Title
Brain-inspired automated visual object discovery and detection

Journal
Proceedings of the National Academy of Sciences of the United States of America, 116(1)

ISSN
0027-8424

Authors
Chen, Lichao
Singh, Sudhir
Kailath, Thomas
et al.

Publication Date
2019-01-02

DOI
10.1073/pnas.1802103115

Peer reviewed

Brain-inspired automated visual object discovery and detection

Lichao Chen[a], Sudhir Singh[a], Thomas Kailath[b,1], and Vwani Roychowdhury[a,1]

[a]Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095; and [b]Department of Electrical Engineering, Stanford University, Stanford, CA 94305

Contributed by Thomas Kailath, April 23, 2018 (sent for review February 12, 2018; reviewed by Rama Chellappa, Shree Nayar, and Erik Sudderth)

Despite significant recent progress, machine vision systems lag considerably behind their biological counterparts in performance, scalability, and robustness. A distinctive hallmark of the brain is its ability to automatically discover and model objects, at multiscale resolutions, from repeated exposures to unlabeled contextual data and then to be able to robustly detect the learned objects under various nonideal circumstances, such as partial occlusion and different view angles. Replication of such capabilities in a machine would require three key ingredients: (*i*) access to large-scale perceptual data of the kind that humans experience, (*ii*) flexible representations of objects, and (*iii*) an efficient unsupervised learning algorithm. The Internet fortunately provides unprecedented access to vast amounts of visual data. This paper leverages the availability of such data to develop a scalable framework for unsupervised learning of object prototypes—brain-inspired flexible, scale, and shift invariant representations of deformable objects (e.g., humans, motorcycles, cars, airplanes) comprised of parts, their different configurations and views, and their spatial relationships. Computationally, the object prototypes are represented as geometric associative networks using probabilistic constructs such as Markov random fields. We apply our framework to various datasets and show that our approach is computationally scalable and can construct accurate and operational part-aware object models much more efficiently than in much of the recent computer vision literature. We also present efficient algorithms for detection and localization in new scenes of objects and their partial views.

computer vision | brain-inspired learning | brain-inspired object models | machine learning | brain memory models

Visual object classification and recognition is of fundamental importance for (almost) all living animals, and evolution has made the underlying systems highly sophisticated, enabling abstractions and specificity at multiple levels of the perception hierarchy. The design of unsupervised, scalable, and accurate computer vision (CV) systems, inspired by principles gleaned from biological visual-processing systems, has long been a cherished goal of the field. For example, the recent success of the Deep Neural Network (DNN) framework has largely been attributed to its brain-inspired architecture, comprised of layered and locally connected neuron-like computing nodes that mimic the organization of the visual cortex (1–5). The features that a DNN automatically discovers are considered to be its primary advantage (2, 5), and it outperforms more conventional classifiers driven by hand-crafted features [such as scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG) (6, 7)].

While the deep learning (DL) framework is undoubtedly a significant achievement, especially in simultaneously learning the visual cues of more than a thousand object categories, it is widely acknowledged that brains are much more efficient and that their operating principles are fundamentally different from those of DNNs or other existing machine-learning platforms (8, 9). Some key limitations of existing DNN-like platforms are as follows: (*i*) a predominantly supervised framework, where one must train them using large manually labeled training sets, and (*ii*) lack of a formal framework for bringing in the higher levels of abstraction necessary for developing a robust perceptual framework, such as recognizing the persistent identity of an object category (e.g., humans, cars, and animals) that is invariant under different views and under variabilities in their shape and form. To put it another way, these platforms lack a framework for a contextual understanding of scenes where different objects and concepts occur together. On the other hand, biological vision systems (*i*) are largely unsupervised learning systems that can learn highly flexible models for objects based purely on familiarity and repeated visual exposures in different contexts, (*ii*) can detect such learned objects at various scales and resolutions, and (*iii*) are highly computationally efficient. Therefore, exploring potential synergies between biological and CV systems remains a topic of considerable ongoing interest.

In this paper, we consider unsupervised machine-learning scenarios—imitating what humans encounter—such as the following: An automated probe browsing the Internet encounters a large body of contextual images, which we also refer to as perceptual data, where a majority of the images contain discernible and high-quality instances of objects from an unknown set of categories. For example, images obtained from videos of real-world scenes show the same objects persistently in their natural environments. Similarly, contextual visual browsing based on text

Significance

The Internet contains millions of videos and images of objects, captured contextually, just as they would appear in their natural environments. Such access to large-scale data has injected an urgency to an age-old query: Can one automatically browse these data, as a human would, to construct brain-like and computationally tractable models of the visual world? If successful, many of the scalability and robustness limitations of existing computer vision methodologies, which primarily rely on supervised frameworks requiring manually labeled exemplars, will be removed. As a step toward solving this problem, we leverage widely accepted findings in the fields of cognitive psychology and neurophysiology to develop an unsupervised framework for automated isolation of objects and their efficient detection and localization in new environments.

Author contributions: L.C., S.S., T.K., and V.R. designed research; L.C., S.S., T.K., and V.R. performed research; L.C. and V.R. analyzed data; and L.C., T.K., and V.R. wrote the paper.

Reviewers: R.C., University of Maryland, College Park; S.N., Columbia University; and E.K., University of California, Irvine.

The authors declare no conflict of interest.

Data deposition: The in-house dataset used in the paper is shared publicly at https://www.ee.ucla.edu/wp-content/uploads/ee/cele_images_lite.zip.

[1]To whom correspondence may be addressed. Email: kailath@stanford.edu or vwani@ucla.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1802103115/-/DCSupplemental.

Published online December 17, 2018.

tags, can provide such large-scale perceptual data. This is exactly the perceptual learning world view with which, for example, an infant is faced, and this is what the Internet makes available to computers and machines for the first time. It is worth reiterating that in our unsupervised learning framework, and unlike in supervised training scenarios, no labels or bounding boxes of any kind are used to tag these images. Given such perceptual data, the tasks are (*i*) to discover and isolate the underlying object categories just by processing these unlabeled images, (*ii*) to build visual models of the discovered categories, and then (*iii*) to detect instances of the said objects in new scenarios, all in a robust manner not affected by operations such as scaling, occlusion, and different viewpoints. Humans and many other animals routinely execute these and much more complex visual tasks.

As a step toward developing such an unsupervised contextual learning framework, we first abstract some of the key aspects of biological vision systems. The immediate goal is not to emulate the exact granular feature-generating brain hardware, such as neurons and their layered interconnections, but to try to computationally capture the basic principles that have been strongly hypothesized as being used in brains and to integrate them into an end-to-end CV framework.

Object Prototypes–SUVMs

The related cognitive science review is outlined in more detail in *SI Appendix*, section 1, p. 1. We have incorporated only certain specific abstractions of the object prototype theory of perception (10) in defining what we shall call a structural unsupervised viewlets model (SUVM) that has two interacting parts to it. (*i*) Viewlets: There is strong evidence for the presence of neurons (e.g., those in the inferotemporal cortex) that fire selectively in response to views of different parts of objects. Neurons in this cortex respond selectively to stimuli from color and texture and even from complex views such as faces (10). It is as if the brain breaks up an object into visually distinct but potentially overlapping jigsaw pieces of different sizes. Each such view is a building block in our model, and to emphasize that such views are not necessarily distinct functional parts, we refer to them as "viewlets." Thus, for our modeling purposes, viewlets are multiscale visual cues representative of different appearances of the object under different circumstances—for example, in the case of humans, different views of the head or arms in different poses or a half body view or just the legs in different poses and partially or fully covered. As explained in *Methodology*, each viewlet is modeled as a distribution over a feature space, allowing for variations in the appearance of exemplars belonging to the same category. (*ii*) A set of models that determine how these viewlets are geometrically organized to create an entire or a partial image of an object, which includes the following models.

