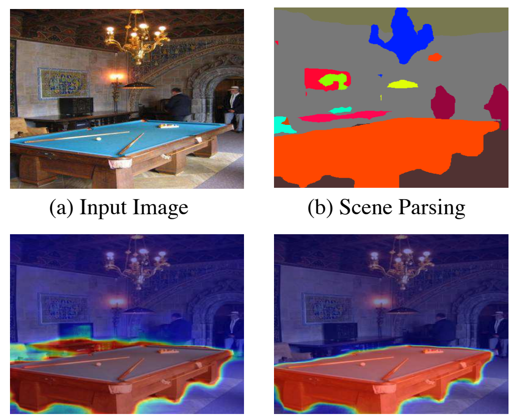
Open Vocabulary Scene Parsing

Hang Zhao[1], Xavier Puig[1], Bolei Zhou[1], Sanja Fidler[2], Antonio Torralba[1]
[1]Massachusetts Institute of Technology, USA
[2]University of Toronto, Canada

Abstract

Recognizing arbitrary objects in the wild has been a challenging problem due to the limitations of existing classification models and datasets. In this paper, we propose a new task that aims at parsing scenes with a large and open vocabulary, and several evaluation metrics are explored for this problem. Our approach is a joint image pixel and word concept embeddings framework, where word concepts are connected by semantic relations. We validate the open vocabulary prediction ability of our framework on ADE20K dataset which covers a wide variety of scenes and objects. We further explore the trained joint embedding space to show its interpretability.



(a) Input Image (b) Scene Parsing (c) Concept Retrieval (d) Concept Synthesis

Figure 1. We propose an open vocabulary framework such that given (a) an input image, we can perform (b) scene parsing, (c) concept retrieval (”table”), and (d) concept synthesis (intersection of “game equipment” and “table”) through arithmetic operations in the joint image-concept embedding space.

1. Introduction

One of the grand goals in computer vision is to recognize and segment arbitrary objects in the wild. Recent efforts in image classification/detection/segmentation have shown this trend: emerging image datasets enable recognition on a large scale [6, 30, 32], while image captioning can be seen as a special instance of this task [12]. However, nowadays most recognition models are still not capable of classifying objects at the level of a human, in particular, taking into account the taxonomy of object categories. Ordinary people or laymen classify things on the entry-levels, and experts give more specific labels: there is no object with a single correct label, so the prediction vocabulary is inherently open-ended. Furthermore, there is no widely-accepted way to evaluate open-ended recognition tasks, which is also a main reason this direction is not pursued more often.

In this work, we are pushing towards open vocabulary scene parsing: model predictions are not limited to a fixed set of categories, but also concepts in a larger dictionary, or even a knowledge graph. Considering existing image parsing datasets only contain a small number of categories (~100 classes), there is much more a model can learn from those images given extra semantic knowledge, like WordNet dictionary (~100,000 synsets) or *Word2Vec* from external corpus.

To solve this new problem, we propose a framework that is able to segment all objects in an image using open vocabulary labels. In particular, while the method strives to label each pixel with the same word as the one used by the human annotator, it resorts to a taxonomy when it is not sure about its prediction. As a result, our model can make plausible predictions even for categories that have not been shown during training, *e.g.* if the model has never seen *tricycle*, it may still give a confident guess on *vehicle*, performing more like a human.

Our framework incorporates hypernym/hyponym relations from WordNet [18] to help with parsing. More concretely, word concepts and image pixel features are embedded into a joint high-dimentional vector space so that (1) hypernym/hyponym relations are preserved for the concepts, (2) image pixel embeddings are close to concepts related to their annotations according to some distance measures. This framework offers three major advantages: (1) predictions are made in a structured way, *i.e.*, they can be intermediate nodes in WordNet, and thus yielding more reasonable mistakes; (2) it is an end-to-end trainable system, its vocabulary can be huge and is easily extensible; (3) the framework leaves more freedom to the annotations: inconsistent annotations from workers with different domain knowledge have less of an affect on the performance of the model.

We explore several evaluation metrics, which are useful measures not only for our open vocabulary parsing tasks, but also for any large-scale recognition tasks where confusions often exist. The open vocabulary parsing ability of the proposed framework is evaluated on the recent ADE20K dataset [33]. We further study the properties of the embedding space by loosing classification boundary, concept retrieval, and concept synthesis with arithmetics.

1.1. Related work

Semantic segmentation and scene parsing. Due to astonishing performance of deep learning, in particular CNNs [14], pixel-wise dense labeling has received significant amount of attention. Popular architectures include fully convolutional neural network (FCN) [17], deconvolutional neural network [19], encoder-decoder SegNet [2], dilated neural network [3, 31], *etc*. These networks perform well on datasets like PASCAL VOC [8] with 20 object categories, Cityscapes [4] with 30 classes, and a recently released benchmark SceneParse150 [33] covering 150 most frequent daily objects. However, they are not easily adaptable to new objects. In this paper we aim at going beyond this limit and to make predictions in the wild.

