Understanding Convolutional Neural Networks in Terms of Category-level Attributes

Makoto Ozeki, Takayuki Okatani

Tohoku University, Japan

Abstract. It has been recently reported that convolutional neural networks (CNNs) show good performances in many image recognition tasks. They significantly outperform the previous approaches that are not based on neural networks particularly for object category recognition. These performances are arguably owing to their ability of discovering better image features for recognition tasks through learning, resulting in the acquisition of better internal representations of the inputs. However, in spite of the good performances, it remains an open question why CNNs work so well and/or how they can learn such good representations. In this study, we conjecture that the learned representation can be interpreted as *category-level attributes* that have good properties. We conducted several experiments by using the dataset AwA (Animals with Attributes) and a CNN trained for ILSVRC-2012 in a fully supervised setting to examine this conjecture. We report that there exist units in the CNN that can predict some of the 85 semantic attributes fairly accurately, along with a detailed observation that this is true only for visual attributes and not for non-visual ones. It is more natural to think that the CNN may discover not only semantic attributes but non-semantic ones (or ones that are difficult to represent as a word). To explore this possibility, we perform zero-shot learning by regarding the activation pattern of upper layers as attributes describing the categories. The result shows that it outperforms the state-of-the-art with a significant margin.

1 Introduction

It has been recently reported in a number of literatures that convolutional neural networks (CNNs) show state-of-the-art performances in many benchmark tests, such as object category recognition, handwritten character recognition, medical image applications etc.; [13, 9] to name a few. The main reason for such high performance of CNNs is arguably due to their ability of learning features. This ability is considered to be particularly advantageous for difficult problems, such as object category recognition, for which it is unclear what features should be extracted from images. Paying attention on how the inputs are represented internally in the networks as a result of learning, one may think that they learn the representations themselves [1].

Despite their success, we lack understanding of why CNNs work so well. For example, it is unclear what in the images the learned networks actually look at and how the input image is represented in them.

This is in stark contrast with the recent accelerated improvements of methods for training deep networks [2, 8, 13, 6]. This lack of understanding leads to real problems; for example, a lot of trial-and-errors are necessary when designing the network architecture for each problem.

There are only a few studies that have contributed to the understanding of convolutional and similar networks [12, 11, 15]. They share the same view that the features are extracted in a hierarchical manner, in order of simpler to more complex features, in their layers. Although it is interesting as it agrees with the findings of neuroscience, these results are merely "visualization" of the learned features and is far from the full understanding of convolutional networks.

In this paper, towards their better understanding, we consider a different approach, which is to attempt to understand them in terms of category-level attributes. Category-level attributes are various types of properties possessed by the categories to be recognized (e.g., general objects) such that they describe multiple categories in a distinguishable manner [10, 14, 5]. They are used as intermediate representations connecting the images and the categories to be recognized. A major application is zero-shot learning, i.e., learning to recognize new categories for which no sample is given.

Our approach is based on a conjecture that there should be some connection between the learned representation of CNNs and the category-level attributes. Good attributes which are useful for category recognition tasks such as zero-shot learning are required to describe the categories compactly as well as discriminatively. This requirement is almost the same as the requirement for good internal representations. Therefore, if CNNs can learn good internal representation, they should be good attributes, too.

In this study, we conducted a series of experiments to verify this conjecture by using AwA [10], one of the standard dataset for studying attributes; see Fig. 1. In the experiments, we use DeCAF (Deep Convolutional Activation Features) of Donahue et el. [4] to analyze a CNN trained for the 1,000 object category recognition task of ILSVRC-2012. We show through experiments that some of theses attributes have a correlation with internal units of particularly higher layers. For example, there automatically emerges in the network a "stripe" neuron (i.e., a unit), which is highly responsive to categories possessing a "stripe" attribute. We also perform zero-shot learning by regarding the activation of a high layer as new attributes. The result shows that this approach outperforms the state-of-the-art method [14] that tailors attributes for the specific task of zero-shot learning.

2 Related work

2.1 Visualization of convolutional networks

Lee et al. [12] propose convolutional DBNs (deep belief networks), which implements convolution and pooling in the framework of DBNs. Training them in an unsupervised manner, they visualize what features are learned by the networks.