Spatial Relationship Network. It has been hypothesized that viewlets that have stable geometrical relationships to each other are indexed in the brain according to their relative spatial locations (11). Exemplars are recognized as class members if and only if the structural information is close enough to that of the prototype (12). We capture this feature through the spatial relationship network (SRN), which uses a variation of the spring network model (13): This is a graph in which nodes are the viewlets and edges impose relative distance and scale/size constraints that the viewlets should satisfy. To encode variations, the edges are represented as springs of varying stiffness and length. Since not all parts directly connect to each other in a physical object, our model naturally allows for sparsity: It introduces springs only among key viewlet pairs that are needed to maintain the integrity of the whole object. Collectively, the entire object model is then defined by the spring-viewlet ensemble.

Configuration-Independent Parts Clustering. A configuration-independent parts clustering (CIPC) that captures certain semantic structures of the object prototype by grouping viewlets with distinct appearances into higher level concepts of parts. For example, the notion of the "left arm" is captured by a set of viewlets corresponding to different configurations of the arm—for example, hanging down or elbows out. Such viewlets look very different from each other in appearance and yet can be structurally identified as configurations of the same part since they occupy the same relative position with respect to other body parts such as the torso and the head.

Global Positional Embedding. A global positional embedding (GPE), in which each viewlet is assigned its own 2D location and a scale value. An optimization algorithm computes these viewlet-specific location and scale coordinates so that they yield best fits to the relative location and scale constraints in the structural relationship network (SRN). GPE brings out the underlying hierarchical semantic structure of the object—that is, how the viewlets are organized both spatially and hierarchically in scale. Thus, for example, the GPE would show that an upper half-body viewlet subsumes viewlets that correspond to the head and shoulder regions. This spatial map can be further segmented into clusters of viewlets that define important regions of the object prototype, which in turn could be interpreted as higher level parts of the object category.

The semantic structures encoded in the CIPC and in the GPE play an important role in robust detection. For example, detection of matching viewlets (those whose relative location and scale values match predictions made by the object prototype) corresponding to two different body parts is a much more robust indicator of the presence of a human than detecting multiple viewlets corresponding to only a single body part. As demonstrated by our results, the SUVM leverages both spatial structure and semantic information to enable high-precision object detection.

A Positive-Only Learning Setup for Estimating SUVMs. In *Methodology*, we formulate a probabilistic model for an SUVM to allow us to develop reliable estimation and learning algorithms. Then we describe how SUVMs for unknown object categories can be automatically learned from large-scale unlabeled perceptual data (a large body of contextual images). Our framework mimics the process of perceptual learning observed in biological systems, wherein category prototypes are learned via repeated exposures to exemplars and examining them from different perspectives in multiple contexts (10). Since for our model building we do not need explicit negative examples, we refer to our setup as positive-only learning. Our perceptual framework corresponds to closely related frameworks in the CV literature ranging from weakly supervised to unsupervised. For example, in the weakly supervised setup of ref. 14, positive exemplars belonging to a single category are unlabeled, but negative exemplars (e.g., background or clutter images, and images from categories other than the one being learned) are explicitly provided as part of the training set (15). Our model automatically learns to create its own negative examples and does not need such external information. In another instance, the unsupervised setup of refs. 16, 17 use data that have a perceptual bias over multiple categories. For example, in ref. 17, automated object category discovery is carried out over contextual data that contain unlabeled exemplars belonging to over 20 different categories; this work, however, does not build stand-alone models for each category that can be used to detect instances from new data (see *SI Appendix*, section 10, p. 23). While our paper reports experimental results involving perceptual data that have exemplars belonging to a single category of interest at a time (thus, closer to the weakly supervised models in CV), there is nothing in the framework that precludes it from

discovering and modeling tens of categories as long as there are sufficiently many instances of each category in the dataset. Our framework will create SUVMs, complete with multiple viewlets and their geometric arrangements, for each such category in the data.

The learning step involves joint estimation of the set of unknown viewlets relevant to the object category, and of the associated models (i.e., SRN, CIPC, and GPE) that constitute the SUVM. We use maximum likelihood estimation (MLE) as our foundational framework and incorporate sparsity constraints and convex relaxations to ensure computational tractability. This leads to an intuitive but mathematically rigorous and computationally simple learning framework.

Object Detection and Localization. Given a learned SUVM, we turn to the task of detecting instances of objects in a new image. Again, we abstract how brains are theorized to detect objects as belonging to a category: by the occurrence of a sufficient number of compatible viewlets that are spatially located as predicted by the SUVM. While we follow an MLE framework of locating a portion of the image that has a high likelihood of being generated by the SUVM, we avoid exponential search complexity (usually associated with the exhaustive combinatorial search required in MLE), by intuitive but careful pruning of the search space based on the structure of the SUVM. This enables a linear-time detection and localization algorithm that can locate multiple instances of the object, unaffected by scale and to the presence of occlusions in the images. The mathematical details are provided in *Methodology*.

Relation to Previous Work. There is a long history of efforts, similar in spirit to ours, aimed at building CV frameworks that develop parts-aware object prototypes (13, 14, 16–22) [see, for example, Geman et al. (23) for a review]. More details on these models are given in *Results*, *Discussion*, and *SI Appendix*, section 10, p. 23. Our conclusion is that the goal of learning persistent and flexible object models, especially when the object is deformable and comprised of multiple configurable parts, in an automated and unsupervised manner is largely unsolved. Most parts-aware approaches require strongly supervised training (18, 21, 22, 24). Moreover, to compensate for computational scalability challenges, they use limited object models such as (*i*) using only a few parts in describing the object and (*ii*) further restricting the kinds of relationship patterns between parts so that they form networks such as trees and stars. Almost all previous attempts at unsupervised (e.g., ref. 17, which can be interpreted as using a star-network spring model) and weakly supervised (e.g., ref. 15) frameworks require datasets where exemplars have very similar views. For example, exemplars must have arms in the same position relative to the body (e.g., see Fig. 1, where we illustrate this point). On the other hand, our SUVM framework is flexible, computationally scalable and allows for models comprising hundreds of viewlets per object, all embedded in a flexible spatial model and a semantic structure that is independent of visual appearances. From a purely modeling perspective, SUVMs can be considered as combining ideas from both the supervised approaches [e.g., Poselets (19)], which use an ensemble of templates with embedded exemplars, with tags at key points, to define an object category, and the flexible mixtures of parts model for human pose detection and estimation (22) (which can be interpreted as grouping viewlets into an ensemble of tree networks, instead of a single SRN) and the probabilistic parts-constellation models used in the weakly supervised approach of refs. 14, 15 and the unsupervised approaches in refs. 16, 17.

Results

Data Description. We experimented with two datasets:

i) The CalTech-4 dataset, comprising faces (435 images), motorbikes (800 images), airplanes (800 images), and cars (800 images). A primary motivation for selecting this dataset

