Visualizing and Comparing AlexNet and VGG using Deconvolutional Layers

Wei Yu W.YU@HIT.EDU.CN
Kuiyuan Yang KUYANG@MICROSOFT.COM
Yalong Bai YLBAI@MTLAB.HIT.EDU.CN
Tianjun Xiao TIXIAO@MICROSOFT.COM
Hongxun Yao H.YAO@HIT.EDU.CN
Yong Rui YONGRUI@MICROSOFT.COM

Abstract

Convolutional Neural Networks (CNNs) have been keeping improving the performance on ImageNet classification since it is firstly successfully applied in the task in 2012. To achieve better performance, the complexity of CNNs is continually increasing with deeper and bigger architectures. Though CNNs achieved promising external classification behavior, understanding of their internal work mechanism is still limited. In this work, we attempt to understand the internal work mechanism of CNNs by probing the internal representations in two comprehensive aspects, i.e., visualizing patches in the representation spaces constructed by different layers, and visualizing visual information kept in each layer. We further compare CNNs with different depths and show the advantages brought by deeper architecture.

1. Introduction

With decades of dedicated research efforts, CNNs recently made another wave of significant breakthroughs in image classification tasks, and achieved comparable error rates to well-trained human on ILSVRC2014[1] image classification task (Russakovsky et al., 2014). The well-trained CNNs on ILSVRC2012 even rival the representational performance of IT cortex of macaques on visual object recognition benchmark created by (Cadieu et al., 2013). CNN was introduced by LeCun et al. (1989) for handwritten digits classification, the designed CNN architecture was inspired by Hubel and Wiesel's discovery of locally-sensitive, orientation-selective neurons in the cat's visual system (Hubel & Wiesel, 1962). With several *big* (in terms of number of filters in each layer) and *deep* (in terms of number of layers) CNNs, Krizhevsky et al. (2012) won the image classification competition in ILSVRC2012 by a large margin over traditional methods. The classification error rate was further significantly reduced by Szegedy et al. (2014); Simonyan & Zisserman (2014); He et al. (2014) with *deeper* CNNs. In order to understand CNN, we attempt to gain the insights into the internal behavior of trained models using the visualization technologies, as illustrated in Figure 1.

Though external classification behavior of CNNs is encouraging, the understanding of CNNs' internal work mechanism is still limited. In this paper[2], we attempt to understand the internal work mechanism by probing the internal representations (a.k.a. internal activations) in two aspects:

1. We visualize representation spaces constructed by internal layers. In CNN, each layer constructs a representation space for image patches with corresponding receptive field size. The representation spaces are visualized by t-SNE (Van der Maaten & Hinton, 2008), where patches with similar representations in a layer are showed in close positions in a 2-dimensional space.

2. We visualize internal representations for an image via deconvolution (Zeiler & Fergus, 2014). In CNN, each layer generates a new representations for an image in an information processing way, the new representation of each layer is projected back to the pixel space for understanding what information is kept.

Considering the deeper CNN designed by (Simonyan & Zisserman, 2014) has achieved significantly better performance than the CNN used by Krizhevsky et al. (2012), we further compare the internal work mechanism of these two

[1]ILSVRC stands for ImageNet Large Scale Visual Recognition Challenge, the challenge has been run annually from 2010 to present.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

[2]This work was done when Wei Yu and Yalong Bai were interns at Microsoft Research.