They report that features are extracted in a hierarchical manner from lower to higher layers. Le et al. [11] consider a sparse deep autoencoder with a repeated structure of a local receptive field layer followed by a pooling layer. Training the autoencoder using a large number of images in an unsupervised manner, they report that there automatically emerge the units that selectively output a high response to specific objects such as cat faces, human faces, body shapes etc. automatically emerge. Zeiler et al. [15] have proposed a method for visualizing the features leaned by convolutional networks in a supervised fashion. For the network of Krizhevsky et al. [9] trained for object category recognition, they show that, similarly to the above studies, the features are extracted in a hierarchical manner corresponding to the layers.

The problem with these approaches is that although they can give us some insight into what features are learned and how they are extracted in the networks, they are merely visualization. It is difficult to use these results to immediately improve performances or to perform further analysis.

2.2 Transfer learning by deep neural networks

The recent advances in the study of deep neural networks are initiated by the study of Hinton et al. [7] on unsupervised pretraining of deep networks. Thus, it has been recognized that the deep neural networks are effective in semi-supervised learning settings, i.e., the case where there are a large number of unlabeled data and a few labeled data. Indeed, in the early studies of feature learning by deep networks [11, 12], the main focus is on unsupervised learning of image features. It was discovered that the features learned by deep networks tend to be similar in lower layers even for different training data (e.g., faces, cars, etc.).

Recently, it is shown by Donahue et al. [4] that the CNN that is trained for ILSVRC-2012 in a fully supervised setting [9] can be repurposed to fairly different tasks of object recognition and achieve the state-of-the-art performances. The methodology is to train a simple classifier such as linear SVM using the activation patterns of a certain (usually higher-level) layer of the CNN for given training samples, which may be a small set of samples. Their study implies that the CNN trained for the specific task has acquired generic representation of objects that will be useful for all sorts of visual recognition tasks.

The methodology used in the present study is similar to Dohanue et al. [4], as we use the same CNN trained for ILSVRC-2012 and use the activation patterns of its certain layer to input images for other purposes. However, our study differs in that we focus on the analyses of the features and representations learned by the CNN. To be specific, we analyze the relation between the layer activation and category-level attributes.

2.3 Category-level attributes

Lampert et al. [10] point out that for object category recognition tasks, it becomes difficult to prepare a sufficient amount of training samples for each object

category with a increasing number of the categories to be recognized. They show how this difficulty is mitigated by using attributes possessed by the object categories, and that it is possible to perform zero-shot learning, i.e., recognizing unknown categories for which no sample image is provided, by learning the intermediate relation between the images and their attributes instead of the direct relation between the images and their categories. They created the dataset AwA (Animals with Attributes), which contains fifty animal categories and their 85 attributes such as skin colors, textures, body shapes, and behaviors, as shown in Fig. 1.

The attributes defined in AwA, which are selected by human, are represented by words and have clear meaning. Thus, they are called *semantic attributes*. On the other hand, there is another type of attributes called *discriminative attributes* [5]. Discriminative attributes, which are usually discovered from data and thus need not be represented by words, are useful for some recognition tasks such as image description and zero-shot learning. Yu et al. [14] have recently proposed a method for designing such discriminative attributes that more directly helps zero-shot learning.

As it is closely related to the present study, we briefly summarize the method of Yu et al. here. Computing image features for the sample images of known categories, it first evaluates pairwise similarities among the known categories. It then determines attributes such that the image-based (dis)similarities among the categories are the best preserved in the (dis)similarities in their attribute values. Next, it determine a mapping from the image features to the attribute values such that it best reproduces their mapping for the known categories. Finally, zero-shot learning is performed using this mapping, which enables the computation of the attribute values from a test input image. There is no sample image for the unknown categories, and their relation to the attributes are unknown. Thus, they propose to use human-created pairwise similarities $\tilde{\mathbf{S}}$ between the known categories and unknown ones, which enables the computation of the attributes of the unknown categories. This method achieves 46.94% recognition rate for the task of zero-shot learning. They further propose an extended method that utilizes the known-unknown category similarity $\tilde{\mathbf{S}}$ for the design of the attributes, which improves the performance to 48.30%.

