS can be completely automatic. This automation was achieved through the use of two deep networks: one to segment and label scene objects, and another to generate a set of candidate scene context labels. We also considered the effects of the number of scene context labels used, as well as different scene context label-generating neural network architectures, on the resulting LASS maps. We found that the two scene context label-generating networks – one a VGG-16-based implementation of VGG-16 and the other a ResNet-50 architecture trained on the same data set – generated relatively similar scene context label distributions, despite differences in their most commonly identified context classes. We also found that the effects of these manipulations on the relationships between LabelMe- and Mask RCNN-derived LASS maps were relatively small.

Next, we demonstrated that while its vocabulary of objects is limited, the label sets Mask RCNN generates are significantly more semantically related to human observer-generated labels for the same image than for randomly paired machine/human label sets or random words. Note that the ability of LASS to operate with different sources of object labels allowed for this comparison between human and automated labels. Despite a degree of noise, human and machine observers therefore identify relatively consistent sets of objects, and LASS is sensitive to this consistency. Such a finding conforms to the design of the training corpus of the network (Microsoft's Common Objects in Context, COCO; Lin et al., 2014), which focused its own crowd-sourced label data on ordinary, easily identified objects such as cars and people. This convergence supports the validity of substituting machine for human observer label data in LASS.

LASS depends not only on object and context labels but also on object segmentation masks for mapping semantic relatedness values into the space of the image. Machine vision-based object detection and segmentation also appear to have significantly improved the quality of these data relative to those provided by human observers. Automatically generated object masks for a given image are typically fewer in number, have a smaller interior area, and take shapes that conform more tightly to the boundaries of the identified objects than human-generated masks for the same image. We argue that all three of these properties reflect significantly reduced segmentation noise relative to human observer-generated data, increasing the accuracy in spatial representations of scene semantically relevant information.

One exception to this is the tendency of the network to identify and segment objects toward the center and right side of the display (Fig. 14), with almost no objects identified in the upper left. Human data-derived semantic similarity maps did not show a similar arcuate or boundary effect (Fig. 13). This trend is evident across radial average plots built using different combinations of scene context source labels and numbers of context labels. Despite this difference, object-contextual semantic similarity maps generated by the machine vision and human observers for the same image are significantly correlated, though the magnitude of these correlations is small and their distribution across the corpus highly skewed. Taken together, these results suggest reasonable agreement between human and machine vision observers' judgments of the size, shape, and content of semantically important scene objects. Given the reduction in noise evident in both mask and object label data provided by the network, automatically generated label and mask information should be preferred to equivalent human observer data when possible.

As a simple initial use case for LASS, we evaluated the semantic similarity of map content as a function of distance from the center of the image using a radial average profile. Both the human observer- and the network-generated data show relatively flat, uniform semantic similarity score distributions as a function of distance from the center of the image, with small but statistically significant effects of the source of the object labels (LabelMe vs. Mask RCNN) and the number of scene context labels used (first label only vs. all five available). The first effect is likely an artifact of the Mask RCNN network's failure to detect objects at the left and upper boundaries of the image when compared with human observers. The second results from the sharp reductions in the object-contextual semantic similarity scores observed when objects are compared only to a single scene context label. A small, statistically nonsignificant but importantly *negative* effect of distance from map center on semantic similarity scores was also found, suggesting that objects may become slightly less semantically related to their scene contexts away from the photographic center of images. This result conforms at least partially to the known tendency of photographic objects of interest to be centered in images, though the magnitude is perhaps smaller than expected. Nevertheless, it raises interesting questions regarding the relationships between photographic composition, objects, and scene contextual understanding.

For example, if we take it to mean that less contextually appropriate or unusual objects are found more frequently at the edges of images, does this imply that many images contain transitions or boundaries between different scene contexts

[8] http://wordlist.aspell.net/ , identified as 'final/english-words.95' in the download

where those objects normally occur? Or does it suggest the existence of a second and perhaps less well understood photographic compositional bias toward centering images over regions that best capture the current scene context? We have also only evaluated radial average profiles for semantic similarity maps averaged across different scene contexts. This is reasonable given the primarily descriptive and exploratory nature of this manuscript, but it leaves an interesting question unaddressed: is the distribution of contextually appropriate objects different across contexts? These questions are interesting targets for further research using LASS.

Though powerful, LASS has several potential limitations that experimenters should consider carefully. First, the automation present in its analytical pipeline can introduce new and different sources of noise compared with Hwang and colleagues' method. While the object detection and labeling threshold was set to be relatively conservative in the Mask RCNN implementation used in this manuscript, we have not estimated a ground-truth false-positive rate for its classifications, though its object classification and segmentation performance on COCO is state of the art (He et al., 2017). It is therefore highly likely that a number of objects are incorrectly identified in our image corpus.