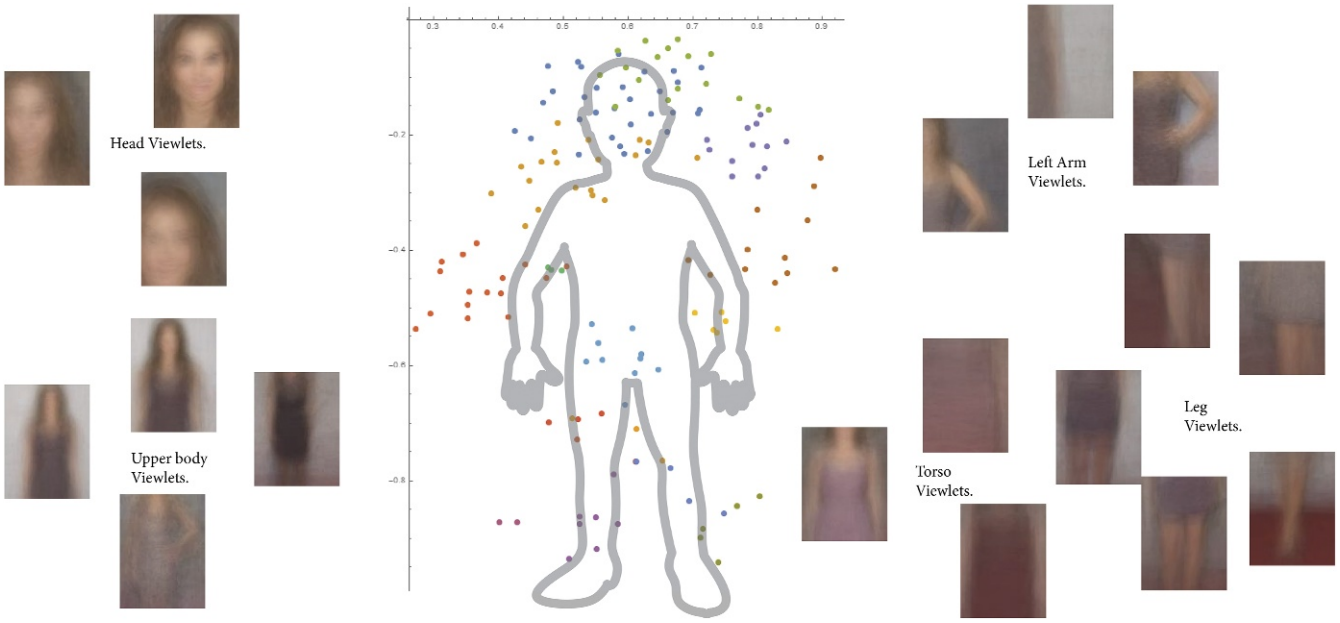

Fig. 1. The colored dots in the figure show estimated average (*X*, *Y*) coordinates of the centers (as determined by the GPE algorithm) of some of the viewlets in our human SUVM. Recall that each viewlet is a distribution over a set of example views/patches that have similar appearances. The patches belonging to the same viewlet are averaged at the pixel level to create a representative visualization patch. Viewlets automatically clustered as comprising a "part" are given the same color code. For example, we automatically group viewlets corresponding to different views of the left arm as belonging to the same part cluster, which thus can be tagged as the left arm part. Three such viewlets corresponding to the arm straight down and with elbow out are shown here. Similarly, multiple viewlets corresponding to different views of head/face, legs, and torso are also shown.

Table 1. Confusion matrices for the CalTech-4 dataset with one multicategory classifier.

	Classifier ⟶ (SUVM + SVM)/[Fergus et al. (15)]			
Query ↓	(F)ace	(M)otorbike	(A)irplane	(C)ar
F	0.982/0.862	0.000/0.073	0.018/0.028	0.000/0.014
M	0.000/0.000	0.990/0.977	0.010/0.013	0.000/0.000
A	0.005//0.003	0.013/0.042	0.967/0.888	0.015/0.060
C	0.000/0.008	0.000/0.092	0.020/0.197	0.980/0.670

The table entry (*i*, *j*) is the percentage of query images belonging to category (*i*) that are classified as belonging to category (*j*). Each table entry is separated with the delimeter "/": the number to the left is the performance of the (SUVM + SVM) approach and the entry to the right is the performance for Fergus et al. (15). For SUVMs, a visual dictionary is learned from all of the images (i.e., a shared visual dictionary is created), but each model is learned only from its category-specific images. A single 4-class SVM classifier is built by combining the outputs of all of the four models as was done in ref. 15.

is that there are existing results, using earlier weakly supervised and unsupervised parts-aware formalisms (14, 15), which we can use to evaluate our framework (see Tables 1 and 2). The other motivation is that it enables us to perform a number of experiments to explore the limits of our SUVM framework. It allows us, for example, to evaluate how our learning framework performs with small-size data (e.g., only 218 images are available to learn models for a face; the other 217 being used for testing). The data do not always contain enough exemplars to capture many of the variations in shape and orientations of the object instances. We also ask questions such as, If the machine's world view is limited only to one object category, how would it interpret the rest of the world? For example, would a face SUVM (learned only from face-related perceptual data and with no exposure to images from other categories) "see" faces everywhere when shown images of motorbikes or cars? Furthermore, it allows us to illustrate another brain-like activity: the ability to improve detection/localization performance by jointly using multiple object prototype models (learned from different perceptual datasets). Table 3 illustrates how the face SUVM learned from the CalTech-4 dataset can be used in conjunction with the full-body human SUVM to obtain face detectors with higher precision.

ii) a celebrity dataset, which we created by crawling 12,047 high-quality images from the web. This is a good example of the types of perceptual data the Internet can provide and is ideally suited to our perceptual framework: The images are from natural settings with diverse backgrounds and resolutions and often have multiple instances of individuals (the unknown category to be modeled) in the same image, who display a wide diversity in clothing and body gestures. For evaluation purposes only, we manually annotated the whole dataset with precise main body part (such as the head and the torso) locations; this information was not used in the learning process. There are no unsupervised approaches to extract object prototypes from such datasets, and hence, we compare our detection and localization performance with those of supervised frameworks. See the detailed discussion on torso detection in *Evaluation of SUVMs via Detection and Localization Tasks*.

SUVM Learning and Visualizations. For the celebrity dataset, we used 9,638 images as a learning set and the rest of the images as a test set. As a first step, we learn a set of visual words, which we shall call a "visual dictionary," that captures the repeated visual patterns in the learning set (see *Methodology*). To build the visual dictionary, we first randomly sample 239,856 image patches (each of size 128×96 pixels) from the dataset using a scale pyramid: We successively scale down each image by a constant multiplicative factor or scale to create a layered "pyramid." Then we select fixed-size windows, located at random locations in each layer, to crop image patches. These patches are then represented in the form of dimension-reduced HOG descriptors (7). These descriptor vectors are then grouped into k clusters using the k-means algorithm. After comparing the results of k-means clustering for different ks, we settled on $k = 1{,}006$, and the corresponding clusters formed our visual dictionary.

We next followed our SUVM learning steps as described in *Methodology* and derived a sparse SRN, containing 566 viewlets. Thus, while the visual dictionary contains 1,006 visual words, only a subset of them are viewlets in the human model, and the rest describe background scenes. These viewlets are then visualized as a weighted average of all of the constituent image patches and are shown in *SI Appendix*, section 8; examples of a few select viewlets are also shown in Fig. 1. As one can determine via visual inspection, each viewlet corresponds to a meaningful human body part. We then (*i*) constructed a CIPC network and automatically found 18 distinct parts (see *SI Appendix*, Fig. S8) and (*ii*) computed a GPE of the viewlets. Fig. 1 illustrates some of the salient aspects of our human SUVM.

The same steps are executed for the CalTech-4 datasets, with one twist: (*i*) separate dictionaries—the visual dictionary for each category is constructed from its category-specific images, and then an SUVM is derived for each category from its own image set—and (*ii*) shared dictionary—a common visual dictionary is derived from all of the learning sets, and then four separate SUVMs are learned by processing their respective category-specific images. The resulting viewlets and their automated groupings as parts are illustrated in *SI Appendix*, section 9.

Evaluation of SUVMs via Detection and Localization Tasks.

CalTech-4 dataset results. The results on the CalTech-4 dataset as summarized in Tables 1 and 2 show that the SUVM

Table 2. Confusion matrices for the CalTech-4 dataset based on four separate category models

	Models			
Query image ↓	F	M	A	C
SUVM(separate)/[Fergus et al. (14)]				
(F)ace	0.980/0.964	0.069/0.33	0.215/0.32	0.252/-
(M)otorbike	0.000/0.50	0.95/0.925	0.370/0.51	0.237/-
(A)irplane	0.000/0.63	0.007/0.64	0.665/0.902	0.025/-
(C)ar	0.000/	0.000/-	0.002/-	0.600/-
SUVM (shared)				
(F)ace	0.972477	0.087156	0.674312	0.073394
(M)otorbike	0.007500	0.960000	0.675000	0.140000
(A)irplane	0.000000	0.002500	0.745000	0.117500
(C)ar	0.000000	0.000000	0.167500	0.970000

Each category model (*j*) only outputs whether a query image contains an exemplar of category (*j*). The table entry (*i*, *j*) is the percentage of query images belonging to category (*i*) that are detected to contain an instance of category (*j*) [by using a category model (*j*)]. For the top half of the table, each table entry is separated with the delimeter "/": the number to the left is the performance of the SUVM(separate) approach and the entry to the right is the performance for Fergus et al. (14). The top half of the table corresponds to the case where SUVMs are created using separate dictionaries, whereas the bottom half of the table corresponds to the shared dictionary case. Column 1 in the top half of the table, for example, shows that the face SUVM returned no FPs when tested on nonface images; the face model in ref. 14, on the other hand, returned 50% FPs on motorbike images. Similarly, the FP rate on face images for the motorbike model is 6.9% for SUVM vs. 33% in ref. 14. The bottom half of the table shows that the face, motorbike, and car models do extremely well, even without a separate multiclass classifier.