Zero-shot learning. Zero-shot learning addresses knowledge transfer and generalization [24, 10]. Models are often evaluated on unseen categories, and predictions are made based on the knowledge extracted from the training categories. Rohrbach [25] introduced the idea to transfer large-scale linguistic knowledge into vision tasks. Socher *et al.* [27] and Frome *et al.* [9] directly embedded visual features into the word vector space so that visual similarities are connected to semantic similarities. Norouzi *et al.* [20] used a convex combination of visual features of training classes to represent new categories. Attribute-based methods map object attribute labels or language descriptions to visual classifiers [22, 1, 16, 15].

Hierarchical classifications. Hierarchical classification addresses the common circumstances that candidate categories share hierarchical semantic relations. Deng *et al.* [7] achieved hierarchical image-level classification by trading off accuracy and gain as an optimization problem. Ordonez *et al.* [21], on the other hand, proposed to make entry-level predictions when dealing with a large number of categories. More recently, Deng *et al.* [5] formulated a label relation graph that could be directly integrated with deep neural networks.

While embedding-based approaches cannot embed knowledge from semantic graphs, optimization-based methods do not have the ability to generalize to new/zero-shot concepts. Our approach on hierarchical parsing is inspired by the order-embeddings work [28], we attempt to construct an asymmetric embedding space, so that both image features and hierarchy from knowledge graphs are effectively and implicitly encoded by the deep neural networks. The major advantage of our approach is that it makes an end-to-end trainable network, which is easily scalable when dealing with larger datasets in practical applications.

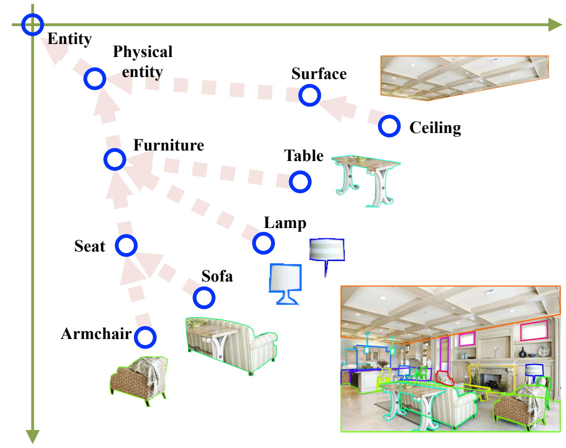


Figure 2. Jointly embedding vocabulary concepts and image pixel features.

2. Learning joint embeddings for pixel features and word concepts

We treat open-ended scene parsing as a retrieval problem for each pixel, following the ideas of image-caption retrieval work [28]. Our goal is to embed image pixel features and word concepts into a joint high-dimensional positive vector space $\mathbb{R}_+^N$, as illustrated in Figure 2. The guiding principle while constructing the joint embedding space is that image features should be close to their concept labels, and word concepts should preserve their semantic hypernym/hyponym relations. In this embedding space, (1) vectors close to origin are general concepts, and vectors with larger norms represent higher specificity; (2) hypernym/hyponym relation is defined by whether one vector is smaller/greater than another vector in all the N dimensions. A hypernym scoring function is crucial in building this embedding space, which will be detailed in Section 2.1.

Figure 3 gives an overview of our proposed framework. It is composed of two streams: a concept stream and an image stream. The concept stream encodes the pre-defined semantics: it learns an embedding function $f(\cdot)$ that maps the words into $\mathbb{R}_+^N$ so that the hypernym/hyponym relationships between word concepts are preserved. The image stream $g(\cdot)$ embeds image pixels into the same space by pushing them close to their labels (word concepts). We describe these two streams in more details in Section 2.2 and 2.3.

