

Learning Deep Representation with Large-scale Attributes

Wanli Ouyang, Hongyang Li, Xingyu Zeng, Xiaogang Wang

Department of Electronic Engineering, The Chinese University of Hong Kong

wlouyang, hyli, xyzeng, xgwang@ee.cuhk.edu.hk

Abstract

Learning strong feature representations from large scale supervision has achieved remarkable success in computer vision as the emergence of deep learning techniques. It is driven by big visual data with rich annotations. This paper contributes a large-scale object attribute database¹ that contains rich attribute annotations (over 300 attributes) for ~180k samples and 494 object classes. Based on the ImageNet object detection dataset, it annotates the rotation, viewpoint, object part location, part occlusion, part existence, common attributes, and class-specific attributes. Then we use this dataset to train deep representations and extensively evaluate how these attributes are useful on the general object detection task. In order to make better use of the attribute annotations, a deep learning scheme is proposed by modeling the relationship of attributes and hierarchically clustering them into semantically meaningful mixture types. Experimental results show that the attributes are helpful in learning better features and improving the object detection accuracy by 2.6% in mAP on the ILSVRC 2014 object detection dataset and 2.4% in mAP on PASCAL VOC 2007 object detection dataset. Such improvement is well generalized across datasets.

1. Introduction

Object representations are vital for object recognition and detection. There is remarkable evolution on representations for objects [20, 17, 43, 35, 15, 37, 34, 11], scenes [48], and humans [45]. Much of this progress was sparked by the creation of datasets [33, 6, 42, 29, 31, 45]. We construct a large-scale object attribute dataset. The motivation is two-folds.

First, it is an important step towards further semantic understanding of images. Since deep learning achieved close or even better performance than human-level on the ImageNet classification dataset [35, 15, 16, 37], semantic understanding of images is drawing much attention [39, 7]. Besides object class names, the attributes of objects pro-

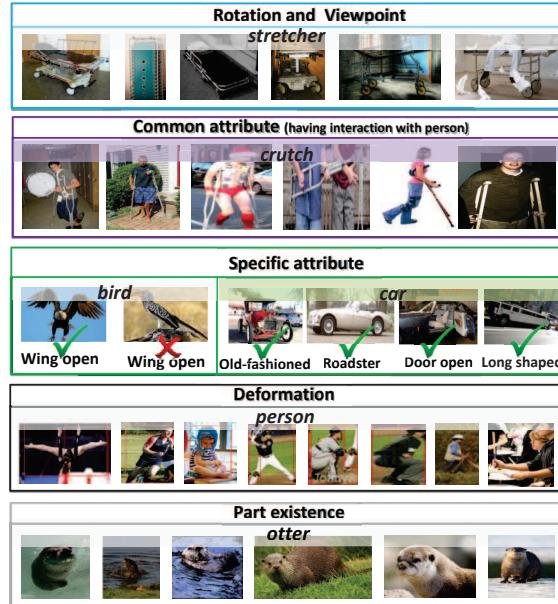


Figure 1. Objects of the same class are very different in appearance caused by the variation in rotation, viewpoint, occlusion, deformation, and other attributes. Some attributes (such as “having interaction with persons”) are common across all the classes. Some are class-specific (e.g. “wing open” is only valid for birds). Their attributes facilitate semantic understanding of images and help to learn better deep representations.

vide richer semantic meanings. For example, with the attributes, we can recognize that a car is “old-fashioned” and has its “door open”, an otter is “floating on the water” and “facing the camera”. As another example, with the location of object parts, we can estimate the action of an object. Although ImageNet has become one of the most important benchmark driving the advance of computer vision because of its large scale and richness on object classes, attribute annotations on it are much smaller in scale. The annotations from our dataset largely enrich the semantic description on ImageNet.

Second, this database provides labels that facilitate analysis on the appearance variation of images. It is well-known that the intra-class variation is one of the most important factors that influence the accuracy in object detection and recognition. As shown in Fig. 1, objects of the same class

¹The dataset is available on www.ee.cuhk.edu.hk/~xgwang/ImageNetAttribute.html

are very different in appearance due to the variation in rotation, viewpoint, part deformation, part existence, background complexity, interaction with other objects, and other factors. On the Pascal VOC dataset, researchers infer the viewpoint change and part existence by using aspect ratio [10]. However, images of the same aspect ratio can be very different in appearance because of the factors mentioned above. A direct way of revealing the factors that influence appearance variation is to explicitly annotate them. Therefore, we annotate the ImageNet object detection data, which has been most widely used in generic object detection nowadays, with these attributes.

Much evidence has shown that powerful generic feature representations can be learned on ImageNet with deep models and the image classification task. With our database, feature learning can be guided by the knowledge of attributes. Bengio *et al.* have pointed out the importance of identifying and disentangling the underlying explanatory factors hidden in images for representation learning [2]. A more effective way would be telling the model these factors during training for better disentangling them. For the examples in Fig. 1, the rotation and viewpoint information helps the deep model to learn different features for representing stretchers with different rotations; the common attribute “having interaction with persons” for crutches helps the deep model to capture this contextual visual pattern; the class specific attributes (e.g. “wing open” for birds and “door open” for cars) help the deep model to learn features for representing these specific visual patterns instead of treating them as noise; the part location and part existence help the deep model to handle deformation and appearance variation caused by part existence.

