Tinkering Under The Hood: Interactive Zero-Shot Learning with Pictorial Classifiers

Vivek Krishnan

CMU-CS-16-123

August 2016

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Deva Ramanan, Co-Chair
Kayvon Fatahalian, Co-Chair

Submitted in partial fulfillment of the requirements for the degree of Master of Philosophy.

Keywords: convolutional neural networks, knowledge transfer, weak supervision, zero-shot learning, visualization, internal semantics, dimensionality reduction, deformable part models, object detection, image classification

Abstract

We consider the task of visual zero-shot learning, in which a system must learn to recognize concepts omitted from the training set. While most prior work make use of linguistic cues to do this, we do so by using a *pictorial language* representation of the training set, implicitly learned by a CNN, to generalize to new classes. We first demonstrate the robustness of pictorial language classifiers (PLCs) by applying them in a weakly supervised manner: labeling unlabeled concepts for visual classes present in the training data. Specifically we show that a PLC built on top of a CNN trained for ImageNet classification can localize humans in Graz-02 and determine the pose of birds in PASCAL-VOC without extra labeled data or additional training. We then apply PLCs in an interactive zero-shot manner, demonstrating that pictorial languages are expressive enough to detect a set of visual classes in MSCOCO that never appear in the ImageNet training set.

Acknowledgments

I am grateful to my advisor Deva Ramanan for mentoring me over the year. He has always been generous with his time and mindful of the challenges facing a student who had just started working in a research setting. In addition, I would like to thank Kayvon Fatahalian for steering me toward a career in computer vision / machine learning (learning GPU programming in 418 has a wide application) I would also like to thank the members of my lab group for sitting through multiple presentations and for reviewing my paper before submission.

Contents

Chapter 1

Introduction

Convolutional Neural Networks (CNNs) have revolutionized modern vision pipelines by replacing hand-crafted features with large set of trainable parameters that learn complex image representations from data. However, these models notoriously require a substantial amount of annotated training data to learn a feature hierarchy that generalizes well. As a result, subsequent performance gains in various visual tasks have been limited by the availability of large datasets such as ImageNet Deng et al. [2009].

Transfer / adaptation: A practical question is how to train such pipelines with *small* training data, readily encountered in such applications as robotics Atkeson and Schaal [1997] and biological image analysis Shamir et al. [2010]. An influential approach is that of transferring knowledge from a data-rich task through fine-tuning a pre-trained model Oquab et al. [2014]. Indeed, virtually all state-of-the-art vision systems make use of models pre-trained on ImageNet Girshick et al. [2014], Yosinski et al. [2014]. A related but distinct formalism is that of domain adaptation, where a pre-trained model is adapted to a novel domain with different statistics from test data Ben-David et al. [2007], Saenko et al. [2010] or entirely different modalities Glorot et al. [2011], Tzeng et al. [2015]. However, in all such systems, some amount of target data for the task of interest is needed to provide a gradient-based signal for learning.

Zero-shot learning: In this work, we consider an extreme form of the problem where *no* labeled data for the target task is available Socher et al. [2013]. In vision, a classic example is recognizing a never-before-seen object category Farhadi et al. [2009], Lampert et al. [2009], Rohrbach et al. [2011], Mensink et al. [2013]. Importantly, some form of a *semantic knowledge base* Palatucci et al. [2009] must be used to specify the relationship of the novel category to labeled concepts encountered during training. For example, one may recognize an instance of a never-before-seen `whale` category given the semantic

description that it is similar to a known animal (such as a `shark`) or that it is composed of known attributes/parts (such as `fins`). The vast majority of past work makes use of *linguistic* semantics, mined from text corpora Fu et al. [2015] or lexical knowledge bases such as Wordnet Miller [1995].

Mental imagery: Our work differs from past zero-shot approaches in that we do *not* make use of linguistic cues. Cognitive psychology suggests that many concepts in one's associative memory may be better represented through visual iconography rather than linguistic tokens Paivio [1969]. Similarly, it is well-known in developmental psychology that children learn spatial concepts visually rather than linguistically Bowerman [1996] Indeed, prior work on image retrieval has shown that sketch-based interfaces can be more effective than textual input for certain queries Antol et al. [2014]. Inspired by such approaches, we demonstrate that CNN visualizations can be used to define a *pictorial language* for zero-shot learning.