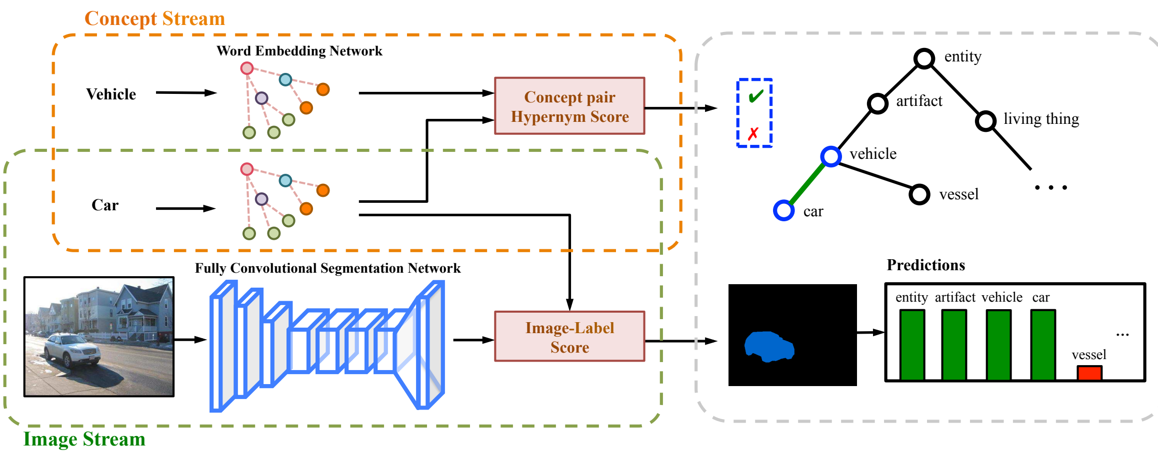


Figure 3. The open vocabulary parsing network. The concept stream encodes word concept hierarchy based on dictionaries like WordNet. The image stream parses images based on the learned hierarchy.

2.1. Scoring functions

In this embedding problem, training is performed on pairs: image-label pairs and concept-concept pairs. For either of the streams, the goal is to maximize scores of matching pairs and minimize scores of non-matching pairs. So the choice of scoring functions $S(x, y)$ becomes important. There are symmetric scoring functions like L_p distance and cosine similarity widely used in the embedding tasks,

$$S_{Lp}(x, y) = -\|x - y\|_p, \quad S_{cos}(x, y) = x \cdot y. \quad (1)$$

In order to reveal the asymmetric hypernym/hyponym relations between word concepts, a hypernym scoring function [28] is indispensable,

$$S_{hyper}(x, y) = -\|max(0, x - y)\|_p. \quad (2)$$

If x is hypernym of y ($x \succeq y$), then ideally all the coordinates of x are smaller than y ($\bigwedge_i (x_i \leq y_i)$), so $S_{hyper}(x, y) = S_{hyper,max} = 0$. Note that due to asymmetry, swapping x and y will result in different scores.

2.2. Concept stream

The objective of the concept stream is to build up semantic relations in the embedding space. In our case, the semantic hierarchy is obtained from WordNet hypernym/hyponym relations. Consider all the vocabulary concepts form a directed acyclic graph (DAG) $H = (V, E)$, sharing a common root $\hat{v} \in V$ "entity", each node in the graph $v \in V$ can be an abstract concept as the unions of its children nodes, or a specific class as a leaf. A visualization of part of the DAG we built based on WordNet and ADE20K labels can be found in Supplementary Materials.

Internally, the concept stream include parallel layers of a shared trainable lookup table, mapping the word concepts u, v to $f(u)$, $f(v)$. And then they are evaluated on hypernym scores $S_{concept}(f(u), f(v)) = S_{hyper}(f(u), f(v))$, which tells how confident u is a hypernym of v. A max-margin loss is used to learn the embedding function $f(\cdot)$,

$$\mathcal{L}_{concept}(u, v) = \begin{cases} -S_{concept}(f(u), f(v)) & \text{if } u \succeq v, \\ \max\{0, \alpha + S_{concept}(f(u), f(v))\} & \text{otherwise} \end{cases}$$

Note that positive samples $u \succeq v$ are the cases where u is an ancestor of v in the graph, so all the coordinates of $f(v)$ are pushed towards values larger than $f(u)$; negative samples can be inverted pairs or random pairs, the loss function pushes them apart in the embedding space. In our training, we fix the root of DAG "entity" as anchor at origin, so the embedding space stays in $\mathbb{R}^N_+$.

2.3. Image stream

The image stream is composed of a fully convolutional network which is commonly used in image segmentation tasks, and a lookup layer shared with the word concept stream. Consider an image pixel at position (i, j) with label $x_{i,j}$, its feature $y_{i,j}$ is the top layer output of the convolutional network. Our mapping function $g(y_{i,j})$ embeds the pixel features into the same space as their label $f(x_{i,j})$, and then evaluate them with a scoring function $S_{image}(f(x_{i,j}), g(y_{i,j}))$.

As label retrieval is inherently a ranking problem, negative labels $x'_{i,j}$ are introduced in training. A max-margin ranking loss is commonly used [9] to encourage the scores of true labels be larger than negative labels by a margin,

$$\mathcal{L}_{image}(y_{i,j}) = \sum_{x'_{i,j}} \max\{0, \beta - S_{image}(f(x_{i,j}), g(y_{i,j})) + S_{image}(f(x'_{i,j}), g(y_{i,j}))\}. \quad (3)$$

In the experiment, we use a softmax loss for all our models and empirically find better performance,

$$\mathcal{L}_{image}(y_{i,j}) = \\ -\log \frac{e^{S_{image}(f(x_{i,j}),g(y_{i,j}))}}{e^{S_{image}(f(x_{i,j}),g(y_{i,j}))} + \sum_{x'_{i,j}} e^{S_{image}(f(x'_{i,j}),g(y_{i,j}))}}. \tag{4}$$

This loss function is a variation of triplet ranking loss proposed in [11].

