

# Unsupervised learning of object semantic parts from internal states of CNNs by population encoding

Jianyu Wang<sup>1</sup>, Zhishuai Zhang<sup>2</sup>, Cihang Xie<sup>2</sup>, Vittal Premachandran<sup>2</sup>, Alan Yuille<sup>1,2</sup>

<sup>1</sup>University of California Los Angeles, <sup>2</sup>Johns Hopkins University

wjyouch@ucla.edu, {zhishuai.zhang, cihangxie306}@gmail.com, {vittalp, alan.yuille}@jhu.edu

## Abstract

We address the key question of how object part representations can be found from the internal states of CNNs that are trained for high-level tasks, such as object classification. This work provides a new unsupervised method to learn semantic parts and gives new understanding of the internal representations of CNNs. Our technique is based on the hypothesis that semantic parts are represented by populations of neurons rather than by single filters. We propose a clustering technique to extract part representations, which we call Visual Concepts. We show that visual concepts are semantically coherent in that they represent semantic parts, and visually coherent in that corresponding image patches appear very similar. Also, visual concepts provide full spatial coverage of the parts of an object, rather than a few sparse parts as is typically found in keypoint annotations. Furthermore, We treat each visual concept as part detector and evaluate it for keypoint detection using the PASCAL3D+ dataset, and for part detection using our newly annotated ImageNetPart dataset. The experiments demonstrate that visual concepts can be used to detect parts. We also show that some visual concepts respond to several semantic parts, provided these parts are visually similar. Note that our ImageNetPart dataset gives rich part annotations which cover the whole object, making it useful for other part-related applications.

## 1. Introduction

Deep Convolutional Neural Networks (CNNs) are very successful at performing visual tasks such as image classification [9, 16], object detection [7], and semantic segmentation [11, 3]. Despite these successes we lack understanding of why CNNs work so well. For example, when CNNs are trained to detect objects it seems likely that the object semantic parts are represented by the internal activity of the filters/neurons. But what is the form of these internal

representation? Are semantic parts represented by individual neurons [23] or perhaps by populations of activity [6]? Understanding the nature of these representations not only makes it easier to understand deep networks, and hence may suggest ways to improve them, but also serves as an unsupervised method for learning object semantic parts. Note, we use the term “semantic parts” to mean object parts, like wheels and windows of cars, which are defined in terms of the three-dimensional object.

We address this problem by studying the internal activity of CNNs when looking at six classes of objects (car, aeroplane, bike, motorbike, bus, and train). Our hypothesis is that object semantic parts are represented by populations of filter/neural activity which we can find by clustering algorithms. We call each cluster a “visual concept”. Although these visual concepts are defined in terms of the feature activity of the CNN they also correspond to a set of image patches (those patches whose feature activity lie within the cluster). We show visually that visual concepts (i.e. the image patches associated with them) correspond well to parts of objects (e.g., the wheels and windows of cars). These clusters are also visually tight, in the sense that image patches corresponding to the same visual concept tend to look very similar. Moreover, the visual concepts give a fairly dense description of each object, in the sense that for most parts of the object we can find a visual concept that looks visually similar.

We proceed to understand and quantify the correspondence between the visual concepts and the semantic parts of the objects. The correspondence is non-trivial because the scale of the visual concepts and semantic parts may differ, hence several visual concepts may correspond to different, or overlapping, regions of the same semantic part. Conversely, some semantic parts may look visually similar (e.g., the bike wheel center and the bike chain ring). In general, we expect the correspondence between visual concepts and semantic parts to be many-to-many.

To study this correspondence we convert each visual concept into a part detector and evaluate them in terms of

part detection and localization. We first compute the distance between the center of the visual concept and the feature response/activity to an input image patch. If the distance is below a threshold we treat this as a detection of that visual concept. We then evaluate the visual concepts for detecting keypoints in the PASCAL3D+ dataset [18]. This gives the promising result that for each keypoint there is typically a visual concept that detects the keypoint fairly well, as measured by average precision (AP), and compared with detection using single filters or a supervised baseline method. But this study is limited because the keypoints are very sparse and only cover a small portion of each object, so many visual concepts do not correspond to them.

To overcome this difficulty we create a new dataset, ImageNetPart, by annotating the six object classes (car, aeroplane, bike, motorbike, bus, and train) in PASCAL3D+ [18] with dense semantic parts and also determining the background regions. We use the ImageNet images provided in PASCAL3D+ dataset. This is a total of roughly 10,000 images and requires roughly 6 times as many labels as the keypoints. As before, we treat the single visual concepts as part detectors and show that for most semantic parts there is a visual concept that detects it fairly well (as measured by AP). But this only tells us what a subset of visual concepts are doing, because the number of visual concepts is much larger than the number of semantic parts (by a factor of 4 or more). By further analysis we show that the remaining visual concepts fall into three classes: (i) those that detect several semantic parts (two to four) which are visually similar, (ii) those that detect background regions of images (e.g., sky for aeroplanes, railway tracks for trains), and (iii) a few which have no obvious correspondence (perhaps due to limitations of the clustering method and CNN feature). We obtain better APs in evaluation when taking into account the fact that visual concepts may respond to a small subset of semantic parts.