Visualizations: Our pictorial approach builds on a body of work that aims to visualize and understand internal representations in CNNs. Common visualization techniques include reconstructing image filters (by deconvolution Zeiler and Fergus [2014]), inverting feature transformations Mahendran and Vedaldi [2015], or computing the set of image regions that activate particular neurons Girshick et al. [2014], Zhou et al. [2014]. Our work differs in that we use a simple grammar to compose neurons together to build models for never-before-seen-objects (such as a "striped aeroplane"). Importantly, our visual grammars allow a user to build zero-shot classifiers composed out of visual tokens that may not have a natural linguistic "name". Our grammars can be seen as spatial part models Girshick et al. [2015], Zhu and Mumford [2007] where part responses are equivalent to individual neural activations or user-defined nonlinear functions of activations (computed through embeddings).

Internal semantics: From another perspective, our work addresses a growing concern that deep networks function as non-interpretable "black boxes". One specific criticism is that while such models work extraordinarily well on data similar to the training set, it is hard to characterize behavior on new data distributions (the "dataset bias" problem Torralba and Efros [2011]). Shalev-Shwartz and Shashua Shalev-Shwartz and Shashua [2016] argue that deep representations learned without internal semantics cannot suffice for high-accuracy tasks such as autonomous navigation, because rare but important scenarios (such as cars running a red light) may be difficult to observe in a finite training set. Our work demonstrates that interpretable semantics can be placed on internal modules *post-hoc*, but crucially, such semantics are pictorial rather than linguistic. One might not be able to "name" what neurons represent, but one can still reconfigure them for novel tasks and scenarios that were not previously encountered.

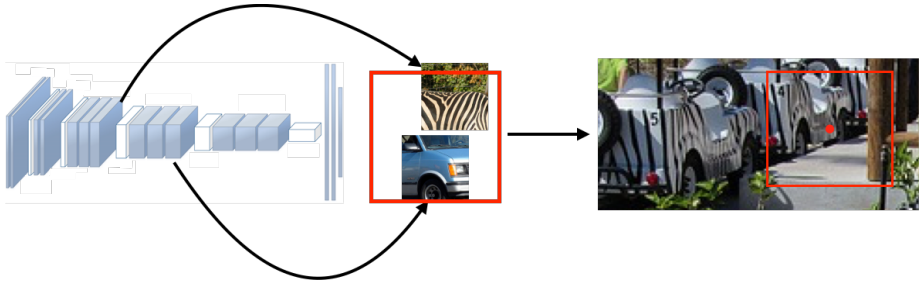

Figure 1.1: We introduce a pictorial language for visualizing, understanding, and *re-wiring* internal representations learned by CNNs to detect never-before-seen concepts, such as a "striped car".

Overview: We begin with a review of techniques for visualizing deep networks (Sec. 2). We describe a pictorial grammar that uses such visualizations to repurpose existing deep networks for novel tasks (Sec. 3). We use these techniques in Sec. **??** to repurpose a network trained for image *classification* (Krizhevsky et al. [2012]) for object *detection*, including never-before-seen objects.

Chapter 2

Visualizing Deep CNNs

In this section, we describe our approach for visualizing and understanding CNNs. Though our methods directly apply to any existing network, we illustrate them on our running example of AlexNet. Our methods fundamentally require both a pre-trained CNN and a collection of validation images with which to process the CNN and diagnose its behaviour. We use the ILSVRC 2012 validation set of 50,000 images Russakovsky et al. [2015]. Importantly, this set is distinct from the test images we use for evaluation.

2.1 Retinotopic embeddings

The heart of our approach requires the ability to interpret the semantics of particular neurons, where neurons correspond to individual activations computed throughout the layers of a CNN. Note that it is not clear that this is even possible; indeed, the well-known "grandmother neuron" hypothesis that individual neurons represent semantic concepts (such as one's grandmother) seems at odds with notion that deep networks are distributed (rather than localized) representations Bengio [2009]. We will shortly offer a solution to this apparent contradiction.

Following established convention, we will denote activations by their layer, such as `conv3`, `conv4`, etc. While numerous techniques have been proposed for understanding the semantics of such neurons, our starting point is a surprisingly simple strategy: characterize a neuron by the *set* of N image patches that maximally activate that particular neuron. Such an approach was popularized by Girshick et al. [2014], Zhou et al. [2014]. We show examples for $N = 6$ for a variety of neurons across different layers in Fig. 2.1; one can readily identify neurons corresponding to different textures, parts, and objects.