Attributes are correlated, e.g. rotation is related to part location, and should be modeled jointly. We cluster samples into attribute groups, which leads to different attribute mixture types. Some clusters are shown in Fig. 3. The deep model is trained to predict the attribute mixture types.

When there are lots of attributes describing various aspects of an object, it is difficult to identify which are the most important ones influencing appearance variation. It is desirable to have a scheme that automatically identifies the main factors in appearance variation. In this paper, a hierarchical cluster tree is constructed by selecting a single attribute factor for division at each time. From the top to the bottom of the hierarchical tree, it is easy to rank the importance of the attribute factors that cause variation. For the example in Fig. 3, the rank is viewpoint first, part existence second, and then rotation.

The contributions of this paper are three-folds:

1. The largest attribute dataset for generic objects. It spans 494 object classes and has 180k samples with rich annotations including rotation, viewpoint, object part location, part occlusion, part existence, 10 common attributes, and 314 class specific attributes. These images selected from the ILSVRC object detection dataset were widely used for

	num. classes	num. samples	part or sample	per class	class group
AwA [18]	50	30k	n	class	a
CORE [8]	28	3k	y	sample	a, v
a-Pascal [9]	10	12k	n	sample	p,a, v, t
a-Yahoo [9]	12	2.6k	n	sample	p,a, v, t
p-Pascal 07 [1]	6	<6k	y	sample	a
a-ImageNet[32]	384	9.6k	n	sample	p,a, v, t
Ours	494	~180k	y	sample	p,a, v, t

Table 1: Comparison of object attribute datasets. Datasets are different in the number of categories, the number of samples, whether part locations are annotated, annotation is per class or per sample. The last column (class group) indicates some datasets only annotated animal classes (a), vehicle classes (v), persons (p), or also include other things (t) like sofa.

fine-tuning deep models in detection literature [11].

2. We show that attributes are useful in discriminating intra-class variation and improving feature learning. The deep representation learned with attributes as supervision improves object detection accuracy on large-scale object detection datasets. Different ways of using attributes are investigated through extensive experiments. We find that it is more effective to learn feature representations by predicting attribute mixture types than predicting attributes directly. There are also other ways to make better use of this attribute dataset in feature learning to be explored in the future.

3. The factor guided hierarchical clustering that constructs semantically meaningful attribute mixture types. The attributes are grouped into several attribute factors. At each step, the attribute factor that best represents the appearance variation is selected for dividing the samples into clusters. With this clustering approach, the importance of attributes in representing variation can be ranked.

2. Related work

Many attribute datasets have been constructed in recent years. The Sun attribute database is for scene recognition [29]. Other datasets describe the attributes of objects from different aspects. A comparison is shown in Table 1. Lampert *et al.* [18] annotated color, texture, shape, body part type, and semantic attributes (like fast and weak) for 50 animal classes. *The attributes were labeled per class instead of per image.* Therefore, these annotations are insufficient in disentangling the factors that cause intra-class appearance variation, as shown by examples in Fig. 1. There are also many datasets that provide attributes per sample [8, 9, 32]. The datasets CORE [8], a-Pascal and a-Yahoo [9] are small in the number of categories and the number of object samples. The ImageNet attribute dataset constructed by Olga and Fei-Fei [32] is an important step towards large-scale attribute labels. However, the number of labeled samples in [32] is still very small. In existing datasets, only the small datasets in [1, 8] have labels on object parts. In comparison,

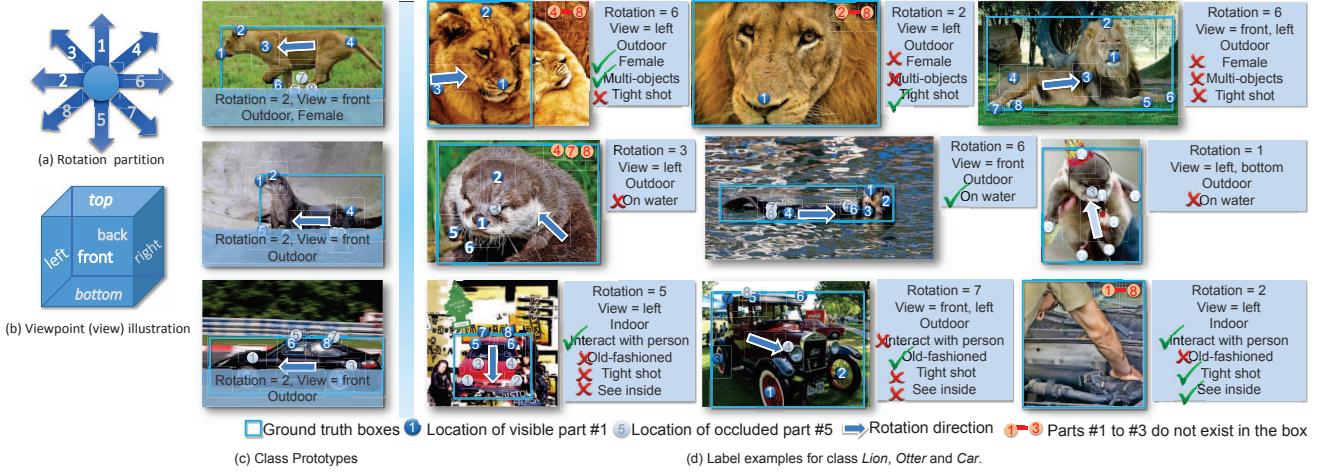


Figure 2. Attribute annotation samples for lion, otter, and car. *Best viewed in color.* Rotation is quantized into 8 directions (a). Viewpoint is a 6 dimensional vector (b), where front means main flat side. The prototypes for orientation and viewpoint are defined (c). And then each bounding box is annotated (d). Outdoor/indoor, interaction with person, tight shot, and see inside are common attributes for all classes. Female for lion, floating on water for otter, and old fashioned for car are class-specific attributes for single or small groups of classes.

our dataset contains 494 object classes with $\sim 180k$ samples labeled by attributes. The number of samples in our dataset is an order of magnitude larger than existing datasets that were annotated per sample. As summarized in Table 1, our dataset is attractive in the large number of object classes and annotated samples, object class variety, and annotation on object part location, occlusion and existence.

