

# OBJECT DETECTORS EMERGE IN DEEP SCENE CNNs

**Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba**

Computer Science and Artificial Intelligence Laboratory, MIT

{bolei, khosla, agata, oliva, torralba}@mit.edu

## ABSTRACT

With the success of new computational architectures for visual processing, such as convolutional neural networks (CNN) and access to image databases with millions of labeled examples (e.g., ImageNet, Places), the state of the art in computer vision is advancing rapidly. One important factor for continued progress is to understand the representations that are learned by the inner layers of these deep architectures. Here we show that object detectors emerge from training CNNs to perform scene classification. As scenes are composed of objects, the CNN for scene classification automatically discovers meaningful objects detectors, representative of the learned scene categories. With object detectors emerging as a result of learning to recognize scenes, our work demonstrates that the same network can perform both scene recognition and object localization in a single forward-pass, without ever having been explicitly taught the notion of objects.

## 1 INTRODUCTION

Current deep neural networks achieve remarkable performance at a number of vision tasks surpassing techniques based on hand-crafted features. However, while the structure of the representation in hand-crafted features is often clear and interpretable, in the case of deep networks it remains unclear what the nature of the learned representation is and why it works so well. A convolutional neural network (CNN) trained on ImageNet (Deng et al., 2009) significantly outperforms the best hand-crafted features on the ImageNet challenge (Russakovsky et al., 2015). But more surprisingly, the same network, when used as a generic feature extractor, is also very successful at other tasks like object detection on the PASCAL VOC dataset (Everingham et al., 2010).

A number of works have focused on understanding the representation learned by CNNs. The work by Zeiler & Fergus (2014) introduces a procedure to visualize what activates each unit. Recently Yosinski et al. (2014) use transfer learning to measure how generic/specific the learned features are. In Agrawal et al. (2014) and Szegedy et al. (2013), they suggest that the CNN for ImageNet learns a distributed code for objects. They all use ImageNet, an object-centric dataset, as a training set.

When training a CNN to distinguish different object classes, it is unclear what the underlying representation should be. Objects have often been described using part-based representations where parts can be shared across objects, forming a distributed code. However, what those parts should be is unclear. For instance, one would think that the meaningful parts of a face are the mouth, the two eyes, and the nose. However, those are simply functional parts, with words associated with them; the object parts that are important for visual recognition might be different from these semantic parts, making it difficult to evaluate how efficient a representation is. In fact, the strong internal configuration of objects makes the definition of what is a useful part poorly constrained: an algorithm can find different and arbitrary part configurations, all giving similar recognition performance.

Learning to classify scenes (i.e., classifying an image as being an office, a restaurant, a street, etc) using the Places dataset (Zhou et al., 2014) gives the opportunity to study the internal representation learned by a CNN on a task other than object recognition.

In the case of scenes, the representation is clearer. Scene categories are defined by the objects they contain and, to some extent, by the spatial configuration of those objects. For instance, the important parts of a bedroom are the bed, a side table, a lamp, a cabinet, as well as the walls, floor and ceiling. Objects represent therefore a distributed code for scenes (i.e., object classes are shared across different scene categories). Importantly, in scenes, the spatial configuration of objects,

Table 1: The parameters of the network architecture used for ImageNet-CNN and Places-CNN.

| Layer   | conv1 | pool1 | conv2 | pool2 | conv3 | conv4 | conv5 | pool5 | fc6  | fc7  |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|
| Units   | 96    | 96    | 256   | 256   | 384   | 384   | 256   | 256   | 4096 | 4096 |
| Feature | 55×55 | 27×27 | 27×27 | 13×13 | 13×13 | 13×13 | 13×13 | 6×6   | 1    | 1    |

although compact, has a much larger degree of freedom. It is this loose spatial dependency that, we believe, makes scene representation different from most object classes (most object classes do not have a loose interaction between parts). In addition to objects, other feature regularities of scene categories allow for other representations to emerge, such as textures (Renninger & Malik, 2004), GIST (Oliva & Torralba, 2006), bag-of-words (Lazebnik et al., 2006), part-based models (Pandey & Lazebnik, 2011), and ObjectBank (Li et al., 2010). While a CNN has enough flexibility to learn any of those representations, if meaningful objects emerge without supervision inside the inner layers of the CNN, there will be little ambiguity as to which type of representation these networks are learning.

The main contribution of this paper is to show that object detection emerges inside a CNN trained to recognize scenes, even more than when trained with ImageNet. This is surprising because our results demonstrate that reliable object detectors are found even though, unlike ImageNet, no supervision is provided for objects. Although object discovery with deep neural networks has been shown before in an unsupervised setting (Le, 2013), here we find that many more objects can be naturally discovered, in a supervised setting tuned to scene classification rather than object classification.

Importantly, the emergence of object detectors inside the CNN suggests that a single network can support recognition at several levels of abstraction (e.g., edges, texture, objects, and scenes) without needing multiple outputs or a collection of networks. Whereas other works have shown that one can detect objects by applying the network multiple times in different locations (Girshick et al., 2014), or focusing attention (Tang et al., 2014), or by doing segmentation (Grangier et al., 2009; Farabet et al., 2013), here we show that the same network can do both object localization and scene recognition in a single forward-pass. Another set of recent works (Oquab et al., 2014; Bergamo et al., 2014) demonstrate the ability of deep networks trained on object classification to do localization without bounding box supervision. However, unlike our work, these require object-level supervision while we only use scenes.