We argue that our work gives some understanding of CNNs and, in particular, of how they encode internal representations of objects by visual concepts, encoded by populations of filter/neuron activity. From another perspective, our approach can be thought of as an unsupervised way to learn part detectors of objects.

## 2. Related Work

Discovering mid-level patches [17, ?] in an unsupervised manner has been studied before CNNs. More recently, [10] used CNN features with association rule algorithm to discover mid-level visual elements, and evaluated them on object and scene classification tasks. But their method used CNN only as a feature extractor without exploring how the CNN filters of intermediate layers capture mid-level visual concepts. [19] and [14] used activations from intermediate layers of the CNN to find object parts, which they then used

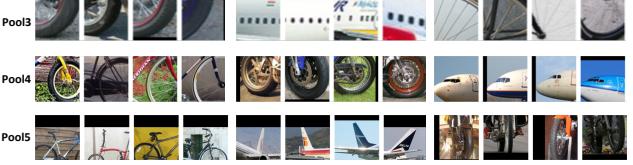


Figure 1: This figure shows example visual concepts on different object categories, for layers pool3, pool4 and pool5. Each row visualizes three visual concepts with four example patches, which are randomly selected from a pool of well-matched patches. These visual concepts are visually and semantically tight. We can easily identify the semantic meaning and parent object class.

for fine-grained object classification. But they assumed that a single filter from an intermediate layer can detect parts. Our work has some similarity to [23], which looked at Scene-CNNs (trained for scene classification), and claimed that object detectors emerge from Scene-CNNs. Their “object detectors” are single filters/neurons which they evaluated using AMT. By comparison, we extract visual concepts at multiple levels of CNN in an unsupervised manner, and evaluate them as part detector automatically.

There is some neuroscience evidence that the brain represents concepts by population activity [6], but there have been few studies of this in CNNs. One work [1] argued for a population code by training linear SVMs on features from Alexnet and showed that several filters were needed to get good performance on object classification. But, unlike us, their approach was fully supervised. They also addressed objects instead of object parts. There has been other supervised methods for object part detection [4, 12] which show good performance.

Several researchers have visualized deep networks. [5, 15] used backpropagation-based techniques to generate images by maximizing certain object class score, enabling them to visualize the notion of an object class as captured by CNN. And [13] showed that incorporating strong natural image priors to the optimization procedure produces images that look more realistic. [21] proposed a visualization strategy that mapped intermediate features activations back to the image space using Deconvolution Networks [22].

## 3. Learning Visual Concepts by Clustering

This section describes our method for finding visual concepts in object-CNNs. We start with a CNN pre-trained for ImageNet object classification. We apply this CNN to a set of images from a specific object class, such as cars. Then we cluster the features at each layer of CNN (ignoring the spatial locations of the features) using K-means++ [2], yielding a dictionary of visual concepts at each layer. We will understand the CNNs internal representations better by restricting the input to objects from the same class, rather than objects from many classes.

More precisely, for an image  $\mathbf{x} \in \mathbb{R}^{W \times H \times 3}$  with spa-

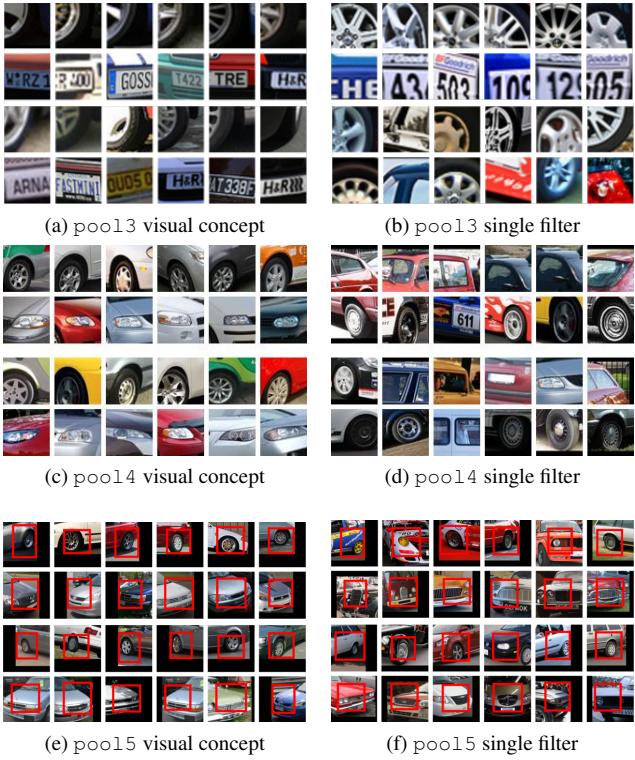


Figure 2: Visual comparison between visual concepts and single filters for layer  $\text{pool}_3$ ,  $\text{pool}_4$  and  $\text{pool}_5$ . In subfigure (a), we visualize two randomly selected visual concepts, where the top two rows correspond to the best 6 patches, and the bottom two rows correspond to 6 randomly selected patches from the top 500 well-matched patches. In subfigure (b), we visualize two randomly selected single filters in the same manner.