The choice of scoring function here is flexible, we can either (1) simply make image pixel features "close" to the embedding of their labels by using symmetric scores $S_{L_p}(f(x_{i,j}), g(y_{i,j}))$, $S_{cos}(f(x_{i,j}), g(y_{i,j}))$; (2) or use asymmetric hypernym score $S_{hyper}(f(x_{i,j}), g(y_{i,j}))$. In the latter case, we treat images as specific instances or specializations of their label concepts, and labels as general abstraction of the images.

2.4. Joint model

Our joint model combines the two streams via a joint loss function to preserve concept hierarchy as well as visual feature similarities. In particular, we simply weighted sum the losses of two streams $\mathcal{L} = \mathcal{L}_{image} + \lambda\mathcal{L}_{concept} (\lambda = 5)$ during training. We set the embedding space dimension to $N = 300$, which is commonly used in word embeddings. Training and model details are described in Section 4.2.

3. Evaluation Criteria

3.1. Baseline flat metrics

While working on a limited number of classes, four traditional criteria are good measures of the scene parsing model performance: (1) pixel-wise accuracy: the proportion of correctly classified pixels; (2) mean accuracy: the proportion of correctly classified pixels averaged over all the classes; (3) mean IoU: the intersection-over-union averaged over all the classes; (4) weighted IoU: the IoU weighted by pixel ratio of each class.

3.2. Open vocabulary metrics

Given the nature of open vocabulary recognition, selecting a good evaluation criteria is non-trivial. Firstly, it should leverage the graph structure of the concepts to tell the distance of the predicted class from the ground truth. Secondly, the evaluation should correctly represent the highly unbalanced distribution of the dataset classes, which are also common in the objects seen in nature.

For each sample/pixel, a score $s(l, p)$ is used to measure the similarity between the label s and the prediction p. The final score is the mean score over all the samples.

3.2.1 Hierarchical precision, recall and F-score

Hierarchical precision, recall and F-score are known as Wu-Palmer similarity, which was originally used for lexical selection [29].

For two given concepts l and p, we define the lowest common ancestor $\texttt{LCA}$ as the most specific concept (i.e. furthest from the root Entity) that is an hypernym of both. Then hierarchical precision and recall are defined by the number of common hypernyms that prediction and label have over the vocabulary hierarchy H, formally:

$$s_{HP}(l,p) = \frac{d_{\texttt{LCA}}}{d_p}, \quad s_{HR}(l,p) = \frac{d_{\texttt{LCA}}}{d_l}, \tag{5}$$

where d is the depth of certain concept node in H.

Combining hierarchical precision and hierarchical recall, we get hierarchical F-score $s_{HF}(l,p)$, defined as the depth of $\texttt{LCA}$ node over the sum of depth of label and prediction nodes:

$$s_{HF}(l,p) = \frac{2 s_{HP}(l,p) \cdot s_{HR}(l,p)}{s_{HP}(l,p) + s_{HR}(l,p)} = \frac{2 \cdot d_{\texttt{LCA}}}{d_l + d_p}. \tag{6}$$

One prominent advantage of these hierarchical metrics is they penalize predictions when being too specific. For example, "guitar" (d_l=10) and "piano" (d_p=10) are all "musical instrument" ($d_{\texttt{LCA}}$=8). When "guitar" is predicted as "piano", $s_{HF} = \frac{2 \cdot 8}{10+10} = 0.8$; when "guitar" is predicted as "musical instrument", $s_{HF} = \frac{2 \cdot 8}{10+8} = 0.89$. It agrees with human judgment that the prediction "musical instrument" is more *accurate* than "piano".

3.2.2 Information content ratio

Performance could be dominated by frequent classes when distribution of data points is unbalanced. *Information content ratio*, which was also used in lexical search, addresses these problems effectively.

According to information theory and statistics, the information content of a message is the inverse logarithm of its frequency $I(c) = -\log P(c)$. We inherit this idea and obtain the pixel frequency of each concept $v \in H$. Specifically, the frequency of a concept is the sum of its own frequency and all its descendents' frequencies in the image dataset. It is expected that the root "entity" has frequency 1.0 and information content 0.

During evaluations, we measure, how much information our prediction gets out of the amount of information in the label. So the final score is determined by the information of the ground truth, prediction and $\texttt{LCA}$:

$$s_I(l,p) = \frac{2 \cdot I_{\texttt{LCA}}}{I_l + I_p} = \frac{2 \cdot \log P(\texttt{LCA})}{\log P(l) + \log P(p)} \tag{7}$$

As information content ratio considers dataset statistics and semantic hierarchy, it rewards both inference difficulty and hierarchical accuracy.