## 2 IMAGENET-CNN AND PLACES-CNN

Convolutional neural networks have recently obtained astonishing performance on object classification (Krizhevsky et al., 2012) and scene classification (Zhou et al., 2014). The ImageNet-CNN from Jia (2013) is trained on 1.3 million images from 1000 object categories of ImageNet (ILSVRC 2012) and achieves a top-1 accuracy of 57.4%. With the same network architecture, Places-CNN is trained on 2.4 million images from 205 scene categories of Places Database (Zhou et al., 2014), and achieves a top-1 accuracy of 50.0%. The network architecture used for both CNNs, as proposed in (Krizhevsky et al., 2012), is summarized in Table 1<sup>1</sup>. Both networks are trained from scratch using only the specified dataset.

The deep features from Places-CNN tend to perform better on scene-related recognition tasks compared to the features from ImageNet-CNN. For example, as compared to the Places-CNN that achieves 50.0% on scene classification, the ImageNet-CNN combined with a linear SVM only achieves 40.8% on the same test set<sup>2</sup> illustrating the importance of having scene-centric data.

To further highlight the difference in representations, we conduct a simple experiment to identify the differences in the type of images preferred at the different layers of each network: we create a set of 200k images with an approximately equal distribution of scene-centric and object-centric images<sup>3</sup>, and run them through both networks, recording the activations at each layer. For each layer, we obtain the top 100 images that have the largest average activation (sum over all spatial locations for

<sup>1</sup>We use *unit* to refer to neurons in the various layers and *features* to refer to their activations.

<sup>2</sup>Scene recognition demo of Places-CNN is available at <http://places.csail.mit.edu/demo.html>. The demo has 77.3% top-5 recognition rate in the wild estimated from 968 anonymous user responses.

<sup>3</sup>100k object-centric images from the test set of ImageNet LSVRC2012 and 108k scene-centric images from the SUN dataset (Xiao et al., 2014).

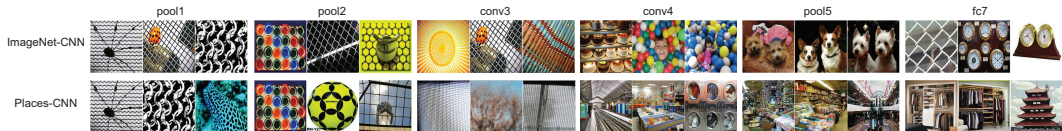


Figure 1: Top 3 images producing the largest activation of units in each layer of ImageNet-CNN (top) and Places-CNN (bottom).

a given layer). Fig. 1 shows the top 3 images for each layer. We observe that the earlier layers such as pool1 and pool2 prefer similar images for both networks while the later layers tend to be more specialized to the specific task of scene or object categorization. For layer pool2, 55% and 47% of the top-100 images belong to the ImageNet dataset for ImageNet-CNN and Places-CNN. Starting from layer conv4, we observe a significant difference in the number of top-100 belonging to each dataset corresponding to each network. For fc7, we observe that 78% and 24% of the top-100 images belong to the ImageNet dataset for the ImageNet-CNN and Places-CNN respectively, illustrating a clear bias in each network.

In the following sections, we further investigate the differences between these networks, and focus on better understanding the nature of the representation learned by Places-CNN when doing scene classification in order to clarify some part of the secret to their great performance.

### 3 UNCOVERING THE CNN REPRESENTATION

The performance of scene recognition using Places-CNN is quite impressive given the difficulty of the task. In this section, our goal is to understand the nature of the representation that the network is learning.

#### 3.1 SIMPLIFYING THE INPUT IMAGES

Simplifying images is a well known strategy to test human recognition. For example, one can remove information from the image to test if it is diagnostic or not of a particular object or scene (for a review see Biederman (1995)). A similar procedure was also used by Tanaka (1993) to understand the receptive fields of complex cells in the inferior temporal cortex (IT).

Inspired by these approaches, our idea is the following: given an image that is correctly classified by the network, we want to simplify this image such that it keeps as little visual information as possible while still having a high classification score for the same category. This simplified image (named minimal image representation) will allow us to highlight the elements that lead to the high classification score. In order to do this, we manipulate images in the gradient space as typically done in computer graphics (Pérez et al., 2003). We investigate two different approaches described below.

In the first approach, given an image, we create a segmentation of edges and regions and remove segments from the image iteratively. At each iteration we remove the segment that produces the smallest decrease of the correct classification score and we do this until the image is incorrectly classified. At the end, we get a representation of the original image that contains, approximately, the minimal amount of information needed by the network to correctly recognize the scene category. In Fig. 2 we show some examples of these minimal image representations. Notice that objects seem to contribute important information for the network to recognize the scene. For instance, in the case of bedrooms these minimal image representations usually contain the region of the bed, or in the art gallery category, the regions of the paintings on the walls.

Based on the previous results, we hypothesized that for the Places-CNN, some objects were crucial for recognizing scenes. This inspired our second approach: we generate the minimal image representations using the fully annotated image set of SUN Database (Xiao et al., 2014) (see section 4.1 for details on this dataset) instead of performing automatic segmentation. We follow the same procedure as the first approach using the ground-truth object segments provided in the database.

This led to some interesting observations: for bedrooms, the minimal representations retained the bed in 87% of the cases. Other objects kept in bedrooms were wall (28%) and window (21%).