Many approaches used predictions on attributes as middle-level features for recognizing new object categories with few or no examples [9, 18, 29, 22]. People aimed at improving the accuracy of attribute prediction [45, 3]. Attributes are also found to be effective for object detection. Farhadi *et al.* [8] used the functionality, superordinate categories, viewpoint, and pose of segments as attributes to improve detection accuracy. Azizpour and Laptev used part location to improve detection [1]. Simultaneous attribute prediction and image classification is done in [41]. However, it is not clear if attributes are helpful in learning generic feature representations with deep models and not clear if attributes are helpful for object detection on very large-scale datasets like ImageNet. Our work shows that attributes are, if used in a proper way, helpful in learning feature representations that improve large-scale object detection.

Deep learning is effective for many vision tasks [17, 34, 43, 14, 19, 11, 26, 28, 23, 25, 44, 47, 21, 49, 5]. It is found that the features learned from large-scale classification data can be applied to many other vision tasks. However, the use of attributes in improving feature learning for object detection is not investigated in literature.

3. The ImageNet Detection Attribute dataset

A few annotations are shown in Fig. 2. The labeled attributes are summarized into the following groups:

1. Rotation. It corresponds to in-plane rotation of an object, as shown in Fig. 2 (a). Rotation is discretized into 8

directions.

2. Viewpoint. It corresponds to out-of-plane rotation of an object, as shown in Fig. 2 (b). Viewpoint can be multi-valued. For example, one can see both front and left side of a car. For both in-plane and out-of-plane rotation, the reference object orientation is chosen such that in most cases the objects undergo no rotation, in frontal view, and have most of their parts not self-occluded. The appearance mixtures obtained in [10] for bicycles and cars correspond to viewpoint change. The viewpoint has semantic meaning on whether a person or animal is facing the camera.

3. Common attributes. These attributes are shared across all the object classes. They includes 10 binary attributes that may result in appearance variation. 1) Indoor or outdoor, which is scene-level contextual attribute. 2) Complex or simple background, which is a background attribute. 3) Tight shot, in which the camera is very close to the object and leads to perspective view change. In this case, usually most object parts do not exist. 4) Internal shot, which is true for images captured in a car and false for images captured out of a car. 5) Almost all parts occluded, in which more than 70% of an object is hidden in the bounding box. 6) Interaction with person, which is an important context for objects like crutch, stretcher, horse, harmonica, and bow. 7) Rotten, corrupted, broken, which is a semantic attribute that results in appearance variation. 8) Flexible shape, which is true for objects like starfish. 9) Multi-objects, which is true when a bounding box include multiple objects, e.g. when a lion hugs its baby. 10) Cut or bitten, which is true when an apple or a lemon is cut into slices. Fig. 2 shows some common attributes like outdoor/indoor, interaction with person.

4. Class-specific attributes. It refers to attributes specifically used for a single class or a small group of classes. We choose attributes that result in large appearance variation. For example, binary attributes “long ear” and “fluffy” for

dog, “mouth open” for hippopotamus, “switched on with content on screen” for monitor, “wings open” for dragon fly and bird, “with lots of books” for bookshelf, and “floating on the water” for whale. Fig. 2 shows some class specific attributes. There are 314 class-specific attributes defined in total. Common attributes and class-specific attributes provide rich semantic information for describing objects.

5. Object part location and occlusion. Different object classes have different parts. For example, for lions and otters as shown in Fig. 2, the parts are mouth, neck, hip, and four legs. For cars as shown in Fig. 2, the parts are the four wheels and the four corners of the car roof. Variation in part location corresponds to deformation of object parts. It is found in [1] on 6 animal classes that part location supervision is helpful. The part location not only is useful in disentangling the factors that influence appearance variation, but also facilitates further applications like action recognition, animation, content based video and image retrieval. Object parts may be occluded, which results in distortion of the visual cues of an object. Therefore, the occlusions of object parts are annotated and represented by gray circles in Fig. 2.

6. Object part existence. For a given object class, its parts may not be in the bounding box because of occlusion or tight-shot. For the example in Fig. 2, a lion image with only head is labeled as lion and a lion image with the full body is also labeled as lion. However, these two images have large appearance variation. The appearance mixtures like half body and full body for persons in [10] correspond to different object part existence.