3 Relation of learned representation to category-level attributes

We conducted several experiments to examine the conjecture that *the internal representations learned by CNNs can be interpreted as category-level attributes?*

3.1 Experimental setup

As mentioned earlier, we used the dataset AwA [10] in our experiments. The dataset consists of fifty animal categories, to which 85 attributes are given. Example images with a few chosen attributes are shown in Fig. 1. All the attributes

Fig. 1. Example of the images of animals and attributes given to them in the dataset of AwA (Animals with Attributes) [10].

are listed in Table 1. We analyze a CNN by using the responses of units in its single layer to input images. Although AwA only provides precomputed image features and not the original images because of the nature of the dataset, the authors kindly provide the original images at out request. For the experiment of predicting attributes by a linear SVM and that of zero-shot learning, the fifty categories are divided into forty and ten categories, and the former is used for training and the latter for testing, as is done in the earlier studies [10, 14].

We use DeCAF [4] to compute the responses of a CNN to input images; the CNN is trained for 1,000 object category recognition task of ILSVRC-2012. (Thus, the CNN analyzed here is the same as [4].) Following [9], we have also succeeded training a similar CNN for ILSVRC-2012 and duplicated a similar result of about 60% top-1 recognition accuracy. As there was practically no difference between DeCAF and our CNN in the results of the analyses described below, we choose to show the results obtained by DeCAF for better repeatability of our results. In any case, the CNN we examined is trained for the object recognition task of ILSVRC-2012 using 1.2 million images of 1,000 object categories. Note that the 1,000 categories of ILSVRC-2012 and 50 animal categories in AwA share 17 categories.

However, there is a slight difference in our use of DeCAF from its standard usage. The features provided by DeCAF are usually the activation patterns of a layer to input images, or equivalently, the *output* of the rectified linear units in that layer. Instead of using these, we use the *inputs* to the same rectified linear units, which are merely the signals before applying the rectified linear function that discards all negative values by setting to zero.

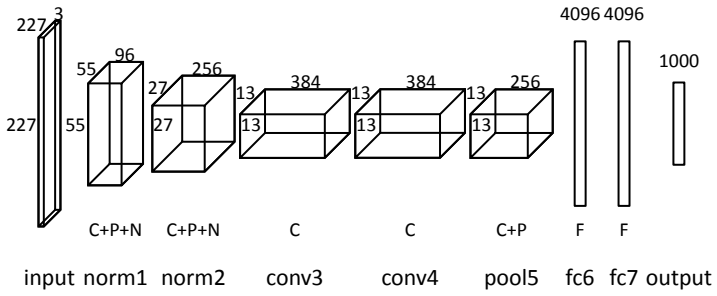

Fig. 2. The architecture of the CNN of DeCAF [4].

3.2 Predicting the semantic attributes of AwA

We first consider predicting the 85 attributes of AwA from the responses of the CNN to input images. These attributes, each of which is represented by a single word, are selected by human and describe the animal categories more or less in a distinguishable manner. Whether or not each category possesses an attribute is represented by a binary value (*yes/no*), as shown in Fig. 1. Some of them are concerned with visual properties of animals such as color, the number of legs, body shape etc., and others are with non-visual properties such as behaviors and food habits of animals.

Prediction by fc7 individual units We examined for individual units in the fc7 layer how its responses to input images relate to their attributes. To be specific, for each image of AwA and for each attribute, we pick the unit that best predicts the attribute and see its prediction accuracy. The accuracy is measured by the overlapped area s of the two histograms of the responses of that unit to the images with and without the attribute. They are normalized before the computation of s. Note that by the response of a unit, we mean the input to the rectified linear activation function, as mentioned above.

The resulting histograms for several selected attributes are shown in Fig. 3; the top row shows the top four attributes; the middle row shows attributes selected from the top 1/3 (but the top four); the bottom row shows attributes with the worst prediction accuracy. Note that the order of the two histograms (red for *yes* and blue for *no*) can be flipped horizontally, as we merely look at the separability of the attributes. Table 1 shows the results for all the attributes in the order of decreasing prediction accuracy.