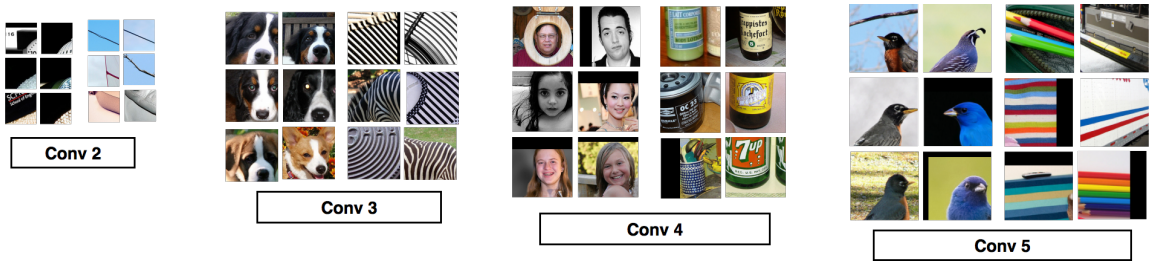

Figure 2.1: Visualization of image features in the VGG_CNN_S model. For layers 2-6, we show the top 6 activations taken from the ILSVRC 2012 Russakovsky et al. [2015] validation set. The size of the image patch corresponds to the theoretical RF of the neuron as computed in Long et al. [2014], Girshick [2015]. We see that (1) a CNN trained for object classification learns a rich, hierarchical set of features, ranging from textures to complex invariances such as pose. (2) Neurons can learn to identify and localize unlabeled concepts (such as face). (3) These images of faces belong to a wide range of classes including Helmet, French Horn, Suit, Bathtub, and Park Bench. Even though there is no category on ILSVRC 2012 for generic human detection, the CNN has automatically determined through correspondence that detecting human faces is useful.

Neurons in lower layers are characterized by smaller receptive fields (defined as the set of pixel values that will affect a given neural activation). Neurons in deeper layers tend to be invariant to various factors such as pose, but perhaps surprisingly, strong semantics appear even at shallower layers (e.g., the dog-face neuron in `conv3`).

Embeddings: Because neurons at higher layers tend to be more invariant, we would like to understand the *variation* in the appearance of image patches that activate a particular neuron. Recall that a neuron activation a is computed from previous layer responses with a (convolutional) linear dot product with weights w and bias b, followed by a (ReLU) nonlinearity:

$$a = \max(0, w \cdot x + b) \tag{2.1}$$

where x corresponds to a local neighborhood of activations from the previous layer. For concreteness, if a was from `conv3`, x is of dimensionality $3 \times 3 \times 256$. We propose to view $x \in \mathcal{X}$ as a point in a $\mathcal{X} = R^{2304}$ dimensional embedding space, where the activation a is given by a linear threshold function in $\mathcal{X}$. We posit that this embedding space is a rich characterization of the invariances learned by this particular neuron - e.g., if neuron a corresponds to a view-invariant object, then different points in $\mathcal{X}$ may correspond to different viewpoints of that object.

Past work: The idea of viewing neural activations as embeddings is certainly not

new. Training deep nets to predict embedding coordinates is a well-studied task Hinton and Salakhutdinov [2006]. Moreoever, it is widely known that one can interpret the final (FC7) activation layer as an embedding, even for nets originally trained for classification Devlin et al. [2015]. We point out that such a perspective also applies to local neighborhoods of activations from intermediate layers, which might offer a solution to the localized (grandmother) versus distributed debate: interpretable semantics might reside in localized neighborhoods of activations. Because such neighborhoods share similar receptive fields (due to the convolutional structure of weights), we refer to such embeddings as *retinotopic embeddings*. This characterization of deep nets as learning progressively invariant embeddings may agree with perspectives of such models as "disentagling" factors of variation from one layer to the next Bengio [2009].

Dimensionality reduction: We use the above observation to visualize N high-scoring patches for a particular neuron. By viewing each patch as a point in $\mathcal{X}$, we can visualize a collection using low-dimensional projections (Fig. 2.2). T-SNE Van der Maaten and Hinton [2008] is a popular technique for visualizing *nonlinear* projections that preserves neighborhood structure in the high-dimensional space. We also experimented with simpler *linear* projections obtained through PCA, which also allows us to visualize neuron a as a 2D line given by the linear threshold function (w, b). Formally speaking, let us write the 2-dimensional projected coordinates of a point x as $(c_1, c_2) = (v_1 \cdot x, v_2 \cdot x)$, where (v_1, v_2) are projection vectors given by PCA. Given that we can approximately reconstruct x from its projection with $x = c_1 v_1 + c_2 v_2$, the linear threshold function can be written as $wx + b = c_1 v_1 \cdot w + c_2 v_2 \cdot w + b$, or a 2D line with a normal vector $(v_1 \cdot w, v_2 \cdot w)$ and offset b. This line is visualized in Fig. 2.2.