ENGINEERING

Table 3. Face detection—SUVM vs the Viola–Jones algorithm (VJA) (25)

		SUVM			
Performance metrics	VJA	H-1	H-2	HF-1	HF-2
True-positive (TP)	3,072	2,965	3,048	2,959	3,047
FP	972	54	301	31	183
Coverage/recall	92.9%	89.7%	92.2%	89.5%	92.2%
Precision	76.0%	98.2%	91.0%	99%	94.3%

Coverage/recall is the ratio of (number of TPs) and (actual number of positives in the labeled test data). Precision is the ratio of (number of TPs) and (number of TPs + number of FPs). For a description of the OpenCV implementation used for VJA and ROC plots, see *SI Appendix*, section 8, p. 19, column 1. The columns labeled H-1 and H-2 represent face detection results for two different settings of parameters, when our human SUVM is used for prediction. A subset of these predicted face patches with high-enough resolution (e.g., those with heights greater than 150 pixels) are then filtered through the face SUVM, derived from the CalTech-4 dataset, and those that do not pass are rejected. The respective results after this filtering are shown in columns HF-1 and HF-2. The human SUVM provides much higher precision while matching the coverage of the well-known algorithm.

framework significantly outperforms the only other comparable unsupervised/weakly supervised framework in the literature (14, 15). As noted earlier, this dataset allows us to explore the workings of SUVMs in more detail. For example, Table 2 shows that a face SUVM, learned solely from face images, makes no errors [i.e., does not give false-positives (FPs) of detecting faces] when fed with images from the rest of the categories. In comparison, the face model in ref. 14 detected faces in 50% of the test motorbike images. Fig. 2*A* shows an instance of what happens when a motorbike image is viewed through the lens of a face SUVM: Each patch in the motorcycle image has to be assigned to one of the visual words derived from the face-only dataset, and consequently, individual face viewlets are detected all over the image. However, they do not have the collective structural integrity to be detected as instances of face. The airplane and car categories are not as discriminative when their visual worlds are based on such separate dictionaries. As shown in the bottom half of Table 2, however, performance improved considerably across three categories (face, motorcycle, and car) when we used a shared dictionary (i.e., we used images from all of the categories to learn a shared set of visual words) but still learned individual SUVMs solely based on their category-specific images, and no negative examples were used. The airplane model is the worst performer: Not enough examples are present in the learning set (comprised of only 400 images) to learn the different shapes and orientations of airplanes (for example, there are several images with planes pointing in opposite directions) in the data. Finally, by combining the four models together using an SVM (support vector machine), one gets almost perfect detection results across all four categories as shown in Table 1.

Human dataset results. Regarding the precision and coverage/recall performance for various detection tasks and comparison with supervised methods, recall that during our object prototype learning process, we automatically break up the given perceptual visual data into a visual dictionary that is comprised of two sets: visual words that are part of an object prototype, which we refer to as viewlets, and the rest that represent visual cues of the background scenes. Since we do not bring in any negative exemplars, everything the model sees is interpreted only in terms of this dictionary, and hence, the quality and diversity of the perceptual data plays a very important role in how the model performs when it sees a new image. During detection, the first step is to decompose the given image into patches at multiple scales using a sliding window and a scale pyramid and then to assign to each patch the likelihood of it being a particular word from the dictionary that was already learned. We have used the *k*-nearest-neighbor (kNN) classifier for this classification task.

The limitations of the kNN classifier are well known, and that is why in most CV applications considerable effort is expended to train much more powerful classifiers using additional negative exemplars that represent other object categories and background scenes. It is, however, instructive to note how well an SUVM does, even with very weak classifiers for viewlets and without the benefits of training with negative exemplars. Fig. 3 illustrates the types of viewlets detected in images with multiple people in them. In *Discussion*, we point out how one can incorporate negative exemplars to improve performance.

Head/face detection. We used a subset of the viewlets in our human SUVM and mapped their detected locations to where the head would be to create a face detector. As summarized in Table 3, our results turned out be vastly superior to those obtained by the VJA (25), which was developed via extensive manual training over multiple years and was the face-detection algorithm of choice until the recent development of superior face detectors, made possible by even more extensive training and DL. The superior performance is clearly because of the structure embedded in our human SUVM: Viewlets corresponding to other body parts can locate the position of the face, even when our face-only viewlet detectors are weak. As explained in Table 3, this experiment allowed us to combine two different SUVMs: The precision of our face detector (based on our human SUVM) can be significantly increased, without compromising recall performance, by further examining the predicted faces via the face model obtained from the CalTech-4 dataset. This emulates how human vision tends to work: First impressions of objects, based on outlines, are further refined by focusing on the details. Fig. 4 illustrates a couple of face detection results under different views.

Torso detection. Torsos are much harder to localize, due to lack of distinctive features and the large variety introduced by dress patterns. These considerations make a rigid template-based torso detector impractical. However, part-based models can detect torsos by mapping other detected parts to where the torso would be. We compare our model with two other notable part-based approaches—namely, Poselets (19) and the Deformable Parts Model (DPM) (18)—and the results are summarized in Table 4. As shown in Table 4, we again outperform these strongly supervised models. Note that the ingenious Poselets approach is based on processing manually tagged data to generate hundreds of distinctive templates (and associated classifiers obtained through extensive supervised training) that have exemplars with various views/poses and scales embedded in them. Such templates, however, do not constitute detailed object prototypes of the kind illustrated in Fig. 1.The slightly lower recall rate in our model is to be expected, as the weak kNN classifer misses many of the viewlets that are otherwise present in an image. As further elaborated in *Discussion* (see the part on

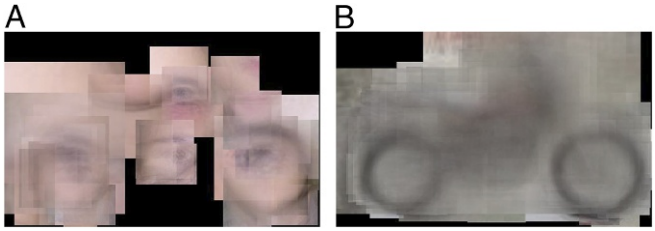

Fig. 2. (*A*) A motorcycle image viewed through the lens of a face model. While it sees face viewlets everywhere, it does not detect any faces because the viewlets do not match structurally. (*B*) The same image viewed through a motorcycle model.

Table 4. Torso detection—SUVM vs. parts-aware supervised approaches

Performance metrics	Approaches			
	DPM (18)	Poselets (19)	SUVM	SUVM, stricter
TP	1,239	3,115	2,935	2,838
FP	5,263	1,678	277	52
Coverage/recall	38.3%	96.3%	90.7%	87.7%
Precision	19.1%	65.0%	91.4%	98.2%

For details of DPM and Poselets, see *SI Appendix*, section 8, p. 20, column 2. SUVM outperforms all of the models in precision [for Poselets, we used the recommended threshold value of 3.6 (19)], while providing a solid recall performance. As discussed in the torso detection section in *Evaluation of SUVMs via Detection and Localization Tasks*, the recall performance of an SUVM can be improved by replacing the kNN classifiers it uses with superior supervised classifiers and introducing negative examples. We have intentionally persisted with the weak kNN classifiers to emphasize the power that an SUVM derives from its structure and its hundreds of viewlets.

dealing with low-resolution images and integrating supervised learning), we can also train classifiers for each visual word in our model to improve our performance. The point of this paper is to show how well the SUVM does even with a limited and self-contained world view.

