To Appear in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications* (SPIE Defense + Commercial Sensing)

A Framework for Explainable Deep Neural Models Using External Knowledge Graphs

Zachary A. Daniels[a], Logan D. Frank[b], Christopher J. Menart[c], Michael Raymer[d], and Pascal Hitzler[e]

[a]Department of Computer Science, Rutgers, the State University of New Jersey, Piscataway, NJ, USA
[b]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
[c]Sensors Directorate, Air Force Research Laboratory, Dayton, OH, USA
[d]Department of Computer Science and Engineering, Wright State University, Dayton, OH, USA
[e]Department of Computer Science, Kansas State University, Manhattan, KS, USA

ABSTRACT

Deep neural networks (DNNs) have become the gold standard for solving challenging classification problems, especially given complex sensor inputs (e.g., images and video). While DNNs are powerful, they are also brittle, and their inner workings are not fully understood by humans, leading to their use as "black-box" models. DNNs often generalize poorly when provided new data sampled from slightly shifted distributions; DNNs are easily manipulated by adversarial examples; and the decision-making process of DNNs can be difficult for humans to interpret. To address these challenges, we propose integrating DNNs with external sources of semantic knowledge. Large quantities of meaningful, formalized knowledge are available in knowledge graphs and other databases, many of which are publicly obtainable. But at present, these sources are inaccessible to deep neural methods, which can only exploit patterns in the signals they are given to classify. In this work, we conduct experiments on the ADE20K dataset, using scene classification as an example task where combining DNNs with external knowledge graphs can result in more robust and explainable models. We align the atomic concepts present in ADE20K (i.e., objects) to WordNet, a hierarchically-organized lexical database. Using this knowledge graph, we expand the concept categories which can be identified in ADE20K and relate these concepts in a hierarchical manner. The neural architecture we present performs scene classification using these concepts, illuminating a path toward DNNs which can efficiently exploit high-level knowledge in place of excessive quantities of direct sensory input. We hypothesize and experimentally validate that incorporating background knowledge via an external knowledge graph into a deep learning-based model should improve the explainability and robustness of the model.

Keywords: Deep Learning, Neural Networks, Knowledge Graphs, External Knowledge, Explainability, Generalizability, Scene Classification

1. INTRODUCTION

In the last decade, deep neural networks (DNNs) have become the gold standard for solving challenging classification problems, especially with complex sensor data (e.g., images and video). The Air Force is conducting

Further author information:
Z.A.D.: E-mail: zad7@cs.rutgers.edu, Telephone: 1 610 417 5761
C.J.M.: E-mail: christopher.menart@us.af.mil, Telephone: 1 937 713 8150

extensive research into this technology for rapid, automated target identification and understanding of the battlefield. One of the striking differences between DNNs and previous classification techniques is that neural methods are significantly more opaque and idiosyncratic in their decision-making. Well-known results in the field have shown that deep neural models often latch onto details and relationships very unlike those utilized by the human visual system.[1] Neural methods confidently classify many images in ways no human observer would. But one of the primary goals of computer vision and deep learning could be described as replicating the human understanding of the visual world. In this work, we propose integrating neural networks with external sources of human-generated semantic knowledge and relationships (in the form of knowledge graphs) so that a neural model's decision-making and reasoning capabilities better align with that human understanding. We hypothesize and experimentally validate that incorporating background knowledge from an external knowledge graph can improve the explainability and robustness of a neural model.

Formalized distillations of human knowledge about objects and the world exist in the form of public knowledge graphs such as WordNet,[2] WikiDate,[3] SUMO,[4] schema.org,[5] and freebase.[6] These include many distillations of "common-sense" knowledge which span a wide range of domains (e.g., life sciences, geographical data, political and government data, scientific data, data about media and publications) and are relevant for tasks already being performed by machine learning-based models. The relationships of people and animals to their body parts, the parts of structures and vehicles, and common context and spatial relationships between objects are among the kinds of knowledge available.

Presently, DNNs must learn all semantic information on their own during training purely from observation. While many DNNs are capable of learning complex tasks, such as image recognition, on their own, they will rarely learn to perform these tasks in the same way as human beings, and thus, will often learn 'around' basic knowledge of the world rather than learning it. Even stat-of-the-art models show behaviors such as e.g. classifying images based on high-frequency information rather than information such as shape that humans use to make similar decisions.[7]

By aligning neural networks with external semantic knowledge, we hope to help constrain them to more human-like behavior, and thus alleviate several of the major issues they face. Neural networks that are grounded to human knowledge may be capable of generating high-level descriptions of complex signals by using reasoning that is comprehensible to human beings, at least in those steps of reasoning where human beings do the same. Such models would possess explainability.[8–11] Second, neural networks have been shown to be vulnerable to adversarial attacks.[12–16] If a neural network classifies signals using the same features as human beings, it will, in theory, not be vulnerable to perturbations which would not be human-detectable. Third, DNNs are frequently among the best machine learning-based methods for classifying in-sample data, but often catastrophically fail when confronted with related but out-of-distribution data or when attempting to classify images without the same distribution of poses, backgrounds, and other incidental conditions as the training data.[17] By grounding neural networks to human-like semantic knowledge, DNNs should exhibit better generalizability in classifying out-of-distribution data. Unlike models in use today, a neural network aligned with human knowledge and reasoning should continue to exhibit human-like performance when exposed to some new situations. On a related note, if the external knowledge really does represent distilled information relevant to a task, neural networks utilizing it should not have to rely on as much raw sensory training data, much as human beings bring prior knowledge to a learning task. Learning-based models that can achieve strong generalization with limited data is crucial in many real-world applications; in real-world settings, classification systems often need to identify targets using limited amounts of lower-than-expected quality data under novel operating conditions. Current neural networks can rarely perform in such situations.

In this paper, we explore the effectiveness of aligning and integrating neural networks with external sources of knowledge (in the form of knowledge graphs). Experiments are conducted on the ADE20K dataset,[18] focusing on the task of scene classification. The ADE20K dataset consists of images of general indoor and outdoor scenes captured at ground-level using standard electro-optic cameras. We choose this dataset because it provides both 1) scene category labels for each image and 2) information about which objects (and their parts) are present in each image. This object information can be used as an additional form of supervision which ties the sensor data (images) to external semantic symbols.

We propose a novel neural architecture that integrates external knowledge directly with a deep convolutional neural network. Furthermore, we investigate how this new approach affects explainability and robustness compared to traditional DNNs and simpler models. Our external knowledge comes in the form of WordNet,[2] a hierarchically-organized lexical database. We align the 1,268 object types (not including object parts) labeled in the ADE20K dataset with their corresponding terms in the WordNet ontology. Once this alignment is complete, we construct a hierarchy of objects and their ancestor categories based on subclass-superclass relationships. Instead of knowing only that a "silver bird statue" is present in an image, for example, we also know that a "metal statue", "statue", "decoration", etc. are present in the image. We train an object recognizer on this "expanded" object set, and use the predicted object probabilities as features that can then be fed into a logistic regression model to perform scene classification. Because the features (object probabilities) are interpretable and the logistic regression model is a simple linear classifier, the model is capable of generating human-understandable explanations. Interestingly, experiments show that our grounded and interpretable model achieves similar performance (in terms of scene classification accuracy) compared to traditional unconstrained/black-box DNNs. We also explore how the structure of the hierarchy can be exploited to further improve object recognition, and conduct experiments analyzing the effect of calibrating the object prediction probabilities/scores, a necessary step for improving trustworthiness. In both cases, we achieve better performance on the object recognition task as measured by F1 score. Finally, we conduct experiments to determine how well the learned representation generalizes to unseen classes compared to a unmodified DNN that directly maps pixels to scene categories. To do so, we formulate a basic few-shot scene classification problem. Disappointingly, the knowledge-driven approach significantly under-performs the traditional scene classification DNN on this task, but this experiment does illuminate a few useful pieces of information. First, incorporating the knowledge graph into the model improves generalizability compared to a model which uses only the objects that come labeled with the ADE20K dataset. Second, compared to the traditional DNN, we gain some interpretability during the cross-domain knowledge transfer. It is difficult to understand why and when the traditional DNN representation generalizes well, because its features carry uninterpretable meaning. However, in our knowledge-based approach, features are grounded to well-defined concepts, allowing us to understand why a representation may or may not work well for a given domain. In future work this interpretability could be used to engineer new knowledge in order to improve a model/representation.

2. RELATED WORK

We present our work in the context of varied efforts to leverage human knowledge for deep learning. Some efforts attempt to exploit knowledge graphs directly, while others operate on prior knowledge imposed directly by the creator of the model or learn to automatically exploit consistency with patterns in previously-seen data (i.e., such models "discover" a knowledge graph). While more tangentially related, we also discuss other approaches to explainability.

