Learning Everything about Anything: Webly-Supervised Visual Concept Learning

Santosh K. Divvala[*,†], Ali Farhadi[†], Carlos Guestrin[†]
[†]University of Washington [*]The Allen Institute for AI
http://levan.cs.uw.edu

Figure 1: We introduce a fully-automated method that, given *any* concept, discovers an exhaustive vocabulary explaining *all* its appearance variations (i.e., actions, interactions, attributes, etc.), and trains full-fledged detection models for it. This figure shows a few of the many variations that our method has learned for four different classes of concepts: object (horse), scene (kitchen), event (Christmas), and action (walking).

Abstract

Recognition is graduating from labs to real-world applications. While it is encouraging to see its potential being tapped, it brings forth a fundamental challenge to the vision researcher: scalability. How can we learn a model for any *concept that exhaustively covers* all *its appearance variations, while requiring minimal or no human supervision for compiling the vocabulary of visual variance, gathering the training images and annotations, and learning the models?*

In this paper, we introduce a fully-automated approach for learning extensive models for a wide range of variations (e.g. actions, interactions, attributes and beyond) within any concept. Our approach leverages vast resources of online books to discover the vocabulary of variance, and intertwines the data collection and modeling steps to alleviate the need for explicit human supervision in training the models. Our approach organizes the visual knowledge about a concept in a convenient and useful way, enabling a variety of applications across vision and NLP. Our online system has been queried by users to learn models for several interesting concepts including breakfast, Gandhi, beautiful, *etc. To date, our system has models available for over 50,000 variations within 150 concepts, and has annotated more than 10 million images with bounding boxes.*

1. Introduction

How can we learn *everything* (visual) about *any* concept? There are two main axes to this question. The *everything* axis corresponds to all possible appearance variations of a concept, while the *anything* axis corresponds to the span of different concepts for which visual models are to be learned.

The conventional paradigm for learning a concept model is to first discover the visual space of variance for the concept (variance discovery), then gather the training data i.e., images and annotations, and finally design a powerful model that can handle all the discovered intra-concept variations (variance modeling). For variance discovery, the common practice is to assume a manually defined vocabulary by relying on benchmark datasets [47]. For variance modeling, which is often approached in isolation from the discovery step, the majority of methods use a *divide and conquer* strategy, where the training data within a category is grouped into smaller sub-categories of manageable visual variance [13]. A variety of cues have been used to partition the data: viewpoint [9], aspect-ratio [18], poselets [5], visual phrases [43], taxonomies [11], and attributes [16, 23].

While the above paradigm has helped advance the recognition community, two fundamental and pragmatic questions remain unanswered: *First,* how can we ensure *everything* about a concept is learned? More specifically, how can

we gather an exhaustive vocabulary that covers all the visual variance within a concept? *Second,* how can we scale the above paradigm to learn everything about *anything*? I.e., is it possible to devise an approach that alleviates the need for human supervision in discovering the vocabulary, gathering training data and annotations, and learning the models?

In this paper, we introduce a novel "webly-supervised" approach to discover and model the intra-concept visual variance. We show how to automatically gather an exhaustive vocabulary of visual variance for any concept, and learn reliable visual models using no explicit human supervision.

1.1. Variance Discovery and Modeling

Almost all previous works have resorted to the use of explicit human supervision for variance discovery and modeling. Using explicit supervision for variance discovery has a couple of drawbacks:

Extensivity: a manually-defined vocabulary of variance cannot enumerate all the visual variances for a concept, and is biased towards the cultural, geographical, or temporal biases of the people compiling them. For example, 'fire-walking' is a popular phenomenon only in some parts of the world, and thus may get excluded in the vocabulary of 'walking'. When sampling the visual space for collecting data, arbitrary and limited vocabularies can result in highly biased datasets [47]. There is always a trade-off between the exhaustiveness of the vocabulary discovered and the complexity of the model used to constrain the visual variance; a more exhaustive vocabulary results in limited variance within each group, and thereby potentially alleviates the need for sophisticated models.

Specificity: pre-defined vocabularies do not typically generalize to new concepts. For example, the action of 'rearing' can modify a 'horse' with very characteristic appearance, but does not extend to 'sheep', while 'shearing' applies to 'sheep' but not to 'horse'. This makes the task of manually defining a vocabulary even more burdensome as one needs to define these vocabularies per concept.

Using explicit supervision for variance modeling has the following additional drawbacks:

Flexibility: the act of explicit human annotation leads to rigid decisions at the time of dataset creation (e.g., the list of attributes [16, 23], or visual phrases [43]). These decisions can seldom be modified once the annotations have been collected and thus often end up dictating the methods used to process the data. For example, a grouping based on horse breeds ('sorrel horse', 'pommel horse', etc) as in Imagenet [11] is not very useful for a shape (HOG)-based 'horse' detector [18]; a grouping based on actions ('jumping horse', 'reining horse', etc) might be preferable. Thus, it will be beneficial if the annotations can be modified based on the feature representation and the learning algorithm.

Scalability: human annotation also presents a hurdle towards learning scalable models. Every new proposal to constrain the visual variance of the data poses a herculean task of either preparing a new dataset (e.g., ImageNet) or adding new annotations to an existing dataset. For example, in the case of phraselets [12] and attributes [16, 23], new annotations had to be added to all the PASCAL VOC images. Furthermore, as the modeling step is typically approached in isolation from the discovery step, the annotations obtained for modeling the intra-concept variance are often different and disjoint from those gathered during variance discovery.