Discussion

Improving SUVMs. Several features can be added to SUVMs, so as to increase both their accuracy and power of representation. For example, currently we have not explicitly assigned prior probabilities to the occurrences of individual viewlets and parts. Doing this would make detection more robust: An occurrence of a distinctive viewlet or a part could be given more weight in making a decision during object detection. Similarly, the assumption that pairwise relative distances among viewlets is unimodal in distribution can be relaxed. For many highly flexible objects such as humans, the separation distance could be bimodal: For example, if we want to capture both standing and sitting postures of humans via a single SUVM, then the relative distance between head and feet viewlets would clearly have to be bimodal in distribution. Furthermore, instead of relying on random sampling and then clustering the samples, the compilation of a visual dictionary and determination of viewlets can be improved, especially using joint-segmentation and unsupervised object discovery techniques introduced, for example, in ref. 17.

We also recognize that complex objects can have such radically different views so as to make them look like two different objects altogether, where they share none or very few viewlets. A sideways profile view versus a full-body frontal view of humans provides such an example. In such situations, one can have two different SUVMs representing the two scenarios and recognize them to be the same object category based on cues other than just images. For example, spatiotemporal continuity or flow of objects in a video, where a person moves from facing the camera to facing perpendicular to it in consecutive frames, would be sufficient to determine that the views belong to the same object category. Using motion to persistently detect objects is a field of considerable interest and can be incorporated into our framework (26). Shared text tags in image databases can also provide such information (27).

Dealing with Low-Resolution Images and Integrating Supervised Learning. Traditionally a lot of the curated and publicly available databases have predominantly low-resolution images—for example, a distant silhouette of someone walking or multiple people in the same image, where only full body views have enough resolution for detection. Detecting objects in such databases is challenging for any parts-aware approach and especially so for ours, where we use weak classifiers: There is not enough resolution to reliably detect individual viewlets corresponding to different parts, thereby losing the advantage afforded by collective decision making by a group of viewlets. For dealing with such situations, one can actively create low-resolution templates from the high-resolution learned object prototypes. That is, once an SUVM is learned from high-resolution datasets, one can methodically subsample the viewlets and embed them in different scenes and build an ensemble of classifiers using the full power of discriminative learning afforded by sophisticated classifiers, such as DNNs. The discriminative power can be further enhanced by including nonobject visual words and image patches corresponding to viewlets belonging to other categories as negative examples. The success of a recent approach for detecting tiny faces based on an ensemble of supervised classifiers (each classifier trained for a particular scale) (28) is a good indicator that our suggested approach would succeed. The training sets needed for our method, however, are automatically generated, minimizing the need for supervised learning.

Scaling up to Detect Multiple Categories. Using the Internet, one can find large-scale perceptual data of the kind analyzed here for almost any category of objects (27). Thus, given enough data and computational power, efficient SUVMs for most individual categories can be reliably built. The preceding discussions, however, highlight that the main challenge will lie in integrating the different SUVMs: While structural information plays a very important role, it is still necessary that viewlets belonging to different object prototypes be mapped to a common feature space—a shared visual world view—so that they can be reliably distinguished. To facilitate such integration, the paradigm of DL with its proven capability to simultaneously learn a thousand or more different categories can be incorporated into our framework. The manually created training sets that are currently used to train DNNs, however, can now be replaced by the automatically generated viewlets. The network will no longer be trained to detect manually tagged categories but to detect automatically generated viewlets. Thus, one can potentially automate both the tasks of object discovery and high-accuracy detection. Recent work on domain adaptation methods (29) provide a proof-point for the validity of such an approach. Instead of generating an object model from a set of contextual unlabeled images (as in our work), these approaches attempt to model the overall bias between two datasets belonging to different

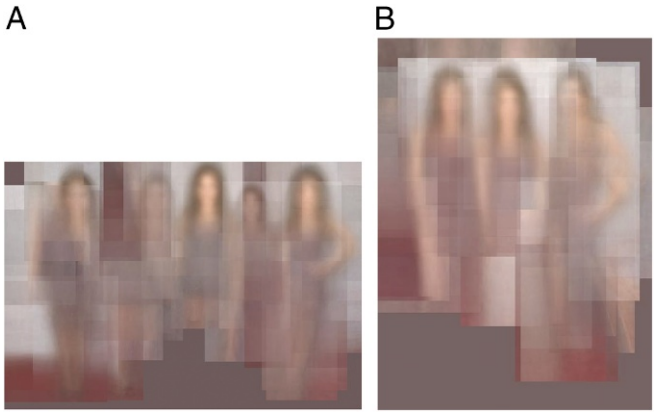

Fig. 3. An illustration of what the human SUVM sees in images with multiple people. For each individual, matching viewlets corresponding to different body parts and their poses are detected even in the presence of occlusion.

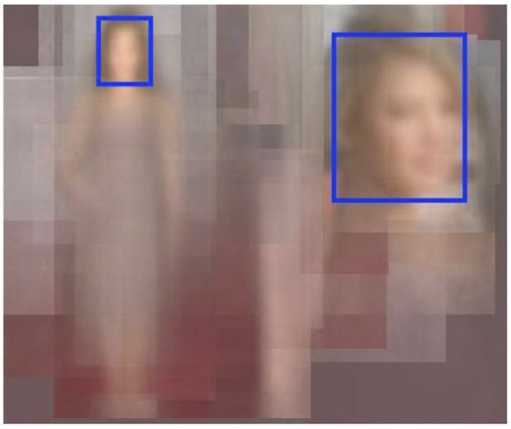

Fig. 4. Face detection at different scales and with different views (front vs. side).

domains. By capturing the differences between a source domain (where the classifier was trained) and the target domain (where it is applied), the performance of updated classifiers can be improved. Here, the target domain can be unlabeled, but the source domain must be labeled and hence supervised. Our unsupervised SUVM framework can benefit from such an approach: The SUVMs for a category can be further tuned and differentiated from SUVMs for a different category in an unsupervised manner.

SUVMs and Recent Work on Unsupervised DL. The unsupervised DL literature can be broadly categorized into three groups: (*i*) Autoencoders (2) generate low-dimensional representations of input signals, which can then be clustered to derive visual words in the dataset. We currently perform this step using non-DL methods: By using predefined features, such as HOG, and then by performing PCA, we obtain low-dimensional feature vectors for our image patches. Then, we cluster these feature vectors using *k*-means to obtain visual words. Our main contribution lies in creating object models that build on these visual words, a step currently not done by the DL methods. In our ongoing work, we are implementing deep autoencoders, which can provide better features and hence a more robust set of visual words. (*ii*) In generative adversarial networks (GANs) (30), a DNN, driven by random noise, generates sample images that try to mimic a given set of images in an adversarial setting. If everything converges (it tends to get stuck in local minima often), then the adversarial network learns to generate outputs/images similar to those in the learning set. It, however, does not generate and is not intended to generate a parts-based persistent model of an object category of the type we do. (*iii*) In sequence prediction, given a sequence of images (for example, in a video setup), one can learn to predict future frames based on current and a few past frames. Thus, the prediction network [such as the long–short-term memory (LSTM) model (31)] can be said to have learned a representation of how objects move and change shape. This is again a useful end-to-end model that learns an overall representation and is not intended to learn parts-aware representational models. In summary, the paradigm of DNNs with its ability to memorize patterns and templates is a powerful tool, and in our future work, we plan to use its power to make our models more accurate and expressive.