The ILSVRC 2013 dataset [31] for object detection is composed of training data (train), validation data (val) and testing (test) data. The val data is split into val₁ and val₂ using the approach in [11]. This dataset evaluates performance on 200 object classes that have their descent object class labels being 494 classes. Our attribute dataset is based on the bounding boxes of the 494 classes provided in this dataset. These bounding boxes do not have attribute annotations. In our dataset, we annotate their attributes. The attributes for all the 494 object classes are annotated. For the samples in train, each object class is constrained to have at most 1000 samples labeled. The indices in [11] for selecting positive training samples are used in our attribute dataset in selecting samples to be labeled. All samples in val₁ are labeled. In total, the 154,886 samples in training data and all the 26,728 samples in val₁ have their attributed annotated. In summary, all the positive samples that were used for finetuning the deep model from ILSVRC 2013 object detection in [11] have been labeled with attributes.

It took \sim 3000 hours/labor for annotating the attributes. Then the other \sim 2000 hours/labor were spent for refining the labels in order to improve the quality of the annotations.

4. Learning deep model with attributes

The following procedure is used to train deep models:

1. Pretrain the deep model for the 1000-class classifica-

tion problem. The 1000-class ImageNet classification and localization dataset is used for pretraining the model because it is found to be effective for object detection.

2. Finetune the deep model for both attribute estimation and 200-class object detection using the following loss:

$$L = L_o + \sum_{j=1}^J b_j L_{a,j}, \quad (1)$$

$$L_o = \sum_{c=1}^{200} \left\{ \frac{1}{2} \|\mathbf{w}_{o,c}\|^2 - b_o \sum_{n=1}^N [\max(0, 1 - \mathbf{w}_{o,c}^\top \mathbf{h}_n)] \right\},$$

where L_o is hinge loss for classifying objects as one of the 200 classes or background. $\mathbf{w}_{o,c}$ is the classifier for object class c , \mathbf{h}_n is the feature from the deep model for the n th sample. $\sum b_j L_{a,j}$ is the loss for attribute estimation, b_j is the pre-defined weight for the loss $\sum b_j L_{a,j}$. When label $y_{j,n}$ is continuous, e.g. for part location, the square loss $L_{a,j} = \sum_n (y_{j,n} - \tilde{y}_{j,n})^2$ is used in (1), where $\tilde{y}_{j,n}$ is the prediction for the j th attribute and n th sample. When label $y_{j,n}$ is discrete, e.g. for part existence or attribute mixture type, the cross-entropy loss $L_{a,j} = -\sum_n y_{j,n} \log(\tilde{y}_{j,n})$ is used. Different preparations of the labels $y_{j,n}$ for obtaining $\sum L_{a,j}$ are detailed in Section 6.3.1. When $b_j = 0$ for $j = 1, \dots, J$, the deep model degenerates to the normal object detection framework without attributes. When attributes are used, we set $b_j = 1$. With the loss function in (1), the deep model not only needs to distinguish the 200 object classes from background for the loss L_o but also needs to predict the labels from attributes for the loss $\sum_j b_j L_{a,j}$. Samples without attribute labels are constrained to not having loss $L_{a,j}$ so that they will not influence the attribute learning.

5. Factor guided hierarchical clustering

We divide the training samples of an object class into many attribute mixture types using the attributes introduced in Section 3. Then the deep model is used for predicting the attribute mixture type of training samples using the cross-entropy loss. The attributes of samples are grouped into 6 semantic factors $\mathbf{f} = \{\mathbf{f}_i\}_{i=1\dots 6} = \{\mathbf{f}_{rot}, \mathbf{f}_{view}, \mathbf{f}_{com}, \mathbf{f}_{spec}, \mathbf{f}_{loc}, \mathbf{f}_{ext}\}$. They correspond to the six factors introduced in Section 3. For example, \mathbf{f}_{rot} denotes rotation and \mathbf{f}_{view} denotes viewpoint.

For the samples of an object class, a hierarchical clustering tree is built. The algorithm is summarized in Algorithm 1. The clustering is done in a divisive way. There is only one cluster that contains all samples initially. Splits are performed recursively as one moves down on the hierarchical tree. At each stage, a cluster C is chosen to be split, and then one of the 6 semantic attribute factors is chosen for splitting the C into several clusters. Then the other cluster is selected for further splitting until no cluster satisfies the requirement on depth and sample size in a cluster. The clustering result obtained for the class bus is shown in Fig. 3.

Since different object classes are different in their distributions of attributes, clustering is done separately for each

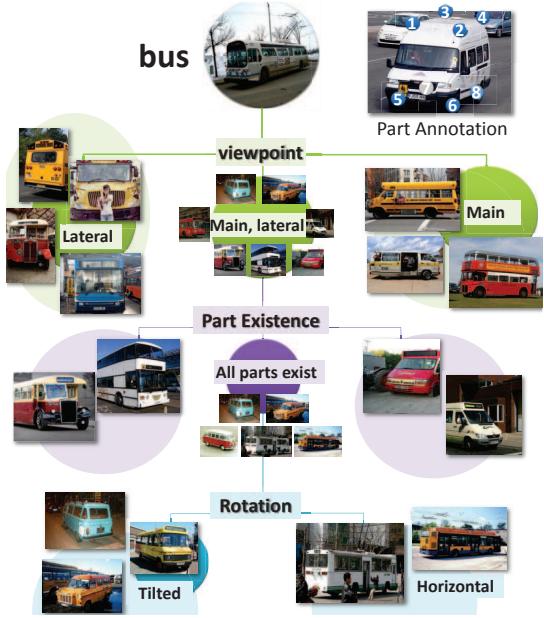


Figure 3. Factor guided hierarchical clustering for the object class bus. For splitting samples into clusters, viewpoint is used first, then part existence is used, and then rotation is used.

class so that different classes can choose different semantic attribute factors.