It is observed from these results that some of the attributes can be predicted with very high accuracy by single unit responses, in spite of its simplicity. Moreover, the visual attributes, such as colors, textures, and body shapes, tend to

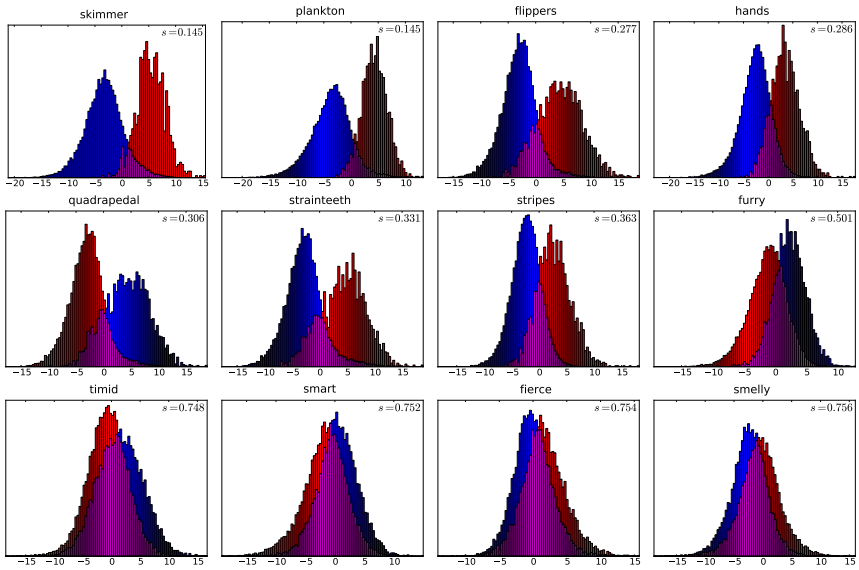

Fig. 3. Histograms of the responses of the fc7 unit that the best predicts each attribute. In each histogram, red bars indicate the images with the attribute and blue bars indicate those without the attribute. The two histograms are normalized. s is the area of their overlap, which measures the prediction (in)accuracy. Top row: the top four attributes with the highest accuracy. Middle row: selected other attributes from the top 1/3. Bottom row: selected four attributes of the worst ten.

be ranked high, whereas the non-visual attributes (shaded in the table), which describe the behaviors and other non-visual properties of the animals, tend to be ranked low. Although there are a few non-visual attributes that are ranked high, such as *swims* and *walks*, it might be possible to predict them from the surrounding environments of the animals. This might be true for the attributes with highest ranks, such as *skimmer* and *plankton*, which are solely given to animals living in water such as whale; they will be able to be predicted by simply detecting *blue* or *ocean*.

There are a few exceptions to the above observations, such as *black*, for which the prediction accuracy is low despite the fact that it is a visual attribute. This might be because of the way of determining the attributes in AwA that the attributes are given to each category, not to each image. For example, a category *sheep* is given an attribute *black*, which merely means that *some* sheep are in black; see Fig. 1 for such examples. In the above analysis, the unit associated with the attribute *black* is supposed to be activated for an input image of sheep that is not black at all.

Fig. 4 shows examples of the prediction for the attributes *hands*, *stripes*, and *blue*.

Table 1. Prediction accuracies of the 85 attributes of AwA by a single unit. Non-visual attributes such as animal's behaviors and natures are displayed in shaded boxes.

1st-22nd	s	23rd-44th	s	45th-66th	s	67th-85th	s
skimmer	0.145	*furry*	0.501	*tail*	0.604	*vegetation*	0.693
plankton	0.145	*hairless*	0.508	*scavenger*	0.626	*meat*	0.693
flippers	0.277	*big*	0.519	*plains*	0.631	*fields*	0.698
hands	0.286	*longneck*	0.524	*pads*	0.634	*nocturnal*	0.699
quadrapedal	0.306	*tree*	0.539	*bulbous*	0.640	*muscle*	0.700
strainteeth	0.331	*hooves*	0.542	*nestspot*	0.640	*patches*	0.703
ocean	0.352	*arctic*	0.542	*fish*	0.641	*brown*	0.703
stripes	0.363	*toughskin*	0.543	*active*	0.646	*inactive*	0.707
desert	0.364	*horns*	0.546	*claws*	0.649	*meatteeth*	0.717
swims	0.366	*paws*	0.546	*grazer*	0.649	*solitary*	0.718
water	0.366	*strong*	0.550	*jungle*	0.652	*hunter*	0.732
coastal	0.375	*small*	0.561	*forager*	0.654	*chewteeth*	0.741
ground	0.385	*fast*	0.563	*lean*	0.655	*spots*	0.742
blue	0.386	*bipedal*	0.568	*bush*	0.664	*timid*	0.748
red	0.401	*stalker*	0.572	*slow*	0.666	*smart*	0.752
walks	0.421	*insects*	0.572	*white*	0.675	*fierce*	0.754
tunnels	0.430	*newworld*	0.574	*group*	0.675	*smelly*	0.756
tusks	0.430	*forest*	0.576	*mountains*	0.677	*gray*	0.767
hops	0.451	*domestic*	0.577	*longleg*	0.678	*black*	0.774
orange	0.453	*hibernate*	0.594	*oldworld*	0.686		
yellow	0.454	*weak*	0.597	*buckteeth*	0.688		
flys	0.481	*cave*	0.599	*agility*	0.689		