From Modeling Object Categories to Modeling Scenes. Once individual object prototypes are learned, one can go to a higher level of abstraction. Instead of viewlets and their relative positions in an SUVM, one can capture the cooccurrence and relative locations and orientations of object instances (belonging to different categories) to define an analogous scene model. (See ref. 23 for an insightful discussion on the importance of this problem.)

Methodology

The SUVM: Representation and Learning. We outline the mathematical and computational formulations of the SUVM (most of the details are deferred to *SI Appendix*, section 2).

Viewlets. Recall that viewlets are multiscale characteristic appearances of views of objects. To account for variations, each viewlet V_i is modeled as a random variable that outputs a patch or a part of an image with an appearance feature vector random variable A_i, which is drawn from a certain distribution over a feature space. For example, in this paper, each sample of a viewlet V_i is represented by a rectangular patch of fixed width (w) and fixed height (h). Moreover, because viewlets can represent larger or smaller sections of the same object category under consideration (e.g., a half-body viewlet will contain the head and hence has a larger size or scale than a head-only viewlet), we associate a relative scale parameter S_i with V_i. To accommodate variations, S_i is itself a random variable. The best way to visualize S_i is to imagine a global scale parameter s for an exemplar embedded in a given image—that is, s determines the overall size of the object in pixels. In such a scenario, any sample of viewlet V_i has a width of $s_i^{(x)} = w * S_i * s$ pixels and a height of $s_i^{(y)} = h * S_i * s$ pixels. The appearance feature vector, A_i, of a sample is a set of features derived from the underlying image patch. Though for our experimental results we use local HOG features (see *Results* and *SI Appendix*, section 8), SUVMs can use any feature set, including those derived using DNNs.

The SRN. Let's recall that the SRN uses a variation of the spring network model to represent pairwise distance and scale variations among the viewlets. Nodes in the network are the viewlets, and edges are the relative distance and scale/size constraints. To represent distances, each sample of a viewlet, V_i, is assigned a location coordinate, $X_i = (x_i, y_i)$, where (x_i, y_i) are the pixel values of the top left corner of the associated rectangular patch that has a width of $s_i^{(x)} = w * S_i * s$ pixels and a height of $s_i^{(y)} = h * S_i * s$ pixels. The relative distance between two viewlets V_i and V_j can then be modeled by the random variable $(X_i - X_j)$. However, to have both scale and translation invariance, we need to normalize the location differences appropriately. In particular, as explained in *SI Appendix*, section 2, since the variances in the perceived/measured location coordinates (i.e., x_i and y_i) depend on the actual lengths, we define a scale-normalized relative distance measure, $\left(\frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}}, \frac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}}\right)$.

To model variations in relative positions, each pair of viewlet nodes V_i and V_j is connected via a spring of stiffness parameter $c_{ij} \geq 0$ and of zero-stress normalized length μ_{ij}. Variations in pairwise relative distances from their respective zero-stress lengths lead to overall stress, which can then be modeled by a potential function defined over the ensemble of springs. Further, assuming an isotropic spring model, where the displacements along the x axis and the y axis are treated separately and independently, a total potential function of a given configuration can be written as $\mathbf{G} = G(\mathbf{X}) + G(\mathbf{Y})$, where

$$\mathbf{G}(\mathbf{X}) = \frac{1}{Z^{(x)}} \exp\left(-\frac{1}{2}\sum_{i \neq j} c_{ij}^{(x)}\left(\frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}} - \mu_{ij}^{(x)}\right)^2\right), \quad [1]$$

$$\mathbf{G}(\mathbf{Y}) = \frac{1}{Z^{(y)}} \exp\left(-\frac{1}{2}\sum_{i \neq j} c_{ij}^{(y)}\left(\frac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}} - \mu_{ij}^{(y)}\right)^2\right), \quad [2]$$

and (*i*) $Z^{(x)}$ and $Z^{(y)}$ are the corresponding normalization terms, also referred to as the partition functions, and (*ii*) $\mu_{ij}^{(x)} = \left(\frac{\mu_i^{(x)}-\mu_j^{(x)}}{s_i^{(x)}+s_j^{(x)}}\right)$, $\mu_{ij}^{(y)} = \left(\frac{\mu_i^{(y)}-\mu_j^{(x)}}{s_i^{(y)}+s_j^{(y)}}\right)$ are the normalized zero-stress lengths. The above expressions are functions of the relative scale parameters S_i, and their pairwise variations can again be modeled via springs. Since scale is a multiplicative factor, we take its logarithm and define an analogous potential function:

$$G(\mathbf{S}) = \frac{1}{Z^{(s)}} \exp\left(-\frac{1}{2}\sum_{i\neq j} c_{ij}^{(s)}\left(\log\frac{S_i}{S_j} - \log\frac{\mu_i^{(s)}}{\mu_j^{(s)}}\right)^2\right), \quad [3]$$

where $\mu_i^{(s)}$ and $\mu_j^{(s)}$ are the respective expected scales of viewlets V_i and V_j.

Gaussian Markov Random Field Model (GMRF). A Markov random field is a set of random variables having a Markov property described by an undirected graph. In particular, each random variable node is independent of the rest of the random variables given its neighbors in the network. If the joint distribution is Gaussian, with joint covariance matrix, Σ, then the GMRF specifies the zero patterns of the precision matrix $\Lambda = \Sigma^{-1}$: $\Lambda_{ij} = 0$ implies that the corresponding random variables are conditionally independent and hence does not have an edge in the GMRF. For quadratic-form potential functions (as in the above equations), we can regard our spring model as a GMRF. The precision matrix $\Lambda^{(x)}$ can be calculated by noting that $\Lambda_{ij}^{(x)}$ is the coefficient of the product terms $(x_i - \mu_i^{(x)})(x_j - \mu_j^{(x)})$ in the exponent of Eq. **1**:

$$\Lambda_{ii}^{(x)} = \sum_{j=1, j\neq i}^{M-1} \frac{c_{ij}^{(x)}}{(s_i^{(x)}+s_j^{(x)})^2} + \frac{c_{iM}^{(x)}}{(s_i^{(x)}+s_M^{(x)})^2}, \quad [4]$$

$$\Lambda_{ij}^{(x)} = -\frac{c_{ij}^{(x)}}{(s_i^{(x)}+s_j^{(x)})^2} \quad i\neq j, \quad [5]$$

where without loss of generality, we have assumed $X_M = 0$ to reduce the degree of freedom to $M-1$ (see *SI Appendix*, section 3). Note that if $c_{ij} = 0$, then $\Lambda_{ij} = 0$, and we know from the properties of multivariate Gaussian distributions that the corresponding location variables are conditionally independent. Now Eq. **1** can be written as a log-likelihood function:

$$\mathcal{L}(X) = \frac{1}{2}\log r|\Lambda| - \frac{1}{2}\sum_{i\neq j} c_{ij}^{(x)}\left(\frac{x_i - x_j}{s_i^{(x)}+s_j^{(x)}} - \mu_{ij}^{(x)}\right)^2, \quad [6]$$

where $|\Lambda|$ is the determinant of the precision matrix and r is the normalization constant for a Gaussian distribution.

Sparsity and Conditional Independence. The direct interactions (i.e., for which $c_{ij} > 0$), combined with node set, V, form the SRN network, $G(V, E)$, and hence, the model complexity of the SRN corresponds to its sparsity. Sparsity has a physical meaning in our model: For most physical objects, locations of parts and the resulting views are indeed not statistically fully connected with each other. Equivalently, a sparse set of springs are enough to constrain deformations in exemplars. We impose such sparsity constraints in the learning process, and a relaxation leads to a convex optimization problem that can be solved efficiently.

Semantic structure. As already explained in the introduction, we use two complementary constructs to capture the semantic structure of the object—namely, the CIPC and the GPE. The specifics of these two constructs are made precise in *Learning SUVMs*. For now, it suffices to mention that together they provide a description of the object prototype in terms of parts (each part being a grouping of viewlets), their locations, and the inclusion/overlap relationships among the viewlets and parts.