User-drawn concepts (PCA): Our embedding visualizations can be used to define *new* linear threshold boundaries, corresponding to user-defined visual concepts. In the case of PCA-embeddings, these user-drawn linear thresholds can be back projected to define new filters and biases. Formally, given a user-drawn line with normal vector (α_1, α_2) and offset β, the corresponding high-dimensional filter producing the same response is given by $\alpha_1 c_1 + \alpha_2 c_2 + \beta = \alpha_1 v_1 \cdot x + \alpha_2 v_2 \cdot x + \beta = w' \cdot x + b'$ where $w' = \alpha_1 v_1 + \alpha_2 v_2$ and $b' = \beta$. We will show examples of zero-shot models built with user-defined filters in our experimental results.

User-drawn concepts (TSNE): Reconstructing the filter from a TSNE embedding is much more difficult. TSNE does not explicitly compute an embedding function, but rather directly outputs an embedding of a fixed set of input points (implying that the embedding cannot be applied to "out-of-sample" points). P-TSNE Maaten [2009] is a parametric extension that essentially trains a feedforward neural net to predict the TSNE embedding obtained for a fixed set of inputs - e.g., $(c_1, c_2) = (f_1(x), f_2(x))$. Once this function

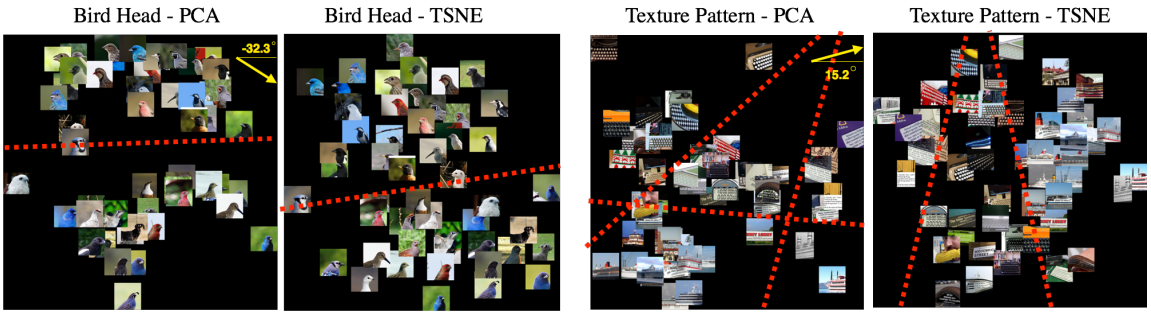

Figure 2.2: We visualize pairs of (PCA,TSNE) embedding of two `conv5` neurons, along with user-drawn linear boundaries (dotted red lines). For PCA, we also visualize the direction w (the yellow arrow) that would maximally excite the given neuron, rather than the linear threshold boundary which lies outside the image. For the **left** pair, we clearly observe a linear separator in the embeddings between bird heads with different orientations. The **right** illustrates a grouping of texture patterns that may not be easily described linguistically, but appear to loosely correspond to keyboard, cruise ship, and text.

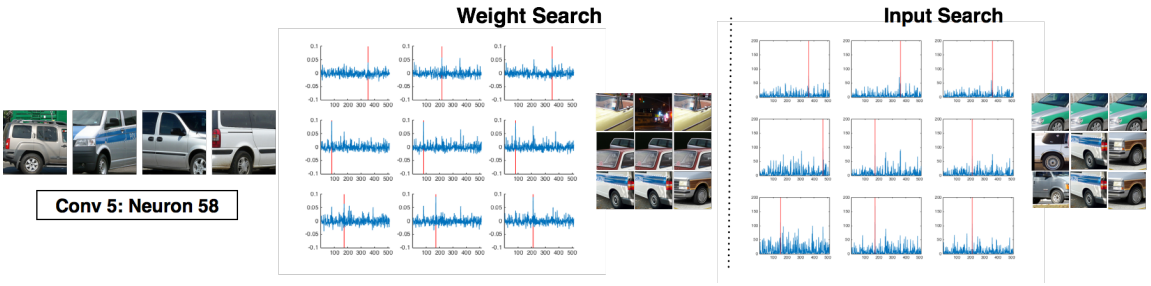

Figure 2.3: We illustrate two search methods based on neuron 58 from `conv5`, which fires on images of vehicles. We provide the 4 image patches in the ILSVRC validation set that maximally activate the neuron (left). Weight Search (**left**) consists of finding the maximum weight for each spatial location in the filter. We plot the learned weight values for each filter and indicate the maximum filter weight with a red line. Input Co-occurence Search (**right**) consists of averaging the input features over the top-100 activations of the neuron. Here we plot the averaged input for each filter and highlight the maximum with a red line. For each search method, we also show the top image at each spatial location corresponding to the retrieved `conv4` filters.