5.1. Clustering for an attribute factor

For an attribute factor, the selected sample set C is split into several clusters such that samples in the same cluster are more similar to each other than to those in other clusters (lines 5-7 in Algorithm 1). The clustering approach in [46] is used for splitting C into N clusters. It constructs directed graph using K -nearest neighbor (K -NN).

Algorithm 1: Factor Guided Hierarchical Clustering

Input: $\Psi = \{\mathbf{f}\}$, training samples with attribute factors for an object class.
 D , maximum Tree depth.
 M , minimum size for splitting set.

Output: V , the clustering result.

- 1 Initially, the cluster set $V = \{C\}$. Denote the depth of the cluster C by $d(C)$ and the size of cluster C by $|C|$.
- 2 **while** 1
- 3 Find a C , s.t. $d(C) < D$ and $|C| \geq M$.
- 4 If C cannot be found, terminate the ‘while’ loop and output V .
- 5 **for** $i = 1$ to 6
- 6 Split C into N clusters $S^i = \{C_{1,i}, \dots, C_{N,i}\}$ using the factor \mathbf{f}_i .
- 7 **end for**
- 8 $S = \text{argmax}_{S^i \in \{S^1, \dots, S^6\}} E(S^i)$.
- 9 $V \leftarrow \{V \setminus C\} \cup S$.
- 10 **end while**

graph, each sample is a node, the directed edge from the n th node to the m th node is used for measuring the similarity between the m th sample and the n th sample as follows:

$$s_i^{n,m} = \begin{cases} \exp(-\frac{\|\mathbf{f}_n^i - \mathbf{f}_m^i\|^2}{\sigma^2}), & \text{if } \mathbf{f}_n^i \text{ is in the } K\text{-NN of } \mathbf{f}_m^i, \\ 0, & \text{otherwise,} \end{cases}$$

where σ^2 is the mean Euclidean distance of all $\|\mathbf{f}_n^i - \mathbf{f}_m^i\|^2$, \mathbf{f}_n^i and \mathbf{f}_m^i are the i th attribute factors for the n th and m th sample respectively. The closeness measure of clusters is defined via the indegree and outdegree on the graph. This approach is adopted because it is found [46] to be better than affinity propagation, spectral clustering, and normalized cut on many benchmark image datasets.

5.2. Choice of attribute factor for splitting

Each attribute factor \mathbf{f}_i is used for obtaining a candidate split $S^i = \{C_{1,i}, \dots, C_{N,i}\}$. The candidate split with the maximum evaluation score $E(S^i)$ among the six candidate splits $\{S^1, \dots, S^6\}$ is selected for splitting C (lines 8-9 in algorithm 1). In our implementation, $E(S^i)$ is the entropy of the split as follows:

$$E(S^i) = E(\{C_{1,i}, \dots, C_{N,i}\}) = -\sum_k p_k \log(p_k),$$

$$\text{where } p_k = |C_{k,i}| / \sum_{\bar{k}} |C_{\bar{k},i}|, \quad (2)$$

$|C_{k,i}|$ denotes the number of elements in $C_{k,i}$.

5.3. Discussion

$E(S^i)$ measures the quality of the candidate split. The reason of dividing samples into clusters is to group samples that are similar in appearance. The candidate splits are obtained for small within-cluster dissimilarity. However, uniformness of the clusters is important but not considered. For example, the ImageNet classification dataset has almost the same number of samples (1300 samples for 90% classes) in each class for training. As another example, the training samples are constrained to be not larger than 1000 for training the deep model on ImageNet detection dataset [11]. The entropy is used in our algorithm for measuring the uniformness of cluster size. The larger the entropy, the more uniform the cluster size, and thus the better the captured variation in attributes. For example, suppose candidate group S^1 have the C split into clusters having samples with percentage 30%, 35% and 35%, candidate group S^2 have the C split into samples with percentage 90%, 9% and 1%. Candidate group S^2 is considered as worse than S^1 . S^2 has 90% samples within a cluster and does not capture the main factor in variation. As another problem, cluster with 2% samples in S^2 has too few samples to be learned well while the cluster with 90% samples will dominate the feature learning. Therefore, the S^1 is a better choice and will be chosen by our approach in this case. By using the approach in [46] for clustering samples with similar factors and then selecting the candidate split that has the best uniformness, both

similarity within a cluster and the ability in identifying the main factor in variation are considered in our clustering algorithm.

There are some classes that do not have variation in certain attribute factors. For example, balls like basketball do not have in-plane or out-of-plane rotation. When a cluster is split using these attribute factors, the returned cluster number will be one and has the minimum entropy. Therefore, these attribute factors will not be selected for clustering.

The cluster C for splitting is constrained to have more than M samples and tree depth less than D . In our experiment, $D = 4$, $M = 300$, $N = 3$ and 1372 sub-classes are obtained. D , M , and N are used for controlling the number of samples within a cluster. If the number of samples within a cluster is too small, it is hard to be well trained.

6. Experimental results

We use the ImageNet 2014 training data and val_1 data as the training data and the val_2 data for evaluating the performance if not specified. The split of val_1 and val_2 is the same as that in [11] because it was downloaded from the authors' web. The additional images of ILSVRC 2014 training data are used in training the detectors, just without the attribute annotations. The attribute annotations are not required at the testing stage because they are only used for supervising feature learning. We only evaluate the performance on object detection instead of attribute prediction because the aim of this work is to study how rich attribute annotation can help feature learning in detection.