For each attribute, the upper row shows the images randomly chosen from the top 0.5% of the entire images sorted in the order of response of the unit; the lower row shows the bottom 0.5% of the sorted images. (The 0.5 percentages correspond to a set of 150 images.) The unit is the same as the one in Fig. 3.

Several observations can be made for the results. For the attribute *hands*, the unit seems to be tuned to detect primates. This is reasonable, as this attribute is solely given to the primates in AwA. Although this might not be so interesting because the unit is unlikely to actually search for hands in images, it will be rather rare that the concept automatically acquired by the CNN through learning happens to be the same as a manually given semantic attribute. However, the attribute *stripes* seems to be such a case; the top images contain zebras, raccoons, tigers, skunks, which do share this visual attribute and do not seem to have any other visual property in common. Thus, this unit is highly likely to detect the presence of stripe texture in the images. For the attribute *blue*, the unit also seems to actually detect this attribute in the images; interestingly, however, the "correct" prediction of the color for the images of *killer whales* are counted as incorrect predictions, since the animals are not given this attribute in AwA.

Differences among layers In the above experiments we have considered only the units of the fc7 layer. To examine the differences among the layers, we computed s for the units of different layers. To be specific, for each of the fully-

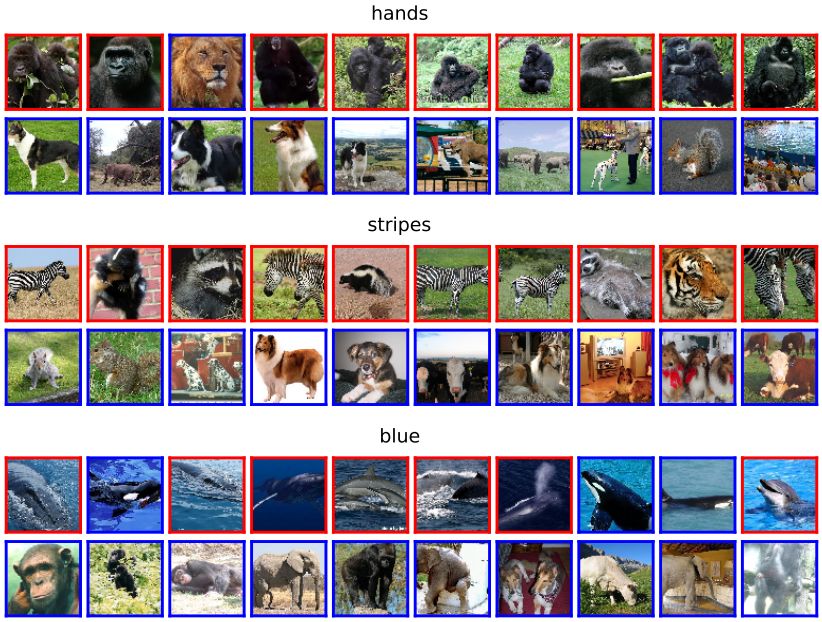

Fig. 4. Examples of the prediction of each attribute by a single unit. For each attribute, the upper row shows the images randomly chosen from the top 0.5% of the images that the most activates the unit; the lower row shows those randomly chosen from the bottom 0.5%. The images surrounded by red lines are *with* the attribute and those surrounded by blue lines are *without* the attributes. Best viewed in color.