is learned, it can be applied to new "out-of-sample" patches. Because the function is nonlinear, halfspaces in the embedded space do not correspond to simple linear filters in $\mathcal{X}$. However, one can still "implement" a user-drawn halfspace by explicitly computing

the feedforward embedding on any query patch, and linearly scoring the resulting 2D projection: $a_{TSNE} = \max(\alpha_1 f_1(x) + \alpha_2 f_2(x) + \beta, 0)$. Note that this same approach could also apply for PCA; instead of reconstructing the original filter, one could compute $a_{PCA} = \max(\alpha_1 v_1 \cdot x + \alpha_2 v_2 \cdot x + \beta, 0)$, which might be ter if processing many user-defined concepts obtained from the same embedding.

2.2 Searching for semantics

The previous section describes an approach for uncovering the semantic concepts of individual neurons, as well as constructing new user-defined concepts. In theory, one could apply the given approach in an exhaustive fashion over all neurons to uncover semantics of interest, but methods to filter the set of neurons can be used to reduce the search time.

We envision two modes of user interaction for our zero-shot pipeline. A user may have a semantic concept in mind ("I want to build a model of 3-wheeled vehicles"), or users may wish to puruse internal concepts captured in a network in an exploratory fashion. In either case, the user will identify a number of visual concepts which will then be assembled into a pictorial grammar (Sec. 3). In order to identify a pool of relevant candidates, a successful pipeline appeared to be first identifying an "interesting" neuron (by random search at layers in `conv5` or higher), and then searching for related neurons as detailed below.

Filter search: The first method utilizes the filter weights learned by the CNN. Recall that activations are computed by linear threshold functions (2.1) where the inputs x themselves must be non-negative (since they are rectified activations themselves). This implies that the magnitude of weight of each input is directly related to the final activation value of a neuron, as well as its importance in positive feature detection for the neuron. Therefore, given a neuron of interest a, we search for overlapping neurons at higher and lower levels associated with large weights w.

Co-occurrence search: Our second method empirically searches for collections of neurons that consistently activate (over a set of images). Given a particular neuron of interest, we compute all non-zero activations on a validation set, and record the average activation of all overlapping neurons in higher and lower layers. We construct a pool of overlapping neurons (ordered by their filter weight or empirical correlation) that serve as candidates for embedding visualizations. We show examples of successfully-identified related neurons in Fig. 2.3.

Chapter 3

Pictorial Language Classifiers

Our previous sections describe an interface for identifying visual tokens of interest. Examples might include textures, part decompositions, and object invariances, that together can represent a wide range of complex objects. In this section, we construct a simple visual grammar for composing these tokens into flexible models for object detection. Importantly, the final visual grammar can be executed using standard CNNs (convolution and max-pooling) implying that the user-constructed models can be efficiently implemented as additional add-on layers to AlexNet. Since the vast majority of computation will be shared, this means that user-defined object detection essentially comes "for-free".

3.1 Model

Our object detection model is based off the deformable parts model (DPMs) described in Felzenszwalb et al. [2010]. In particular, we exploit the insight in Girshick et al. [2015] that DPMs can be implemented in CNN toolboxes. We revisit the derivation with slightly more detail here, pointing out that general tree-structured grammars can be implemented in CNNs. This allows, for example, users to build zero-shot detectors for objects (e.g., a "striped car") as well as spatial arrangements of objects (e.g., "a striped car next to two red bicycles")". Let us write the pixel location of part i as $z_i = (x_i, y_i)$, and the score of a collection of parts $z = \{z_i\}$ as follows:

$$\text{score}(z) = \sum_i \phi(z_i) + \sum_{j \in parent(i)} \psi(z_j - z_i - a_j) \tag{3.1}$$

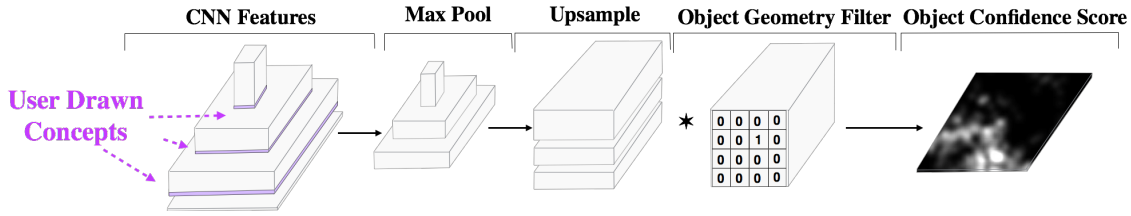

Figure 3.1: We efficiently implement pictorial grammars by augmenting Alexnet with additional user-defined concept filters (from Sec. 2), max-pooling responses, upsampling responses across different layers to the same pixel resolution, and summing max-pooled response maps shifted by the anchor location of a part with respect to its parent (implemented as a sparse object geometry filter Girshick et al. [2015]).