Table 1. Scene parsing performance on 150 classes, evaluated with flat metrics.

Networks	Pixel Accuracy	Mean Accuracy	Mean IoU	Frequency Weighted IoU
Softmax [33]	**73.55%**	**44.59%**	**0.3231**	**0.6014**
Conditional Softmax [23]	72.23%	42.64%	0.3127	0.5942
Word2Vec [9]	71.31%	40.31%	0.2918	0.5879
Word2Vec+	73.11%	42.31%	**0.3160**	0.5998
Image-L2	70.18%	38.89%	0.2174	0.4764
Image-Cosine	71.40%	40.17%	0.2803	0.5677
Image-Hyper	67.75%	37.10%	0.2158	0.4692
Joint-L2	71.48%	39.88%	0.2692	0.5642
Joint-Cosine	**73.15%**	**43.01%**	0.3152	**0.6001**
Joint-Hyper	72.74%	42.29%	0.3120	0.5940

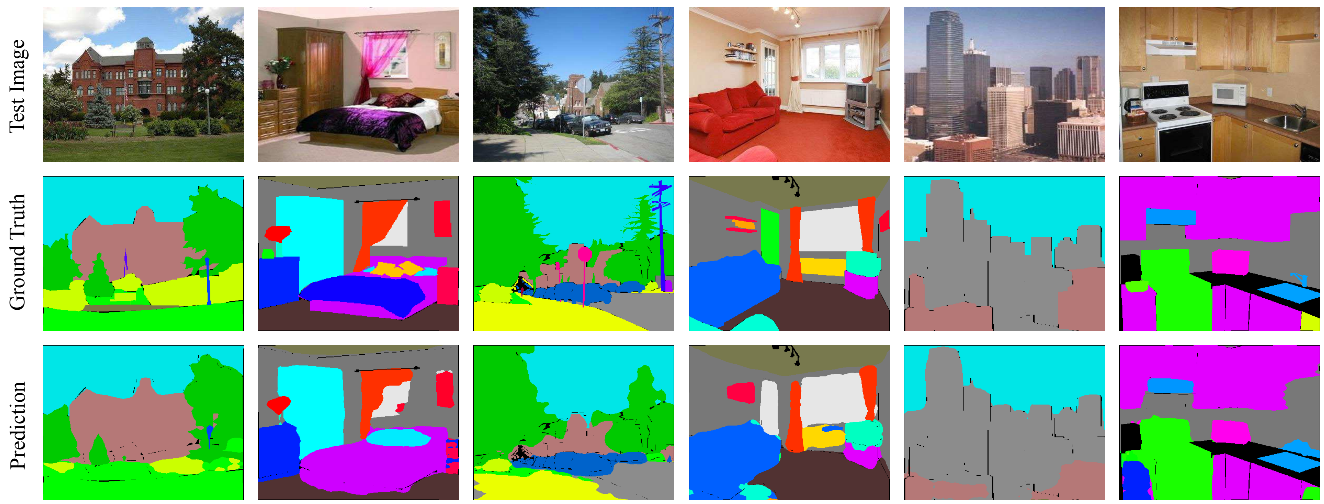


Figure 4. Scene parsing results on 150 classes, images are nearly fully segmented.

4. Experiments

4.1. Image label and concept association

We associate each class in ADE20K dataset with a *synset* in WordNet, representing a unique concept. The data association process requires semantic understanding, so we resort to Amazon Mechanical Turks (AMTs). We develop a rigorous annotation protocol, which is detailed in Supplementary Materials.

After association, we end up with 3019 classes in the dataset having synset matches. Out of these there are 2019 unique synsets forming a DAG. All the matched synsets have *entity.n.01* as the top hypernym and there are in average 8.2 synsets in between. The depths of the ADE20K dataset annotations range from 4 to 19.

4.2. Network implementations

4.2.1 Concept stream

The concept stream takes in positive and negative concept pairs. The positive training pairs are found by traversing the graph H and find all the transitive closure hypernym pairs, *e.g.* "neckwear" and "tie", "clothing" and "tie", "entity" and "tie"; negative samples are randomly generated by excluding these positive samples.

4.2.2 Image stream

Our core CNN in the image stream is adapted from VGG-16 by taking away *pool4* and *pool5* and then making all the following convolution layers dilated (or Atrous) [3, 31]. Considering the features of an image pixel from the last layer of the fully convolutional network *fc7* to be $y_{i,j}$ with dimension 4096, we add a 1×1 convolution layer $g(\cdot)$ with weight dimension of 4096×300 to embed the pixel feature. To ensure positivity, we further add a *ReLU* layer.

To improve the numerical stability of training, we fix the norms of the embeddings of image pixels to be 30, where a wide range of values will work. Intuitively, fixing image to have a large norm makes sense in the hierarchical embedding space: image pixels are most specific descriptions of concepts, while words are general, and closer to the origin.