1.2. Overview

In this work, we propose a new approach to automatically discover and model the visual space of a concept that circumvents the above limitations (see Figure 2). To discover the vocabulary of variance, we leverage vast resources of books available online (Google Books Ngrams [33]). This discovered vocabulary is not only extensive but also concept-specific. Given a term e.g., 'horse', the corpus includes ngrams containing all aspects of the term such as actions ('rearing horse'), interactions ('barrel horse'), attributes ('bridled horse'), parts ('horse eye'), viewpoints ('front horse'), and beyond (see Figure 1, top row).

To model the visual variance, we propose to intertwine the vocabulary discovery and the model learning steps. Our proposal alleviates the need for explicit human annotation of images, thus offering greater flexibility and scalability. To this end, we leverage recent progress in text-based web image search engines, and weakly supervised object localization methods. Image search has improved tremendously over the past few years; it is now possible to retrieve relevant sets of object-centric images (where the object of interest occupies most of the image) for a wide range of queries. While the results are not perfect, the top ranked images for most queries tend to be very relevant [32]. With the recent success of the Deformable Parts Model (DPM) detector [18], weakly-supervised object localization techniques [36, 39] have risen back to popularity. Although these methods do not work well when presented with a highly diverse and polluted set of images, e.g., images retrieved for 'horse', they work surprisingly well when presented with a relatively clean and constrained set of object-centric images, e.g., images retrieved for 'jumping horse'.

Our idea of intertwining the discovery and modeling steps is in part motivated by the observation that the VOC dataset was compiled by downloading images using an explicit set of query expansions for each object (see Table 1 in [15]). However, the VOC organizers discarded the keywords after retrieving the images, probably assuming that the keywords were useful only for creating the dataset and not for model learning purposes, or presumed that since the keywords were hand-chosen and limited, focusing too much on them would produce methods that would not generalize to new classes. In this work, we show how the idea of systematic query expansion helps not only in gathering less biased data, but also in learning more reliable models with no explicit supervision.

Our contributions include: (i) A novel approach for discovering a comprehensive vocabulary (covering actions, interactions, attributes, and beyond), and training a full-fledged detection model for any concept, including scenes, events, actions, places, etc., using no explicit supervision. (ii) Showing substantial improvement over existing weakly-

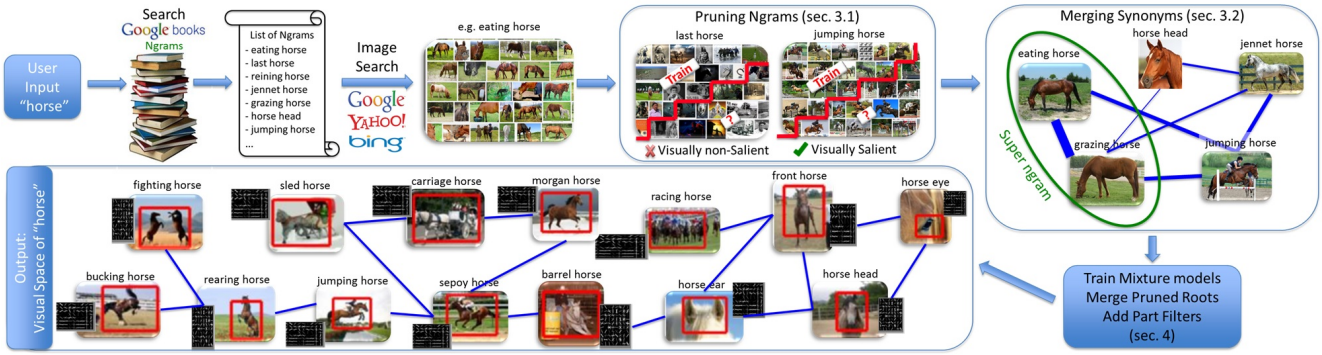

Figure 2: Approach Overview

supervised state-of-the-art methods. For some categories, our results are on par with the *supervised* state-of-the-art. (iii) Presenting impressive results for unsupervised action detection for the first time. (iv) An open-source online system (`http://levan.cs.uw.edu`) that, given any query concept, automatically learns everything visual about it. To date, our system has learned more than 50,000 visual models that span over 150 concepts, and has annotated more than 10 million images with bounding boxes.

2. Related work

Taming intra-class variance: Previous works on constraining intra-class variance have considered simple annotations based on aspect-ratio [18], viewpoint [9], and feature-space clustering [13]. These annotations can only tackle simple appearance variations of an object [51]. Recent works have considered more complex annotations such as phrases [43], phraselets [12], and attributes [16, 23]. While explicit supervision is required to gather the list of phrases and their bounding boxes in [43], the work of [12] needs heavier supervision to annotate joint locations of all objects within the dataset. Although [24, 27] discover phrases directly using object bounding boxes, their phrasal vocabulary is limited to object compositions, and cannot discover complex actions, e.g., 'reining horse', 'bucking horse', etc. Moreover, all of the methods [12, 24, 27] discover phrases only involving the fully annotated objects within a dataset, i.e., they cannot discover 'horse tram' or 'barrel horse' when tram and barrel are not annotated. Attributes [16, 23] are often ambiguous to be used independent of the corresponding object, e.g., a 'tall' rabbit is shorter than a 'short' horse; 'cutting' is an attribute referring to a sport for horses while it has a completely different meaning for sheep. To date, there exists no established schema for listing attributes for a given dataset [37].