The GoogleNet in [37] is chosen as the deep model structure to learn features because it is found to be the best in the ILSVRC 2014 object detection challenge. In all experiments, we have used the same model as that in [37] without any modification like spatial pyramid pooling [14], deformation pooling [27], or batch normalization [16].

6.1. Overall experimental comparison

Since our work is to show the effectiveness of attributes in feature learning but not for showing the best result from model averaging, only single-model results across state-of-the-art methods are reported and compared. Table 2 summarizes the result for RCNN [11] and the results from ILSVRC2014 object detection challenge. It includes the best results on the test data submitted to ILSVRC2014 from GoogLeNet [37], DeepID-Net [24], DeepInsight, UvA-Euvision, and Berkeley Vision, which ranked top among all the teams participating in the challenge. Our single-model performance is already higher than existing approaches that use model averaging. Compared with single model results, it outperforms the winner of ILSVRC2014, GoogleNet, by 10.2% on mAP and the best single model, DeepInsight, by 8% on the ImageNet test dataset.

6.2. Overview of basic components

Table 3 shows the components that arrives at the final result with mAP 48.5% on val_2 . If the attributes are not used and image-level annotation from ImageNet classification dataset is used for training, this corresponds to the approach proposed in [11] and was adopted by many existing approaches [14, 37]. By pretraining with the 1000-class ImageNet classification data and using selective search [36] for proposing candidate regions, the mAP is 37.8% and denoted by res.(a) in Table 3.

If the attributes are not used and bounding box annotation from the 1000-class ImageNet classification dataset is used for pretraining, it corresponds to the approach in [24]. Using the region proposed by selective search, pretraining with bounding box labels, the mAP is 40.4% and denote by res.(b) in Table 3. Based on the setting in res.(b), training the model with attributes using the approach introduced in Section 4 improves the mAP from 40.4% to 43%, which is res.(c) in Table 3. With scale jittering, bounding box regression and context, the final single-model result is 48.5%.

6.3. Ablation study on attributes

In this section, the model pretrained with bounding boxes, without scale jittering, using selective search for candidate region proposal, without context or bounding box regression is used as the basic setting. If attributes are not used, this corresponds to res.(b) in Table 3 with mAP 40.4%. This is used as the baseline in this section. Other results investigated are only different from the res.(b) in Table 3 in using of different loss functions for attributes.

6.3.1. Investigation on approaches in using attributes

We can skip the clustering of attributes and directly the attributes, e.g. part location and part existence, as labels $y_{j,n}$ for the loss function $L_{a,j}$ in (1). The part location labels are continuous, so the square loss is used. The other attributes, e.g. part existences, are discrete, so the cross-entropy loss is used. If the attribute labels are prepared in this way for the loss function in (1), the object detection has mAP 40.1%, no better than the baseline with mAP 40.4%. The partial reasons could be attributes are too complex for the deep model to learn meaningful features and the relationship among attributes is not considered.

Attributes can be used for dividing a object class into several attribute-mixture-types (sub-classes). The deep model is then required to predict the sub-class labels, for which multi-class cross entropy loss is used as the loss $L_{a,j}$ in (1). For example, if 200 object classes are divided into 1200 sub-classes, 1201-class cross entropy loss is used (+1 for the background).

In order to investigate the influence of the single attribute factors introduced in Section 5, we conduct experiment on using only single attribute factor as the feature for clustering samples and obtaining attribute mixture types. In this case,

approach	Flair [38]	RCNN[11]	Berkeley Vision	UvA-Euvision	DeepInsight	DeepID-Net [24]	GoogLeNet[37]	ours
ImageNet val ₂ (sgl)	n/a	31.0	33.4	n/a	40.1	40.1	38.8	48.5
ImageNet test (avg)	22.6	n/a	n/a	n/a	40.5	40.7	43.9	n/a
ImageNet test (sgl)	n/a	31.4	34.5	35.4	40.2	37.7	38.0	48.2

Table 2: Detection mAP (%) on ILSVRC2014 for top ranked approaches with single model (sgl) and model averaging (avg).

$J = 1$ in (1). The “s-cluster i ” for $i = 1, \dots, 6$ in Fig. 4 corresponds to the use of only a single attribute factor \mathbf{f}_i defined in section 5. \mathbf{f}_1 is rotation, \mathbf{f}_2 is viewpoint, \mathbf{f}_3 is common attributes, \mathbf{f}_4 is class specific attributes, \mathbf{f}_5 is part existence, and \mathbf{f}_6 is part location and visibility. According to Fig 4, attribute mixture obtained by rotation is the worst in learning features for object detection, with mAP 40.3%, i.e. s-cluster 1 in Fig. 4. Part location is the best, with mAP 40.7%, i.e. s-cluster 6 in Fig. 4.

If all attribute factors are directly concatenated and clustered using [46] for obtaining attribute mixture types, the mAP is 40.9% and denoted by d-cluster in Fig. 4. The use of all attribute factors performs better than the use of single attribute factor for obtaining attribute mixture type. If the hierarchical clustering approach in Section 5 is used for obtaining attribute mixture types, the mAP is 42.0% (denoted by h-cluster in Fig. 4). The h-cluster considers uniformness in cluster size and performs better than direct concatenation of attribute factors for clustering without considering uniformness, i.e. the d-cluster in Fig. 4.