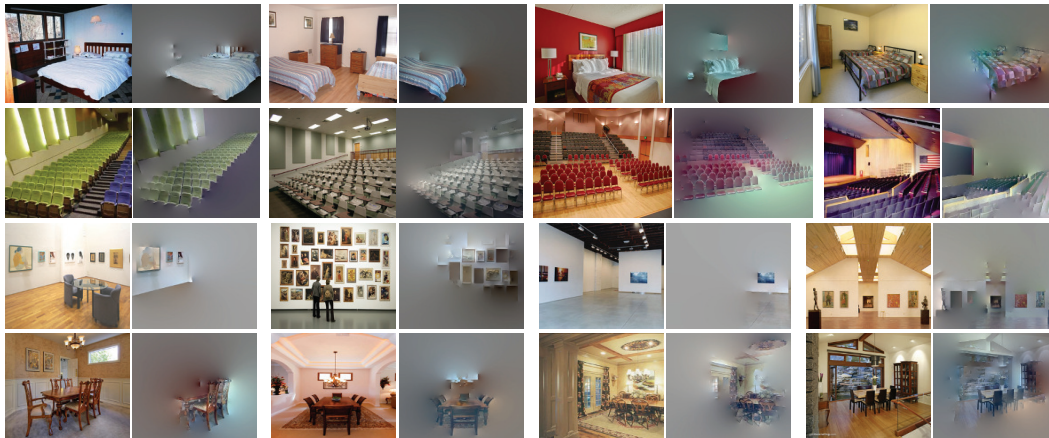


Figure 2: Each pair of images shows the original image (left) and a simplified image (right) that gets classified by the Places-CNN as the same scene category as the original image. From top to bottom, the four rows show different scene categories: bedroom, auditorium, art gallery, and dining room.

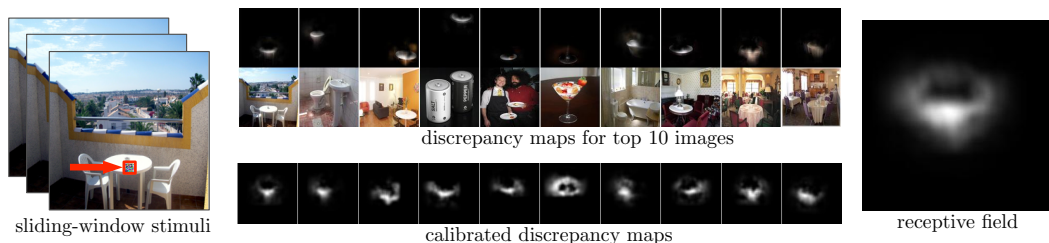


Figure 3: The pipeline for estimating the RF of each unit. Each sliding-window stimuli contains a small randomized patch (example indicated by red arrow) at different spatial locations. By comparing the activation response of the sliding-window stimuli with the activation response of the original image, we obtain a discrepancy map for each image (middle top). By summing up the calibrated discrepancy maps (middle bottom) for the top ranked images, we obtain the actual RF of that unit (right).

For art gallery the minimal image representations contained paintings (81%) and pictures (58%); in amusement parks, carousel (75%), ride (64%), and roller coaster (50%); in bookstore, bookcase (96%), books (68%), and shelves (67%). These results suggest that object detection is an important part of the representation built by the network to obtain discriminative information for scene classification.

### 3.2 VISUALIZING THE RECEPTIVE FIELDS OF UNITS AND THEIR ACTIVATION PATTERNS

In this section, we investigate the shape and size of the receptive fields (RFs) of the various units in the CNNs. While theoretical RF sizes can be computed given the network architecture (Long et al., 2014), we are interested in the actual, or *empirical* size of the RFs. We expect the empirical RFs to be better localized and more representative of the information they capture than the theoretical ones, allowing us to better understand what is learned by each unit of the CNN.

Thus, we propose a data-driven approach to estimate the learned RF of each unit in each layer. It is simpler than the deconvolutional network visualization method (Zeiler & Fergus, 2014) and can be easily extended to visualize any learned CNNs<sup>4</sup>.

The procedure for estimating a given unit’s RF, as illustrated in Fig. 3, is as follows. As input, we use an image set of 200k images with a roughly equal distribution of scenes and objects (similar to Sec. 2). Then, we select the top  $K$  images with the highest activations for the given unit.

<sup>4</sup>More visualizations are available at <http://places.csail.mit.edu/visualization>

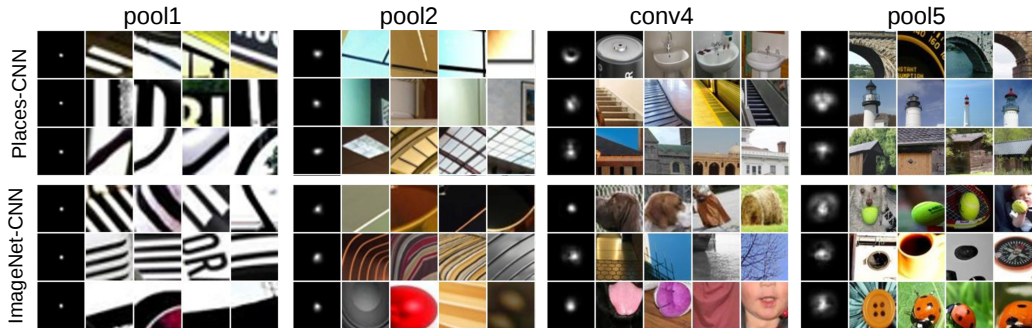


Figure 4: The RFs of 3 units of pool1, pool2, conv4, and pool5 layers respectively for ImageNet- and Places-CNNs, along with the image patches corresponding to the top activation regions inside the RFs.

Table 2: Comparison of the theoretical and empirical sizes of the RFs for Places-CNN and ImageNet-CNN at different layers. Note that the RFs are assumed to be square shaped, and the sizes reported below are the length of each side of this square, in pixels.

|                          | pool1          | pool2          | conv3           | conv4           | pool5           |
|--------------------------|----------------|----------------|-----------------|-----------------|-----------------|
| Theoretic size           | 19             | 67             | 99              | 131             | 195             |
| Places-CNN actual size   | $17.8 \pm 1.6$ | $37.4 \pm 5.9$ | $52.1 \pm 10.6$ | $60.0 \pm 13.7$ | $72.0 \pm 20.0$ |
| ImageNet-CNN actual size | $17.9 \pm 1.6$ | $36.7 \pm 5.4$ | $51.1 \pm 9.9$  | $60.4 \pm 16.0$ | $70.3 \pm 21.6$ |

For each of the  $K$  images, we now want to identify exactly which regions of the image lead to the high unit activations. To do this, we replicate each image many times with small random occluders (image patches of size  $11 \times 11$ ) at different locations in the image. Specifically, we generate occluders in a dense grid with a stride of 3. This results in about 5000 occluded images per original image. We now feed all the occluded images into the same network and record the change in activation as compared to using the original image. If there is a large discrepancy, we know that the given patch is important and vice versa. This allows us to build a discrepancy map for each image.