However, we have also provided data on LASS's behavior across a range of object detection confidence threshold values. We showed that increasing the object detection confidence threshold past Mask RCNN's default value – even slightly – can significantly improve the correlation between human observer-generated (LabelMe) object label LASS maps and those created using objects identified and segmented using Mask RCNN. Increasing the confidence threshold in this way unfortunately led to a significant increase in the number of images where Mask RCNN failed to identify any objects, and also reduced the semantic similarity of the object labels generated by Mask RCNN to the labels taken from LabelMe for the same image. Given the relative magnitude of the performance gains versus losses resulting from increasing the confidence threshold in this way, we recommend using a slightly elevated confidence threshold, though researchers should evaluate this trade-off relative to their own needs and modify their code base accordingly.

The use of fastText instead of LSA also gives LASS a significantly increased scope of application relative to Hwang and colleagues' approach. However, it is clear that permitting partial matches between terms at several scales may also inflate the estimates of semantic similarity between them (Fig. 10). Distributions of semantic similarity scores between the random words, randomly paired images, and correctly paired images should ideally exhibit greater separation than that shown in these data, despite the fact that the distributional separation tests were statistically significant. This issue can be addressed by using only one or a small number of larger *n*-gram sub-word vectors in the fastText model, though this would require researchers to train a fastText model themselves.

Researchers should also consider whether the default training corpus used for our implementation of fastText – a large dump of Wikipedia data, see Bojanowski et al. (2017) – is suitable to their needs. In the case of narrow, specific, or highly unusual object or context vocabularies of interest, an appropriate existing or custom corpus should be assembled instead. LASS will work regardless of training corpus, but for specialized or rare words that may only co-occur frequently in specific corpora, the Wikipedia corpus is likely to underestimate their semantic similarity.

A final issue arises in the range of values that LASS generates by default. Specifically, LASS confounds the zero of the semantic similarity space with the zero value assigned to regions of the image that contain no object labels. This is only a *potential* issue, as the likelihood of semantic similarity scores being *exactly* zero as they are for unlabeled regions is small. Nevertheless, it remains possible to incorrectly interpret true zero scores in semantic similarity maps as regions containing objects that are entirely semantically unrelated to the scene context. Simple qualitative examination of maps generated using LASS against their source images demonstrates that this is untrue: there are clearly recognizable objects in image regions identified in this way. This and the related issues of object mask discreteness and the often modest object-level pixel coverage produced by Mask RCNN make careful object detection model parameter selection crucial for deploying LASS, and without modification, make it less suitable for use cases such as gaze prediction.

Nevertheless, both our findings and these identified limitations underscore LASS's greatest asset: its *flexibility*. With the exception of the use of fastText and the need to map semantic information to image space, no component of the described analytical pipeline is strictly necessary to guarantee performance consistent with that reported here. Experimenters are free to substitute machine vision-generated label and segmentation mask data for equivalents produced by human observers, adjust the *n*-gram size and size range used in the fastText model, or combine information across context-specific maps as they wish.

Perhaps the best example of this flexibility is that LASS's behavior acts as a superset of Hwang and colleagues': simply substitute object-contextual for object-object label comparisons. Because the analytical pipeline described here is fully automatic and uses only relatively lightweight computer vision tools, this flexibility can be quickly and easily leveraged to develop new and highly context-specific variants suited to a wide range of interesting research questions and areas, while also retaining interpretive consistency across results. This combination of speed, flexibility, and power is unique among existing approaches for studying scene and object semantic effects, and can help open new avenues of research in these and many other domains.

References

Altszyler, E., Sigman, M., Ribeiro, S., & Slezak, D. F. (2016). Comparative study of LSA vs Word2vec embeddings in small corpora: A case study in dreams database. *arXiv preprint arXiv: 1610.01520*.

Anderson, S. E., Chiu, E., Huette, S., & Spivey, M. J. (2011). On the temporal dynamics of language-mediated vision and vision-mediated language. *Acta Psychologica*, *137*(2), 181–189.

Becker, M. W., Pashler, H., & Lubin, J. (2007). Object-intrinsic oddities draw early saccades. *Journal of Experimental Psychology: Human Perception and Performance*, *33*(1), 20–30.

Biederman, I., Mezzanotte, R. J., & Rabinowitz, J. C. (1982). Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, *14*(2), 143–177.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information.