We also investigate other clustering methods without requiring attribute annotations. If each object class is randomly partitioned into 6 sub-classes, the mAP is 40.5%. Appearance features can also be used for clustering [4]. If the 1024 appearance features of last hidden layer in the baseline GoogleNet with mAP 40.4% are used as the features for k-means clustering, the mAP is 40.6%.

6.3.2 Investigation on using multiple mixture sets

When the j th attribute factor \mathbf{f}_j is used for obtaining the attribute mixture, we have the j th mixture type set ψ_j . For the six factors, we have six sets ψ_j for $j = 1, \dots, 6$. They can be used together for jointly supervising the deep model. In this experiment, the loss for set ψ_j , $j = 1, \dots, 6$ is used as the $L_{a,j}$ in (1) and denoted by L_j in Fig. 5. The loss from the cluster set obtained using the approach in Section 5 is used as the $L_{a,7}$ in (1) and denoted by L7. Fig. 5 shows the experimental results on using multiple mixture sets. The use of all mixture sets from L1 to L7 has the best result, with mAP 43%. If we do not use $L_{a,7}$ and only use the losses from single factors, i.e. $L_{a,j}$ for $j = 1, \dots, 6$, mAP is 41.1%. Thus $L_{a,7}$ obtained by our clustering approach is important in the combination.

Fig. 6 visualizes the feature maps obtained by the model learned with and without attributes. For visualization, we consider the response maps from the last hidden layer before average pooling. We directly average 50 response maps that has the largest positive classification weights. It can be

denotation	res.(a)	res.(b)	res.(c)
pretrain label	image	bbox	bbox
scale jittering	n	n	n
Attribute	n	n	y
mAP (%)	37.8	40.4	43

Table 3: Detection mAP on ILSVRC2014 val₂ for different usage of basic components.

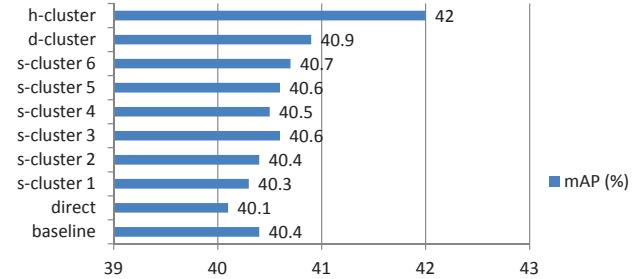


Figure 4. Investigation on different approaches in using attributes on ILSVRC2014 val₂.

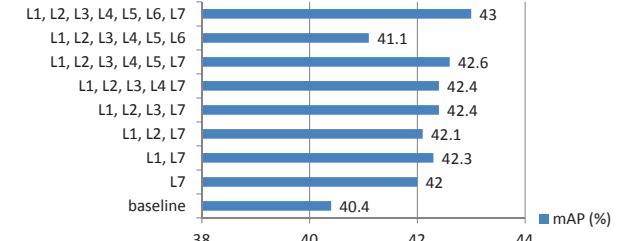


Figure 5. Investigation on using multiple attribute mixture sets on ILSVRC2014 val₂.

seen that the features learned with attribute is better in distinguishing background from objects. Fig. 7 shows some examples which have high prediction scores on attribute mixture types in four object classes. In order to evaluate attribute prediction performance, we randomly select 200 images for each object category from the training positive dataset for testing and the rest of the training data for training. The accuracy for rotation estimation is 58.5%. For part location estimation, the mean square error is 0.105 (part locations are normalized to [0 1]). The mean average precision (precision vs recall) are, respectively, 50.1%, 67.3%, 86.3%, 84.3%, 83.8% for part occlusion, part existence, viewpoint, common attributes and class-specific attributes.

After attribute mixture types have been obtained for multiple mixture sets, the predicted attribute mixture types can be used as features for object detection. If we *only* use these attribute mixture types as features and learn linear SVM for object detection, mAP is 40.4%. Note that the attribute mixture types are not supervised by negative background samples. Although no negative sample has been used for esti-

	HOG-DPM [12]	HSC-DPM [30]	Regionlet [40]	Flair [38]	DP-DPM [13]	SPP [14]	RCNN (VGG) [11]	ours-b	ours-a	ours-a-j
	33.7	34.3	41.7	33.3	45.2	59.2	66.0	64.4	66.8	68.5

Table 4: Detection mAP (%) on the PASCAL VOC-2007 test set. *ours-b* denotes our baseline which uses features learned without attributes with settings the same as that of the res.(b) in Table 3, *ours-a* denotes learning features from attributes while other settings are the same as *ours-b*, *our-a-j* denotes the results of adding scale jittering to pretraining on top of *ours-a*.