The local function $\phi(z_i)$ denotes the score associated with placing part i at pixel location i, which is found by evaluating (a possibly-upsampled) activation map.The pairwise function $\psi(z_j - z_i - a_j)$ encodes a spatial model that denotes valid relative locations of part j to its parent i, where its rest anchor location is given by a_j. We use a simplified variant where $\psi(z_j - z_i - a_j) = 0$ if $||z_j - z_i - a_j||_\infty \leq r$ and $-\infty$ otherwise. It is well known that the best-scoring configuration of parts can be found with dynamic programming for tree-structured spatial constraints. By using the particular spatial model given above, partial scores of each part can be efficiently computed by repeating the following updates from the leaf to the root:

$$\text{score}_i(z_i) = \phi(z_i) + \sum_{j \in kids(i)} m_j(z_i + a_j), \quad \text{where} \quad m_j(z_i) = \max_{\{z_j : ||z_j - z_i||_\infty \leq r_j\}} \text{score}_j(z_j)$$

where $\text{score}_i(z_i)$ is initialized to $\phi(z_i)$ for leaf parts and $\text{score}_i(z_i)$ represents the true max-marginal score for the root part. Note that $m_j(z_i)$ is computed by max-pooling while $\text{score}_i(z_j)$ is computed by summing up shifted response maps (which can be implemented with a sparse, binary convolution operation). This implies that tree-structured dynamic programming operations can be implemented as standard layers in a CNN (Fig. 3.1).

User-defined spatial models: Though a tree-structured formulation is flexible, it may be noninuitive for a user to manually specify. For that reason, we limit our results to "star" structured models that consist of a single root filter and a collection of child filters. We use an empty root filter of a fixed size $H \times W$, which specifies the bounding box dimension of a candidate detection. We allow users to specify the a_j and r_j, which corresponds to the "rest" location of part within the bounding box, and the amount of valid displacements about that location. To find objects of different sizes, we process an image pyramid.

Model	aP (IOU $\geq$ 0.3)	aP (IOU $\geq$ 0.5)	aP (IOU $\geq$ 0.8)
PLC-Face	0.152	0.123	0.0015
PLC-Lower Body	0.412	0.036	0.006
PLC-Upper Body	0.110	0.099	0.012
PLC-All (Bag)	0.478	0.223	0.021
PLC-All (Spatial)	0.696	0.499	0.110
Discriminatively trained DPM Felzenszwalb et al. [2010]	-	0.880	-
Fast R-CNN Girshick [2015]	-	0.882	-

Figure 4.1: Evaluation of a human detector on Graz-O2. For each image, we run the detector (which is equivalent to running Alexnet convolutionally) at 5 scales and report the highest-scoring bounding boxes after NMS. We explore variants of a PLC with 3 parts, including individual parts, a bag-of-parts (without a spatial term), and a full 3-part model with star-spatial structure (which outperforms all variants). For comparison, we show average precision results of two supervised methods. While not state-of-the-art, our results are impressive given that no person labels were ever used. Qualitative results are included in Fig. 4.2.

Chapter 4

Evaluation

We evaluate our pictorial zero-shot framework with a variety of tasks of increasing difficulty, in terms of novelty with respect to AlexNet's training set. We first learn a model of categories that do appear in the training set but are *not labeled*. We then learn a *fine-grained* subcategory model for labeled categories, and finally conclude with a true zero-shot model of *never-before-seen* categories.

Figure 4.2: On the **right**, we show a visual representation of the parts and spatial constraints used to construct our PLC for human detection. At **left**, we show the top scoring detections for 3 instances of Graz-02.

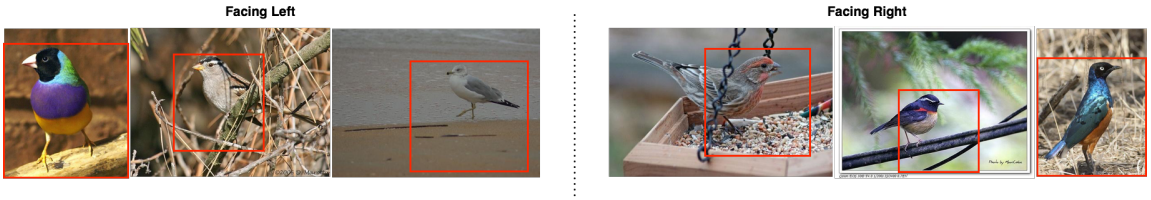

Figure 4.3: We apply two *pose-specific* PLCs composed of 4 parts to detect birds on PASCAL VOC 2007, showing the top-3 results for each detector. Our detectors are able to identify the correct on 87.5% of the detected bird instances, which is impressive given that no pose labels were ever used.