2.1 Combining Knowledge Graphs and Deep Neural Networks for Computer Vision Tasks

There has been some recent interest in combining knowledge graphs with neural networks for computer vision-based tasks. Marino et al.[19] use structured prior knowledge in the form of knowledge graphs to improve performance on image classification. They train a neural network to predict every node in some knowledge graph and then propagate information between nodes to refine predictions. Goo et al.[20] utilize hierarchical taxonomies to relate objects based on subclass-superclass relations in order to learn better features that are more discriminative for classifying often confused sub-classes belonging to the same superclass. Guo et al.[21] learn a hierarchical classifier which combines a convolutional neural network for feature extraction with a recurrent neural network for exploiting relationships between the predicted classes. Srivastava et al.,[22] Fan et al.,[23] Kuang et al.,[24] and Zhang et al.[25] each propose different tree-structured concept ontologies which organize large numbers of concept classes (often objects) based on coarse-to-fine labels. Some of these methods also automatically discover inter-related learning tasks. Roy et al.[26] learn fine-to-coarse tree-structured DNN classifiers as well, but their approach learns incrementally, so new classes can be added without having to retrain the entire network. Yan et al.[27] utilize DNNs for hierarchical classification. Deng et al.[28] introduce Hierarchy and Exclusion graphs, which capture semantic relations based on mutual exclusion, overlap, and subsumption between two labels applied to the same object. Other approaches[29,30] utilize hierarchies and knowledge graphs to learn semantic embeddings. Finally, some approaches attempt to exploit ontologies in order to better explain the behavior of deep neural networks,[11] and improve deep learning image interpretation.[31] In contrast to other approaches, which exploit knowledge graphs to refine predictions given in a black-box fashion, our method integrates knowledge graphs earlier inside the neural network itself, producing more robust features for downstream tasks.

2.2 Improving the Explainability and Interpretability of Deep Neural Networks

Most neural networks are treated as black-box models: models that take in data and output decisions without providing further explanations or evidence to support them. While black-box neural networks are powerful tools that achieve state-of-the-art accuracy on a wide range of prediction tasks (especially visual recognition tasks), such models are not well-suited for safety-critical applications (e.g., those used in defense) and applications requiring trust in an autonomous or semi-autonomous agent. These applications require the use of explainable and interpretable models. Lipton et, al.[8] define several properties of interpretable models, including 1) model transparency and 2) post-hoc explainability. Model transparency is concerned with understanding how a model works at the level of 1) the entire model, 2) its individual components (e.g., features, parameters, etc.), and 3) the learning algorithm. However, useful explanations don't always require understanding the exact process by which a model operates. Post-hoc explainability is concerned with interpreting opaque models "after-the-fact" without sacrificing predictive performance. The methods we explore in this paper primarily focus on building the explanation into the neural network by grounding the decisions made by the network to human-understandable semantic knowledge.

Recently, there has been great interest in improving the interpretability and explainability of deep neural networks. The most popular and widely used methods focus on improving post-hoc explainability. Much of this work focuses on identifying the pixels and edges in an image that are most informative with respect to a neural network's final decision.[32–38] The major issue with these methods is that they tell us *where* a net is attending to without telling us *why* that region is important. Interpretable Basis Decomposition (IBD)[39] helps to alleviate this problem by training separate classifiers for each concept in some known set of concepts (e.g., using objects as explanations for scene classification) and also training a completely separate classifier for the target task. In a greedy manner, the concept classifier that explains most of the direction of the target classifier is selected and a residual is computed. Then, the concept classifier that explains most of the direction of the residual is selected, and a new residual is computed. This process is repeated until the residual cannot be explained by any of the remaining concept classifiers. Kim et al. propose a similar approach, "Quantitative Testing with Concept Activation Vector (TCAV)",[40] which provides an interpretation of a neural net's internal state in terms of human-friendly concepts. They show how to use directional derivatives to quantify the degree to which a user-defined concept is important to a classification result, e.g., how sensitive a prediction of zebra is to the presence of stripes.

There are also a number of methods proposed recently for making neural networks more transparent by grounding the decisions of the networks to some semantic knowledge. Sarker et al.[11] have proposed using knowledge graphs to improve deep learning explainability but provide only a very preliminary study. Daniels and Metaxas[41] explore an approach that is similar to IBD, but instead of learning the concept classifiers disjointly from the target classifiers, they train a single model that first identifies and predicts meaningful sets of objects (termed "scenarios") and then use these scenarios as interpretable features that are fed into a linear classification model for the target task. Hendricks et al.[42] proposes an alternative approach to improving model transparency whereby an auxiliary model is learned that uses the features of a trained convolutional neural network as input to a language-based model (e.g., a recurrent neural network) to generate captions that explain the decisions made by the visual recognition model. However, this approach suffers a major flaw: it is unknown if the explanation generator really "explains" which features the visual classification model uses or if it just finds new (and sometimes incorrect) plausible explanations based on shared features. Sometimes, the generated explanations describe information that is not even present in the image. To correct for this, in their follow up work,[43] Hendricks et al. ground the natural language explanations to visual cues by selecting explanations that are both image- and class-relevant. Finally, instead of grounding neural network decisions to specific concepts present in an image, one can also ground them to other "prototypical" instances of each class, e.g., Li et al.[44] combine DNNs with case-based reasoning. Our approach is most similar to Daniels and Metaxas' approach.[41] We try to improve model transparency by expanding the set of semantic concepts (e.g., objects) present in an image using an external knowledge graph, learning to recognize these concepts using a DNN, and using these predicted concepts as interpretable features for some downstream task (e.g., scene classification).

2.3 Exploiting Object Information for Visual Recognition Tasks

For our motivating application (scene classification), we explore utilizing object-based representations. Researchers in the computer vision community have long been interested in incorporating object-based information into visual recognition pipelines in order to improve visual recognition performance and model interpretability, especially for scene understanding. One common approach is to model contextual information about scenes based on object relations using a probabilistic graphical model.[45–51] These approaches involve either exploiting information about multiple tasks (e.g., scene classification, object recognition, semantic segmentation) to improve the performance of each individual task, or exploiting relationships between objects and the scene context to improve performance on some task. Our approach is different because it learns sequential models for object recognition and scene classification instead of jointly learning a single model, but it is similar in that we use a graph-based model to refine object predictions in order to make them consistent with a given knowledge graph.

Another way to exploit relationships between objects in order to improve scene understanding is by learning to organize objects (and sometimes their parts) into hierarchies and taxonomies, and then exploiting these hierarchies to improve performance on some other scene understanding task. For example, one can try to model objects by their parts, e.g., if one is trying to identify houses, he or she might first search for roofs, doors, windows, and walls.[52–54] Alternatively, one can learn how objects are naturally grouped into either tree structures or sets, e.g., see 55–57. Similarly, tree-based hierarchical context models have shown promise[50, 55, 56] for refining object predictions and detecting out-of-context objects. Other methods for exploiting hierarchies of concepts have been successfully applied to more specific applications such as content-based image retrieval.[58–60] In future work, we'd like to explore how to combine learning the relationships that exist between objects with our existing approach that assumes a known knowledge graph.

ObjectBank[61, 62] proposed using the output of generic object detectors as feature extractors for higher-level scene classification tasks. This is similar to our approach, but we rely on multi-object recognition instead of detection, and we incorporate external knowledge graphs to expand the object set and refine the object predictions.

Finally, object information can be helpful for explaining how otherwise complex models arrive at their decisions. The Interpretable Basis Decomposition[39] previously mentioned is one such method. Xie et al.[63] also tries to explain the decisions of neural networks for scene classification by grounding DNNs to known information about objects, a major focus of this work.