Finally, to consolidate the information from the  $K$  images, we center the discrepancy map around the spatial location of the unit that caused the maximum activation for the given image. Then we average the re-centered discrepancy maps to generate the final RF.

In Fig. 4 we visualize the RFs for units from 4 different layers of the Places-CNN and ImageNet-CNN, along with their highest scoring activation regions inside the RF. We observe that, as the layers go deeper, the RF size gradually increases and the activation regions become more semantically meaningful. Further, as shown in Fig. 5, we use the RFs to segment images using the feature maps of different units. Lastly, in Table 2, we compare the theoretical and empirical size of the RFs at different layers. As expected, the actual size of the RF is much smaller than the theoretical size, especially in the later layers. Overall, this analysis allows us to better understand each unit by focusing precisely on the important regions of each image.



Figure 5: Segmentation based on RFs. Each row shows the 4 most confident images for some unit.

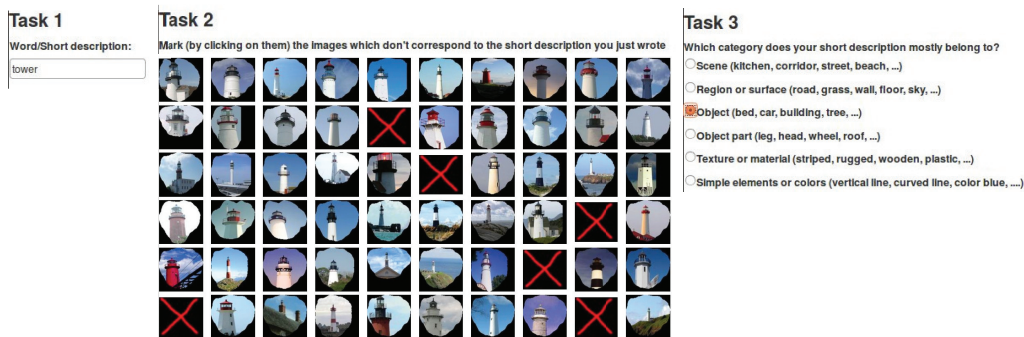


Figure 6: AMT interface for unit concept annotation. There are three tasks in each annotation.

### 3.3 IDENTIFYING THE SEMANTICS OF INTERNAL UNITS

In Section 3.2, we found the exact RFs of units and observed that activation regions tended to become more semantically meaningful with increasing depth of layers. In this section, our goal is to understand and quantify the precise semantics learned by each unit.

In order to do this, we ask workers on Amazon Mechanical Turk (AMT) to identify the common theme or *concept* that exists between the top scoring segmentations for each unit. We expect the tags provided by naive annotators to reduce biases. Workers provide tags without being constrained to a dictionary of terms that could bias or limit the identification of interesting properties.

Specifically, we divide the task into three main steps as shown in Fig. 6. We show workers the top 60 segmented images that most strongly activate one unit and we ask them to (1) identify the concept, or semantic theme given by the set of 60 images e.g., car, blue, vertical lines, etc, (2) mark the set of images that do not fall into this theme, and (3) categorize the concept provided in (1) to one of 6 semantic groups ranging from low-level to high-level: simple elements and colors (e.g., horizontal lines, blue), materials and textures (e.g., wood, square grid), regions and surfaces (e.g., road, grass), object parts (e.g., head, leg), objects (e.g., car, person), and scenes (e.g., kitchen, corridor). This allows us to obtain both the semantic information for each unit, as well as the level of abstraction provided by the labeled concept.

To ensure high quality of annotation, we included 3 images with high negative scores that the workers were required to identify as negatives in order to submit the task. Fig. 7 shows some example annotations by workers. For each unit, we measure its precision as the percentage of images that were selected as fitting the labeled concept. In Fig. 8.(a) we plot the average precision for ImageNet-CNN and Places-CNN for each layer.

In Fig. 8.(b-c) we plot the distribution of concept categories for ImageNet-CNN and Places-CNN at each layer. For this plot we consider only units that had a precision above 75% as provided by the AMT workers. Around 60% of the units on each layer were above that threshold. For both networks, units at the early layers (pool1, pool2) have more units responsive to simple elements and colors, while those at later layers (conv4, pool5) have more high-level semantics (responsive more to objects and scenes). Furthermore, we observe that conv4 and pool5 units in Places-CNN have higher ratios of high-level semantics as compared to the units in ImageNet-CNN.

Fig. 9 provides a different visualization of the same data as in Fig. 8.(b-c). This plot better reveals how different levels of abstraction emerge in different layers of both networks. The vertical axis indicates the percentage of units in each layer assigned to each concept category. ImageNet-CNN has more units tuned to simple elements and colors than Places-CNN while Places-CNN has more objects and scenes. ImageNet-CNN has more units tuned to object parts (with the maximum around conv4). It is interesting to note that Places-CNN discovers more objects than ImageNet-CNN despite having no object-level supervision.