Generative model and calculating object likelihoods. An SUVM defined by its parameter set $\theta = \left(\{c_{ij}^{(x)}\}, \{c_{ij}^{(y)}\}, \{c_{ij}^{(s)}\}, \{\mu_i^{(x)}\}, \{\mu_i^{(y)}\}, \{\mu_i^{(s)}\}\right)$ that specifies the SRN and the accompanying CIPC and GPE models is the key representational and generative tool we use. As a generative model, any exemplar can be viewed as being created by a four-step process: (*i*) first, picking the parts or regions that are to be rendered in the exemplar from the CIPC and GPE, let D_p be the set of parts that is picked. (*ii*) Then picking N_G viewlets, numbered $1, \ldots, N_G$, that go together for the picked parts; for example, for configurable parts, certain viewlets are mutually exclusive and should not be picked together. Let V_G be the set of picked viewlets. The probability $P(V_G|\theta)$ is stated in *SI Appendix*, section 3, where we provide a detailed description of our detection algorithms. For each picked viewlet, $V_i \in V_G$, an appearance feature vector A_i is drawn by sampling its appearance distribution, and a corresponding image patch is created; let $A = \{A_1, \ldots, A_{N_G}\}$. (*iii*) Then choosing scaling factors by sampling the joint scale distribution (Eq. **3**), let $S = \{S_1, \ldots, S_{N_G}\}$, and finally (*iv*), locating these N viewlets spatially by sampling the joint distribution specified by the SRN (Eqs. **1** and **2**), let $X = \{x_1, \ldots, x_{N_G}\}$ and $Y = \{y_1, \ldots, y_{N_G}\}$ be the set of these location coordinates. Note that in our model, given S and V_G, X and Y are picked independently.

Each step has its own likelihood, allowing us to calculate the likelihood of any such generated exemplar:

$$\begin{aligned} &P(\text{Generated Exemplar}|\theta) \\ &\quad = P(A, X, Y, S, V_G|\theta) \\ &\quad = P(Y|S, V_G, \theta) \times P(X|S, V_G, \theta) \\ &\qquad \times P(A|V_G, \theta) \times P(S|V_G, \theta) \times P(V_G|\theta)\,. \end{aligned} \quad [7]$$

Learning SUVMs. Given a learning dataset comprising unlabeled instances of the unknown category, we need to construct an SUVM—that is, appearance feature vectors of the viewlets, the SRN, and the semantic structures CIPC and GPE.

Learning a visual dictionary. In this step, we determine a set of visual words, or a dictionary, from the given images. We randomly sample all images in the learning set using a scale pyramid and using a fixed-size rectangular patch, then convert all patches into image feature vectors, and then extract a visual vocabulary out of them using an unsupervised clustering algorithm. Note that each visual word is a cluster of feature vectors and the visual dictionary naturally comes with a classification algorithm. For example, for *k*-means clustering, one can use the kNN algorithm to assign a word label to any candidate image patch. Also note that a visual word represents only a potential viewlet in the object models to be extracted from the learning set.

A maximum likelihood (ML) framework for learning SRNs. We have already simplified our model in the form of a sparse network, or a GMRF, which can be determined by an edge set E, and the related parameters $\left(\{c_{ij}^{(x)}\}, \{c_{ij}^{(y)}\}, \{c_{ij}^{(s)}\}, \{\mu_i^{(x)}\}, \{\mu_i^{(y)}\}, \{\mu_i^{(s)}\}\right)$. Since we have already created a set of visual words, we first go back to the original image corpus (e.g., in the celebrity dataset, 9,638 images are in the learning set; see *Results*) and detect in each image the visual words that appear in it. That is, in every image, we first perform a dense scan (using a scaling pyramid so that we capture viewlets that have inherently larger scale) with a fixed-size sliding window (the same size as used to determine the visual dictionary) and assign a visual word to each resulting patch using a kNN algorithm. Then, for each detected visual word, V_i, we have its size in pixels $(s_i^{(x)}.s_i^{(y)})$ and its location coordinates (x_i, y_i). For a pair of visual words,

V_i and V_j, detected in the same image, we have samples of the SUV model outputs: $Z_{ij}^{(s)} = \frac{S_j}{S_i} = \frac{s_j^{(x)}}{s_i^{(x)}} = \frac{s_j^{(y)}}{s_i^{(y)}}$, $Z_{ij}^{(x)} = \frac{(x_j - x_i)}{(s_i^{(x)} + s_j^{(x)})}$, and $Z_{ij}^{(y)} = \frac{(y_j - y_i)}{(s_i^{(y)} + s_j^{(y)})}$. We need to infer now an edge set E and the related spring parameters such that the data likelihood, as captured by Eqs. **1–3**, is maximized. For example, to estimate the set of parameters $\{c_{ij}^{(x)}\}$, it follows from preceding discussions on GMRF that the empirical likelihood function is given by $\log P(X) = \text{const} + \frac{1}{2}\log|\Lambda| - \frac{1}{2}\sum_{i \neq j} c_{ij}^{(x)} \text{Var}(Z_{ij})$, where $\text{Var}(Z_{ij})$ is the empirically observed variance of the random variable Z_{ij}.

Approximate Sparse Estimation of c_{ij}s. To maximize $\log P(X)$ while making c_{ij}s sparse, we reverse the sign to get a minimization problem and add an L_1 regularization term to obtain $\mathcal{L}(X) = -\frac{1}{2}\log|\Lambda| + \frac{1}{2}\sum_{i \neq j} c_{ij}(\text{Var}(Z_{ij}) + \lambda)$, where $\lambda > 0$ is the regularization parameter and $c_{ij} \geq 0$. This is a convex optimization problem and can be solved efficiently. Staying true to our spirit of performing simple computations, we analyze this convex optimization problem, and using the Karush–Kuhn–Tucker (KKT) conditions, we prove an upper bound on the optimal values of c_{ij}: $c_{ij}^* \leq \frac{1}{\text{Var}(Z_{ij}) + \lambda}$ (see *SI Appendix*, section 3). Thus, c_{ij}^* decreases monotonically with increases in both the observed variance, $\text{Var}(Z_{ij})$, and the sparsity parameter, λ. This bound then leads to an approximate but efficient algorithm to directly impose sparsity: If we say that all those edges for which the optimal c_{ij}^* is less than say a target value of c will be removed from the network, then it implies from the above equation that all edges with empirical $\text{Var}(Z_{ij}) > \frac{1}{c} - \lambda$ should be disconnected or their corresponding $c_{ij} = 0$. Thus, we have derived a simple threshold rule on the pairwise variances, and by lowering the threshold (that is, by increasing the sparsity parameter λ), we get increasingly sparse SRNs. In our implementation, we defined a combined variance for an edge $V = \text{Var}(Z_{ij}^x) + \text{Var}(Z_{ij}^{(y)}) + \text{Var}(\log Z_{ij}^{(s)})$ and then imposed a threshold on it until the SRN is sparse enough. Note that as the edge set becomes sparse, the initial network, comprising all visual words, gets disconnected and the giant connected components correspond to the SRNs of the underlying object categories.

Extracting parts using CIPC. We first point out two ways in which a part in the object category gets encoded in terms of viewlets and their structure in our model: (*i*) two or more viewlets that are replaceable in making up the whole object or, equivalently, two or more viewlets that are mutually exclusive (in terms of co-occurrence and hence shares no edge in the SRN) and yet have nearly identical geometrical relationships with other viewlets (representing other parts). Pairs of such nodes/viewlets can be identified efficiently by processing the SRN (e.g., by sequentially examining each viewlet node and finding other nodes that are not connected to it by an edge but share neighbors in common). (*ii*) Viewlets that share a very stable edge in the SRN between them and have the same geometrical relationships with viewlets corresponding to other parts of the object. This scenario arises when two viewlet nodes are only slightly shifted versions of each other, representing the persistent presence of a part in the object. Again such pairs can be efficiently detected from the SRN. We construct a CIPC network, where each pair of viewlets, satisfying type *a* or *b* relationship, is connected by an edge. Each connected component in the resulting CIPC network then corresponds to a distinct configurable and stable part of the underlying object category. The results shown in Fig. 1 and in *SI Appendix*, section 8 demonstrate the effectiveness of this methodology.