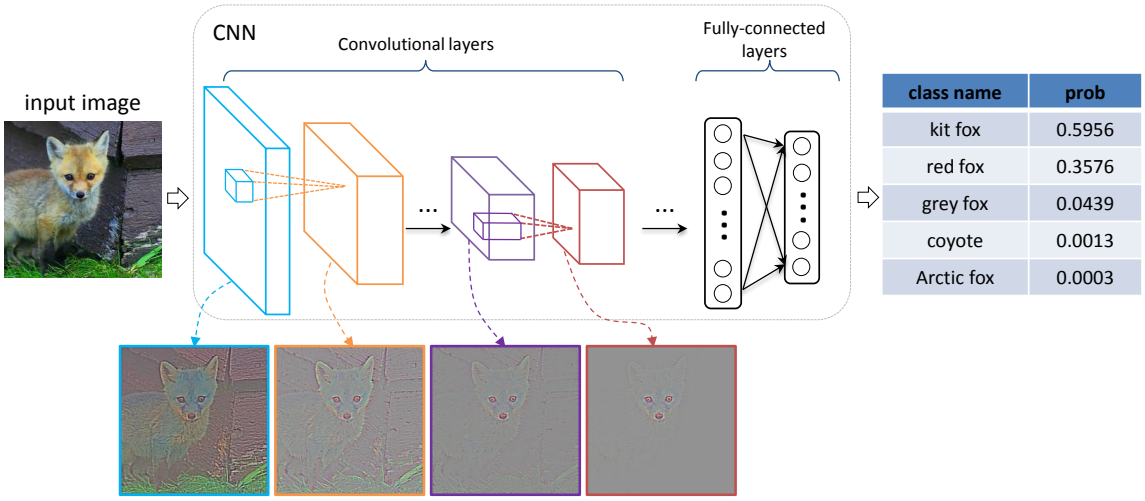

Figure 1. The illustration of external and internal behavior of a CNN. The external behavior is output prediction categories for input images. The internal behavior is to be probed by visualizing the representation spaces constructed by each layer and the visual information kept in each layer.

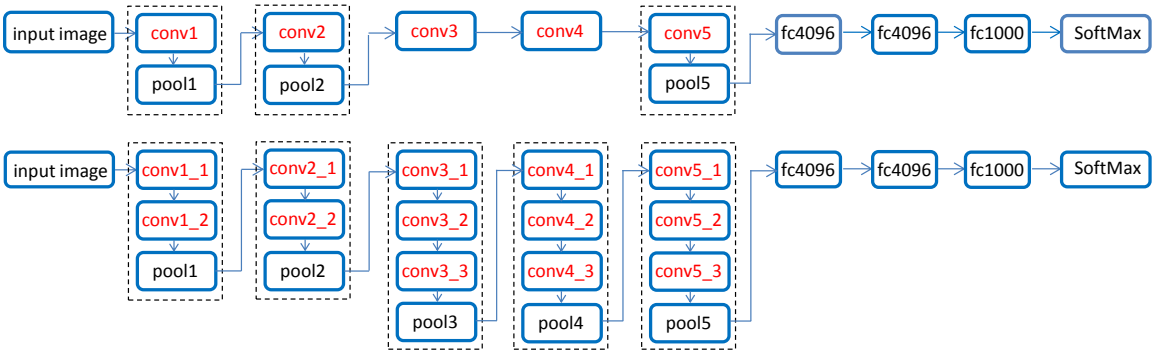

Figure 2. The architectures of AlexNet and VGG-16. The top part is the architecture of AlexNet, and the bottom part is the architecture of VGG-16

CNNs (named as VGG-16 and AlexNet respectively). The results show that VGG-16 is better at removing unrelated background information.

The rest of the paper is organized as follows. We cover related work in Section 2 and then describe the architectures of VGG-16 and AlexNet in Section 3. The visualization of internal representations is introduced section 4. The comparison of VGG-16 and AlexNet is present in Section 5. We discuss the conclusion in Section 6.

2. Related work

In order to open the "black box" of CNN, researchers have proposed several approaches to visualize the filters[3] to probe what kinds of patterns are these filters favoring. Krizhevsky et al. (2012) directly visualized the filters learned in the first layer to judge whether the parameters of a trained CNN is apart from randomness. Since filters in high layers receive inputs from their previous layers instead of pixels, there is no direct way to visualize them in pixel space. Girshick et al. (2013); Yu et al. (2014) used a non-parametric method, a filter is visualized by image patch-

[3]In CNN, neurons are organized by layer, each neuron receives neuron activations from previous layer and weighted by weights in the connections. In fully-connected layer, each neuron is connected to all neurons in previous layers with its own weights. While in convolutional layer, neurons are further organized by feature map and only locally connected to neurons in previous layer. Moreover, all neurons in a feature map share the same filter (weights bank), so neurons in a feature map are favoring the same kind of pattern.