Weakly-supervised object localization: The idea of training detection models from images and videos without bounding boxes has received renewed attention [2, 36, 39, 45] due to the recent success of the DPM detector [18]. While it is encouraging to see progress, there are a few limitations yet to be conquered. Existing image-based methods [2, 36, 45] fail to perform well when the object of interest is highly cluttered or when it occupies only a small portion of the image (e.g., bottle). Video-based methods [39] rely on motion cues, and thus cannot localize static objects (e.g., tvmonitor). Finally, all existing methods train their models on a weakly-labeled dataset where each training image or video is assumed to contain the object. To scale to millions of categories, it is desirable to adapt these methods to directly learn models from noisy web images.

Learning from web images: Due to the complexity of the detection task and the higher supervision requirements, most previous works [4, 19, 28, 38, 44, 48] on using web images have focused on learning models only for image classification. The work of [21, 41] focuses on discovering commonly occurring segments within a large pool of web images, but does not report localization results. The work of [49] uses active learning to gather bounding box annotations from Turkers. The work of [7] aims at discovering common sense knowledge from web images, while our work focuses on learning exhaustive semantically-rich models to capture intra-concept variance. Our method produces well-performing models that achieve state-of-the-art performance on the benchmark PASCAL VOC dataset.

3. Discovering the Vocabulary of Variance

In order to obtain all the keywords that modify a concept, we use the Google books ngram English 2012 corpora [33]. We specifically use the dependency gram data, which contains parts-of-speech (POS) tagged *head=>modifier* dependencies between pairs of words, and is much richer than the raw ngram data (see section 4.3 in [30]). We choose ngram data over other lexical databases (such as Wordnet or Wikipedia lists [1]) as it is much more exhaustive, general, and includes popularity (frequency) information. Using the books ngram data helps us cover all variations of any concept the human race has ever written down in books.

Given a concept and its corresponding POS tag, e.g., 'reading, verb', we find all its occurrences annotated with that POS tag within the dependency gram data. Using the POS tag helps partially disambiguate the context of the query, e.g., reading action (verb) vs. reading city (noun). Amongst all the ngram dependencies retrieved for a given concept, we select those where the modifiers are tagged either as noun, verb, adjective, or adverb[1]. We marginalize over years by summing up the frequencies across differ-

[1]Conjunctions, determiners, pronouns, numbers, and particles are ignored as they seldom carry visually relevant information, e.g., 'the horse', '100 horse', etc.

ent years. Using this procedure, we typically end up with around 5000 ngrams for a concept.

Not all the ngrams gathered using the above procedure are visually salient, e.g., 'particular horse', 'last horse', etc. While our model learning procedure (section 4) is robust to such noise, it would be unnecessary to train full-fledged detectors for irrelevant ngrams. To avoid wasteful computation, we use a simple and fast image-classifier based pruning method. Our pruning step can be viewed as part of a cascade strategy that rejects irrelevant ngrams using a weak model before training strong models for relevant ngrams.

3.1. Classifier-based Pruning

The goal here is to identify visually salient ngrams out of the pool of all discovered ngrams for a concept. Our main intuition is that visually salient ngrams should exhibit predictable visual patterns accessible to standard classifiers. This means that an image-based classifier trained for a visually salient ngram should accurately predict unseen samples of that ngram.

We start by retrieving a set of images I_i for each ngram i. To maintain low latency, we only use thumbnails (64×64 pixels) of the first 64 images retrieved from Image Search. We ignore all near-duplicate images. We then randomly split this set into equal-sized training and validation sets $I_i = \{I_i^t, I_i^v\}$, and augment the training images I_i^t with their mirrored versions. We also gather a random pool of background images $\bar{I} = \{\bar{I}^t, \bar{I}^v\}$. For each ngram, we train a linear SVM [6] W_i with I_i^t as positive and $\bar{I}^t$ as negative training images, using dense HOG features [18]. This classifier is then evaluated on a combined pool of validation images $\{I_i^v \cup \bar{I}^v\}$.

We declare an ngram i to be visually salient if the Average Precision (A.P.) [15] of the classifier W_i computed on $\{I_i^v \cup \bar{I}^v\}$ is above a threshold. We set the threshold to a low value (10%) to ensure all potentially salient ngrams are passed on to the next stage, and only the totally irrelevant ones are discarded. Although our data is noisy (the downloaded images are not manually verified to contain the concept of interest), and the HOG+linearSVM framework that we use is not the prevailing state-of-the-art for image classification, we found our method to be effective and sufficient in pruning irrelevant ngrams. After the pruning step, we typically end up with around 1000 ngrams for a concept.

3.2. Space of Visual Variance

Amongst the list of pruned ngrams there are several synonymous items. For example, 'sledge horse' and 'sleigh horse', 'plow horse' and 'plough horse', etc. Further, some non-synonymous ngrams correspond to visually similar entities, e.g., 'eating horse' and 'grazing horse' (see Figure 2) [31]. To avoid training separate models for visually similar ngrams, and to pool valuable training data across them, we need to sample the visual space of a concept more carefully. How can we identify representative ngrams that span the visual space of a concept? We focus on two main criteria: quality and coverage (diversity).