connected layers (i.e., fc6 and fc7), we pick a single unit with the maximum s in the layer, as in the same way as above. As mentioned above, the value s is the overlapped area of the two normalized histograms of the responses of a single unit to images *with and without* each attribute. For the lower layers (i.e., norm1, norm2, conv3, conv4, pool5), we pick a map instead of a single unit. By a map, we mean the outputs of a single filter in the convolutional layer. (For pooling layers (pool5) and contrast normalization layers (norm1, norm2), we use their pooled and normalized signals.) To be specific, we calculate the maximum response of the units in each map and use it to create the histograms. Thus, for these layers, each attribute is related not to a single unit but to a filter. This is because the units in these layers are considered not only to represent the presence of a feature but also to convey the positional information of the feature; thus, a single unit is not likely to represent an attribute. Using a map instead of a single unit indeed contributes to raise the prediction accuracies of these lower layers.

Fig. 5 shows the prediction accuracies $(1-s)$ of the different layers for several selected attributes. We have found from the results for all the attributes that

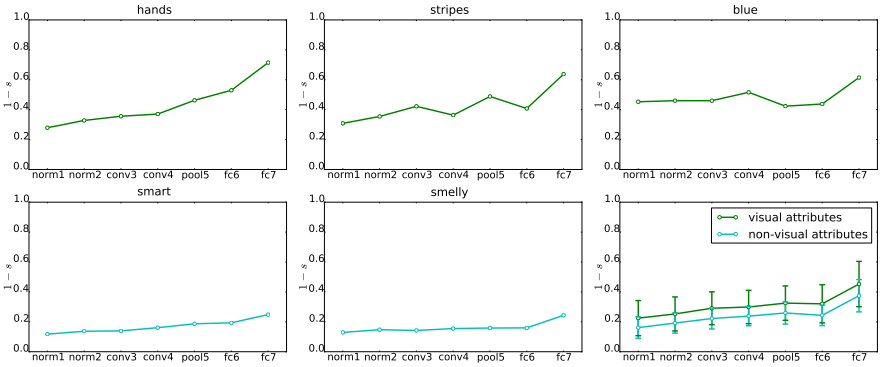

Fig. 5. Prediction accuracies of attributes by different layers. From top left to bottom right: *hands* (4), *stripes* (8), *blue* (14), *smart* (81), *smelly* (83), and the averaged accuracies for visual and non-visual attributes. The numbers in the parentheses indicate the rank in Table 1.

the accuracy curves can be categorized into several types. The first type is that accuracy increases sharply with the height of layers, as seen in *hands* of Fig. 5. The second is that accuracy is already high at lower layers and continues to be high at higher layers, as shown in *blue*. Their difference is not necessarily clear and thus there are attributes of intermediate type, as seen in *stripes* of Fig. 5. The last is the type that accuracy tends to be low throughout entire layers, as in *smart* and *smelly* of Fig. 5. It should also be noted that there is no attribute such that accuracy decreases with the height of layers.

These differences among layers and attributes may be explained by how difficult it is to represent the attributes. Some attributes, such as colors, are easy to judge their presence in images. They are associated with low level features, which can be correctly extracted even by the lower layers. Some attributes, such as those related to body shapes like *hands*, are more difficult to judge their presence in images (even if it could be translated into *primates* in the CNN as mentioned earlier). They may need complicated feature extraction, which could only be performed at higher layers. Non-visual attributes, such as *smart* and *smelly*, cannot be correctly estimated even at higher layers.

Prediction by linear SVM In the above, we have considered the possibility that a single unit represents a particular attribute. It is more natural to think that each attribute is represented by a combination of multiple unit activations, e.g., a linear combination in the simplest case. Thus, we trained a linear SVM to predict each attribute from the responses of the entire fc7 units. We used the forty categories for training and the remaining ten categories for test, as in the standard procedure of the zero-shot learning. Table 2 shows the results, i.e., the prediction accuracies for the 85 attributes, sorted in their order. Apart from the the top ten attributes are predicted with more than 90% accuracies, which is much better than the single unit results, the two results share the order of the

Table 2. Prediction accuracies of the 85 attributes of AwA by linear SVM using all the responses of the fc7 units. Non-visual attributes are in shaded boxes.