4.2.3 Training and inference

In all the experiments, we first train the concept stream to get the word embeddings, and then use them as initializations in the joint training. Image stream is initialized by pre-trained weights from VGG-ImageNet [26].

Adam optimizer [13] with learning rate 1e-3 is used to update weights across the model. The margin of loss functions is default to $\alpha = 1.0$.

Table 2. Zero-shot parsing performance, evaluated with hierarchical metrics.

Networks	Hierarchical Precision	Hierarchical Recall	Hierarchical F-score	Information content ratio
Softmax [33]	0.5620	0.5168	0.5325	0.1632
Conditional Softmax [23]	0.5701	0.5146	0.5340	0.1657
Word2Vec [9]	0.5782	0.5265	0.5507	0.1794
Convex Combination [20]	0.5777	0.5384	0.5492	0.1745
Word2Vec+	0.6138	0.5248	0.5671	0.2002
Image-L2	0.5741	0.5032	0.5375	0.1650
Image-Hyper	0.6318	0.5346	0.5937	0.2136
Joint-L2	0.5956	0.5385	0.5655	0.1945
Joint-Hyper	**0.6567**	**0.5838**	**0.6174**	**0.2226**

Figure 5. Zero-shot parsing results on the infrequent object classes.

In the inference stage, there are two cases: (1) While testing on the 150 training classes, the pixel embeddings are compared with the embeddings of all the 150 candidate labels based on the scoring function, the class with the highest score is taken as the prediction; (2) While doing zero-shot predictions, on the other hand, we use a threshold on the scores to decide the cutoff score, concepts with scores above the cutoff are taken as predictions. This best threshold is found before testing on a set of 100 validation images.

4.3. Results on scene parsing

In this section, we report the performance of our model on scene parsing task. Training is performed on the most frequent 150 classes of stuffs and objects in the ADE20K dataset, where each of the class has at least 0.02% of total pixels in the dataset.

We have trained some models in the references and several variants of our proposed model, all of which share the same core CNN to make fair comparisons. *Softmax* is the baseline model that does classical multi-class classification.

Conditional Softmax is a hierarchical classification model proposed in [23]. It builds a tree based on the label relations, and softmax is performed only between nodes of a common parent, so only conditional probabilities for each node are computed. To get absolute probabilities during testing, the conditional probabilities are multiplied following the paths to root.

Word2Vec regresses the image pixel features to pretrained word embeddings, where we use the GoogleNews vectors. Cosine similarity and max-margin ranking loss with negative samples are used. This model is a direct counterpart of DeViSe[9] in our scene parsing settings.

Word2Vec+ is our improved version of *Word2Vec* model. Max margin loss is replaces by a softmax loss as mentioned in Section 2.3;

There are 6 variants of our proposed model. Model names with *Image-** refer to the cases where only image stream is trained, by fixing the concept embeddings. In models *Joint-** we train two streams together to learn a joint embedding space. Three aforementioned scoring functions are used for the image stream, their corresponding models are marked as **-L2*, **-Cosine* and **-Hyper*.

4.3.1 Performance on 150 classes

Evaluating on the 150 training classes, our proposed models offer competitive results. Baseline flat metrics are used to compare the performance, as shown in Table 1. Without surprise, the best performance is achieved by the *Softmax* baseline, which agrees with the observation from [9], classification formulations usually achieves higher accuracy than regression formulations. At the same time, our proposed models *Joint-Cosine* and *Word2Vec+* fall short of *Softmax*

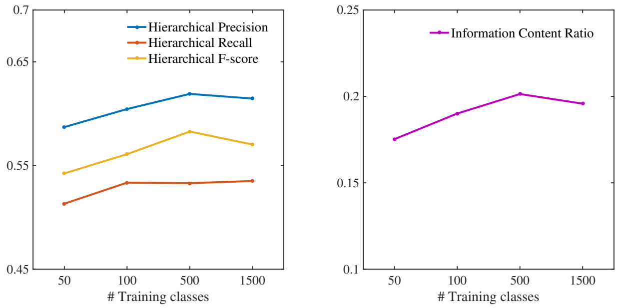


Figure 6. Diversity test, evaluated with hierarchical metrics.