tial resolution  $W \times H$ , an intermediate layer,  $\ell$ , produces a response,  $\mathbf{f}_\ell \in \mathbb{R}^{W_\ell \times H_\ell \times N_\ell}$ , where  $W_\ell \times H_\ell$  is the spatial resolution of the intermediate layer  $\ell$ 's response and  $N_\ell$  is the number of filters on that layer. We randomly sample a number of feature population responses,  $\mathbf{p}_\ell^{i,j} \in \mathbb{R}^{N_\ell}$ , over the spatial grid  $(i, j)$  of the layer's response  $\mathbf{f}_\ell$ . Each  $\mathbf{p}_\ell^{i,j}$  corresponds to a patch from the original image with the theoretical receptive field, which can be computed for any spatial grid  $(i, j)$  and any intermediate layer  $\ell$ . We extract the population responses,  $\mathbf{p}_\ell^{i,j}$ , from all images in the training dataset. Next we cluster the population responses,  $\mathbf{p}_\ell^{i,j}$ , using the K-means++ algorithm. At each level of the CNN, this gives a dictionary of visual concepts. The intuition is that this clustering will identify image patterns that frequently occur in images of cars and hence which may correspond to parts such as car wheels or license plates. An important issue is to determine the number of visual concepts  $K$ . We adopt a two-step strategy where we first set  $K$  in kmeans++ to be the number of filters in the layer, and then we reduce the dictionary size by a greedy cluster merg-

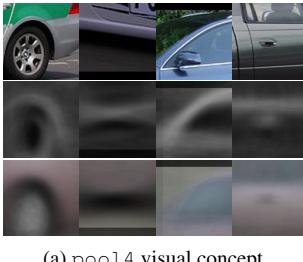
ing algorithm. We analyze the effect of  $K$  in the diagnostic experiment.

### 3.1. Visualization of Visual Concepts

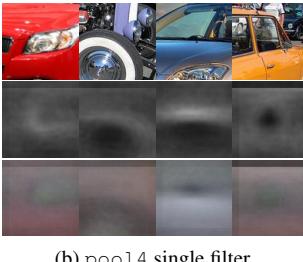
We can visualize the visual concepts by observing the image patches which are assigned to each cluster by K-means++. Our first finding is that the image patches for each visual concept appear to roughly correspond to semantic parts of the object classes with larger parts at higher pooling levels, see figure 1. Most of the visual concepts are tight in the sense that the image patches corresponding to them look visually similar. But demonstrating this in a conference paper is difficult for two reasons (i) it is impractical to show all the image patches for each visual concept, and (ii) we also cannot show all the visual concepts. To address the second difficulty we show all the visual concepts of cars in the supplementary material with comparisons to the individual filters.

We address the first difficulty in two ways. Firstly, in figure 2 we show a small number of the best image patches for each visual concept (i.e. the best 6 whose feature vectors are closest to the visual concept center) and a random sample of the other patches (e.g., the random 6 from best 500). For comparison, we show similar results for individual filters (chosen from the small fraction of filters that appear to correspond to semantic parts). We observe that best patches and the random samples for visual concepts are often very visually similar, but this is less true for the individual filters. Secondly, for each visual concept we compute (a) the mean of the edge maps of the best 500 image patches using HED [20] and (b) the average of the (color) intensity of the best 500 patches. Examples of these means are given in figure 3. These both show that most visual concepts are visually tight. Furthermore figure 4 is a schematic showing that visual concepts give a spatially complete description of object parts and cover most of the object. The supplementary material extends figures 2 to all the visual concepts at  $\text{pool}_4$  and all the filters at that level. Roughly speaking, around seventy percent of the visual concepts are tight but this is only true for roughly thirty percent of the filters.

Finally we mention an artifact that arises when we visualize the visual concepts and has implications when evaluating them as part detectors. We observed that the visual concepts appeared less tight at the highest levels of the hierarchy (e.g.  $\text{pool}_5$ ). We conjecture that one reason for this is that the “effective receptive field of the neurons is not the same as their theoretical receptive fields. Intuitively, the effective receptive field captures the region of pixels which are most responsible for the response of visual concept or single filter. We obtain the effective receptive field by applying the deconvolutional operation in (Zeiler and Fergus 2014), and then threshing the deconv responses. We find out that, for visual concepts at  $\text{pool}_5$ , the image patches



(a) pool4 visual concept



(b) pool4 single filter

Figure 3: This figure shows four visual concept examples (left four) and four single filter examples (right four). The top row contains example image patches, and the bottom two rows contain average edge maps and intensity maps respectively obtained using top 500 patches. Observe that the means of the visual concepts are sharper than the means of the single filters for both average edge maps and intensity maps, showing visual concepts capture patterns more tightly than the single filters.

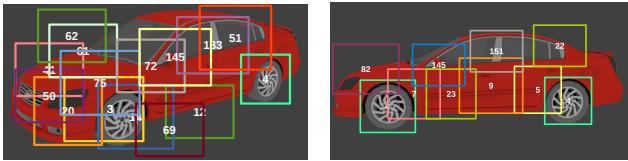


Figure 4: The figure shows example cars rendered by 3D model. We overlay the image with bounding boxes that are denoted by the visual concept number at the center, which would be the best match at that location. We see that the learned visual concepts for cars can cover most of the car region.

look more similar within the effective receptive field (red bounding box) than looking at the entire patch (which is very large).