3. DATA

3.1 ADE20K

The ADE20K dataset[18] consists of images of indoor and outdoor scenes captured at ground-level using an electro-optic camera. We chose this dataset because it includes a rich amount of semantic information, facilitating its

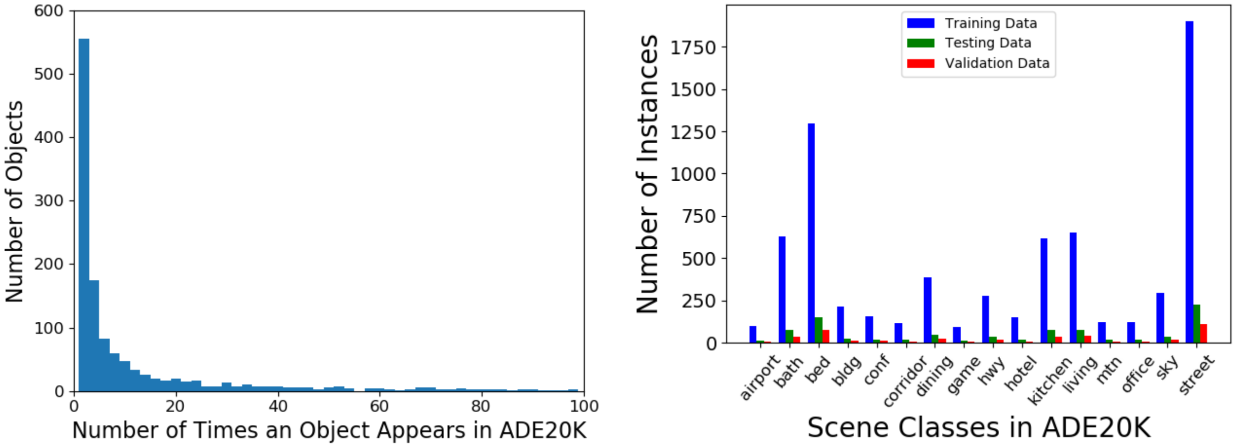

Figure 1. Left: A histogram relating each object to the number of times it appears in the subset of the ADE20K dataset used in our experiments. Right: The scene class distribution used in our experiments

alignment with existing external sources of knowledge. Each image in the ADE20K dataset has an associated scene label (e.g. bedroom, kitchen, street), a text file that identifies the objects and parts present in an image, and pixel-level segmentations for objects and parts. In our experiments, we utilize information about all 1,268 unique first-level objects (i.e., we exclude parts-of-objects) provided by the dataset. The pixel-level segmentations were also not used in our experiments. In Fig. 1, we show how frequently each object appears in the dataset. Note that most objects appear very rarely (i.e., less than ten times in the entire dataset), which poses challenges discussed later.

For most of our experiments, we use the subset of scene classes that have at least 100 total images (between train, test, and validation splits), resulting in 16 scene classes and 8,446 total images split into 7,131 training, 876 testing, and 439 validation. For the few-shot learning experiments, we use a disjoint subset of 26 classes and sample 50 instances from each class for a total of 1,300 additional images. Fig. 1 shows the scene classes used in the majority of our experiments and the distribution of images into each of these classes. Note that the scene class data is imbalanced, but we find that this presents fewer challenges than the imbalance in the object data. The only major issue we encountered due to this imbalance is that when two classes are both visually and semantically similar (e.g., bedroom and hotel room), the trained model is likely to default to the class with more training instances.

We also artificially inflate the size of our training data using data augmentation. When we sample an image from the dataset, we randomly crop the image so the area of the cropped image is 80% of the original image, we flip the image randomly with 50% probability, and randomly jitter the brightness, contrast, and saturation. Likewise, to improve training efficiency, we resize each image to 224 pixels-by-224 pixels and normalize the image using the mean pixel value and standard deviation.

3.2 WordNet

We used WordNet,[2] a hierarchically-organized lexical database, as our source of external knowledge. WordNet groups nouns, verbs, adjectives, and adverbs into sets of "cognitive synonyms" called synsets, and then organizes these synsets into a hierarchy based on hypernym (subclass-superclass/is-a) relations. For example, {feline, felid}, {cat, true cat}, and {cat, big cat} are several examples of synsets because the words in each set carry equivalent semantic meaning. Likewise, "feline" is a direct hypernym of both "true cats" (domestic cats and wild cats) as well as "big cats" (e.g., lions and tigers) because all "true cats" and "big cats" are felines, and thus, we can say that "true cats" and "big cats" are related by a shared parent class "feline". In this work, we will

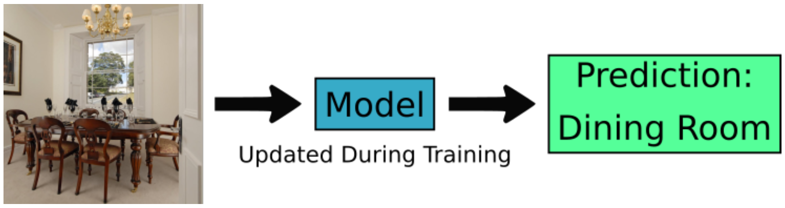

Figure 2. An overview of traditional scene classification which involves using a model to map images to a prediction of the image's scene category

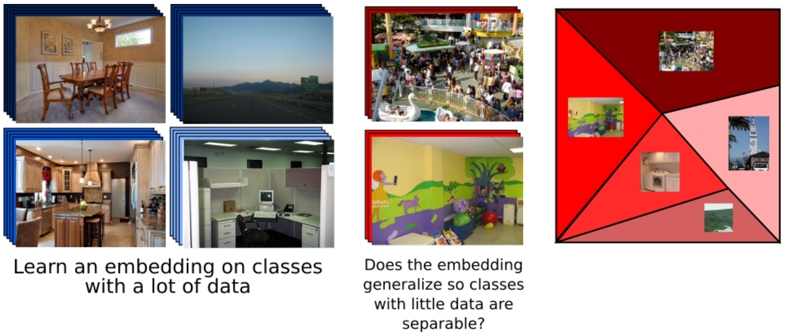
Learn an embedding on classes with a lot of data
Does the embedding generalize so classes with little data are separable?

Figure 3. An overview of the few-shot scene classification task where a representation is learned on classes with a lot of data and applied to a disjoint set of classes with little data and minimal fine-tuning

align the objects in the ADE20K dataset to their corresponding synsets in WordNet and use this information to 1) expand the set of object labels to include all parent objects, thus, giving us more complete, less noisy, and richer semantic information, and 2) hierarchically organize the objects, so we can exploit this known structure to improve our object predictions and the interpretability and trustworthiness of our model.

4. PROBLEM

In our experiments, we consider two problems: traditional scene classification and few-shot scene classification. We provide a brief overview of these problems in the following sections.

4.1 Traditional Scene Classification

Scene classification is a standard visual recognition task. In Fig. 2, we show a typical pipeline where an image is fed into a model, and the model makes a prediction about the identity of the scene category (e.g., dining room, kitchen, park, street, etc.).

4.2 Few-Shot Scene Classification

Few-shot scene classification is a related problem (see Fig. 3 for an overview). In few-shot learning, there exists one set of classes with ample training data, and a second, disjoint set of classes for which very little training data is available. These are analogous to *source* and *target* domains in transfer learning. The goal is to, by exploiting the source data, learn a model which can classify the target classes given only a small *support set* of examples for each class. In this paper, we specifically explore how well the *representations* learned on the source data generalize to the target classes, and whether this generalization is improved by the addition of external knowledge. Our experiments use 5-shot learning, where only five examples of each target class are provided for training.

5. METHODOLOGY

5.1 A Simple Object-Based Model for More Explainable Scene Classification

The first stage of our method involves predicting which objects are present in each scene image. To do so, we use deep convolutional neural networks (CNNs),[64,65] the current gold standard for visual recognition. Convolutional neural networks are a class of neural networks that perform feature extraction and prediction (in our case, classification) in an end-to-end manner. They are especially efficient at exploiting patterns in chain- and grid-structured data (e.g., images), and are thus frequently used for visual recognition tasks. In this work, we utilize the popular ResNet-18[66] architecture as our feature extraction/recognition model. While it is possible to train a CNN to directly predict the scene class from pixel-level visual data, such a model is not ideal if we care about explainability. Instead, our approach is to decompose the classification problem into multiple steps based on the external knowledge and supervision available. In the case of this specific problem, we have knowledge and supervision pertaining to physical objects, which should contain most of the information necessary for the final task of scene classification. So our neural network will 1) predict all of the objects present in a scene, and 2) use these predictions as features to a (linear) logistic regression model which performs the actual scene classification. Note that each of these models must be trained seperately, otherwise the intermediate features intended to encode the probabilities of object presence may encode other, hidden (and potentially brittle) information.

In this way, intermediate "features" of our scene classification model are able to be understood by humans (i.e., how likely is it that each object is in the image?), and the classifier is also interpretable because it is a simple linear model (i.e., if an object is associated with a large positive weight for a specific scene class, then it is strong evidence in favor of predicting that scene class, and the opposite holds for objects associated with large negative weights). Unlike traditional object recognition which involves predicting a single object that is generally the focus of an image (i.e, centered and consisting of the majority of the image's pixels), we need to learn to simultaneously predict the presence of all objects in a scene, and these objects vary in size and are spread throughout the image, making multi-object recognition in scene images a more challenging problem than traditional object recognition. This multi-object recognition problem is an example of a binary multi-label classification problem, so we optimize our network by trying to minimize the multi-label binary cross entropy loss:

$$\text{Goal: } \min_{\vec{\hat{o}}} loss_{bmce},\ loss_{bmce} = -\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(o_i^{(j)}log(\hat{o}_i^{(j)}) + (1-o_i^{(j)})log(1-\hat{o}_i^{(j)})\right) \tag{1}$$

where $o_i^{(j)} \in \{0,1\}$ is the true label for an object i in a given scene instance j, $\hat{o}_i^{(j)} \in [0,1]$ is the probability output by the neural network that object i is present in scene instance j, M is the total number of objects, and N is the total number of training examples in a mini-batch. We train the network using a learning rate of 1.0e-3, a weight decay value of 1.0e-5, and a batch size of 16. ADAM is used as our optimization algorithm. We train the net until convergence is achieved on a held out validation set.