Table 1. Size and stride of receptive fields in each layer of VGG-16.

layer	**c1_1**	**c1_2**	**p1**	**c2_1**	**c2_2**	**p2**	**c3_1**	**c3_2**	**c3_3**
size	3	5	6	10	14	16	24	32	40
stride	1	1	2	2	2	4	4	4	4
layer	**p3**	**c4_1**	**c4_2**	**c4_3**	**p4**	**c5_1**	**c5_2**	**c5_3**	**p5**
size	44	60	76	92	100	132	164	196	212
stride	8	8	8	8	16	16	16	16	32

Table 2. Size and stride of receptive fields in each layer AlexNet.

layer	**c1**	**p1**	**c2**	**p2**	**c3**	**c4**	**c5**	**p5**
size	11	15	47	55	87	119	151	167
stride	4	8	8	16	16	16	16	32

es with highest activations to this filter. Zeiler & Fergus (2014) also visualize filters by patches with highest activations, together with their reconstructed versions via deconvolution network. The reconstructed patch only focuses on the discriminant structure in original patch, and better exhibit the filters' favored patterns.

In contrast to the above non-parametric methods, Erhan et al. (2009) visualised deep neural networks by finding an image which maximises the neuron activation of interest by carrying out an optimisation using gradient ascent in the image space. The method was later used by Le et al. (2012) to visualize the "cat" neuron learned in a deep unsupervised auto-encoder. Recently, Simonyan et al. (2013) employed this method to visualize neurons corresponding categories in last layer. Mahendran & Vedaldi (2014) generalize this method to find images in the image space with similar activations in some layer to an input image.

Existing methods mostly focus on visualizing individual filter or neuron, and only partially reveal the internal work mechanism of CNN. In this paper, we do visualization in more comprehensive ways, where the representation spaces constructed by all filters of a layer are visualized, and all activations of a layers are used to reconstruct the image via deconvolution network.

3. CNN configuration details

3.1. Architecture

In this section, we first introduce the architectures of two CNNs (AlexNet and VGG-16). We used the released VGG-16[4] which has achieved 29.5% top-1 error rate on ILSVRC2012 validation set with single centre-view prediction (Simonyan & Zisserman, 2014). In particular, we retrain a model of AlexNet without local response normalization layers, which achieved 42.6% top-1 error rate with single center-view prediction (Krizhevsky et al., 2012). Both CNNs receive RGB image with fixed size of 224×224 subtracted by the mean image computed on training set.

The overall architectures of these two CNNs are illustrated in Figure 2. AlexNet consists of 8 weight layers including 5 convolutional layers and 3 fully-connected layers, and three max-pooling layers are used following the first, second and fifth convolutional layers. The first convolutional layer has 96 filters of size 11×11 with a stride of 4 pixels and padding with 2 pixels. The stride and padding of other convolutional layers are set as 1 pixel. The second convolutional layer has 256 filters of size 5×5. The third, fourth and fifth convolutional layers have 384, 384 and 256 filters with size of 3×3 respectively.

The used VGG-16 is much deeper which consists of 16 weight layers including thirteen convolutional layers with filter size of 3×3, and 3 fully-connected layers. The configurations of fully-connected layers in VGG-16 are the same with AlexNet. The stride and padding of all convolutional layers are fixed to 1 pixel. All convolutional layers are divided into 5 groups and each group is followed by a max-pooling layer as showed in Figure 2. Max-pooling is carried out over a 2×2 window with stride 2. The number of filters of convolutional layer group starts from 64 in the first group and then increases by a factor of 2 after each max-pooling layer, until it reaches 512.

3.2. Receptive field

The receptive field of a neuron is its covered region in the image plane. The size and stride of receptive field of a neuron is determined by the CNN architecture. Table 1 and Table 2 list the receptive field size and stride of neurons of different layers in VGG-16 and AlexNet, respectively. Although both CNNs output feature maps with the same size in last pooling layer, the neurons of VGG-16 cover the receptive field with larger size.

4. Internal work mechanism of CNN

In this section, we focus on visualizing the representation spaces constructed by different layers and visual information extracted in different layers.