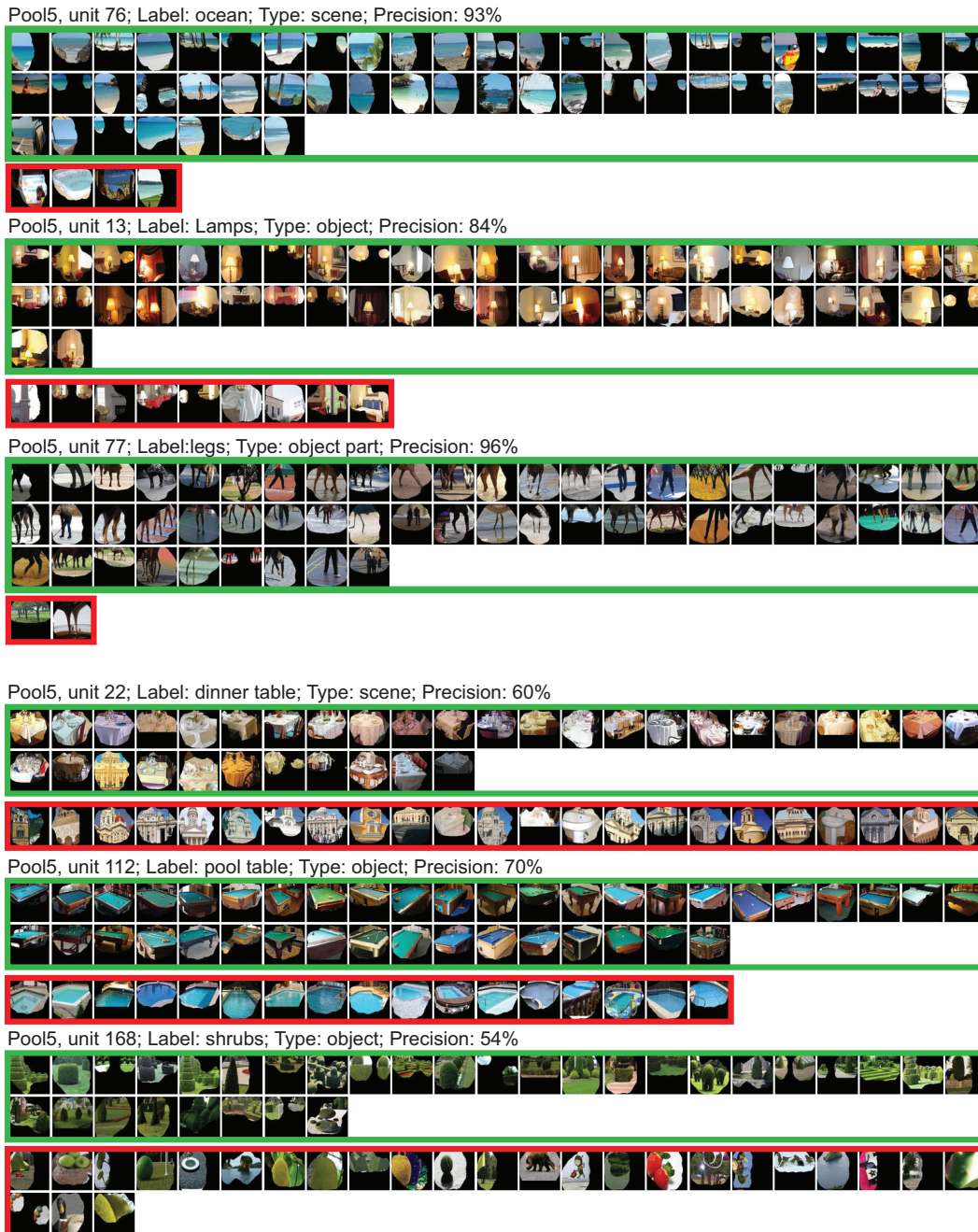


Figure 7: Examples of unit annotations provided by AMT workers for 6 units from pool5 in Places-CNN. For each unit the figure shows the label provided by the worker, the type of label, the images selected as corresponding to the concept (green box) and the images marked as incorrect (red box). The precision is the percentage of correct images. The top three units have high performance while the bottom three have low performance ( $< 75\%$ ).

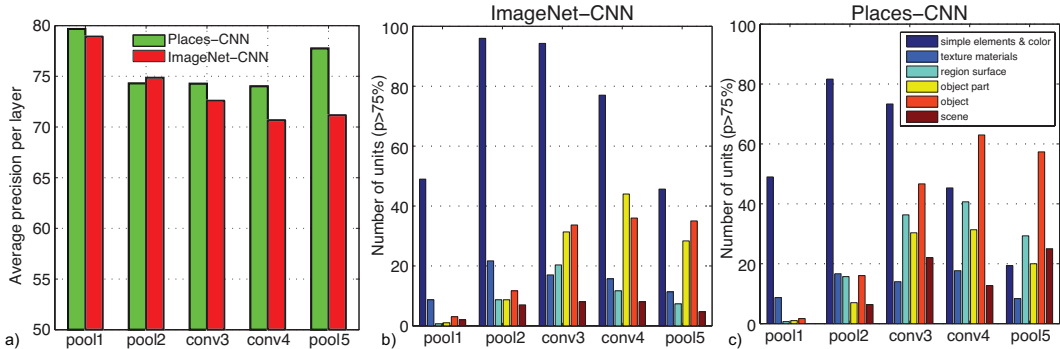


Figure 8: (a) Average precision of all the units in each layer for both networks as reported by AMT workers. (b) and (c) show the number of units providing different levels of semantics for ImageNet-CNN and Places-CNN respectively.