These object recognition probabilities are input to a linear multinomial logistic regression model to perform scene classification. To train this model, we use the standard multi-class variant of the cross entropy loss:

$$\text{Goal: } \min_{\vec{\hat{s}}} loss_{ce},\ loss_{ce} = -\frac{1}{N}\sum_{j=1}^{N} log\left(\hat{s}_{true}^{(j)}\right) \tag{2}$$

where $\hat{s}_{true}^{(j)} \in [0,1]$ is the probability output by the model for the true scene class for a given scene instance j, and once again, N is the total number of training examples in a mini-batch.

5.2 Aligning ADE20K to the WordNet Knowledge Base

Several challenges exist to the basic approach discussed in the previous section. First, the object labels provided by the ADE20K dataset are often noisy and incomplete. There are ambiguous labels, e.g., "bowl" and "bowls" are treated as separate labels. Second, some labels are extremely specific, e.g. the dataset contains an object labeled as "silver bird statue". Third, there are also human errors in labeling where the human annotator sometimes incorrectly labels object that don't exist in a scene or vice versa. Fourth, some objects are very difficult to learn to recognize from visual data because they appear very infrequently (the majority of objects in the dataset appear fewer than ten times), and so the data-hungry learning-based model cannot capture all of the variations of appearance for such data-limited object classes. Fifth, some objects are very small in the scene images, so it is too difficult for the CNN to learn the fine-grained visual patterns necessary to accurately identify them from images. We will now discuss how we can exploit the freely-available WordNet knowledge base in order to alleviate many of these issues. For an overview of the structure of the WordNet knowledge base, please refer to section 3.2.

Each of the 1,268 objects in the ADE20K dataset are semi-automatically mapped (with manual corrections) to their corresponding synsets in WordNet. If no relevant direct synset exists for an object, it is mapped to its closest matching ancestor synset (hypernym). Once this mapping is complete, an object hierarchy can be constructed by recursively traversing the direct hypernyms of each term in WordNet. For example, starting with the "chair" object/synset, we can add the "seat" object/synset followed by the "furniture" object/synset, and so on and so forth until the root node for WordNet is reached. Once the full hierarchy is constructed, it is pruned to remove redundant nodes and edges. Consider the following example: "wall" is-a "partition" is-a "structure". Suppose the only subclass of "partition" that appears in the data is "wall,"* then it would be redundant to predict both "wall" and "partition" because they are effectively identical terms, and similarly, it would be incorrect to learn a model for "partition" that is identical to wall since in the real world there are non-wall partitions. Thus, to simplify the knowledge graph, improve semantic correctness, and remove redundancy, we prune the "partition" class and make "structure" the direct parent class of "wall". Furthermore, we can perform an additional different type of pruning to remove nodes if their corresponding object appears fewer than k times in the subset of the ADE20K dataset used in our experiments. This is useful for times when we need some minimum number of examples to train a relatively accurate object recognition model. It should be noted that this type of pruning can result in valid chains in the graph because the parents of the pruned children still capture information about the pruned children, so no two nodes are exactly equivalent. For example, we might encounter the chain "painting" is-a "art" is-a "creation". In this case, there might be a "sculpture" object that appears less than the desired number of times in the data, so it gets pruned, but the "art" object is still labeled as present if either the "painting" or "statue" object appears in a scene, and so "painting" and "art" are not synonomous terms. In Fig. 4, we show a very small subgraph of the aligned knowledge graph to demonstrate how much additional semantic information is gained by aligning the objects in the ADE20K dataset to WordNet.

Once we have a final pruned object hierarchy, we can generate an expanded object set. Whereas the original object set would consist mostly of the leaf nodes of the object hierarchy, the new expanded object set treats every node in the hierarchy as its own object/label. Then, we can train an object recognition model to predict this expanded object set. By expanding the graph and predicting the expanded object set, we can solve some of the issues previously discussed:

- Ambiguous labels are merged, e.g., the "bowl" and "bowls" objects would be mapped to the same "bowl" synset term in WordNet.
- Object information is captured at multiple levels of granularity, so instead of just predicting extremely fine-grained labels such as "silver bird statue", we now also know that this object is an example of a "statue", which also makes it an example "art", and so on and so forth. Likewise, to make coarse predictions about a scene category (e.g., kitchen), it might be unnecessary to know fine-grained information about the objects in the scene, i.e., instead of needing to know the "fork", "knife", and "spoon" objects are present, it is *sufficient* to know that "silverware" is present.

*This is just an example. In practice, there are non-wall partitions our dataset.

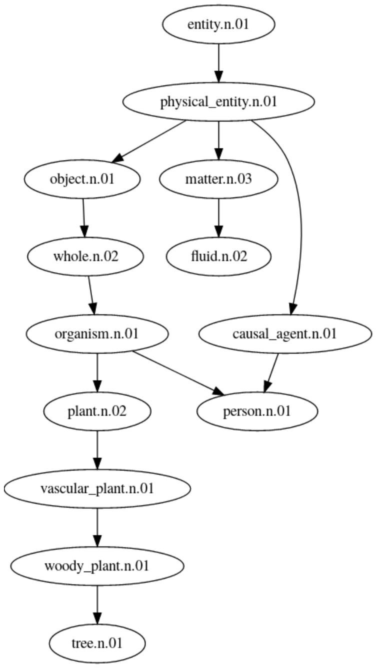

Figure 4. A small subgraph (after heavy pruning) of the aligned knowledge graph demonstrating how much additional semantic information is captured by aligning ADE20K to WordNet

- Higher-level categories appear much more frequently than some of their children. This means while there might not be enough training data to learn to recognize some object, there might be enough training data to recognize one of its ancestors, and so *some* information about the object is still preserved which might have otherwise been lost.

5.3 Calibrating Object Recognition Scores

Deep neural networks are generally powerful tools for making predictions and decisions, but they have imperfections. One flaw in modern networks is that since they have an incredibly large number of parameters and are highly nonlinear, they have a proclivity to overfit to the training data and be overly confident in their predictions. Many real world applications, especially those in defense, require learning-based models to not only be highly accurate but also be able to indicate when a prediction might be wrong, i.e., the neural network should be able to provide a realistic measure of confidence for an output prediction (a calibrated confidence[67]). Using calibrated confidences is especially important from an interpretability perspective. When a network says an object is present with probability greater than 0.5, it should mean that the net believes the object is actually present in the image. In this section, we discuss an approach for calibrating the object prediction scores output by our object recognition model.

First, the neural network is trained to convergence for object recognition. These parameters are frozen before moving onto the next step. It is crucial that these do not change during the calibration process or during training for the final task. Then, the logits (values output by the network before they are passed to the sigmoid function) are extracted for each object for each *validation* instance. Our goal is to learn scaling a and shift b parameters for a sigmoid function that maps the logits l_i to a calibrated score $\hat{o}'_i$ for each object i.:

$$\hat{o}'_i = \frac{1}{1 + e^{-a_i(l_i - b_i)}} \tag{3}$$

Unlike traditional confidence calibration methods like Platt scaling[68] and temperature scaling[67] which use the negative log-likelihood as the supervisory single (i.e., they perform maximum likelihood estimation), we take a slightly different approach and calibrate by minimizing a continuous approximation[69] of the f1-measure:

$$\text{Goal: } \min_{\vec{a},\vec{b}} loss_{f1},\ loss_{f1} = -\frac{1}{M}\sum_{i=1}^{M}\left(\frac{2 * \sum_{j=1}^{N}\hat{o}_i'^{(j)} o_i^{(j)}}{\sum_{j=1}^{N}\hat{o}_i'^{(j)} + \sum_{j=1}^{N} o_i^{(j)}}\right) \tag{4}$$

where M is the total number of objects, N is the number of validation instances, $o_i^{(j)} \in \{0, 1\}$ is the true label for object i in instance j, and $\hat{o}_i'^{(j)}$ is the calibrated score for object i in instance j. We use a continuous approximation of the f1-measure because our object labels tend to be very imbalanced, i.e., most objects appear very infrequently, and so maximum likelihood estimation is often overly aggressive about predicting probabilities closer to zero whereas the f1-measure considers the tradeoff between precision and recall.

5.4 Exploiting the Hierarchical Structure of the Knowledge Graph to Refine Object Predictions

In addition to providing a means of generating a larger object set that captures more semantic information, the knowledge graph also provides tools for helping humans understand when the network makes certain mistakes. Using this information, methods for structured prediction can be used to refine the object predictions output by the neural network leading to more accurate predictions and subsequently more interpretable and trustworthy models. In this section, we propose one simple method (as a proof of concept) for showing how knowing the structure of the knowledge graph can improve object recognition.