[4] http://www.robots.ox.ac.uk/~vgg/research/very_deep/

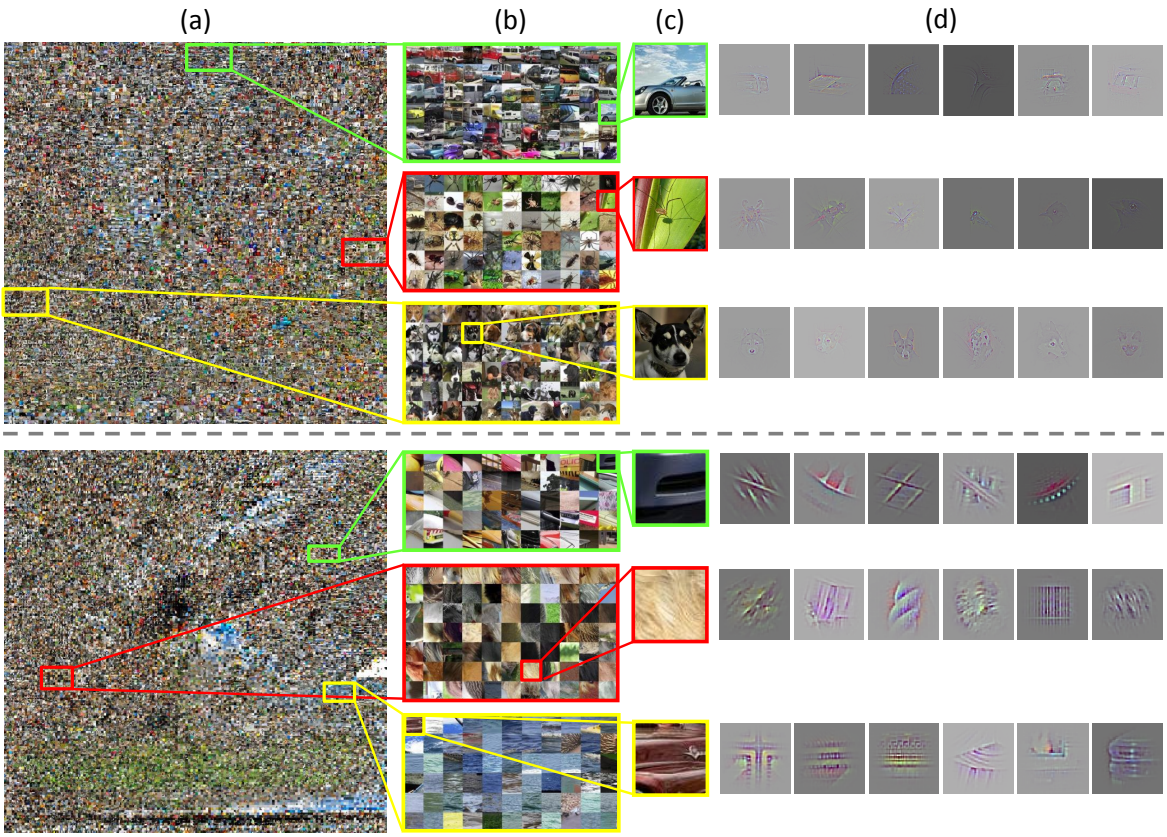

Figure 3. Representation space of c5_3 (top part) and c4_1 (bottom part). Column (a) is the embedding plane by dimensionality reduction, column (b) is the three subregions sampled from whole plane, column (c) is the selected patches and column (d) shows filters with high activations of corresponding patches in column (c). Filters showed by the reconstructed results of patches with highest activations in ILSVRC2012 validation set. (Best viewed in color)

4.1. Representation space

As each filter generates an activation for a patch located in its receptive field, all filters in a layer actually construct a representation space for patches with size of the corresponding receptive field. Visualizing filters by their highest activated patches only partially shows each dimension of the representation space. Here, we utilize t-SNE (Van der Maaten & Hinton, 2008) to visualize the representation space through dimension reduction, where patches close in the representation space are embedded close in the 2-dimensional space. As there are lots of empty and overlapping regions in original embedding, we fill every patch with its nearest neighbor in original embedding. Figure 3 illustrates the representation spaces of two selected layers.

In the representation space of c5_3, semantic-level similar patches are embedded close, e.g. the three zoom-in subregions are about *car, insect, dog face*. The filters with highest activations for patches in these subregions also showed semantic-level consistence. Meanwhile, ways to representing patches are different. In the *car* example, the filters with high activations are car parts, such as car window (1^{st}, 5^{th} and 6^{th} filter), the part of bonnet (2^{nd} filter), wheel (3^{rd} and 4^{th} filter). In the *dog* example, the filters with high activations are the dogs with different appearances or poses.