We represent the space of all ngrams by a graph $G = \{V, E\}$ where each node represents an ngram and each edge represents the visual similarity between them. Each node has a score d_i that corresponds to the quality of the ngram classifier W_i. We set the score d_i as the A.P. of the classifier W_i on its validation data $\{I_i^v \cup \bar{I}^v\}$. The edge weights e_{ij} correspond to the visual distance between two ngrams i, j and is measured by the score of the jth ngram classifier W_j on the ith ngram validation set $\{I_i^v \cup \bar{I}^v\}$. To avoid issues with uncalibrated classifier scores, we use a rank-based measure. Our ranking function $(R : \mathbb{R}^{|\bar{I}^v \cup I_i^v|} \mapsto \mathbb{N}^{|I_i^v|})$ ranks instances in the validation set of an ngram against the pool of background images. In our notation $R_{i,j}$ corresponds to ranks of images in I_i^v against $\bar{I}^v$ scored using W_j. We use the normalized median rank as the edge weight $e_{i,j} = \frac{Median(R_{i,j})}{|I_j^v|}$. We scale the $e_{i,j}$ values to be between $[0\ \ 1]$.

Concept	Discovered superngrams
Cancer	{subglottic cancer, larynx cancer, laryngeal cancer} {rectum cancer,colorectal cancer,colon cancer}
Kitchen	{kitchen bin, kitchen garbage, kitchen wastebasket} {kitchen pantry, kitchen larder}
Gandhi	{gandhi mahatma, gandhi mohandas} {indira gandhi, mrs gandhi}
Christmas	{christmas cake, christmas pie, christmas pudding} {christmas crowd, christmas parade, christmas celebration}
Angry	{angry screaming, angry shouting} {angry protesters, angry mob, angry crowd}
Doctor	{doctor gown,doctor coat} {pretty doctor,women doctor,cute doctor} {examining doctor, discussing doctor, explaining doctor}
Apple	{apple crumble, apple crisp, apple pudding} {apple trees, apple plantation, apple garden} {apple half, apple slice, apple cut}
Jumping	{jumping group, jumping kids, jumping people} {jumping dancing, jumping cheering} {wave jumping, boat jumping}
Running	{running pursues, running defenders, running backs} {fitness running, exercise running} {running shirt, running top, running jacket}

Table 1: Examples of the vocabulary discovered and the relationships estimated for a few sample concepts.

The problem of finding a representative subset of ngrams can be formulated as searching for the subset $\mathcal{S} \subseteq V$ that maximizes the quality $\mathcal{F}$ of that subset:

$$\max_{\mathcal{S}} \mathcal{F}(\mathcal{S}), \quad \text{such that } |\mathcal{S}| \leq k, \tag{1}$$

$$\text{where } \mathcal{F}(\mathcal{S}) = \sum_{i \in V} d_i \cdot \mathcal{O}(i, \mathcal{S}). \tag{2}$$

$\mathcal{O}$ is a soft coverage function that implicitly pushes for diversity:

$$\mathcal{O}(i, \mathcal{S}) = \begin{cases} 1 & i \in \mathcal{S} \\ 1 - \prod_{j \in \mathcal{S}} (1 - e_{i,j}) & i \notin S \end{cases} \tag{3}$$

This formulation searches for a subset of ngrams that are visually manageable (have reliable ngram classifiers) and cover the space of variance within a concept (similar to [3, 22]). Fortunately, this objective function is submodular, hence there exists a greedy solution within a constant approximation of the optimal solution. We use an iterative greedy solution that adds at each stage the ngram

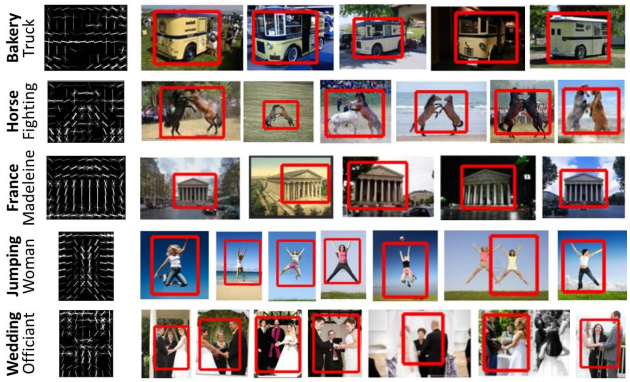

Figure 3: Sample localization results using our approach. Each row shows the learned HOG template and a few training instances.

i that provides the maximum gain over the current subset ($\arg\max_i \mathcal{F}(\mathcal{S} \cup i) - \mathcal{F}(\mathcal{S})$).

This algorithm provides the subset of representative ngrams that best describes the space of variance under a fixed budget k. We can use the same algorithm to also merge similar ngrams together to form *superngrams*. By setting the cost of adding similar ngrams in $\mathcal{S}$ to a really high value, each ngram $l \notin \mathcal{S}$ can be merged to its closest member in $\mathcal{S}$. Our merging procedure reveals interesting relations between ngrams by merging visually similar actions, interactions, and attributes. For example, our method discovers the following ngrams of 'horse' as visually similar: {tang horse, dynasty horse}, {loping horse, cantering horse}, {betting horse, racing horse}, etc. Table 1 shows more examples for other concepts. Using this procedure, we reduce the number of ngrams to around 250 superngrams.