1st-22nd	s	23rd-44th	s	45th-66th	s	67th-85th	s
flys	0.999	*bipedal*	0.856	*fast*	0.737	*bush*	0.623
red	0.999	*swims*	0.855	*hibernate*	0.721	*grazer*	0.621
desert	0.999	*water*	0.855	*gray*	0.718	*domestic*	0.599
plankton	0.965	*hooves*	0.853	*pads*	0.713	*hunter*	0.598
hands	0.961	*strainteeth*	0.849	*nocturnal*	0.711	*smelly*	0.594
yellow	0.953	*blue*	0.847	*tail*	0.697	*black*	0.593
tunnels	0.947	*longleg*	0.844	*mountains*	0.697	*slow*	0.590
longneck	0.946	*insects*	0.837	*muscle*	0.691	*meat*	0.589
skimmer	0.941	*hairless*	0.833	*forest*	0.686	*timid*	0.582
tusks	0.926	*weak*	0.824	*small*	0.684	*group*	0.580
cave	0.926	*paws*	0.821	*smart*	0.675	*patches*	0.579
hops	0.918	*scavenger*	0.818	*chewteeth*	0.672	*meatteeth*	0.707
flippers	0.917	*coastal*	0.816	*tree*	0.672	*brown*	0.576
quadrapedal	0.914	*plains*	0.815	*white*	0.661	*lean*	0.573
ocean	0.896	*furry*	0.804	*forager*	0.660	*active*	0.572
horns	0.889	*toughskin*	0.789	*agility*	0.658	*fierce*	0.560
orange	0.886	*strong*	0.789	*jungle*	0.656	*nestspot*	0.544
stripes	0.881	*big*	0.781	*solitary*	0.653	*fish*	0.529
ground	0.881	*fields*	0.767	*inactive*	0.647	*spots*	0.527
oldworld	0.879	*newworld*	0.762	*buckteeth*	0.641		
walks	0.872	*stalker*	0.761	*vegetation*	0.634		
arctic	0.871	*claws*	0.740	*bulbous*	0.630		

attribute including the tendency that the visual attributes are easier to predict than non-visual ones.

3.3 Zero-shot learning

In the above experiments we consider the relation of layer activations to semantic attributes. These attributes are arbitrarily chosen by human. It could be possible that the CNN finds more general attributes than the 85 semantic attributes, some of which might not be even represented by words. To examine this possibility, we perform zero-shot learning by regarding the layer activation for an input image as its attributes. Based on the results in the last section, we choose the responses of the 4096 units in the fc7 layer.

Unlike the 85 semantic attributes, no relation is provided in advance between the discovered attributes and the unknown categories, and thus it is impossible to recognize the categories without additional information. To fulfill this missing link, Yu et al. [14] propose to use a similarity matrix between the 40 known categories and 10 unknown categories that are created by human subjects. (They used this matrix to perform zero-shot learning by their attributes, which are generated from the training data by their method.) Following their method, we borrow their similarity matrix that are publicly available at the authors'

webpage [1]. Computing the similarities between an input image and the known 40 categories using its responses, we evaluate the correlation between them and the similarity matrix to classify the input image.

The details of the method is as follows. Before testing, we compute the responses of the fc7 units to each image of the known forty categories, which yields forty point sets in a 4096-dimensional space. At the time of testing, we compute the responses $[r_1, \ldots, r_{4096}]$ to the input image, and then evaluate its similarity to the j-th known category ($j = 1, \ldots, 40$) by the following distance metric:

$$d_j = \sum_{i=1}^{4096} \|r_i - \mathrm{NN}_j(r_i)\|^2, \tag{1}$$

where $\mathrm{NN}_j(r_i)$ is the response of the i-th unit that is the nearest to r_i among all the samples belonging to the j-th category. More rigorously, we use its inverse as a similarity. Finally, we compare the resulting similarity vector against the 10×40 similarity matrix $\tilde{\mathbf{S}}$ of Yu et al. [14] to determine into which of the ten categories the input image is classified. To be specific, the comparison is performed by the normalized correlation between the input similarity vector and each row vector of $\tilde{\mathbf{S}}'$:

$$\hat{c} = \underset{c}{\mathrm{argmax}} \sum_j \frac{\tilde{S}_{cj} \cdot (1/d_j)}{(\sum_k \tilde{S}_{cj})(\sum_k (1/d_j))}. \tag{2}$$