#### 4. Visual Concepts as Part Detectors

Our visualization studies suggest that the visual concepts roughly correspond to semantic parts of objects and we now quantify this by evaluating them as part detectors.

We treat a visual concept as a part detector as follows. We compute the distance in the CNN feature space between the center of visual concept and the filter response to an image patch containing, or not containing, the part. If this distance is below a threshold we treat this as a detection for that visual concept. We also use non-maximal suppression (NMS) to suppress duplicate detections. If the detected patch center is sufficiently close (e.g., within 56 pixels) to the (keypoint or semantic part) groundtruth position, we treat it as a true positive. In this way, for each visual concept and semantic part we compute the number of true positives and false positives as a function of the threshold. By varying this threshold we obtain precision recall curves, and calculate the average precision (AP).

To evaluate keypoint detection in PASCAL3D+, or se-

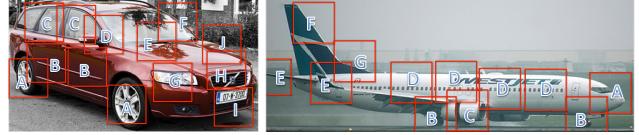


Figure 5: The figure illustrates several semantic part annotations for cars (left) and aeroplanes (right) from a particular viewpoint. We have defined in total 35 semantic parts for cars and 17 for aeroplanes. Car – A: wheels, B: side body, C: side window, D: side mirror and side window and wind shield, E: wind shield and engine hood, F: oblique line from top left to bottom right, wind shield, G: headlight, H: engine hood and air grille, I: front bumper and ground, J: oblique line from top left to bottom right, engine hood. Aeroplane – A: nose pointing to the right, B: undercarriage, C: jet engine, mainly tube-shape, sometimes with highly turned ellipse/circle, D: body with small windows or text, E: wing or elevator tip pointing to the left, F: vertical stabilizer, with contour from top left to bottom right, G: vertical stabilizer base and body, with contour from top left to bottom right.

mantic part detection in ImageNetPart, we select the visual concept which has highest AP. This “best visual concept” evaluation strategy is problematic if the visual concept is good at detecting several keypoints or semantic parts. Later on we modify our evaluation method to deal with this. By visualization we find that visual concepts from pool4 layer corresponds best to the keypoint annotations in PASCAL3D+ and the semantic part annotations in ImageNetPart, so we report them in this paper. The results for pool5 are fairly similar, while the results for pool3 are worse presumably because their visual concepts are too small.

#### 5. ImageNetPart Dataset and Semantic Parts

The problem of PASCAL3D+ dataset for evaluating visual concepts is that there are limited number of keypoint annotations, much fewer than the number of visual concepts, and they only cover a small amount of the object. This makes it hard to understand what all the visual concepts are doing. This motivated us to construct a new dataset, called ImageNetPart, which provides richer semantic part annotations using ImageNet images from PASCAL3D+. We focus on six vehicle categories: car, aeroplane, bicycle, motorbike, bus, and train. These annotations cover most of the objects and hence are more suitable for evaluating the visual concepts and understanding their correspondence to semantic parts.

Our annotation strategy is as follows. Firstly, we define semantic parts based on the object structure using terms like “side mirror and side window and side body of car from left side”. These semantic parts have some viewpoint information since the appearance of an object part may vary with viewpoint. Each defined semantic part represents one visual pattern. We manually construct the semantic part dictionary for each object class. Secondly, we ask labelers to locate

each semantic part of specific category by putting a bounding box in the correct region of the image. To help the labelers understand the semantic parts, we provide them with two types of information during labeling: 1) one-sentence verbal description; 2) typical labeled bounding boxes on several prototype images with various viewpoints. Figure 5 visualizes annotations for several semantic parts, and we give the verbal descriptions for all the semantic parts of six object classes in the supplementary material.

## 6. Experimental Results

**Dataset:** We use the PASCAL3D+ dataset [18]) and our newly annotated ImageNetPart dataset for quantitative evaluation. For PASCAL3D+ we experimented with six vehicle categories: car, aeroplane, bicycle, motorbike, bus, and train. ImageNetPart dataset provides much richer part annotations using ImageNet images in PASCAL3D+. For each category, we use the training subset for learning the visual concepts and the validation subset for testing.

The keypoint annotations from PASCAL3D+ are side-dependent, e.g., the left-front tires and the right-front tires have different labels. However the local image patches are not big enough to make such fine-grained distinctions possible. Therefore we merge some keypoint annotations. The merging is done in the following cases. a) We merge the left/right fine-scale keypoints (e.g. left/right headlight) into a single class (e.g. headlight). b) We merge visually similar keypoints into a single class (e.g. the four wheels of a car). And, c) we discard keypoints that rarely occur in the images.