4. Model Learning

The images for training the detectors are gathered using Image Search with the query phrases as the ngrams constituting the superngram. We download 200 full-sized, 'full' color, 'photo' type images per query. We resize images to a maximum of 500 pixels (preserving aspect-ratio), discard all near-duplicates, and ignore images with extreme aspect-ratios (aspect ratio > 2.5 or < 0.4). We split the downloaded images into training and validation sets.

Training a mixture of roots: Pandey et al., [36] demonstrated the possibility of training the DPM detector [18] using weak supervision. Directly applying their method to all the images within a concept (pooled across all ngrams) results in a poor model[2]. Therefore we train a separate DPM for each ngram where the visual variance is constrained.

[36] initializes the DPM with the full image as the bounding box. We found using this initialization often leads to the bounding box getting stuck to the image boundary during the latent reclustering step[3]. To circumvent this problem, we initialize our bounding box to a sub-image within the image that ignores the image boundaries. Using this initialization also avoids the two-stage training procedure used in [36], where in the first stage latent root placements are identified and cropped, for training the DPM in the second stage.

Similar to [18], [36] also initialized their components using the aspect-ratio heuristic. This is sub-optimal in the weakly supervised setting as image aspect-ratio is a poor heuristic for clustering object instances. To address this limitation, we initialize the model using feature space clustering as proposed in [13]. While our ngram vocabulary helps segregate the major appearance variations of a concept, the downloaded images per superngram still have some remaining appearance variations. For example, the 'jumping horse' ngram has images of horses jumping in different orientations. To deal with such appearance variations, we use a mixture of components initialized with feature space clustering. In the presence of noisy web images, this procedure provides a robust initialization. Some of the mixture components act as noise sinks, thereby allowing cleaner models to be learned [51]. In our experiments, we typically found 70% of the components per ngram to act as noise sinks. It is wasteful to train a full parts-based model for such noisy components. Therefore, we first train root filters for each component and subsequently prune the noisy ones.

Pruning noisy components: To prune noisy components, we run each component detector on its own validation set and evaluate its performance. Given that the positive instances within the validation set for each ngram neither have the ground-truth bounding boxes nor the component labels, we treat this task as a latent image classification problem. Specifically, we first run the ngram mixture-of-components detector on its full validation set (held-out positive images as well as a random pool of background images). We then record the top detection for each image and use the component label of that detection to segregate the images. We now have a segregated pool of validation images per ngram component. In the absence of ground-truth boxes, we assume the top detections of positive images are true and negative images are false, and therefore compute the average precision (A.P.) by only using the detection scores (ignoring overlap). We declare a component to be noisy either if its A.P. is below a threshold (10%) or if its training or validation data has too few (< 5) positive instances. The second condition helps us discard *exemplar* components that overfit to incidental images. While a root-only filter model is relatively weak compared to the parts model, we found that it does an effective job here for pruning noisy components.

Merging pruned components: Some of the components across the different ngram detectors end up learning the same visual concept. For example, a subset of 'hunter horse' instances are quite similar to a subset of 'jumping horse' instances. The merging step in section 3.2 considered a monolithic classifier trained with full image features.

[2] A.P. of $\{10.8\%, 12.7\%, 12.1\%, 10.6\%, 11.1\%, 10.1\%\}$ for $K = \{6, 12, 18, 25, 50, 100\}$ components, respectively, for the *horse* category (c.f. 30.6% using our method, see Table 2)

[3] This is due to the artifact of zero-padding within the HOG descriptor at the image boundaries, and the non-convex optimization of the latent SVM. In some cases, the latent SVM obtains a lower objective function value by preferring a larger box that includes the image boundary during the reclustering step.

As the mixture of component models are more refined (by way of localizing instances using detection windows), they can identify subtle similarities that cannot be found at the full image level. To pick a representative subset of the components and merge similar ones, we follow a similar procedure as outlined in section 3.2. Specifically, we represent the space of all ngram components by a graph $G = \{V, E\}$, where each node represents a component and each edge represents the visual similarity between them. The score d_i for each node now corresponds to the quality of the component. We set it to the A.P. of the component (computed during the above pruning step). The weight on each edge $e_{i,j}$ is defined similarly as the median rank obtained by running the jth component detector on the ith component validation set. (We continue to use only the top detection score per image, assuming top detections on positives are true and on negatives are false.) We solve for the same objective function as outlined in equation (1) to select the representative subset. We found this subset selection step results in roughly 50% fewer components. The final number of components averages to about 250 per concept. Figure 3 shows some of our discovered components.

Given the representative subset of components, we finally augment them with parts as described in [18], and subsequently merge all the components to produce the final detector.

5. Results

Our proposed approach is a generic framework that can be used to train an extensive detection model for any concept. To quantitatively evaluate the performance of our approach, we present results for object and action detection.

Object detection: We evaluated the performance of our trained detection model for the 20 classes in the PASCAL VOC 2007 testset [15]. We picked this dataset as recent state-of-the-art weakly supervised methods have been evaluated on it. In our evaluation, we ensured that none of the test images of the VOC 2007 testset existed in our trainset.