The results are shown in Table 3. Our approach significantly outperforms [2] the accuracy reported in [10], where the 85 semantic attributes are used, and is even much better than the method of Yu et al., in which attributes are designed particularly for the purpose of zero-shot learning. Note that the accuracy of 48.30% reported in [14] is achieved by utilizing $\tilde{\mathbf{S}}$ to design the attributes, meaning that the discovered attributes could be ineffective for other unknown categories. Thus, it is more appropriate to compare the accuracy of 46.94% with the accuracy of 62.40% achieved by our method. It should also be noted that our CNN is trained only for the purpose of the category recognition, not for zero-shot learning, and nevertheless this high performance is attained. This fact shows the goodness of the internal representation of CNNs as attributes for zero-shot learning. It is particularly important that CNNs can *automatically* discover attributes having good properties, as compared with the manually designed attributes and the ones discovered by a dedicated method.

4 Summary

Toward a better understanding of convolutional neural networks (CNNs), we conjecture that the internal representation learned by CNNs should have simi-

[1] https://github.com/felixyu/category/tree/master/zero_shot_data

[2] It should be noted that these comparisons might not be fair, as these studies [10, 14] focused on how to use or how to generate attributes, given a set of image features. In other words, their method should work with the CNN activations used in our study, instead of the traditional hand-designed features.

Table 3. Results of zero-shot learning. The last row indicates the accuracy obtained when the output of the fc7 layer units are directly used for r_i's of (1). The accuracy one row above is obtained by using the inputs to the rectified linear function for r_i's.

Method	# Attributes	Accuracy
Lampert et al. [10]	85	40.5
Yu et al. [14]	200	46.94
Yu et al. (Adaptive) [14]	200	48.30
Our method	4096	**62.40**
Our method (after ReLU)	4096	59.14

larities to category-level attributes possessing good qualities. The experimental results support this conjecture. Despite the fact that the CNN is trained for a specific category recognition task, there automatically emerge units in the CNN that can predict some of the semantic attributes that are hand-designed. We also test zero-shot learning by treating the responses of units in a layer of the CNN as category-level attributes. The method shows much better performances than the state-of-the-art method that designs attributes particularly for the purpose of zero-shot learning based on traditional hand-designed image features.

Acknowledgement.

This work was supported by JSPS KAKENHI Grant Numbers 25135701, 25280054.

References

1. Bengio, Y., Courville, A. C., and Vincent, P.: Representation learning: a review and new perspectives, Computing Research Repository abs/1206.5538 (2012).
2. Cireşan, D., Meier, U., and Schmidhuber, J.: Multi-column deep neural networks for image classification, *CVPR* (2012).
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database, *CVPR* (2009).
4. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, *ICML* (2014).
5. Farhadi, A., Endres, I., Hoiem D., and Forsyth, D.: Describing objects by their attributes, *CVPR* (2009).
6. Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y.: Maxout networks, *ICML* (2013).
7. Hinton, G. E., and Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks, Science (2006).
8. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors, Computing Research Repository abs/1207.0580 (2012).
9. Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet classification with deep convolutional neural networks, *NIPS* (2012).

10. Lampert, C. H., Nichisch, H., and Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer, *CVPR* (2009).
11. Le, Q. V., Ranzato, M., Monga, R., Devin, M., Corrado, G. S., Dean, J., and Ng, A. Y.: Building high-level features using large scale unsupervised learning, *ICML* (2012).
12. Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *ICML* (2009).
13. Wan, L., Zeiler, M. D., Zhang, S., LeCun, Y., and Fergus, R.: Regularization of neural networks using dropconnect, *ICML* (2013).
14. Yu, F. X., Cao, L., Feris, R. S., Smith, J. R., and Chang, S.-F.: Designing category-level attributes for discriminative visual recognition, *CVPR* (2013).
15. Zeiler, M. D., and Fergus, R.: Visualizing and understanding convolutional networks, *ECCV* (2014).