**Baselines:** We provide two baselines for comparison on PASCAL3D+. i) We evaluate keypoint detection performance for single filters. We use the filter responses, instead of distance to center of visual concept, as the detection scores when computing AP values. We report results for the best performing filters for each keypoint. ii) To provide an upper-bound on the detection capacity of unsupervised part-learning techniques, we report results obtained by using strong supervision. We extract the population responses for patches centered at the keypoint position (and randomly sampled patches for a separate background class) and train a multi-class linear SVM to discriminate the patches of different keypoints. Note that this baseline is very similar to the keypoint prediction method in [12] but with a different performance measure.

**Implementation Details:** In the experiments, we use the Caffe toolbox [8] and VGG-16 network [16] pre-trained on ImageNet object classification as our object-CNN. We build separate dictionaries for different object classes. For each image, we crop the object instance using the bounding box annotation and resize the short side to 224 pixels. We normalize each population response (feature vector) using  $\ell_2$ -normalization for both learning visual concepts and detec-

	Car							Motorbike					
	1	2	3	4	5	6	7	mAP	1	2	3	4	mAP
SF	.86	.44	.30	.38	.19	.33	.13	.38	.26	.60	.30	.22	.35
VC	.92	.51	.27	.41	.36	.46	.18	.45	.35	.75	.43	.25	.45
SS	.97	.65	.37	.76	.45	.57	.30	.58	.37	.77	.50	.60	.56

	Bus						Bicycle						
	1	2	3	4	5	6	mAP	1	2	3	4	5	mAP
SF	.45	.42	.23	.38	.80	.22	.42	.23	.67	.25	.43	.43	.40
VC	.41	.59	.26	.29	.86	.51	.49	.32	.78	.30	.55	.57	.50
SS	.74	.70	.52	.63	.90	.61	.68	.37	.80	.34	.71	.64	.57

	Train					Aeroplane						
	1	2	3	4	5	mAP	1	2	3	4	5	mAP
SF	.39	.33	.24	.16	.15	.25	.41	.25	.22	.13	.31	.26
VC	.41	.30	.30	.28	.24	.30	.21	.47	.31	.16	.34	.30
SS	.71	.49	.50	.36	.39	.49	.72	.60	.50	.32	.49	.53

Table 1: AP values for keypoint detection on PASCAL3D+ dataset for six object categories using the “best visual concept”. “SF” refers to single filter; “VC” refers to visual concept; “SS” refers to strong supervision. Visual concept achieves much higher AP than single filter method. The keypoint number-name mapping is provided below. Cars – 1: wheel 2: wind shield 3: rear window 4: headlight 5: rear light 6: front 7: side; Bicycle – 1: head center 2: wheel 3: handle 4: pedal 5: seat; Motorbike – 1: head center 2: wheel 3: handle 4: seat; Bus – 1: front upper corner 2: front lower corner 3: rear upper corner 4: rear lower corner 5: wheel 6: front center; Train – 1: front upper corner 2: front lower corner 3: front center 4: upper side 5: lower side; Aeroplane – 1: nose 2: upper rudder 3: lower rudder 4: tail 5: elevator and wing tip.

tion. We do not use  $\ell_2$ -normalization for evaluation of single filter method. Detailed comparison is in the diagnostic section.

### 6.1. Results on PASCAL3D+

We concentrate on the results using the visual concepts learnt from `poo14` with cluster merging described. We have 211, 237, 201, 176, 257 and 246 visual concepts for aeroplane, bicycle, bus, car, motorbike and train respectively. We show results from other layers in the diagnostic section. Note that here we report the results only for the images in PASCAL3D+ which lie in ImageNet subset (we obtained similar results for the much smaller subset of images which lie in PASCAL).

Table 1 reports the average precision (AP) values for keypoint detection in PASCAL3D+. For each keypoint from a particular object class, we show the AP achieved by the best visual concept and the best single filter (“best” meaning the one with highest AP for that keypoint). “SF” refers to the baseline where we test single filters as keypoint detectors, and “SS” refers to the baseline where we use strong supervision to train linear SVMs as keypoint detectors. The general trend shown in table 1 is that visual