Extracting semantic structure using GPE. In this step, we use the pairwise scale and location relationships to embed the viewlets in the SRN in a 3D space: Each viewlet V_i is assigned an absolute 2D position $(x(i), y(i))$ and scale $S(i)$ such that the pairwise constraints obtained from data are best satisfied. We use a mean squared error-based optimization function, similar to that used in multidimensional scaling (MDS) (32) and derive an iterative approach to calculate these mean positions and scales of viewlets (see *SI Appendix*, section 4).

From Fig. 1, we notice that viewlets, clustered by CIPC as belonging to the same part, have very similar global spatial values and cluster together in the GPE. Our ability to reverse-engineer human body parts, for example, demonstrates that we are able to identify the semantic structure of objects automatically, instead of hand-coding such knowledge via manual tagging.

From Models to Detection. We start with a dense scanning of the given image using a scale pyramid and obtain N patches; note that N can easily be in the thousands. Next, we design complementary algorithms for two different tasks for any given image: (*i*) task 1: detection and localization of object instances (e.g., cars, humans), and (*ii*) task 2: detection and localization of specific parts (e.g., human head/face or torso) of a learned object category. In both cases, there could be multiple occurrences in the same image. For task 1, following the probabilistic interpretation introduced in Eq. **7**, we do the following search: (*i*) We map each image patch to a visual word and consider only those that are mapped to viewlets in the SUVM, and then (*ii*) we group the viewlet patches into clusters so that each cluster of patches maximizes the likelihood of representing an object instance, as given Eq. **7**. This can be accomplished via an exponential search over all possible assignments of image patches to visual words (15, 18, 24). In accordance with our neuroscience inspirations, however, we do a restricted search and we consider an object to be detected if sufficiently many parts (as determined by CIPC and GPE) that match the relative distances and scale requirements are detected with high confidence. Thus, we look at all of the detected viewlets and then start grouping them together based on whether they structurally match our model. This linear time (in N and n_f) heuristic search algorithm is agglomerative in nature rather than exhaustive, making it highly scalable. Moreover, this natural detection framework allows one to find multiple occurrences of objects in the same image efficiently. The details are given in *SI Appendix*, section 6.

For task 2, where the goal is to detect and localize targeted parts of learned object instances, we again use the structure of the underlying SUVM to compute a geometric mapping between any given pair of viewlets, V_i and V_j (see *SI Appendix*, section 6 for details). Thus, a reliable target part and its location are detected by mapping multiple detected viewlets to where the part should be.

ACKNOWLEDGMENTS. The authors thank Prof. Lieven Vandenberghe for his input on the optimization formulations used in the paper and the referees for helpful suggestions and especially for pointing us to relevant prior work.

1. Bengio Y (2009) Learning deep architectures for AI. *Machine Learn* 2:1–127.
2. Hinton GE (2007) Learning multiple layers of representation. *Trends Cogn Sci* 11:428–434.
3. LeCun Y et al. (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1:541–551.
4. Arel I, Rose D, Karnowski T (2010) Deep machine learning-A new frontier in artificial intelligence research. *IEEE Comput Intell Mag* 5:13–18.
5. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Machine Intell* 35:1798–1828.
6. Lowe D (1999) Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Available at https://ieeexplore.ieee.org/document/790410. Accessed November 28, 2018.
7. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Available at https://ieeexplore.ieee.org/document/1467360. Accessed November 28, 2018.
8. Cauwenberghs G (2013) Reverse engineering the cognitive brain. *Proc Natl Acad Sci USA* 110:15512–15513.

9. Ullman S, Assif L, Fetaya E, Harari D (2016) Atoms of recognition in human and computer vision. *Proc Natl Acad Sci USA* 113:1–6.
10. Logothetis NK, Sheinberg DL (1996) Visual object recognition. *Annu Rev Neurosci* 19:577–621.
11. Desimone R, Albright TD, Gross CG, Bruce C (1984) Stimulus-selective properties of inferior temporal neurons in the macaque. *J Neurosci* 4:2051–2062.
12. Biederman I, Cooper EE (1992) Size invariance in visual object priming. *J Exp Psychol Hum Perception Perform* 18:121–133.
13. Fischler MA, Elschlager RA (1973) The representation and matching of pictorial structures. *IEEE Trans Comput* 22:67–92.
14. Fergus R, Perona P, Zisserman A (2003) Object class recognition by unsupervised scale-invariant learning. *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Available at https://ieeexplore.ieee.org/document/1211479. Accessed November 28, 2018.
15. Fergus R, Perona P, Zisserman A (2007) Weakly supervised scale-invariant learning of models for visual recognition. *Int J Comput Vis* 71:273–303.
16. Sivic J, Russell BC, Efros AA, Zisserman A, Freeman WT (2005) Discovering objects and their location in images. *International Conference on Computer Vision*. Available at https://ieeexplore.ieee.org/document/1541280. Accessed November 28, 2018.
17. Cho M, Kwak S, Schmid C, Ponce J (2015) Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. *IEEE Conference on Computer Vision and Pattern Recognition*. Available at https://ieeexplore.ieee.org/document/7298724. Accessed November 28, 2018.
18. Felzenszwalb P, McAllester D, Ramanan D (2008) A discriminatively trained, multiscale, deformable part model. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (IEEE). Available at https://ieeexplore.ieee.org/document/4587597. Accessed November 28, 2018.
19. Bourdev L, Malik J (2009) Poselets: Body part detectors trained using 3D human pose annotations. *2009 IEEE 12th International Conference on Computer Vision*. Available at https://ieeexplore.ieee.org/document/5459303. Accessed November 28, 2018.
20. Chen X, et al. (2014) Detect what you can: Detecting and representing objects using holistic models and body parts. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Available at https://ieeexplore.ieee.org/document/6909651. Accessed November 28, 2018.
21. Chang LB, Jin Y, Zhang W, Borenstein E, Geman S (2011) Context, computation, and optimal ROC performance in hierarchical models. *Int J Comput Vis* 93:117–140.
22. Yang Y, Ramanan D (2013) Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal Machine Intell* 35:2878–2890.
23. Geman D, Geman S, Hallonquista N, Younes L (2012) Visual Turing test for computer vision systems. *Proc Natl Acad Sci USA* 112:3618–3623.
24. Felzenszwalb PF, Girshick RB, McAllester D (2010) Cascade object detection with deformable part models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE). Available at https://ieeexplore.ieee.org/document/5539906. Accessed November 28, 2018.
25. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Available at https://ieeexplore.ieee.org/document/990517. Accessed November 28, 2018.
26. Ullman S, Harari D, Dorfman N (2012) From simple innate biases to complex visual concepts. *Proc Natl Acad Sci USA* 109:18215–18220.
27. Kang H, Hebert M, Efros AA, Kanade T (2012) Connecting missing links: Object discovery from sparse observations using 5 million product images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (LNCS). Available at https://link.springer.com/chapter/10.1007%2F978-3-642-33783-3_57. Accessed November 28, 2018.
28. Hu P, Ramanan D (2017) Finding tiny faces. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Available at https://ieeexplore.ieee.org/document/8099649. Accessed November 28, 2018.
29. Gong B, Grauman K, Sha F (2017) Geodesic flow kernel and landmarks: Kernel methods for unsupervised domain adaptation. *Advances in Computer Vision and Pattern Recognition*. Available at https://link.springer.com/chapter/10.1007%2F978-3-319-58347-1_3. Accessed November 28, 2018.
30. Goodfellow I, et al. (2014) Generative adversarial nets. *Advances in Neural Information Processing Systems* 27: 2672–2680.
31. Donahue J, et al. (2015) Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Available at https://ieeexplore.ieee.org/document/7558228. Accessed November 28, 2018.
32. Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29:1–27.