The hierarchical structure of the knowledge graph tells us that there should never be a case where a child object is predicted as being present when its parent is predicted as being absent. However, up to this point, our method has treated the prediction of the expanded object set as a “flat” classification problem, i.e., the network has no knowledge of the relations that exist between objects and essentially, must learn these from data. Because the network is not constrained to perform hierarchical classification, it very occasionally (in $\sim 1\%$ of predictions) makes this type of error. We can attempt to correct these known and easily identifiable mistaken predictions in a post-hoc manner. To do so, we formulate the following optimization problem:

$$\text{Goal: } \min_{\vec{\hat{o}}''} loss_{refine},\ loss_{refine} = \sum_{j=1}^{N}\left(\sqrt{\sum_{i=1}^{M}(\hat{o}_i''^{(j)} - \hat{o}_i'^{(j)})^2} + \sum_{(q,r)\in(child,parent)} max(\hat{o}_q''^{(j)} - \hat{o}_r''^{(j)}, 0)\right) \tag{5}$$

where M is the number of objects, N is the number of instances, $\hat{o}_i''^{(j)}$ is the refined prediction score for object i in instance j, $\hat{o}_i'^{(j)}$ is the calibrated prediction score for object i in instance j, and $(child, parent)$ is the set of all (child, parent) object relations. The idea behind this optimization problem is to minimally change the scores of the predictions (enforced by the first term) while fixing cases where the score for the child is larger than the score for the parent (enforced by the second term), i.e., how can the predictions be changed to fix any violations of the knowledge graph-based constraints in a minimally-disruptive manner? Note that this method is unsupervised because it doesn’t have any knowledge of what the correct predictions actually are; instead, it is simply exploiting the confidence scores of each object prediction and the known relationships that exist between objects.

5.5 Summary of Approach

Our complete approach can be summarized as:

1. Align the objects in ADE20K to synsets in WordNet.

Table 1. Accuracy on the scene classification task for several baseline models

Approach	**Scene Classification Accuracy**
Unmodified ResNet	0.817
Ground Truth Objects (Initial Set) + Logistic Regression	0.910
Ground Truth Objects (Expanded Set) + Logistic Regression	0.910

2. Using WordNet, generate and prune an object hierarchy knowledge graph.
3. Using the mined knowledge graph, generate an expanded object label set.
4. Train an object recognition CNN to predict the expanded object label set.
5. Calibrate the object prediction scores.
6. Refine the object prediction scores to fix violations in constraints imposed by the knowledge graph.
7. Train a linear logistic regression model for the scene classification task using the refined object prediction scores as features.

6. EXPERIMENTS AND RESULTS

In the following sections, we perform a quantitative investigation into the effectiveness of each component of our approach.

6.1 Experiment 1: The Importance of Utilizing Grounded, Semantic Information

Neural networks are good at learning highly-discriminative visual features that often capture some semantic information. In this first set of experiments, we want to see if there is any benefit to using grounded, semantic information (in our case, the objects present in a scene) over (just) the features discovered by a neural network. If there is, then it should be beneficial to train a network guided by this additional knowledge (objects) instead of relying on labels for the target task (scene classification) as the only means of supervision.

We train an unmodified ResNet-18 CNN to perform end-to-end scene classification as our baseline. We then train a logistic regression model that uses the *ground truth* object labels (initially, only using those present in the ADE20K dataset) for each scene image as features for scene classification. We also want to see how much additional information can be obtained by expanding the object set using the WordNet ontology, so we train another logistic regression model for scene classification which uses the knowledge-graph expanded object set for each scene image as features. Results appear in Table 1.

This experiment tells us several pieces of useful information. First, while the unmodified ResNet model performs very well, it still notably under-performs (by about 10%) compared to a simple linear model that has *perfect* information about the objects present in a scene. This is especially interesting because the object-based model is much easier to interpret than the baseline unmodified ResNet. Second, this experiment tells us that if we have perfect knowledge of the initial set of objects then we gain little to no additional useful information w.r.t. scene classification if we use the expanded object set derived from the WordNet ontology. One possible explanation for this result is that with perfect information, the machine learning model (even a simple linear classifier) might be able to naturally discover and exploit most of the relevant relationships between objects. The semantic nodes provided by the knowledge graph used here can be expressed as OR functions (i.e. piecewise linear functions) of the objects.

However, in practice, we do not have perfect knowledge of the objects in a scene; instead, we have to recover this information from a sensor (in this case, a camera) by performing object recognition/detection using a separate

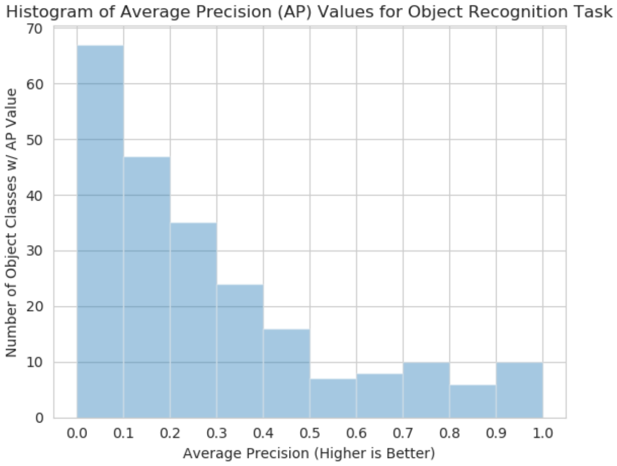

Figure 5. Evaluating the performance of a DNN trained to perform multi-object recognition on the objects that appear in at least 25 training instances in the subset of the ADE20K dataset used in our experiments

learning-based model (in this case, another ResNet-18 neural network). In the next several experiments, we will see: 1) multi-object recognition in scene images can be very imperfect and 2) when object recognition is noisy, the structure provided by the knowledge graph leads to notably improved performance.

6.2 Experiment 2: Understanding the Limitations and Impact of Noisy Object Recognition

Multi-object recognition in scene images can be very noisy. There are many different reasons for this, including that 1) the object recognition model is trained on imperfect and sometimes ambiguous labels; 2) unlike traditional object recognition, the model must be able to at least roughly localize the object in the image (or pick up on surrounding context clues); 3) some objects are extremely small, so there are not enough details to learn the fine-grained patterns needed to distinguish between certain objects; and 4) most importantly, there often is not enough data for a given object to learn all of its variations in appearances. To understand just how good or bad a DNN is for performing object recognition on our data, we train a ResNet-18 model to try to predict all objects that appear at least 25 times in the training data (because we assume there is little chance of learning an accurate classifier for the remaining objects which appear fewer than 25 times). We compute the average precision (a summary statistic of the area under the precision-recall curve) for each object. This is a useful metric because 1) as a metric averaged across classes, it is suited for classification tasks with significant class imbalance, and 2) it doesn't require us to threshold output scores as opposed to other common metrics like the f1-measure and accuracy. Results appear in Fig. 5.

The general message that can be taken away from this experiment is that, in general, most objects in the dataset (even after heavily pruning the object set to only consider those objects with at least 25 appearances) are recognized very poorly. Thus, when we use predicted objects as features for scene classification, we should not anticipate achieving the high levels of accuracy obtained in the previous example when using the ground truth object data.

An interesting observation we have found is that the quality of the object recognition model tends to be proportional to the number of training examples for each object class. In Fig. 6, we see that as we prune the object set to those objects which appear most frequently, the object recognition mean average precision (mAP) value increases. Thus, we can restrict our scene classification features by only considering the objects that are able to be accurately recognized from visual data.