Car	1	2	3	4	5	6	7	8	9	10	11	12		
SF	.86	.87	.84	.68	.70	.83	.83	.82	.72	.18	.22	.49		
VC	.94	.97	.94	.93	.94	.95	.94	.92	.94	.42	.48	.58		
Car	13	14	15	16	17	18	19	20	21	22	23	24		
SF	.33	.25	.22	.18	.37	.18	.28	.13	.16	.14	.30	.20		
VC	.45	.48	.46	.54	.60	.34	.38	.19	.26	.37	.42	.32		
Car	25	26	27	28	29	30	31	32	33	34	35	mAP		
SF	.16	.17	.15	.27	.19	.25	.18	.19	.16	.07	.07	.36		
VC	.26	.40	.29	.40	.18	.41	.53	.31	.57	.36	.26	.53		
Aeroplane	1	2	3	4	5	6	7	8	9					
SF	.32	.42	.08	.61	.58	.42	.49	.31	.23					
VC	.26	.21	.07	.84	.61	.44	.63	.42	.34					
Aeroplane	10	11	12	13	14	15	16	17		mAP				
SF	.09	.07	.20	.12	.29	.29	.07	.09		.28				
VC	.15	.11	.44	.23	.59	.65	.08	.15		.37				
Bike	1	2	3	4	5	6	7	8	9	10	11	12	13	mAP
SF	.77	.84	.89	.91	.94	.92	.94	.91	.91	.56	.53	.15	.40	.75
VC	.91	.95	.98	.96	.96	.96	.97	.96	.97	.73	.69	.19	.50	.83
Motor	1	2	3	4	5	6	7	8	9	10	11	12	13	mAP
SF	.69	.46	.76	.67	.66	.57	.54	.70	.68	.25	.17	.22	.53	
VC	.89	.64	.89	.77	.82	.63	.73	.75	.88	.39	.33	.29	.67	
Bus	1	2	3	4	5	6	7	8	9		mAP			
SF	.90	.40	.49	.46	.31	.28	.36	.38	.31		.43			
VC	.93	.64	.69	.59	.42	.48	.39	.32	.27		.53			
Train	1	2	3	4	5	6	7	8						
SF	.58	.07	.20	.15	.21	.15	.27	.43						
VC	.66	.50	.32	.28	.24	.15	.33	.72						
Train	9	10	11	12	13	14			mAP					
SF	.17	.27	.16	.25	.17	.10			.23					
VC	.36	.41	.27	.45	.27	.47			.39					

Table 2: AP values for keypoint detection on ImageNetPart dataset for cars (top), aeroplanes (middle) and bicycle (bottom). We see that visual concept by population encoding achieves much higher AP than single filter method. The semantic part names are provided in the supplementary material.

concepts are better as part detector than single filters. Unsurprisingly, the strongly supervised method is better than two unsupervised part-learning methods. Moreover, we see that the visual concepts obtained in our unsupervised way achieve reasonably good AP values for all categories.

## 6.2. Results on ImageNetPart

The richer annotations on ImageNetPart allows us to do more detailed experiments which make it much clearer how the visual concepts correspond to the object semantic parts. We start by evaluating the “best visual concepts” for each semantic part following the strategy discussed above for PASCAL3D+. The results are fairly similar showing that

	1	2	3	4	5	6	7	mAP
VC-512	.94	.51	.32	.53	.36	.48	.22	.48
VC-256	.94	.47	.29	.54	.31	.45	.20	.45
VC-128	.94	.46	.27	.51	.17	.47	.19	.43
VC-64	.93	.47	.27	.29	.17	.46	.18	.40
Merge	.92	.51	.27	.41	.36	.46	.18	.45
SF	.86	.44	.30	.38	.19	.33	.13	.38

Table 3: The effect of  $K$  on keypoint detection performance. “VC- $K$ ” refers to setting  $K$  as the number of clusters in kmeans++. All the VC-based methods are without merging.

	1	2	3	4	5	6	7	mAP
VC-NL2	.92	.33	.17	.34	.12	.39	.16	.35
VC-L2	.94	.51	.32	.53	.36	.48	.22	.48
SF-NL2	.86	.44	.30	.38	.19	.33	.13	.38
SF-L2	.80	.42	.24	.42	.19	.31	.18	.37

Table 4: The effect of  $\ell_2$ -normalization for cluster and single filter. “NL2” refers to not using  $\ell_2$ -normalization, and “L2” refers to using  $\ell_2$ -normalization. We can see that  $\ell_2$ -normalization is crucial to population encoding.

Layer	1	2	3	4	5	6	7	mAP
pool3	.80	.28	.13	.22	.26	.22	.16	.29
pool4	.92	.51	.27	.41	.36	.46	.18	.45
pool5	.83	.61	.27	.55	.29	.45	.29	.47

Table 5: AP values obtained using visual concepts learnt from different layers for cars. Different layers of the CNN are capturing different scales of object parts. pool5 is better on “big” parts while pool4 is better on “small” parts.

Method	1	2	3	4	5	6	7	mAP
SF	.96	.65	.61	.59	.59	.38	.27	.58
VC	.99	.75	.61	.73	.70	.54	.39	.67
SS	.99	.87	.82	.88	.84	.62	.41	.78

Table 6: AP values of viewpoint control for cars. The results are higher than the car results in Table 1 of the paper after selecting the best viewpoint for each keypoint.

visual concepts can be very effective at detecting most semantic parts and generally perform better than single filters. From table 2 we observe the same trends as for the keypoint results, the “best visual concept” results are best on cars, bicycles, motorbikes, and buses.

## 6.3. Diagnostic Experiments

In this section we present results from some diagnostic experiments to enable a better understanding of the visual concepts obtained from population encoding. Currently, our experiments are performed on cars using PASCAL3D+ dataset. Experiments on other object categories or other dataset is left as future work.

**Effect of the number of clusters  $K$ :** An important issue is to determine a good value for the number of clusters

$K$ . This is partly influenced by the amount of data available because we risk over-fitting if we try to make  $K$  too large. But it is also affected by the variability in appearance and geometry of the object. The greater the variability of the object, presumably, the larger the number of visual concepts. But it is unclear how to measure the variability of the object in advance. Determining the number  $K$  of clusters is important in order to prevent over-fitting while capturing the variability of the feature responses. We resort to a two-pronged strategy: (i) we choose a range of values for  $K$  (64, 128, 256, 512) to see how the results change with  $K$ , and (ii) we use a second stage to reduce the number of clusters by a merging algorithm.