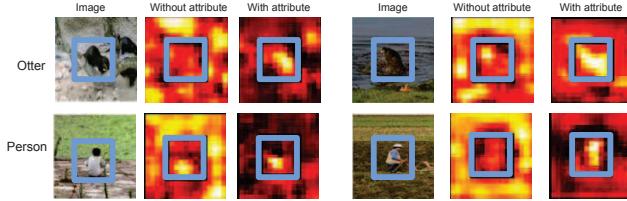


Figure 6. Visualizing feature maps that are most correlated to the object classes otter and person. The feature maps learned with attributes are better in handling the appearance variation and distinguishing objects from background. *Best viewed in color.*

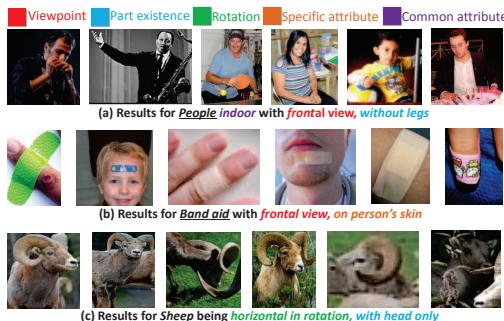


Figure 7. Examples with high prediction scores on attribute mixture types. The images are cropped so the attributes can be seen better. *Best viewed in color.*

mating attribute mixture types, they are still useful in distinguishing foreground from background and attaining the same accuracy as the baseline.

6.4. Results on PASCAL VOC 2007

In order to prove that the features learned from ImageNet with our annotated attributes can also be well generalized to datasets, we conduct evaluation on PASCAL VOC 2007. We follow the approach in [11] for splitting the training and testing data. Table 4 shows the experimental results on VOC-2007 testing data. Many existing works are included for comparison [12, 30, 40, 38, 10, 11, 14, 13]. We only report our single-model result as other approaches. We directly use the models finetuned on the ImageNet detection data for extracting features and learning SVM without any finetuning on Pascal VOC for all of our results. If the setting of res.(b) in Table 3 is used to learn features, mAP is 64.4%. These features are not learned from attributes and it can be used as our baseline. If attributes are added to learn features from ILSVRC, mAP is 66.8%. All the other settings are the same as res.(b) in Table 3. It shows that the features learned by the attributes provided by our dataset are stronger in object detection. It is effective not only on ImageNet test data but also on PASCAL VOC 2007. Such improvement can be generalized across datasets. If the scale jittering is used

for pretraining and attributes are used for learning features from ILSVRC, mAP is 68.5%. It is better than state-of-the-art on PASCAL VOC 2007, such as RCNN that uses VGG for finetuning on Pascal in [11].

7. Conclusion

In this paper, we present a large-scale attribute dataset based on the ImageNet object detection dataset. It spans 494 object classes and has large number of samples labeled with rich annotations. These attributes are useful towards further semantic understanding of images. Based on this dataset, we provide a deep learning scheme that uses the attributes to learn stronger feature representations and improve object detection accuracy. Such improvement can be generalized across datasets. Our empirical study also finds that, in order to learn better feature representations, when training the deep model, it is better to group attributes into clusters, forming attribute mixture types, for prediction instead of separately predicting them. The ImageNet detection attribute dataset will be released to the public.

Acknowledgment: This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK417011, and CUHK14207814).

References

- [1] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 2, 3, 4
- [2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. PAMI*, 35(8):1798–1828, 2013. 2
- [3] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *CVPR*, 2014. 3
- [4] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *ECCV Workshops and Demonstrations*, 2012. 7
- [5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 3
- [6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1
- [7] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. *arXiv preprint arXiv:1411.4952*, 2014. 1
- [8] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2, 3

- [9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785. IEEE, 2009. 2, 3
- [10] P. Felzenszwalb, R. B. Grishick, D. McAllister, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. PAMI*, 32:1627–1645, 2010. 2, 3, 4, 8
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3, 4, 5, 6, 7, 8
- [12] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. <http://www.cs.berkeley.edu/rbg/latent-v5/>. 8
- [13] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. *arXiv preprint arXiv:1409.5403*, 2014. 8
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 3, 6, 8
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. 1
- [16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1, 6
- [17] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3
- [18] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2, 3
- [19] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 3
- [20] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1
- [21] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In *CVPR*, 2012. 3
- [22] D. Mahajan, S. Sellamanickam, and V. Nair. A joint learning framework for attribute models and object descriptions. In *ICCV*, 2011. 3
- [23] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *CVPR*, 2014. 3
- [24] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, et al. Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. *arXiv preprint arXiv:1409.3505*, 2014. 6, 7
- [25] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *CVPR*, 2012. 3
- [26] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *ICCV*, 2013. 3
- [27] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *CVPR*, 2015. 6
- [28] W. Ouyang, X. Zeng, and X. Wang. Modeling mutual visibility relationship in pedestrian detection. In *CVPR*, 2013. 3
- [29] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 1, 2, 3
- [30] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013. 8
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1, 4
- [32] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *Trends and Topics in Computer Vision*, pages 1–14. 2012. 2
- [33] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008. 1
- [34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1, 3
- [35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1
- [36] A. Smeulders, T. Gevers, N. Sebe, and C. Snoek. Segmentation as selective search for object recognition. In *ICCV*, 2011. 6
- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1, 6, 7
- [38] K. E. A. van de Sande, C. G. M. Snoek, and A. W. M. Smeulders. Fisher and vlad with flair. In *CVPR*, 2014. 7, 8
- [39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014. 1
- [40] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 8
- [41] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010. 3
- [42] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1
- [43] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013. 1, 3
- [44] X. Zeng, W. Ouyang, and X. Wang. Multi-stage contextual deep learning for pedestrian detection. In *ICCV*, 2013. 3
- [45] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. *CVPR*, 2014. 1, 3
- [46] W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In *ECCV*, 2012. 5, 7
- [47] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015. 3
- [48] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 1
- [49] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity preserving face space. In *ICCV*, 2013. 3