4.1 Unlabeled category: pedestrians

Perhaps surprisingly, pedestrians are not an explicit category in the ILSVRC12's training set. We use our interface to construct a simple model for pedestrians using three user-defined parts, roughly corresponding to face, lower-body, and upper-body (visualized in Fig. 4.2). We benchmark our model as a pedestrian detector on the Graz-02 Marszatek and Schmid [2007] dataset, which is composed of natural-scene images that contain cluttered backgrounds and complex objects in a variety of poses. We use the standard detection metric of average precision (AP) computed over various intersection-over-union (IOU) thresholds in Fig. 4.1. We refer the reader to the caption for more details, but in summary, star-structured part models significantly outperform any single part, as well as a "bag-of-parts" model without any spatial constraints. Our zero-shot models don't quite match the previously published performance of supervised models. One reason is that our models are tuned for frontal people. But our performance is impressive in that we use **zero** labeled examples and **no** linguistic knowledge!

4.2 Subcategory Pose Analysis on Pascal VOC

We next learn novel subcategory for the labeled category of birds. We specifically use the user-drawn boundary in Fig. 2.2-(a) to define left-facing and right-facing bird models. We evaluate our two pose-specific bird models on Pascal VOC 2007 testset, and show the top-3 hits for each model in Fig. 4.3. When we evaluate pose accuracy on the set of detected birds in PASCAL 2007-val, we obtain an accuracy of 87.5%. These results empirically support the claim that CNNs trained for image classification implicitly learn pose invariance, but this knowledge might be "tangled" in internal neural activations. Our pictorial visualizations allow a user to "untangle" such cues and explicitly build fine-grained detectors **without** any labeled examples.

4.3 Concept Discovery on MS COCO

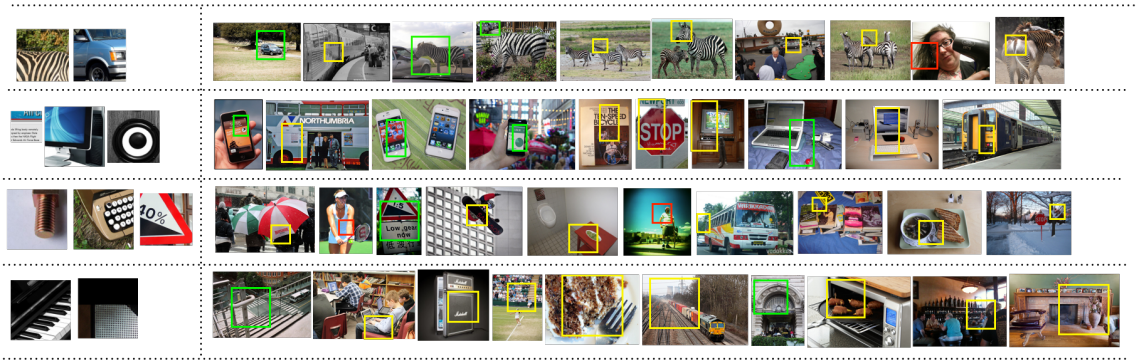

Figure 4.4: Concept discovery on MSCOCO. Each row presents a set of neurons that composed into a "bag-of-parts" PLC, followed by the top 10 detections from MSCOCO-val. We evaluate results with a user-study where participants are prompted with the following: `Interpret the query on the left as a set of visual concepts such as textures, parts, and objects. For each image on the right, determine whether the detection enclosed by the bounding box (i) represents all the queried visual concepts, (ii) represents at least one, or (iii) does not represent any.` Boxes are colored green (i), yellow (ii), or red (iii) based on the majority response.