In the representation space of c4_1, near patches are with similar texture or simple shape. Patches in the first subregion are oblique lines or arcs. Patches in the second subregions are mainly about animal furs, while patches in the third subregion are mainly about water texture.

4.2. Visual information extraction

In CNN, each layer forms a new representation for an input image by gradually extracting discriminative information. Here, we visualize the new representation of a layer via deconvolution network (Zeiler & Fergus, 2014). The visualization reveals the discriminant image structure that generates that representation. Figure 4 shows several examples of visual results reconstructed from representations

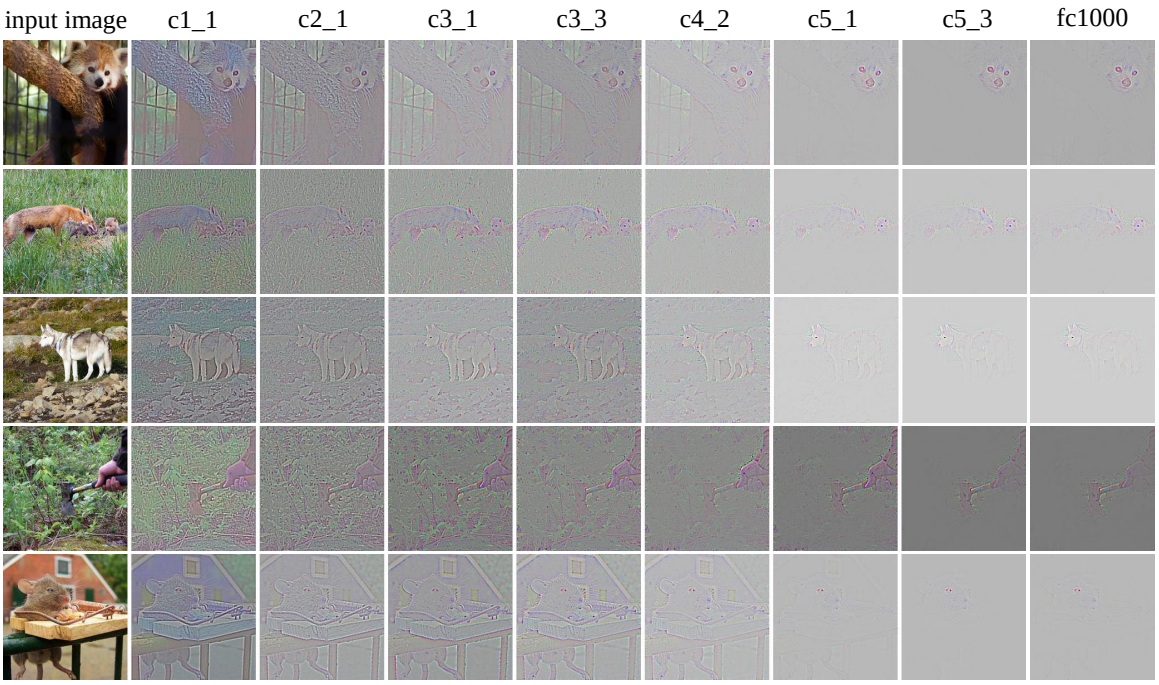

Figure 4. Visualization of visual information processing through different layers. The first column shows five input images, the following 7 columns show the reconstruction results of every two convolutional layers of VGG-16, and the last column shows the reconstruction result of last fully-connect layer. From top to bottom, the images are from *lesser panda, kit fox, Siberian husky, hatchet* and *mousetrap* respectively. (Best viewed electronically)

from several layers. It can be observed that, unrelated information is gradually removed from low layers to high layers (from left to right in the figure). The reconstructions of last layer only keeps the most discriminate parts. The last row shows an interesting case where mouse head is kept as discriminative part for prediction mousetrap, this is due to mouse and mousetrap have high co-occurrence rate in images, and mouse is more discriminant in this image.

5. Comparisons between CNNs

In this section, we attempt to compare the prediction processes of VGG-16 with AlexNet through analyzing visual information kept in different layers.

Figure 5 shows the representation sparsity of all convolutional layers and max-pooling layers for VGG-16 and AlexNet. The sparsity is measured by the proportion of zero activations of a layer on ILSVRC2012 validation set. In general, the sparsity increases from low layers to high layers. To be noted that the decrease of sparsity in max-pooling layer is caused by the max operator which decreases the number of zero activations. The high layers of VGG-16 are with higher sparsity than AlexNet, which also demonstrates VGG-16 is with better representation ability and removing unrelated information.