Brockmole, J. R., & Le-Hoa Vo, M. (2010). Semantic memory for contextual regularities within and across scene categories: Evidence from eye movements. *Attention, Perception & Psychophysics*, *72*(7), 1803–1813.

Chollet, F. (2015). Keras.

Coco, M. I., Araujo, S., & Petersson, K. M. (2017). Disentangling stimulus plausibility and contextual congruency: Electro-physiological evidence for differential cognitive dynamics. *Neuropsychologia*, *96*, 150–163.

Coco, M. I., & Keller, F. (2014). Classification of Visual and Linguistic Task Features using Eye-movement Features. *JOV*, *14*(3).

Cribari-Neto, F., & Zeileis, A. (2010). Beta Regression in R. *Journal of Statistical Software*, *34*(2).

Draschkow, D., Wolfe, J. M., & Vo, M. L. H. (2014). Seek and you shall remember: Scene semantics interact with visual search to build better memories. *Journal of Vision*, *14*(8), 10–10.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '88* (pp. 281–285). Washington, D.C., United States: ACM Press.

Hayhoe, M., & Ballard, D. (2005). Eye movements in natural behavior. *Trends in Cognitive Sciences*, *9*(4), 188–194.

Hayhoe, M. M., Shrivastava, A., Mruczek, R., & Pelz, J. B. (2003). Visual memory and motor planning in a natural task. *Journal of vision*, *3*(1), 6–6.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Computer Vision (ICCV), 2017 IEEE International Conference on* (pp. 2980–2988). IEEE.

Henderson, J. M., & Ferreira, F. (Eds.). (2004). *The interface of language, vision, and action: Eye movements and the visual world*. New York: Psychology Press.

Hollingworth, A. (1998). Does consistent scene context facilitate object perception? *Journal of Experimental Psychology: General*, *127*(4), 398.

Hwang, A. D., Wang, H.-C., & Pomplun, M. (2011). Semantic guidance of eye movements in real-world scenes. *Vision Research*, *51*(10), 1192–1205.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (2013). *Handbook of Latent Semantic Analysis*.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., et al. (2014). Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*. Retrieved from http://arxiv.org/abs/1405.0312

Loftus, G. R., & Mackworth, N. H. (1978). Cognitive Determinants of Fixation Location During Picture Viewing. *Journal of Experimental Psychology: Human Perception and Performance*, *4*(4), 562–572.

Mamassian, P., Knill, D. C., & Kersten, D. (1998). The perception of cast shadows. *Trends in Cognitive Sciences*, *2*(8), 288–295.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mohr, J., Seyfarth, J., Lueschow, A., Weber, J. E., Wichmann, F. A., & Obermayer, K. (2016). BOiSBerlin Object in Scene Database: Controlled Photographic Images for Visual Search Experiments with Quantified Contextual Priors. *Frontiers in Psychology*, *7*.

Öhlschläger, S., & Võ, M. L.-H. (2017). SCEGRAM: An image database for semantic and syntactic inconsistencies in scenes. *Behavior Research Methods*, *49*(5), 1780–1791.

Olshausen, B. A., & Field, D. J. (2005). How close are we to understanding V1? *Neural computation*, *17*(8), 1665–1699.

Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, *21*(5), 1112–1130.

Rehurek, R., & Sojka, P. (2011). GensimPython framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic.

Richardson, D. C., Dale, R., & Spivey, M. J. (2007). Eye movements in language and cognition. In M. Gonzalez-Marquez (Ed.), *Methods in cognitive linguistics*, Human cognitive processing. Amsterdam ; Philadelphia: John Benjamins Pub.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe a database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173.

Rust, N. C., & Movshon, J. A. (2005). In praise of artifice. *Nature Neuroscience*, *8*(12), 1647–1650.

Sahlgren, M. (2008). The distributional hypothesis. *Rivista di Linguistica*, *20*(1), 18.

Thalenberg, B. (2008). Distinguishing Antonyms from Synonyms in Vector Space Models of Semantics, 6.

Underwood, G., & Foulsham, T. (2006). Visual saliency and semantic incongruency influence eye movements when inspecting pictures. *Quarterly Journal of Experimental Psychology*, *59*(11), 1931–1949.

Võ, M. L.-H., & Henderson, J. M. (2011). ObjectScene inconsistencies do not capture gaze: Evidence from the flash-preview moving-window paradigm. *Attention, Perception, & Psychophysics*, *73*(6), 1742–1753.

Võ, M. L. H., & Wolfe, J. M. (2013). Differential Electrophysiological Signatures of Semantic and Syntactic Scene Processing. *Psychological Science*, *24*(9), 1816–1823.

Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., & Oliva, A. (2016). Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*.