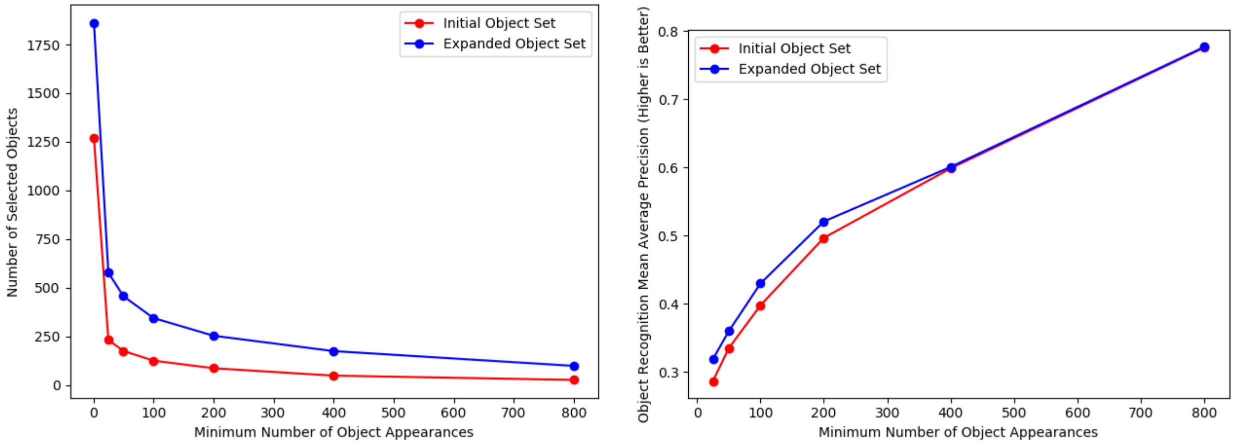

Figure 6. Understanding the effect of sample size on multi-object recognition

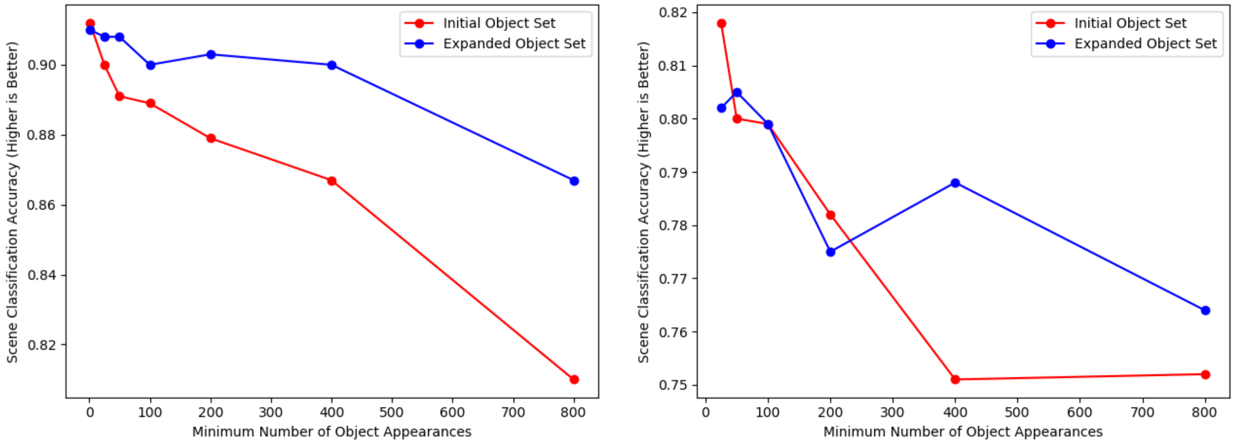

Figure 7. Understanding how scene classification performance is affected by object set size and quality. Left: evaluating models trained on ground truth object data; Right: evaluating models trained when using object recognition scores

Next, we must understand how scene classification performance is affected by pruning the object set based on minimum number of appearances. The left chart in Fig. 7 shows how scene classification quality degrades as objects are pruned when using the *ground truth* object data as features. The right chart in Fig. 7 shows how scene classification quality degrades as objects are pruned when using the *predicted* object probabilities as features. Note that in both cases, as we prune objects, we lose useful information, and scene classification accuracy quickly degrades.

6.3 Experiment 3: Improving Performance by Utilizing Knowledge Graphs

In section 5.2, we listed several of the ways that aligning the objects in the ADE20K dataset to the WordNet knowledge base and using this alignment to generate an object hierarchy and expanded object set should be beneficial for improving the robustness and explainability of the object-based representation. To summarize, we hypothesized: 1) ambiguous labels would be merged, 2) object information would be captured at multiple levels of granularity, and 3) since higher-level categories appear much more frequently than some of their children, while

Table 2. Evaluating the performance of various approaches for scene classification

Approach	**Min. Obj. Appears.**	**Obj. Rec. mAP**	**Scene Class. Acc.**
Unmodified ResNet	N/A	N/A	0.817
OR + LR: Initial	400	0.599	0.751
OR + LR: Expanded	400	0.601	0.788
OR + LR: Initial	800	0.776	0.752
OR + LR: Expanded	800	0.777	0.764

Table 3. Evaluating how object recognition and scene classification are affected by calibrating the object prediction scores

Approach	**Min. Obj. Appears.**	**Calib.?**	**Obj. Rec. Macro-F1**	**Scene Class. Acc.**
OR + LR: Expanded	400	No	0.557	0.788
OR + LR: Expanded	400	Yes	0.587	0.773
OR + LR: Expanded	800	No	0.723	0.764
OR + LR: Expanded	800	Yes	0.741	0.754

there might not be enough training data to learn to recognize some objects, there might be enough training data to recognize their ancestors, so *some* information about pruned objects (which might have otherwise been lost) can still be preserved.

Recall Figs. 6 and 7. Fig. 6 shows that as we prune objects based on the number of times they appear in the training dataset, the expanded object set 1) generally achieves equal or slightly higher object recognition results despite 2) working with a larger number of selected objects (i.e., because some of ancestor objects appear much more frequently than the leaf objects). Fig. 7 shows that while performance on the scene classification task still degrades as objects are pruned, the effect is less severe when using the knowledge graph-expanded object set. One interesting thing to note is the "bump" in the right chart of Fig. 7 when the threshold for minimum number of object appearances is set to 400. We hypothesize that this is the point where there is a good tradeoff between the amount of information preserved in the selected objects while the object recognition accuracy remains somewhat decent (mAP ≈ 0.6).

Finally, we summarize how our initial object-based model (OR + LR: Initial) compares with the knowledge graph-expanded object-based model (OR + LR: Expanded) and also how the unmodified ResNet-18 DNN for scene classification compares with our object-based approaches in Table 2. It should be noted that the unmodified ResNet-18 model outperforms our object-based model, but the best object-based model only under-performs by a few percentage points and remains significantly more interpretable.

6.4 Experiment 4: Understanding the Effects of Object Prediction Score Calibration

Up to this point, the experiments have utilized models with uncalibrated probability estimates. In this experiment, we wish to see if the calibration method proposed in section 5.3 is effective, and want to understand the impact of utilizing calibrated object prediction scores on the scene classification task. Unlike the previous experiments where we used mAP as our metric for evaluating the quality of the multi-object recognition, here we will use the macro-f1-measure which averages the f1-measure over all object classes. We make this change because we specifically want to know how the object recognition model performs when the prediction scores are thresholded at 0.5 because most humans intuitively assume a score greater than 0.5 means the object is present in an image and assume a score less than 0.5 means the object is absent from an image. The results in Table 3 suggest that the calibration does indeed work (i.e., it improves the f1-measure by several percentage points in each tested case). However since the calibration method manipulates the parameters of a sigmoid, some information is lost near the asymptotes of the sigmoid, so we do see a minor (1%) decrease in scene classification performance.

Table 4. Evaluating the effect of the object refinement strategy on scene classification accuracy

Approach	**Min. Obj. Appears.**	**Refined?**	**Scene Class. Acc.**
OR + LR: Expanded	400	No	0.773
OR + LR: Expanded	400	Yes	0.774
OR + LR: Expanded	800	No	0.754
OR + LR: Expanded	800	Yes	0.758

6.5 Experiment 5: Refining Object Predictions by Exploiting the Known Structure of the Knowledge Graph

In this experiment, we evaluate the object prediction refinement strategy proposed in section 5.4. First, we consider the calibrated model for the object recognition DNN trained to predict the knowledge graph-expanded set of objects with at least 400 examples in the training data. On the test dataset, 150,325 total predictions are made about objects. Of these predictions, there are 784 violations of the constraints imposed by the known knowledge graph (i.e., a child is predicted present while its parent is predicted absent). 1,547 object predictions are involved in these 784 violations. Initially, 730 of these object predictions are correct and 817 of the involved object predictions are incorrect. After refinement, 828 of the involved object predictions are correct, and 719 of the predictions are still incorrect.

Next, we consider the calibrated model for the object recognition DNN trained to predict the knowledge graph-expanded set of objects with at least 800 examples in the training data. On the test dataset, 85,041 total predictions are made about objects. Of these predictions, there are 441 violations of the constraints imposed by the known knowledge graph (i.e., a child is predicted present while its parent is predicted absent). 876 object predictions are involved in these 441 violations. Initially, 418 of these object predictions are correct and 458 of the involved object predictions are incorrect. After refinement, 484 of the involved object predictions are correct, and 392 of the predictions are still incorrect.

There are several things to note. First, the object recognition models are extremely good at making knowledge graph-consistent predictions despite having no prior knowledge of the relationships that exist between objects. Only 1-2% of object predictions violate the constraints imposed by the graph. Second, even the simple refinement strategy proposed in section 5.4 is effective at correcting some of the incorrect object predictions, providing supplemental evidence that there are additional benefits to combining knowledge graphs with learning-based models for visual recognition tasks. That being said, the refinement strategy corrects such a small proportion of mistakes that it ultimately doesn't dramatically effect the downstream scene classification task (see Table 4 for empirical validation of this).