Our results, Table 3, show that although performance increases with  $K$ , the improvement is relatively slow from  $K = 64$  to  $K = 512$ . Note that even at  $K = 64$  the performance by visual concept is still better than single filter (“SF”) method. Also the merging algorithm (“Merge”) achieves the same AP as VC-256, but with much fewer visual concepts (176).

**Effect of  $\ell_2$ -normalization:** We make a comparison between using  $\ell_2$ -normalization as a preprocessing step and without  $\ell_2$ -normalization. From Table 4, we can see that  $\ell_2$ -normalization is critical for visual concepts obtained from population encoding method. We then hypothesize that it is the population response direction that captures visual information, instead of the population response magnitude. For the single filter case, we have also tried similar  $\ell_2$ -normalization across filters for each spatial position, and used the normalized response as detection score. But we find that  $\ell_2$ -normalization in single filter case makes little difference (slightly worse) in terms of the AP values.

**Effect of Intermediate Layers:** In Table 5, we present the results obtained using the visual concepts learnt from different layers (pool3 - pool15). Looking at the numbers from pool3, it is clear that its concept dictionaries are not able to capture semantic level of the part annotations. This is because the receptive field of pool3 is too small to capture the entire part. Surprisingly, pool15 gives similar results as pool4 layer. But on careful examination, we see that pool5 does better than pool4 when the keypoints correspond to “big” parts (e.g. windshield) and worse than pool4 when the keypoints correspond to smaller parts (e.g. wheels), the size of which is more appropriate for pool4. This shows that different layers of the CNN are capturing different scales of object parts.

**Viewpoint Control:** While visualizing the clusters, we find that many visual concepts are viewpoint specific. Therefore, we test the performance of visual concepts by controlling for viewpoint. We divide the test set into five viewpoint bins: front, front-side, side, rear-side, and rear. Then for each viewpoint we run the same evaluation procedure as described above. Table 6 shows the best viewpoint

AP result for each keypoint. The numbers clearly shows a significant jump in detection capabilities of both single filters and our visual concepts compared to the car results in Table 1 in the main paper. Note that population encoding still outperforms single-filter detectors. These results suggest the lack of viewpoint invariance in the internal representation of the CNNs.

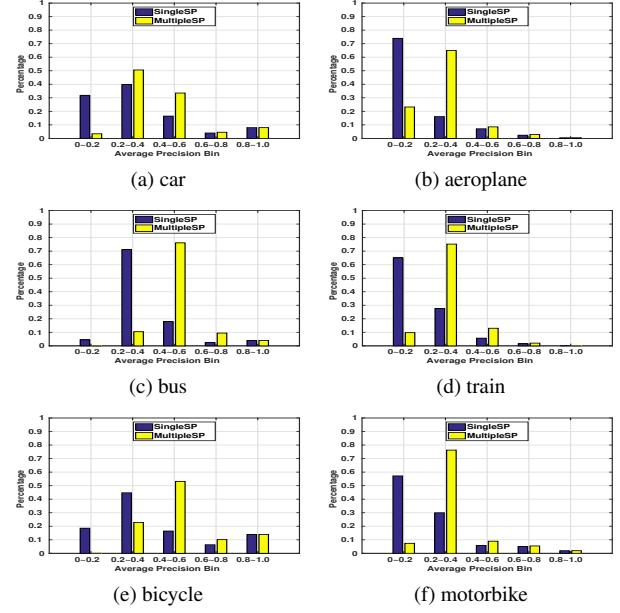


Figure 6: The histograms of the AP responses for all visual concepts. For “SingleSP”, we evaluate each visual concept by reporting its AP for its best semantic part (the one it best detects). Some visual concepts have very low APs when evaluated in this manner. For “MultipleSP”, we allow each visual concept to detect a small subset of semantic parts (two, three, or four) and report the AP for the best subset (note, the visual concept is penalized if it does not detect all of them). The APs rise considerably using the MultipleSP evaluation, suggesting that some of the visual concepts detect a subset of semantic parts. The remaining visual concepts with very low APs correspond to the background.

#### 6.4. The correspondence between Visual Concepts and Semantic Parts

The “best visual concept” evaluation strategy is problematic because it assumes that each visual concept detects a single semantic part only. Moreover, the number of visual concepts is much larger than the number of semantic parts so it does not help understand what the majority of the visual concepts are doing. Hence for each visual concept we analyze how well it responds to one semantic part, or a small subset. Firstly, for each visual concept we determine which semantic part it best detects, calculate the AP, and plot the histogram as shown in figure 6 (SingleSP). Next, we allow each visual concept to select a small subset – two, three, or