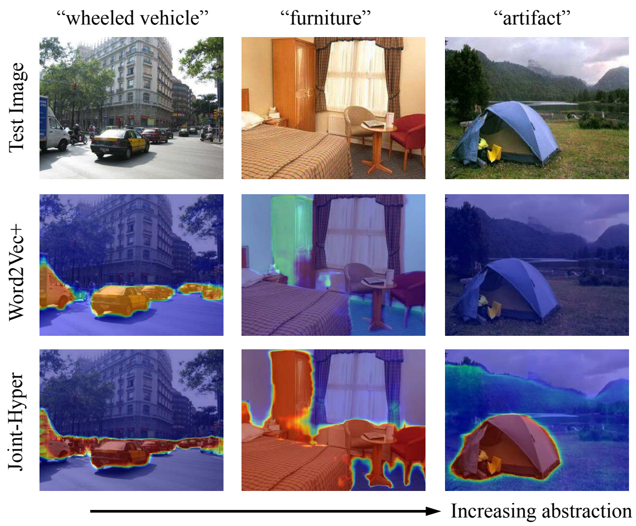


Figure 7. Pixel-level concept search with increasing abstraction.

by only around 1%, which is an affordable sacrifice given the zero-shot prediction capability and interpretability that will be discussed later. Visual results of the best proposed model *Joint-Cosine* are shown in Figure 4.

4.3.2 Zero-shot predictions

We then move to the zero-shot prediction tasks to fully leverage the hierarchical prediction ability of our models. The models are evaluated on 500 less frequent object classes in the ADE20K dataset. Predictions can be in the 500 classes, or their hypernyms, which could be evaluated with our open vocabulary metrics.

Softmax and *Conditional Softmax* models are not able to make inferences outside the training classes, so we take their predictions within the 150 classes for evaluation.

Convex Combination [20] is another baseline model: we take the probability output from *Softmax* within the 150 classes, to form new embeddings in the word vector space, and then find the nearest neighbors in vector space. This approach does not require re-training, but still offers reasonable performance.

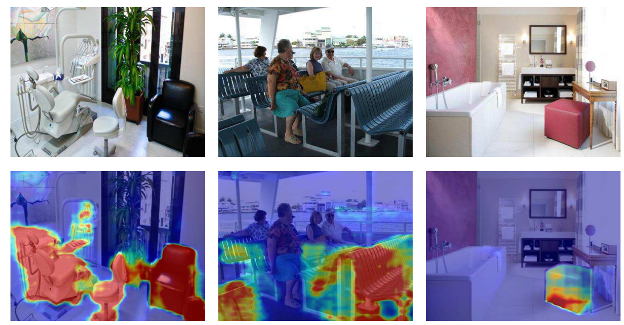


Figure 8. Sittable objects have high scores while retrieving "chair", indicating abstract attributes encoded in the embedding space.

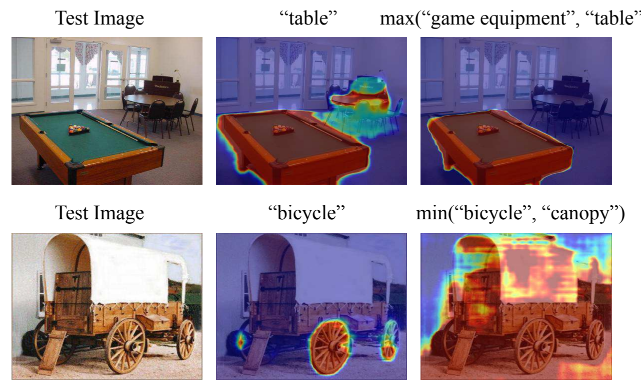


Figure 9. Pixel-level search with synthesized concepts through arithmetic operations. Intersections and unions are achieved in the embedding space by `max` and `min`.

Most of our proposed models can retrieve the hypernyms of the testing classes, except **-Cosine* as they throw away the norm information during scoring, which is important for hypernym predictions.

Table 2 shows results on zero-shot predictions. In terms of the hierarchical metrics, *Joint-Hyper* gives the best performance. And our proposed models in general win by a large margin over baseline methods. It confirms us that modeling the asymmetric relations of data pairs better represents the hierarchy. Figure 5 shows some prediction samples of our best model *Joint-Hyper* (see Supplementary Materials for full predictions of our model). In each image, we only show one ground truth category to make clear visualizations, different colors represent different predictions. Though the model does not always get the ground truth labels exactly correct, it gives reasonable predictions. Another observation is that predictions are sometimes noisy, we get 2-3 predictions on a single objects. Some of the inconsistencies are plausible though, *e.g.* in the first row, the upper part of the "rocking chair" is predicted as "chair" while the lower part is predicted as "furniture". As the pixels in the upper segment are *closer* to ordinary chairs while the lower segment does not, so in the latter case the model gives a more general prediction.

4.4. Diversity test

The open vocabulary recognition problem naturally raises a question: how many training classes do we need to generalize well on zero-shot tasks? To answer this question, we do a diversity test in this section.

Different from the previous experiments, we do not take the most frequent classes for training, instead uniformly sample training and testing classes from the histogram of pixel numbers. For better comparison, we fix the number of zero-shot test set classes to be 500, and the training classes range from 50 to 1500. In the training process, we offset the unbalance in pixel numbers by weighting the training class loss with their corresponding information content, so the less frequent classes contribute higher loss.