Previous section has shown the process of visual information extraction using visual results reconstructed from the internal representations of different layers. It was shown that unrelated parts are gradually removed and the discriminative parts gradually stand out. In Figure 6, we compare the process carried by VGG-16 and AlexNet through several examples. In contrast to VGG-16, AlexNet retains more unrelated background information in last convolutional layer, which often disturbs the final prediction.

6. Conclusion

In this paper, we probe the internal work mechanism of CNN via visualizing the internal representations formed by different layers in two aspects. The visualizations of representation spaces constructed from different layers demonstrate the ability of CNN in sorting patterns gradually from low level to high level. The visualizations of the reconstructed images from representations of different layers show CNN's ability in gradually extracting discriminant information. Through comparison of CNNs with different depths, it shows that deeper CNN is better at extracting the discriminant information, which improves the predic-

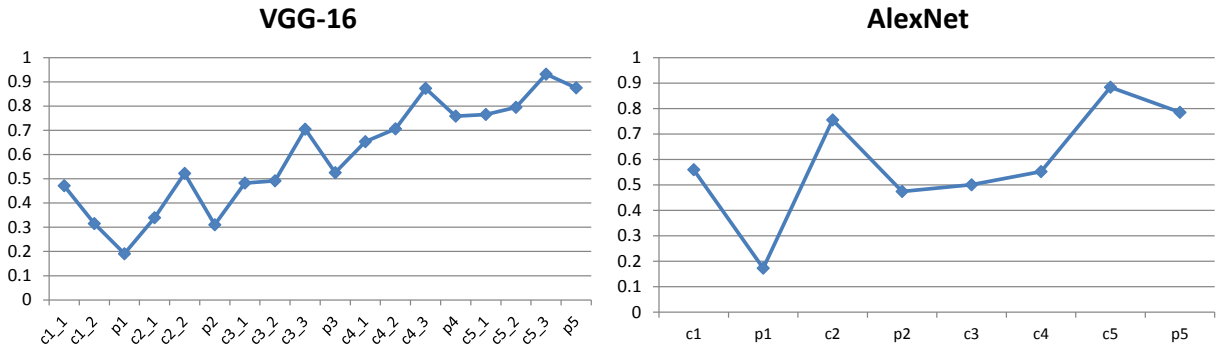

Figure 5. The sparsity of each layer. The left part is the sparsity of VGG-16 and right part is the sparsity of AlexNet.

tion performance.

References

Cadieu, Charles F, Hong, Ha, Yamins, Dan, Pinto, Nicolas, Majaj, Najib J, and DiCarlo, James J. The neural representation benchmark and its evaluation on brain and machine. *arXiv preprint arXiv:1301.3530*, 2013.

Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Visualizing higher-layer features of a deep network. *Dept. IRO, Université de Montréal, Tech. Rep*, 2009.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pp. 346–361. 2014.

Hubel, David H and Wiesel, Torsten N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1): 106, 1962.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. ImageNet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Le, Quoc, Ranzato, Marc'Aurelio, Monga, Rajat, Devin, Matthieu, Chen, Kai, Corrado, Greg, Dean, Jeff, and Ng, Andrew. Building high-level features using large scale unsupervised learning. In Langford, John and Pineau, Joelle (eds.), *ICML*, ICML '12, pp. 81–88, New York, NY, USA, July 2012. Omnipress. ISBN 978-1-4503-1285-1.

LeCun, Yann, Boser, Bernhard, Denker, John S, Henderson, Donnie, Howard, Richard E, Hubbard, Wayne, and Jackel, Lawrence D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551, 1989.

Mahendran, Aravindh and Vedaldi, Andrea. Understanding deep image representations by inverting them. *arXiv preprint arXiv:1412.0035*, 2014.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. ImageNet Large Scale Visual Recognition Challenge. *arXiv preprint arXiv:1409.0575*, 2014.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *arXiv:1409.4842*, 2014.

Van der Maaten, Laurens and Hinton, Geoffrey. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Yu, Wei, Yang, Kuiyuan, Bai, Yalong, Yao, Hongxun, and Rui, Yong. DNN Flow: DNN feature pyramid based image matching. In *BMVC*, 2014.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *ECCV*, pp. 818–833. 2014.