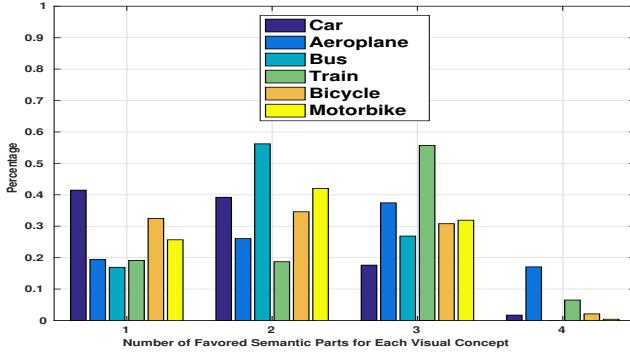


Figure 7: Distribution of number of semantic parts favored by each visual concept for 6 objects. Most visual concepts like one, two, or three semantic parts.

four – of semantic parts that it can detect (penalizing it if it fails to detect all of them). We measure how well the visual concept detects this subset using AP and plot the same histogram as before. This generally shows much better performance than before. From figure 6 (MultipleSP) we see that the histogram of APs gets shifted greatly to the right, when taking into account the fact that one visual concept may correspond to one or more semantic parts. Figure 7 shows the percentage of how many semantic parts are favored by each visual concept. If a visual concept responds well to two, or more, semantic parts, this is often because those parts are visually similar, see figure 8. For some object classes, particularly aeroplanes and trains, there remain some visual concepts with low APs even after allowing multiple semantic parts. Our analysis shows that many of these visual concepts are detecting background (e.g., the sky for the aeroplane class, and railway tracks or coaches for the train class), see figure 9. A few others have no obvious interpretation and are probably due to limitations of the clustering algorithm and CNN feature.

Note that our results also imply that there are several visual concepts for each semantic part. In other work, in preparation, we show that these visual concepts typically correspond to different subregions of the semantic parts (with some overlap) and that combining them yields better detectors for all semantic parts with a mean gain of 0.25 AP.

## 7. Conclusion and Future Work

This paper hypothesizes that CNNs have internal representations of object parts which are encoded by population activity of features/neurons. To validate this hypothesis we used clustering to determine a set of visual concepts for each object class at each layer of the CNN. We showed that these visual concepts are visually tight and appear to correspond to semantic parts of the object. This is supported by quantitative analysis which finds the “best visual concept” for each keypoint (PASCAL3D+) and shows that they have

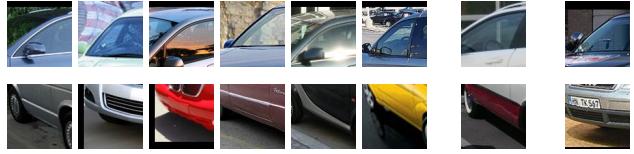


Figure 8: This figure shows two examples of visual concepts that correspond to multiple semantic parts. The first six columns show example patches from that visual concept, and the last two columns show prototypes from two different corresponding semantic parts. We see that the visual concept in the first row corresponds to “side window” and “front window”, and the visual concept in the second row corresponds to “side body and ground” and “front bumper and ground”. Note that these semantic parts look fairly similar.

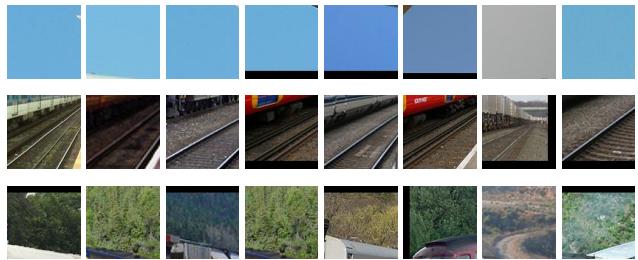


Figure 9: The first row shows a visual concept corresponding to sky from the aeroplane class, and the second and third rows are visual concepts corresponding to railway track and tree respectively from the train class.

high AP for detecting the keypoints. To further understand the relationship between visual concepts and semantic parts we constructed a new dataset ImageNetPart by providing dense annotations in terms of semantic parts. This dataset showed good results for the “best visual concept” but also enabled us to show that most visual concepts were fairly effective at detecting a small subset of semantic parts with similar visual appearance (ranging from one to four). We conclude that visual concepts help understand the internal representations of deep networks used for object detection and that our clustering method for discovering visual concepts can be thought of as unsupervised learning of object parts. Future work involves building on ideas in [24] to learn models, by grouping visual concepts which co-occur.

## References

- [1] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*. Springer, 2014.
- [2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.
- [3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.
- [4] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*. IEEE, 2014.
- [5] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. In *Technical Report 1341*. 2009.
- [6] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.
- [7] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*. IEEE, 2014.
- [8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *MM*. ACM, 2014.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [10] Y. Li, L. Liu, C. Shen, and A. van den Hengel. Mid-level deep pattern mining. In *CVPR*. IEEE, 2015.
- [11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*. IEEE, 2015.
- [12] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014.
- [13] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*. IEEE, 2015.
- [14] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*. IEEE, 2015.
- [15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.
- [16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [17] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*. Springer, 2012.
- [18] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*. IEEE, 2014.
- [19] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*. IEEE, 2015.
- [20] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*. IEEE, 2015.
- [21] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. Springer, 2014.
- [22] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*. IEEE, 2011.
- [23] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.
- [24] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *PAMI*, 32(6):1029–1043, 2010.