We now use zero-shot visualizations to discover novel concepts in MSCOCOLin et al. [2014].Fig. 4.4 shows 4 visual concept queries and the top 10 detections for each. Each

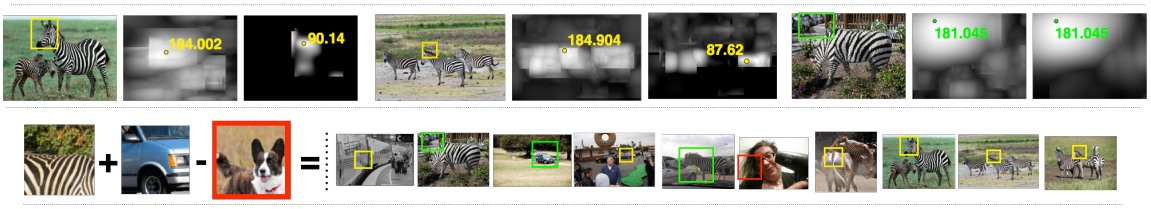

Figure 4.5: We explore the notion of negative parts, by subtracting an *animal head* part score from a *stripes+car front*. On the **top**, we show three images corresponding to the query from the first row of Fig. 4.4, followed by score maps of the original two-part model and the two-parts-minus-the-head. Scores associated with zebra regions decrease, but the striped car detection (on the **top-right**) remains high. This translates to a better re-ranking of retrieved images (**bottom**).

concept is user-defined to be a ”bag of (2-3) parts”. To evaluate results, we ask a set of participants whether or not the returned detections (i) encompass all of the visual parts, (ii) contain at least one of the visual parts, or (iii) are false detections. Our mAP for the four models at a recall of 10 is 0.401 for (i) and 0.931 for (ii). User-defined models tend to fire on image regions with a single strong part activation. We posit that since these parts rarely (if ever) co-occured in the training set, their activations are not properly calibrated, implying that better normalization of their activations might help. One interesting error mode is the *striped car* query (first row), which actually fires on a striped car (the ranked-4 result) but also finds zebras. During our semantic exploration, it is quite apparent that an *animal head*s co-occur with *stripes*. We built a PLC with a ”negative” part simply by subtracting the score of the best *animal head* within a candidate bounding box. We find that this leads to noticeably better score maps and retrievals (Fig. 4.5), suggesting an avenue for making our grammar more flexible.

Chapter 5

Conclusions

We have shown that a CNN trained on ImageNet classification learns an image representation that can be transferred to build part-based detectors for (both previously seen but unlabeled and never-before-seen) objects without additional training data. We also present a simple interface for users to specify a set of neurons and spatial constraints that can perform object detection, retrieval, and concept discovery on large datasets. Our current interface implementation is not real-time, though we believe this is readily obtainable with intelligent feature caching (possible because our model can make use of off-the-shelf Alexnet features).

Bibliography

Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. Zero-shot learning via visual abstraction. In *ECCV*. 2014. 1

Christopher G Atkeson and Stefan Schaal. Learning tasks from a single demonstration. In *ICRA*, volume 2, pages 1706–1712. IEEE, 1997. 1

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *NIPS*, 19:137, 2007. 1

Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. 2.1, 2.1

Melissa Bowerman. The origins of childrens spatial semantic categories: Cognitive versus linguistic determinants. *Rethinking linguistic relativity*, pages 145–176, 1996. 1

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009. 1

Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015. 2.1

Alireza Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785. IEEE, 2009. 1

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 3.1, 4

Zhenyong Fu, Tao Xiang, Elyor Kodirov, and Shaogang Gong. Zero-shot object recognition by semantic manifold distance. In *CVPR*, pages 2635–2644, 2015. 1

Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 2.1, 4

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1, 2.1

Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *CVPR*, pages 437–446, 2015. 1, 3.1, 3.1

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520, 2011. 1

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2.1

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958. IEEE, 2009. 1

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014. 4.3

Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 2.1

Laurens Maaten. Learning a parametric embedding by preserving local structure. In *International Conference on Artificial Intelligence and Statistics*, pages 384–391, 2009. 2.1

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196. IEEE, 2015. 1

Marcin Marszatek and Cordelia Schmid. Accurate object localization with shape masks. In *CVPR*, pages 1–8. IEEE, 2007. 4.1

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *PAMI*, 35 (11):2624–2637, 2013. 1

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 1

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014. 1

Allan Paivio. Mental imagery in associative learning and memory. *Psychological review*, 1969. 1

Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*. 2009. 1

Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, pages 1641–1648. IEEE, 2011. 1

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. 2, 2.1

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226. Springer, 2010. 1

S. Shalev-Shwartz and A. Shashua. On the Sample Complexity of End-to-end Training vs. Semantic Abstraction Training. *ArXiv e-prints*, April 2016. 1

Lior Shamir, John D Delaney, Nikita Orlov, D Mark Eckley, and Ilya G Goldberg. Pattern recognition software and techniques for biological image analysis. *PLoS Comput Biol*, 6(11):e1000974, 2010. 1

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013. 1

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011. 1

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015. 1

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008. 2.1

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014. 1

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV*, pages 818–833. Springer, 2014. 1

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 1, 2.1

Song-Chun Zhu and David Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007. 1