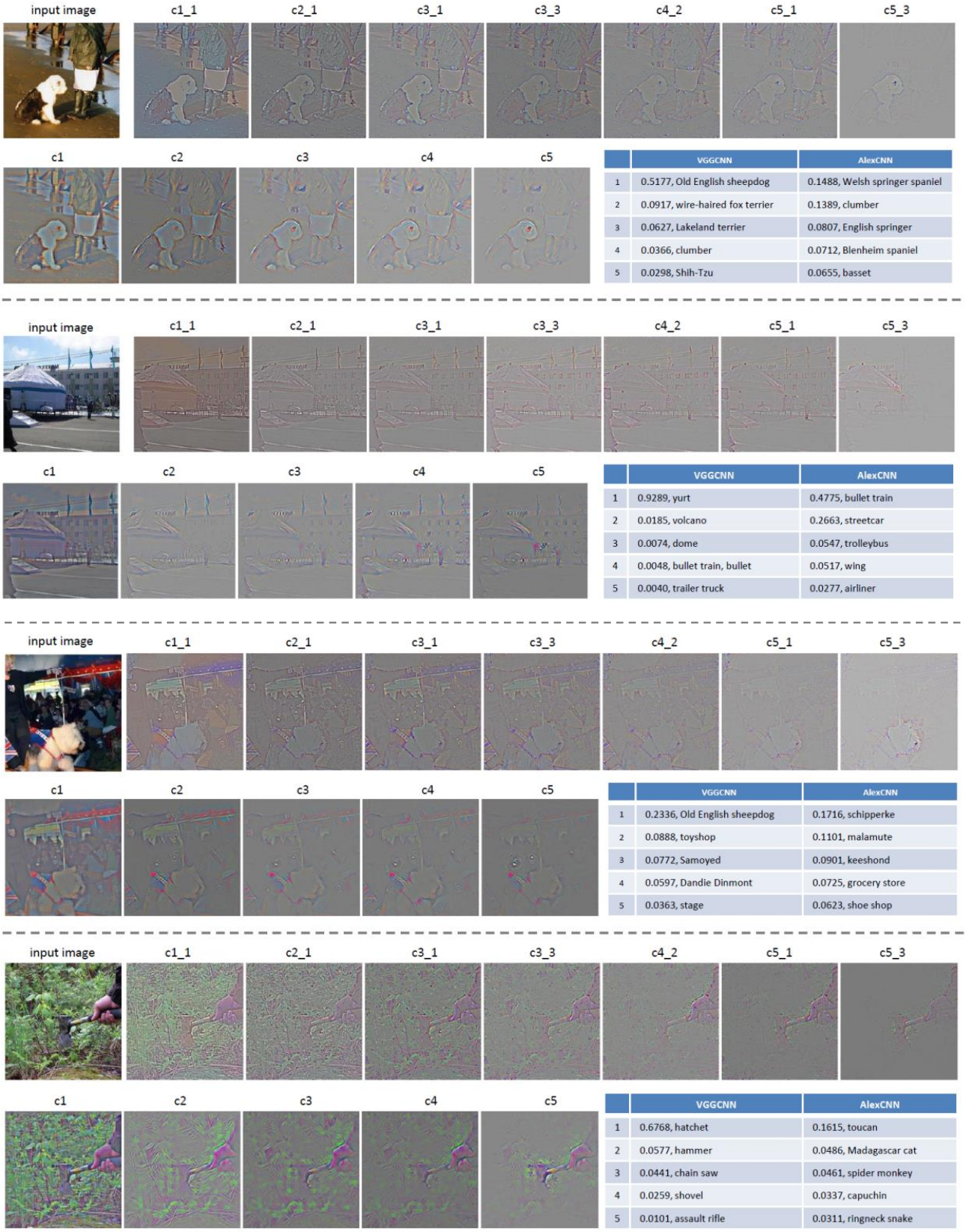

Figure 6. Comparison of visual information extraction. The extraction process of VGG-16 and AlexNet are visualized for four exemplar images. For each exemplar image, the first row show the input image followed by the reconstructed images of different layers of VGG-16, the second row shows the reconstructed images of different layers of AlexNet followed by the top-5 prediction results on the image.