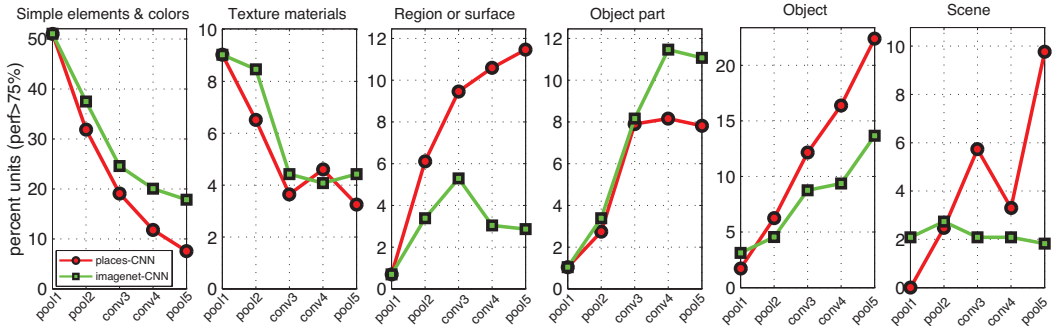


Figure 9: Distribution of semantic types found for all the units in both networks. From left to right, each plot corresponds to the distribution of units in each layer assigned to simple elements or colors, textures or materials, regions or surfaces, object parts, objects, and scenes. The vertical axis is the percentage of units with each layer assigned to each type of concept.

#### 4 EMERGENCE OF OBJECTS AS THE INTERNAL REPRESENTATION

As shown before, a large number of units in pool5 are devoted to detecting objects and scene-regions (Fig. 9). But what categories are found? Is each category mapped to a single unit or are there multiple units for each object class? Can we actually use this information to segment a scene?

##### 4.1 WHAT OBJECT CLASSES EMERGE?

To answer the question of why certain objects emerge from pool5, we tested ImageNet-CNN and Places-CNN on fully annotated images from the SUN database (Xiao et al., 2014). The SUN database contains 8220 fully annotated images from the same 205 place categories used to train Places-CNN. There are no duplicate images between SUN and Places. We use SUN instead of COCO (Lin et al., 2014) as we need dense object annotations to study what the most informative object classes for scene categorization are, and what the natural object frequencies in scene images are. For this study, we manually mapped the tags given by AMT workers to the SUN categories.

Fig. 10(a) shows the distribution of objects found in pool5 of Places-CNN. Some objects are detected by several units. For instance, there are 15 units that detect buildings. Fig. 11 shows some units from the Places-CNN grouped by the type of object class they seem to be detecting. Each row shows the top five images for a particular unit that produce the strongest activations. The segmentation shows the regions of the image for which the unit is above a certain threshold. Each unit seems to be selective to a particular appearance of the object. For instance, there are 6 units that detect lamps, each unit detecting a particular type of lamp providing finer-grained discrimination; there are 9 units selective to people, each one tuned to different scales or people doing different tasks.



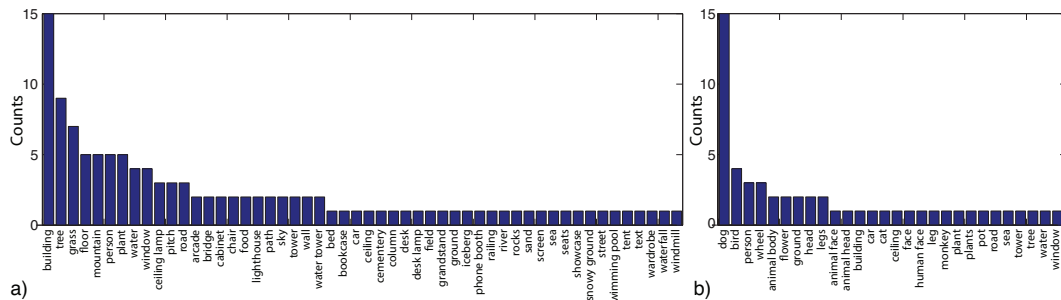


Figure 10: Object counts of CNN units discovering each object class for (a) Places-CNN and (b) ImageNet-CNN.

Fig. 10(b) shows the distribution of objects found in pool5 of ImageNet-CNN. ImageNet has an abundance of animals among the categories present: in the ImageNet-CNN, out of the 256 units in pool5, there are 15 units devoted to detecting dogs and several more detecting parts of dogs (body, legs, ...). The categories found in pool5 tend to follow the target categories in ImageNet.

Why do those objects emerge? One possibility is that the objects that emerge in pool5 correspond to the most frequent ones in the database. Fig. 12(a) shows the sorted distribution of object counts in the SUN database which follows Zipf’s law. Fig. 12(b) shows the counts of units found in pool5 for each object class (same sorting as in Fig. 12(a)). The correlation between object frequency in the database and object frequency discovered by the units in pool5 is 0.54. Another possibility is that the objects that emerge are the objects that allow discriminating among scene categories. To measure the set of discriminant objects we used the ground truth in the SUN database to measure the classification performance achieved by each object class for scene classification. Then we count how many times each object class appears as the most informative one. This measures the number of scene categories a particular object class is the most useful for. The counts are shown in Fig. 12(c). Note the similarity between Fig. 12(b) and Fig. 12(c). The correlation is 0.84 indicating that the network is automatically identifying the most discriminative object categories to a large extent.

Note that there are 115 units in pool5 of Places-CNN not detecting objects. This could be due to incomplete learning or a complementary texture-based or part-based representation of the scenes. Therefore, although objects seem to be a key part of the representation learned by the network, we cannot rule out other representations being used in combination with objects.

## 4.2 OBJECT LOCALIZATION WITHIN THE INNER LAYERS

Places-CNN is trained to do scene classification using the output of the final layer of logistic regression and achieves state-of-the-art performance. From our analysis above, many of the units in the inner layers could perform interpretable object localization. Thus we could use this single Places-CNN with the annotation of units to do both scene recognition and object localization in a single forward-pass. Fig. 13 shows an example of the output of different layers of the Places-CNN using the tags provided by AMT workers. Bounding boxes are shown around the areas where each unit is activated within its RF above a certain threshold.