We only experiment with our best model *Joint-Hyper* for this diversity test. Results in Figure 6 suggest that performance could saturate after training with more than 500 classes. We conjecture that training with many classes with few instances could introduce sample noises. So to further improve performance, more high quality data is required.

5. Interpreting the embedding space

The joint embedding space we trained earlier features different properties from known spaces like *Word2Vec*. In this section, we conduct three qualitative tests to explore these properties.

Concept search. In our framework, the joint training does not require all the concepts to have corresponding image data, as semantics can be propagated. This enables us to search with concepts that are not trained with images at test time, and visualize their activations in images. Given a search concept, we obtain its embedding $f(x)$ from the concept stream, and calculate per-pixel score of target image features $g(y_{i,j})$ according to scoring function. Results are shown in Figure 7, with heatmaps representing the scores. *Joint-Hyper* and *Word2Vec+* perform equally well when searching for specific concepts. But as the search concepts become increasingly abstract, our model far outperforms *Word2Vec+*, indicating the effective encoding of hierarchical information in our embedding space.

Implicit attributes encoding. One intriguing property of feature embeddings is that it is a continuous space, and classification boundaries are flexible. So we explore the vicinity of some concepts. In Figure 8, we show score maps when searching for the concept "chair". Interestingly, it is a common phenomenon that objects like "bench" and "ottoman", which are not hyponyms of "chair" in WordNet, get reasonable response. We conjecture that the embedding space implicitly encodes some abstract attributes by clustering them, *e.g. sittable* is an affordance attribute. So by loosing classification threshold of "chair", one can detect regions where one can sit on.

Concept synthesis with arithmetics. Similar to *Word2Vec*, in our joint embedding space, new concepts or object detectors can be synthesized with arithmetics. Given two concepts, we take elementwise `min` or `max` operations on their embeddings $f(x_1)$ and $f(x_2)$ to synthesize a new embedding, and then search for the synthesized concepts in the images, results are shown in Figure 9. It can be seen that `max` operation takes the intersection of the concepts, *e.g.* the "pool table" is a common hyponym of "table" and "game equipment"; and `min` takes the union, *e.g.* the "cart" is composed of attributes of "bicycle" and "canopy". These observations agree with the fact that the embedding space encodes hypernym/hyponym relations.

6. Discussions

Benefits on annotations. Our learning framework offers more freedom for open vocabulary annotations: annotators can freely find a closest concept in the dictionary. People with different domain knowledge might label an object at different depths of the knowledge graph, *e.g.* labeling "Husky" as "dog". This inconsistency does not harm the training of our model as our formulation inherently considers hierarchical relations.

Making general or specific predictions? In hierarchical classification problems, there is no consensus on whether to make general or specific predictions. Human are more tolerant of general concepts than incorrect specific concepts. In our framework, it is dependent on the cutoff threshold in the inference stage, so we could choose to balance precision and recall.

Limitations. Similar to other zero-shot learning frameworks, the system suffers when the target objects share few visual or context similarities with the training data. We are also limited by the scarcity of training data, the image dataset is very small comparing to the large label set. As discussed in Section 4.4, we expect diverse and abundant data could further improve the generalizability. So we hope the community could put more efforts on open-ended classification problems and dataset collection.

7. Conclusion

We introduced a new challenging task: open vocabulary scene parsing, which aims at parsing images in the wild. And we proposed a framework to solve it by embedding concepts image pixel features into a joint vector space, where the hierarchical semantics is preserved.

Acknowledgement: This work was supported by Samsung and NSF grant No.1524817 to AT. SF acknowledges the support from NSERC. BZ is supported by Facebook Fellowship. We thank Wei-Chiu Ma and Yusuf Aytar for insightful discussions.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv:1511.00561*, 2015. 2

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv:1606.00915*, 2016. 2, 5

[4] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *CVPR Workshop on The Future of Datasets in Vision*, 2015. 2

[5] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *European Conference on Computer Vision*, pages 48–64. Springer, 2014. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1

[7] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3450–3457. IEEE, 2012. 2

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int'l Journal of Computer Vision*, 2010. 2

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 2, 3, 5, 6

[10] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3202–3209. IEEE, 2012. 2

[11] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 4

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 1

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 2

[15] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 2

[16] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015. 2

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 2

[18] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1

[19] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proc. ICCV*, 2015. 2

[20] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013. 2, 6, 7

[21] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg. From large scale image categorization to entry-level categories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2768–2775, 2013. 2

[22] D. Parikh and K. Grauman. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE, 2011. 2

[23] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 5, 6

[24] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1641–1648. IEEE, 2011. 2

[25] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 910–917. IEEE, 2010. 2

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[27] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 2

[28] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015. 2, 3

[29] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994. 4

[30] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010. 1

[31] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 5

[32] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 1

[33] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016. 2, 5, 6