Table 2 displays the results obtained using our algorithm and compares to the state-of-the-art baselines [39, 45]. [45] uses weak human supervision (VOC data with image-level labels for training) and initialization from objectness [2] that is in turn trained on a VOC 2007 subset[4]. In comparison, our method uses *web* supervision as not even the images are supplied for training[5]. Nonetheless, our result substantially surpasses the previous best result in weakly-supervised object detection. Figure 1 shows some of the actions, interactions, and attributes learned for 'horse'. Figure 4 shows the models learned for other concepts.

Action detection: The VOC challenge [15] also hosts the *action classification* task where the bounding boxes for the human performing the action are assumed to be known (both in test and train images) and the human activity has to be identified. We consider the more challenging task of action detection, where not only does the action in an image has to be *identified*, but also *localized* with a bounding box. Further, we attempt to perform action detection in the unsupervised setting where even the training images are not provided. To our knowledge, this is the first attempt at webly-supervised action detection on the VOC dataset. The recent work of [12] reports results obtained for strongly-supervised action detection on the VOC 2011 val set. Table 3 reports our results and compares them to the supervised baseline reported in [12].

Figures 1,4 display some of the models learned for the VOC action classes 'jumping', 'running', and 'walking'. For each action category, our approach learns a detailed vocabulary that reveals several fine-grained variations, e.g., 'model walking' vs. 'race walking'. Similar to a dictionary or an encyclopedia providing different lexical connotations of a concept, our method generates its different visual connotations. We also ran our experiments on the composite 'ridingbike' category but found our model performed poorly (A.P. of 4.5% vs. 41.6% by [18]) as the VOC ground-truth box only covers the person performing

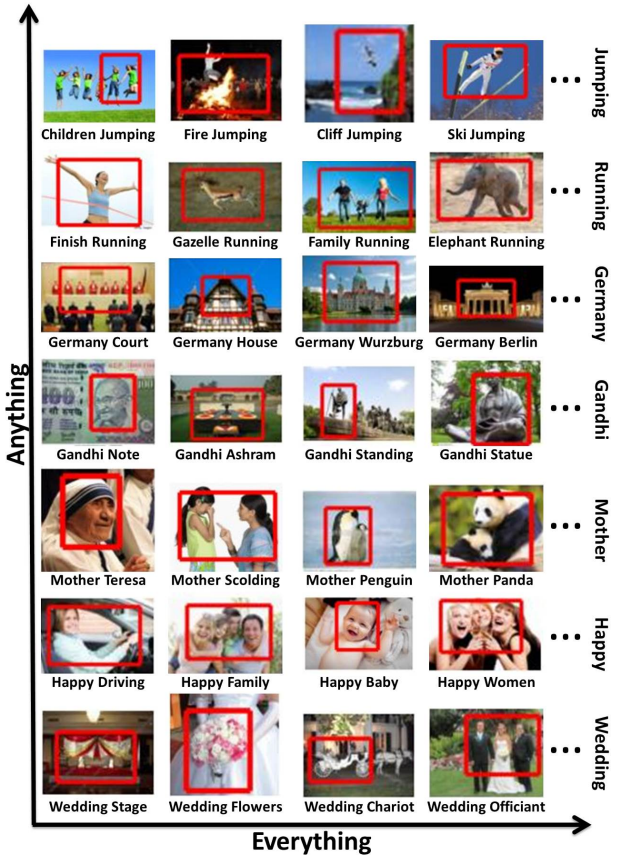

Figure 4: Our approach can learn extensive models for any concept (y-axis). Each row shows four of the many variations that our method has learned. Full results (detailed vocabulary and trained models) are available on our project website.

[4]Objectness [2] uses 'meta-training' images with ground-truth object annotations for learning its parameters. While objectness could be used as an initialization even in our approach, we chose not to use it here as our goal is to learn models for any concept (scenes, events, etc) and not just for the 20 VOC classes.

[5]While our method does not need any explicit supervision, it does download and use two orders of magnitude more images than the PASCAL VOC dataset

Method	Supervision	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	pers	plant	sheep	sofa	train	tv
[45]	weak	13.4	**44.0**	3.1	3.1	0.0	31.2	**43.9**	7.1	0.1	9.3	**9.9**	1.5	29.4	**38.3**	4.6	0.1	0.4	3.8	**34.2**	0.0
[39]	weak	**17.4**	-	9.3	9.2	-	-	35.7	**9.4**	-	9.7	-	3.3	16.2	27.3	-	-	-	-	15.0	-
Ours	web	14.0	36.2	**12.5**	**10.3**	**9.2**	**35.0**	35.9	8.4	**10.0**	**17.5**	6.5	**12.9**	**30.6**	27.5	**6.0**	**1.5**	**18.8**	**10.3**	23.5	**16.4**
[18]	*full*	*33.2*	*59.0*	*10.3*	*15.7*	*26.6*	*52.0*	*53.7*	*22.5*	*20.2*	*24.3*	*26.9*	*12.6*	*56.5*	*48.5*	*43.3*	*13.4*	*20.9*	*35.9*	*45.2*	*42.1*

Table 2: Results (A.P.) on VOC2007 (test) object detection. Rows 1 and 2 show state-of-the-art results for *weakly-supervised* object detection. [45] trains on VOC2007 training data with image-level labels and uses objectness for initialization. [39] trains on manually selected videos but without bounding boxes and shows results on $10/20$ classes (ignores classes without motion). Row 3 shows our webly-supervised results, i.e., not even the training images supplied. Row 4 shows the results of current state-of-the-art for *fully supervised* object detection that is a possible upper bound for weakly supervised approaches. Our method outperforms the supervised DPM on birds and dogs and is almost on par with DPM on sheep.