In Fig. 14 we provide the segmentation performance of the objects discovered in pool5 using the SUN database. The performance of many units is very high which provides strong evidence that they are indeed detecting those object classes despite being trained for scene classification.

## 5 CONCLUSION

We find that object detectors emerge as a result of learning to classify scene categories, showing that a single network can support recognition at several levels of abstraction (e.g., edges, textures, objects, and scenes) without needing multiple outputs or networks. While it is common to train a network to do several tasks and to use the final layer as the output, here we show that reliable outputs can be extracted at each layer. As objects are the parts that compose a scene, detectors tuned to the objects that are discriminant between scenes are learned in the inner layers of the network. Note

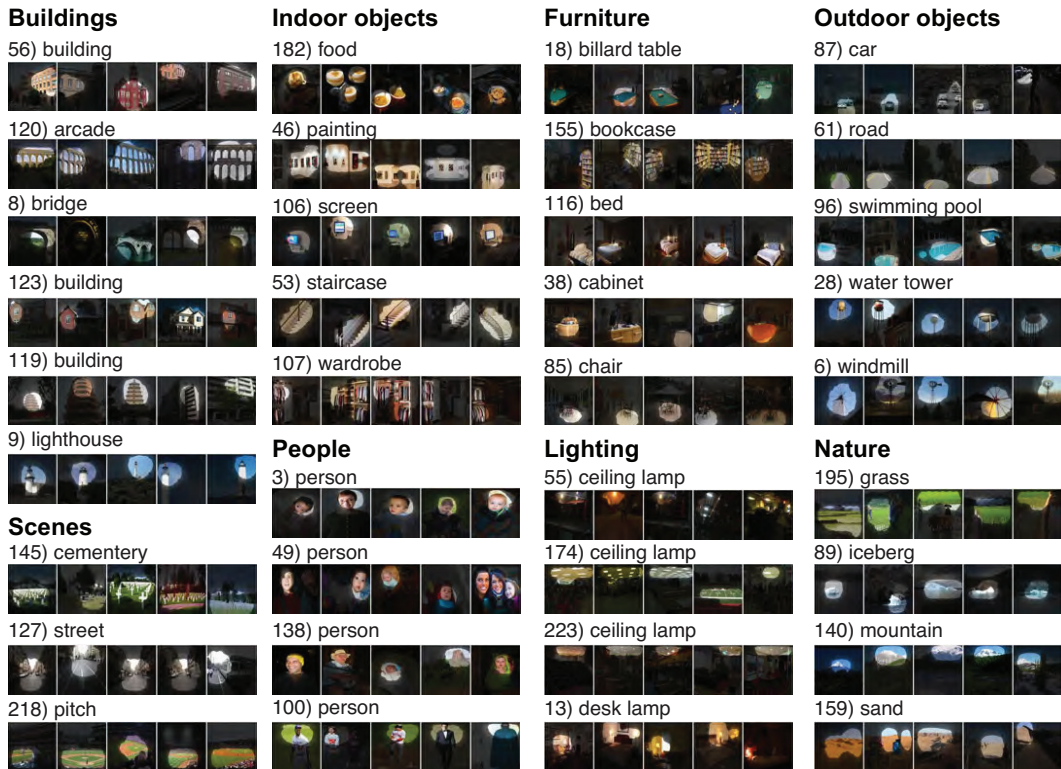


Figure 11: Segmentations using pool5 units from Places-CNN. Many classes are encoded by several units covering different object appearances. Each row shows the 5 most confident images for each unit. The number represents the unit number in pool5.

that only informative objects for specific scene recognition tasks will emerge. Future work should explore which other tasks would allow for other object classes to be learned without the explicit supervision of object labels.

#### ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. 1016862 to A.O, ONR MURI N000141010933 to A.T, as well as MIT Big Data Initiative at CSAIL, Google and Xerox Awards, a hardware donation from NVIDIA Corporation, to A.O and A.T.

#### REFERENCES

- Agrawal, Pulkrit, Girshick, Ross, and Malik, Jitendra. Analyzing the performance of multilayer neural networks for object recognition. *ECCV*, 2014.
- Bergamo, Alessandro, Bazzani, Loris, Anguelov, Dragomir, and Torresani, Lorenzo. Self-taught object localization with deep networks. *arXiv preprint arXiv:1409.3964*, 2014.
- Biederman, Irving. *Visual object recognition*, volume 2. MIT press Cambridge, 1995.
- Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes challenge. *IJCV*, 2010.
- Farabet, Clement, Couprie, Camille, Najman, Laurent, and LeCun, Yann. Learning hierarchical features for scene labeling. *TPAMI*, 2013.