6.6 Experiment 6: Understanding the Generalizability of Object-Based Representation Using Few-Shot Scene Classification

In our final experiment, we want to measure the generalizability and robustness of our learned representations when confronted with out-of-distribution data. We consider the few-shot scene classification problem. Please see section 4.2 for an overview of this problem. We use the models learned on our large standard dataset of 16 classes as generic feature extractors. For the unmodified ResNet-18, this involves outputting the 512-dimensional features that are extracted before the classification layers. We also consider the *calibrated* object prediction scores output by the object recognition model trained only on the initial ADE20K-only object set (OR: Initial + Calibrated), the calibrated object prediction scores output by the object recognition model trained on the knowledge graph-expanded object set (OR: Expanded + Calibrated), and the *refined* calibrated object prediction scores output by the object recognition model trained on the knowledge graph-expanded object set (OR: Expanded + Refined). However, for each of the object-based feature sets, we add an additional pre-processing step, and binarize the features using a threshold of 0.5 to remove some noise and help the generalizability of the representation.

Table 5. Understanding the generalizability of various learned representations on a 5-shot recognition problem with 26 previously unseen classes

Approach	**Min. Obj. Appears.**	**Top-1 Acc.**	**Top-3 Acc.**	**Top-5 Acc.**
Unmodified ResNet	N/A	0.297	0.550	0.687
OR: Initial + Calibrated	400	0.202	0.453	0.612
OR: Expanded + Calibrated	400	0.217	0.467	0.632
OR: Expanded + Refined	400	0.217	0.464	0.625
OR: Initial + Calibrated	800	0.196	0.459	0.622
OR: Expanded + Calibrated	800	0.214	0.458	0.622
OR: Expanded + Refined	800	0.209	0.457	0.619

We run 10-fold cross validation where each fold consists of five support instances per class (for 26 total classes) and 45 query instances per class. We learn a naïve Bayes classifier on the support set. We choose to use the naïve Bayes classifier because generative models generally work better than purely discriminative models (like logistic regression) in low-data regimes, and the naïve Bayes classifier is considered to be the simplest generative model. For the unmodified ResNet features, we use a Gaussian naïve Bayes model, and for the object-based representation, we use a Bernoulli naïve Bayes model. We measure the top-1, top-3, and top-5 accuracies where the top-k accuracy denotes how often the correct class appears in the top-k most likely predicted classes. Results appear in Table 5.

We see that the representation learned on the unmodified ResNet significantly outperforms the object-based representations. There are several possibilities for why this might be the case including 1) the dimensionality of the unmodified ResNet's feature space is much larger; 2) the unmodified ResNet features are designed to well-separate classes whereas the object-based representations have no scene class-level supervision when they are being learned; 3) the unmodified ResNet features are continuous whereas the object-based features are binary, and 4) since we only learn to recognize the objects that appear frequently in a very small set of 16 classes, there might not be enough semantic variability in this set of classes to generalize well to other scenes, but there might be enough visual variability (which the unmodified ResNet can exploit), and thus, the unmodified ResNet features work better. However, compared to the traditional DNN, we gain some interpretability during the cross-domain knowledge transfer. We do not know why the traditional DNN representation generalizes well because we do not have an understanding of how to easily and meaningfully interpret the features. With the knowledge-based approach, since the features are grounded to human-understandable concepts, we have the ability to analyze in which cases these features are used successfully and likewise, in which cases these features are used unsuccessfully. This enables us to form hypotheses about what additional information is needed in order to improve the generalizability of the representation when applied to some new domain and thus, provides a tool that can be used to help engineer new knowledge in order to improve the model/representation.

When we compare the object-based representations learned on the initial object set to the knowledge graph-expanded object set, we see that the extra information typically improves generalizability. We also see that the unrefined expanded object set-based representation slightly outperforms the refined object set-based representation. We do not know why this is the case, but the differences are so small (i.e., less than a percent in every case) that it could be random noise.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated the possibility of leveraging formal, symbolic knowledge from a public database in combination with a deep artificial neural network. This was done via decomposing the machine learning task in accordance with the available external knowledge (by first predicting some set of atomic concepts, e.g., objects, and then using the predicted concepts as features for a downstream task, e.g., scene classification) and utilizing additional supervision generated by establishing connections between the training data and semantic concepts in the external database. These structural changes nominally limited the input/output mappings learnable by the network, but in fact, they introduced additional human-interpretable nodes to the network without significantly changing performance. Furthermore, we experimentally validated the hypothesis that using a knowledge graph to process these intermediate conclusions can result in a more robust representation for downstream tasks.

We believe this suggests several important avenues for future work. Our process for exploiting external knowledge here uses additional supervision above and beyond what is normally required to train a fully-supervised model for a given task. In these experiments, instance-level object annotations were used, annotations much denser than the simple labels normally required for traditional image classification. This is costly and limits the possible data-efficiency benefits of leveraging external knowledge, so future work will seek to reduce the amount of extra supervision required. It is possible that a model which already has an existing framework of known concepts will be able to assimilate new concepts (including new end tasks) with limited supervision by exploiting their relationships to known concepts. It may be the case that learning 'sub-tasks' requires additional upfront effort, but is more re-usable, producing compounding benefits later on in a sufficiently large system.

There may also be more general ways to exploit concept relationships in external knowledge. The current approach operates exclusively on 'is-a' relationships, where the presence of concepts in a specific category also implies the presence of concepts in a more general category (i.e. chair implying furniture). But many other kinds of relationships exist and are generally captured in knowledge graphs. Prominent examples include part-of relationships or causal relationships. It is not yet clear how to leverage arbitrary semantic relationships, or if different classes of relationships must be individually considered.

Because our model is explainable, it also helps us understand when its knowledge might be insufficient when applied to an existing or new domain. An interesting future direction would include exploring how these explanations can be used as feedback for helping humans generate new knowledge in order to improve the generalizability of a model.

Finally, future work should investigate alternative neural architectures incorporating nodes which represent externally-defined semantic concepts. In the current work, the placement of these nodes directly prior to the prediction layer of our network here has exceptionally clear benefits for human interpretation. It may be advantageous, however, for a model to potentially learn a non-linear function of semantic concepts, which can be accomplished by inserting multiple dense layers between a semantically-defined layer of a deep network and its final output. It should also be possible for multiple semantically-aligned layers to exist, with 'higher-level' concepts at later layers existing as a function of 'lower-level' concepts in earlier ones. The multilevel object-part relationships in ADE20K provide a possible test case for this sort of reasoning. But whether such structures must be hand-engineered or can be derived automatically from the structure of a knowledge graph remains for future exploration.

ACKNOWLEDGMENTS

We would like to thank Air Force Research Laboratory for funding and access to computing resources, and the Autonomy Technology Research Center at Wright State University for the logistics which made this research possible.

CITATION

Daniels, Zachary A., Frank, Logan D., Menart, Christopher J., Raymer, Michael, and Hitzler, Pascal. "A Framework for Explainable Deep Neural Models Using External Knowledge Graphs," Proceedings Volume 11413, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications; 1141338 (2020). International Society for Optics and Photonics. DOI Link Not Yet Available

REFERENCES

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., "Intriguing properties of neural networks," *International Conference on Learning Representations* (2014).

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

[3] Vrandečić, D. and Krötzsch, M., "Wikidata: A free collaborative knowledgebase," *Commununications of the ACM* **57**, 78–85 (Sept. 2014).

[4] Pease, A., Niles, I., and Li, J., "The suggested upper merged ontology: A large ontology for the semantic web and its applications," *AAAI Workshop on Ontologies and the Semantic Web* , 2002 (2002).

[5] Guha, R. V., Brickley, D., and Macbeth, S., "Schema.org: Evolution of structured data on the web," *Communications of the ACM* **59**(2), 44–51 (2016).

[6] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J., "Freebase: A collaboratively created graph database for structuring human knowledge," *ACM SIGMOD International Conference on Management of Data* , 1247–1250 (2008).

[7] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W., "Imagenet-trained cnns are biased towards texture: Increasing shape bias improves accuracy and robustness," *International Conference on Learning Representations* (2018).

[8] Lipton, Z. C., "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490* (2016).

[9] Doran, D., Schulz, S., and Besold, T. R., "What does explainable ai really mean? a new conceptualization of perspectives," *International Workshop on Comprehensibility and Explainability in Artificial Intelligence and Machine Learning* (2017).

[10] Ras, G., van Gerven, M., and Haselager, P., "Explanation methods in deep learning: Users, values, concerns and challenges," 19–36, Springer (2018).

[11] Sarker, M. K., Xie, N., , D., Raymer, M., and Hitzler, P., "Explaining trained neural networks with semantic web technologies: First steps," *International Workshop on Neural-Symbolic Learning and Reasoning* (2017).

[12] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P., "Deepfool: A simple and accurate method to fool deep neural networks," *Computer Vision and Pattern Recognition* , 2574–2582 (2016).

[13] Goodfellow, I. J., Shlens, J., and Szegedy, C., "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572* (2014).