	jumping	phoning	walking	takingphoto
[18] (supervised)	*6.1*	*4.1*	*10.9*	*1.1*
Ours	12.8	3.6	10.7	0.2

Table 3: Results (A.P.) on VOC2011 (val) action detection. Top row shows state-of-the-art results for *fully-supervised* action detection obtained using [18] (as reported in [12]), while the bottom row shows our results. The evaluation protocol is the same as that for object detection (overlap $> 50\%$ is success.) Our result beats the fully supervised result for jumping and is almost on par in the remaining 3 classes.

the action, while our unsupervised approach also localizes the bike along with the person. (Our method gets an A.P. of 31.6% when the overlap criteria with ground-truth is reduced to 25%).

What are the sources of errors that prevent our model from performing on par with the supervised state-of-the-art [18]? We have found a couple of issues:

Extent of overlap: Our final detection model is a multi-component model (number of components ≥ 250 on average). Given a test image, there could be several valid detections by our model, e.g., an image of horse-drawn carriage would not only have the 'profile horse' detection but also the 'horse carriage' and 'horse head' detections. As the VOC criterion demands a single unique detection box for each test instance that has 50% overlap, all the other valid detections are declared as false-positives either due to poor localization or multiple detections. Selecting the *correct* box from a pool of valid detections in a completely unsupervised setting is a challenging research problem.

Polysemy: Our framework learns a generic model for a concept, e.g., the car model includes some bus-like car components, while the VOC dataset exclusively focuses on typical cars (and moreover, discriminates cars from buses). Such polysemy is an achilles' heel when dealing with lexical resources as the same term can refer to two completely different concepts (see Figure 5). To alleviate these concerns, it might be possible to tune our model to account for dataset biases and thereby improve its performance [20, 47]. Tuning biases in a completely unsupervised setting is also an interesting research direction.

Testing our model involves convolving each test image with an exhaustive model of around 250 components. This testing step can be easily sped up by leveraging recent fast detection methods [10, 14, 46].

Figure 5: Our approach learns models for all visual connotations of a concept. For example, the noun 'train' could either refer to a dress or a locomotive, and the noun 'chair' could either refer to a piece of furniture or a designation. Future work could focus on exploring the polysemy discovered by our approach.

6. Conclusion & Potential Applications

We have presented a fully automated approach to discover a detailed vocabulary for any concept and train a full-fledged detection model for it. We have shown results for several concepts (including objects, scenes, events, actions and places) in this paper, and more concepts can be obtained by using our online system. Our approach enables several future applications and research directions:

Coreference resolution: A core problem in NLP is to determine when two textual mentions name the same entity. The biggest challenge here is the inability to reason about semantic knowledge. For example, the Stanford state-of-the-art system [25] fails to link 'Mohandas Gandhi' to 'Mahatma Gandhi', and 'Mrs. Gandhi' to 'Indira Gandhi' in the following sentence: Indira Gandhi was the third Indian prime minister. Mohandas Gandhi was the leader of Indian nationalism. Mrs. Gandhi was inspired by Mahatma Gandhi's writings. Our method is capable of relating Mahatma Gandhi to Mohandas Gandhi and Indira Gandhi to Mrs Gandhi (See Table 1). We envision that the information provided by our method should provide useful semantic knowledge for coreference resolution.

Paraphrasing: Rewriting a textual phrase in other words while preserving its semantics is an active research area in NLP. Our method can be used to discover paraphrases. For example, we discover that a 'grazing horse' is semantically very similar to a 'eating horse'. Our method can be used to produce a semantic similarity score for textual phrases.

Temporal evolution of concepts: Is it possible to also model the visual variance of a concept along the temporal axis? We can use the year-based frequency information in the ngram corpus to identify the peaks over a period of time and then learn models for them (see figure 6). This can help in not only learning the evolution of a concept [26], but also in automatically dating detected instances [35].

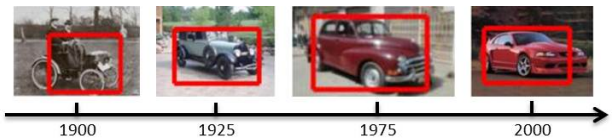

Figure 6: Our approach can be used to learn the temporal evolution of a concept. Figure shows instances of the concept 'car' with separate components trained for every generation (25 years) since 1900. Our merging algorithm merged the model for year 1950 with 1925 and 1975 (possibly indicating no major change in the appearance of cars during that time).

Deeper image interpretation: Recent works have emphasized the importance of providing deeper interpretation for object detections rather than simply labeling them with bounding boxes [34, 43]. Our work corroborates this line of research by producing enhanced detections for any concept. For example, apart from an object bounding box (e.g., 'horse'), it can provide object part boxes (e.g., 'horse head', 'horse foot', etc) and can also annotate the object action (e.g., 'fighting') or the object type (e.g., 'jennet horse'). Since the ngram labels that we use correspond to real-world entities, it is also possible to directly link a detection to its corresponding wikipedia page to infer more details [42].