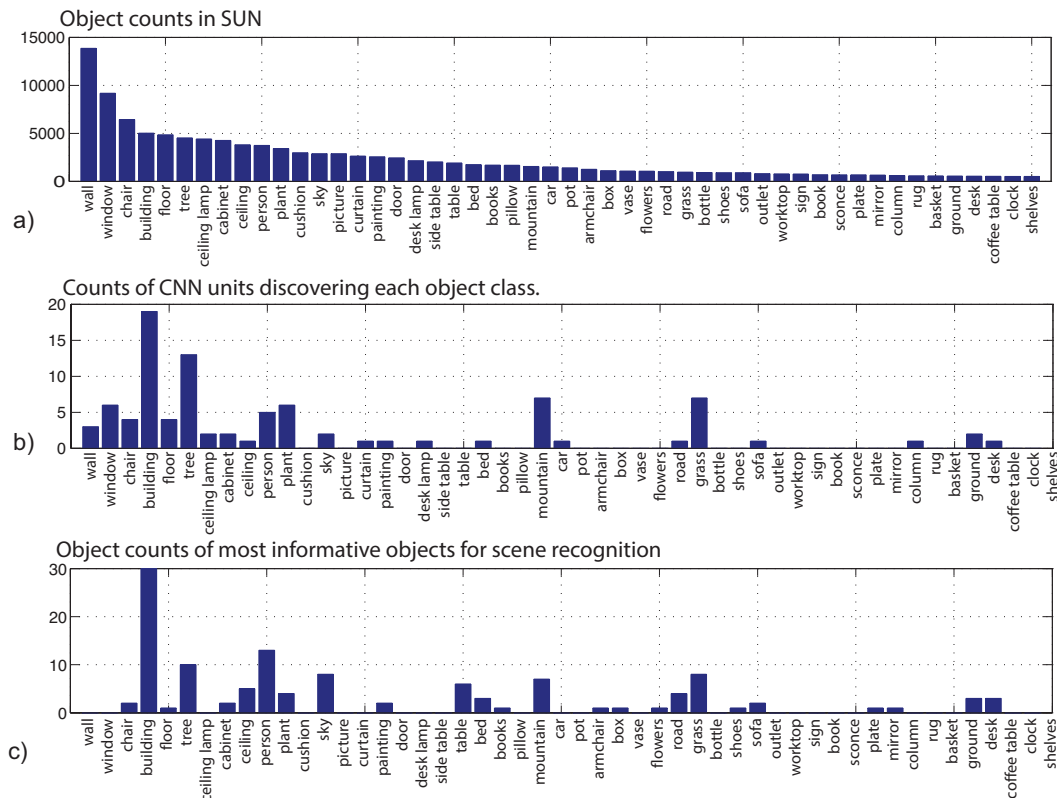


Figure 12: (a) Object frequency in SUN (only top 50 objects shown), (b) Counts of objects discovered by pool5 in Places-CNN. (c) Frequency of most informative objects for scene classification.



Figure 13: Interpretation of a picture by different layers of the Places-CNN using the tags provided by AMT workers. The first shows the final layer output of Places-CNN. The other three show detection results along with the confidence based on the units' activation and the semantic tags.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

Grangier, D., Bottou, L., and Collobert, R. Deep convolutional networks for scene parsing. *TPAMI*, 2009.

Jia, Yangqing. Caffe: An open source convolutional architecture for fast feature embedding, 2013.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Lazebnik, Svetlana, Schmid, Cordelia, and Ponce, Jean. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

Le, Quoc V. Building high-level features using large scale unsupervised learning. In *ICASSP*, 2013.

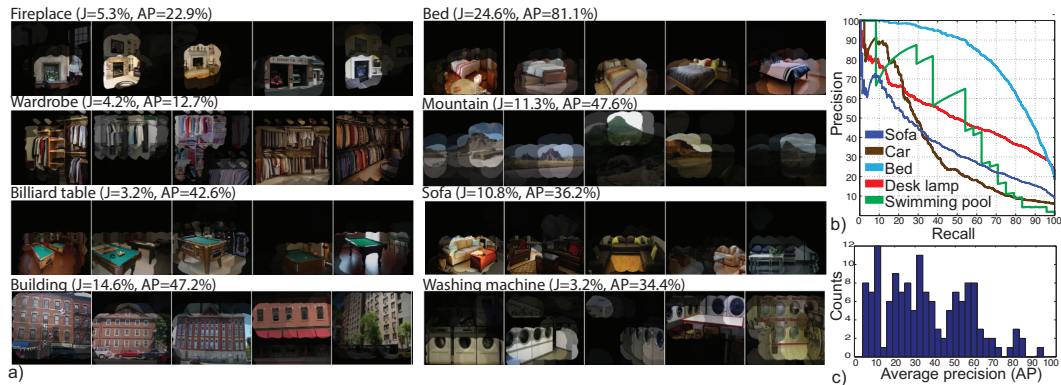


Figure 14: (a) Segmentation of images from the SUN database using pool5 of Places-CNN ( $J$  = Jaccard segmentation index, AP = average precision-recall.) (b) Precision-recall curves for some discovered objects. (c) Histogram of AP for all discovered object classes.

Li, Li-Jia, Su, Hao, Fei-Fei, Li, and Xing, Eric P. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pp. 1378–1386, 2010.

Lin, Tg-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollr, Piotr, and Zitnick, C. Lawrence. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Long, Jonathan, Zhang, Ning, and Darrell, Trevor. Do convnets learn correspondence? In *NIPS*, 2014.

Oliva, A. and Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 2006.

Oquab, Maxime, Bottou, Léon, Laptev, Ivan, Sivic, Josef, et al. Weakly supervised object recognition with convolutional neural networks. In *NIPS*. 2014.

Pandey, M. and Lazebnik, S. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011.

Pérez, Patrick, Gangnet, Michel, and Blake, Andrew. Poisson image editing. *ACM Trans. Graph.*, 2003.

Renninger, Laura Walker and Malik, Jitendra. When is scene identification just texture recognition? *Vision research*, 44(19):2301–2311, 2004.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tanaka, Keiji. Neuronal mechanisms of object recognition. *Science*, 262(5134):685–688, 1993.

Tang, Yichuan, Srivastava, Nitish, and Salakhutdinov, Ruslan R. Learning generative models with visual attention. In *NIPS*. 2014.

Xiao, J, Ehinger, K A., Hays, J, Torralba, A, and Oliva, A. SUN database: Exploring a large collection of scene categories. *IJCV*, 2014.

Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lipson, Hod. How transferable are features in deep neural networks? In *NIPS*, 2014.

Zeiler, M. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

Zhou, Bolei, Lapedriza, Agata, Xiao, Jianxiong, Torralba, Antonio, and Oliva, Aude. Learning deep features for scene recognition using places database. In *NIPS*, 2014.