[14] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A., "The limitations of deep learning in adversarial settings," *IEEE European Symposium on Security and Privacy* , 372–387 (2016).

[15] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., "Intriguing properties of neural networks," *International Conference on Learning Representations* (2013).

[16] Su, J., Vargas, D. V., and Sakurai, K., "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation* (2019).

[17] Alcorn, M., Li, Q., Gong, Z., Wang, C., Mai, L., shinn Ku, W., and Nguyen, A., "Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects," *Computer Vision and Pattern Recognition* (2019).

[18] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A., "Scene parsing through ade20k dataset," *Computer vision and pattern recognition* , 633–641 (2017).

[19] Marino, K., Salakhutdinov, R., and Gupta, A., "The more you know: Using knowledge graphs for image classification," *Computer Vision and Pattern Recognition* , 2673–2681 (2017).

[20] Goo, W., Kim, J., Kim, G., and Hwang, S. J., "Taxonomy-regularized semantic deep convolutional neural networks," *European Conference on Computer Vision* , 86–101 (2016).

[21] Guo, Y., Liu, Y., Bakker, E. M., Guo, Y., and Lew, M. S., "Cnn-rnn: a large-scale hierarchical image classification framework," *Multimedia Tools and Applications* **77**(8), 10251–10271 (2018).

[22] Srivastava, N. and Salakhutdinov, R. R., "Discriminative transfer learning with tree-based priors," *Advances in Neural Information Processing Systems* , 2094–2102 (2013).

[23] Fan, J., Zhao, T., Kuang, Z., Zheng, Y., Zhang, J., Yu, J., and Peng, J., "Hd-mtl: Hierarchical deep multi-task learning for large-scale visual recognition," *IEEE Transactions on Image Processing* **26**(4), 1923–1938 (2017).

[24] Kuang, Z., Yu, J., Li, Z., Zhang, B., and Fan, J., "Integrating multi-level deep learning and concept ontology for large-scale visual recognition," *Pattern Recognition* **78**, 198–214 (2018).

[25] Zhang, J., Mei, K., Zheng, Y., and Fan, J., "Learning multi-layer coarse-to-fine representations for large-scale image classification," *Pattern Recognition* **91**, 175–189 (2019).

[26] Roy, D., Panda, P., and Roy, K., "Tree-cnn: A hierarchical deep convolutional neural network for incremental learning," *arXiv preprint arXiv:1802.05800* (2018).

[27] Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., and Yu, Y., "Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition," *International Conference on Computer Vision* , 2740–2748 (2015).

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

[29] Ge, W., "Deep metric learning with hierarchical triplet loss," *European Conference on Computer Vision* , 269–285 (2018).

[30] Zhang, Z. and Saligrama, V., "Zero-shot learning via semantic similarity embedding," *International Conference on Computer Vision* , 4166–4174 (2015).

[31] Donadello, I., Serafini, L., and Garcez, A. D., "Logic tensor networks for semantic image interpretation," *International Joint Conference on Artificial Intelligence* , 1596–1602 (2017).

[32] Oquab, M., Bottou, L., Laptev, I., and Sivic, J., "Learning and transferring mid-level image representations using convolutional neural networks," *Computer Vision and Pattern Recognition* , 1717–1724 (2014).

[33] Zeiler, M. D. and Fergus, R., "Visualizing and understanding convolutional networks," *European Conference on Computer Vision* , 818–833 (2014).

[34] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M., "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806* (2014).

[35] Oquab, M., Bottou, L., Laptev, I., and Sivic, J., "Is object localization for free? weakly-supervised learning with convolutional neural networks," *Computer Vision and Pattern Recognition* , 685–694 (2015).

[36] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A., "Learning deep features for discriminative localization," *Computer Vision and Pattern Recognition* , 2921–2929 (2016).

[37] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D., "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Conference on Computer Vision* , 618–626 (2017).

[38] Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N., "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *IEEE Winter Conference on Applications of Computer Vision* , 839–847 (2018).

[39] Zhou, B., Sun, Y., Bau, D., and Torralba, A., "Interpretable basis decomposition for visual explanation," *European Conference on Computer Vision* , 119–134 (2018).

[40] Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al., "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," *International Conference on Machine Learning* , 2673–2682 (2018).

[41] Daniels, Z. A. and Metaxas, D., "Scenarionet: An interpretable data-driven model for scene understanding," *IJCAI Workshop on Explainable Artificial Intelligence* , 33 (2018).

[42] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T., "Generating visual explanations," *European Conference on Computer Vision* , 3–19 (2016).

[43] Anne Hendricks, L., Hu, R., Darrell, T., and Akata, Z., "Grounding visual explanations," *European Conference on Computer Vision* , 264–279 (2018).

[44] Li, O., Liu, H., Chen, C., and Rudin, C., "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," *AAAI Conference on Artificial Intelligence* (2018).

[45] Galleguillos, C. and Belongie, S., "Context based object categorization: A critical survey," *Computer Vision and Image Understanding* **114**(6), 712–722 (2010).

[46] Galleguillos, C., Rabinovich, A., and Belongie, S., "Object categorization using co-occurrence, location and appearance," *Computer Vision and Pattern Recognition* , 1–8 (2008).

[47] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. J., "Objects in context," *International Conference on Computer Vision* **1**(2), 5 (2007).

[48] Murphy, K., Torralba, A., Freeman, W., et al., "Using the forest to see the trees: A graphical model relating features, objects and scenes," *Advances in Neural Information Processing Systems* **16**, 1499–1506 (2003).

[49] Carbonetto, P., de Freitas, N., and Barnard, K., "A statistical model for general contextual object recognition," 350–362, Springer (2004).

[50] Choi, M. J., Torralba, A., and Willsky, A. S., "Context models and out-of-context objects," *Pattern Recognition Letters* **33**(7), 853–862 (2012).

[51] Singhal, A., Luo, J., and Zhu, W., "Probabilistic spatial context models for scene content understanding," *Computer Vision and Pattern Recognition* **1**, I–235 (2003).

[52] Ahuja, N. and Todorovic, S., "Learning the taxonomy and models of categories present in arbitrary images," *International Conference on Computer Vision* , 1–8 (2007).

[53] Sudderth, E. B., Torralba, A., Freeman, W. T., and Willsky, A. S., "Learning hierarchical models of scenes, objects, and parts," *International Conference on Computer Vision* **2**, 1331–1338 (2005).

[54] Lan, T., Raptis, M., Sigal, L., and Mori, G., "From subcategories to visual composites: A multi-level framework for object detection," *International Conference on Computer Vision* , 369–376 (2013).

[55] Choi, M. J., Lim, J. J., Torralba, A., and Willsky, A. S., "Exploiting hierarchical context on a large database of object categories," *Computer Vision and Pattern Recognition* , 129–136 (2010).

[56] Choi, M. J., Torralba, A., and Willsky, A. S., "A tree-based context model for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(2), 240–252 (2012).

[57] Cinbis, R. G. and Sclaroff, S., "Contextual object detection using set-based classification," 43–57, Springer (2012).

[58] Fan, J., Gao, Y., and Luo, H., "Hierarchical classification for automatic image annotation," *ACM SIGIR Conference on Research and Development in Information Retrieval* , 111–118 (2007).

[59] Fan, J., Gao, Y., and Luo, H., "Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation," *IEEE Transactions on Image Processing* **17**(3), 407–426 (2008).

[60] Fan, J., Gao, Y., Luo, H., and Jain, R., "Mining multilevel image semantics via hierarchical classification," *IEEE Transactions on Multimedia* **10**(2), 167–187 (2008).

[61] Li, L.-J., Su, H., Fei-Fei, L., and Xing, E. P., "Object bank: A high-level image representation for scene classification & semantic feature sparsification," *Advances in Neural Information Processing Systems* , 1378–1386 (2010).

[62] Li, L.-J., Su, H., Lim, Y., and Fei-Fei, L., "Object bank: An object-level image representation for high-level visual recognition," *International Journal of Computer Vision* **107**(1), 20–39 (2014).

[63] Xie, N., Sarker, M. K., Doran, D., Hitzler, P., and Raymer, M., "Relating input concepts to convolutional neural network decisions," *NIPS Workshop on Interpreting, Explaining, and Visualizing Deep Learning* (2017).

[64] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., "Backpropagation applied to handwritten zip code recognition," *Neural Computation* **1**(4), 541–551 (1989).

[65] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems* , 1097–1105 (2012).

[66] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition* , 770–778 (2016).

[67] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q., "On calibration of modern neural networks," *International Conference on Machine Learning* , 1321–1330 (2017).

[68] Platt, J. et al., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers* **10**(3), 61–74 (1999).

[69] Ye, N., Chai, K., Lee, W., and Chieu, H., "Optimizing f-measures: A tale of two approaches," *International Conference on Machine Learning* **1**, 289–296 (2012).