Understanding actions: Actions and interactions (e.g., 'horse fighting', 'reining horse') are too complex to be explained using simple primitives. Our methods helps in discovering a comprehensive vocabulary that covers all (subtle) nuances of any action. For example, we have discovered over 150 different variations of the walking action including 'ball walking', 'couple walking', 'frame walking' (see Figure 1, bottom row). Such an exhaustive vocabulary helps in generating fine-grained descriptions of images [17, 29, 34, 40, 50].

Segmentation & discovery: Each component in our model has training instances that are all tightly aligned in the appearance space (see Figure 3). Hence it is possible to *cosegment* the instances, and learn a foreground segmentation model for each component using cosegmentation [8, 21, 41]. This enables extending our approach to perform unsupervised pixel-level segmentation, and obtain a rich semantic segmentation model for any concept.

Acknowledgments: This work was supported by ONR N00014-13-1-0720, ONR MURI 1141221-293180, ONR PECASE N00014-13-1-0023, NSF IIS-1218683, & NSF IIS-1258741. We thank Neeraj Kumar for his helpful comments.

References

[1] Wikipedia list of lists of lists. http://en.wikipedia.org/wiki/List_of_lists_of_lists. 3
[2] B. Alexe, T. Deselares, and V. Ferrari. Measuring the objectness of image windows. In *PAMI*, 2013. 3, 6
[3] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-best solutions in markov random fields. In *ECCV*, 2012. 4
[4] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010. 3
[5] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 1
[6] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. 2001. 4
[7] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013. 3
[8] X. Chen, A. Shrivastava, and A. Gupta. Enriching Visual Knowledge Bases via Object Discovery and Segmentation. In *CVPR*, 2014. 8
[9] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 1, 3
[10] T. Dean et al. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013. 7
[11] J. Deng et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2
[12] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012. 2, 3, 6, 7
[13] S. K. Divvala, A. A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *ECCV Workshop on Parts and Attributes*, 2012. arXiv:1206.3714. 1, 3, 5
[14] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. In *ECCV*, 2012. 7
[15] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, 2010. 2, 4, 6
[16] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2, 3
[17] A. Farhadi et al. Every picture tells a story: Generating sentences for images. In *ECCV*, 2010. 8
[18] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. http://www.cs.berkeley.edu/ rbg/latent. 1, 2, 3, 4, 5, 6, 7
[19] R. Fergus, F.-F. L., P. Perona, and A. Zisserman. Learning object categories from internet image searches. In *Proc. of IEEE*, 2010. 3
[20] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 7
[21] G. Kim and E. P. Xing. On multiple foreground cosegmentation. In *CVPR*, 2012. 3, 8
[22] A. Kulesza and B. Taskar. k-DPPs: Fixed-size determinantal point processes. In *ICML*, 2011. 4
[23] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 2, 3
[24] T. Lan, M. Raptis, L. Sigal, and G. Mori. From subcategories to visual composites: A multi-level framework for object detection. In *ICCV*, 2013. 3
[25] H. Lee et al. Deterministic coreference resolution based on entity-centric, precision-ranked rules. In *Computational Linguistics*, 2013. 7
[26] Y. J. Lee, A. A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *ICCV*, 2013. 7
[27] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *CVPR*, 2012. 3
[28] L.-J. Li and L. Fei-Fei. Optimol: Automatic online picture collection via incremental model learning. In *IJCV*, 2010. 3
[29] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *Computational Natural Language Learning*, 2011. 8
[30] Y. Lin et al. Syntactic annotations for the google books ngram corpus. In *ACL*, 2012. http://books.google.com/ngrams/datasets. 3
[31] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008. 4
[32] E. Mezuman and Y. Weiss. Learning about canonical views from internet image collections. In *NIPS*, 2012. 2
[33] J.-B. Michel et al. Quantitative analysis of culture using millions of digitized books. In *Science*, 2010. 2, 3
[34] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg. From large scale image categorization to entry-level categories. In *ICCV*, 2013. 8
[35] F. Palermo, J. Hays, and A. Efros. Dating historical color images. In *ECCV*, 2012. 7
[36] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 2, 3, 5
[37] D. Parikh, A. Farhadi, K. Grauman, T. Berg, and A. Gupta. Attributes. In *CVPR Tutorial*, 2013. https://filebox.ece.vt.edu/ parikh/attributes/. 3
[38] O. Parkhi, A. Vedaldi, and A. Zisserman. On-the-fly specific person retrieval. In *Image Analysis for Multimedia Interactive Services*, 2012. 3
[39] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 3, 6, 7
[40] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where and why semantic relatedness for knowledge transfer. In *CVPR*, 2010. 8
[41] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013. 3, 8
[42] B. C. Russell et al. 3d wikipedia: Using online text to automatically label and navigate reconstructed geometry. In *Siggraph Asia*, 2013. 8
[43] M. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 1, 2, 3, 8
[44] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *PAMI*, 2011. 3
[45] P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICV*, 2011. 3, 6, 7
[46] H. O. Song, R. Girshick, and T. Darrell. Discriminatively activated sparselets. In *ICML*, 2013. 7
[47] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1, 2, 7
[48] D. Tsai et al. Large-scale image annotation using visual synset. In *ICCV*, 2011. 3
[49] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 2011. 3
[50] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. In *PAMI*, 2013. 8
[51] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012. 3, 5