Learn How to Choose: Independent Detectors versus Composite Visual Phrases

Guy Rosenthal*
Tel Aviv University
guyrose3@gmail.com

Ariel Shamir
The Interdisciplinary Center
arik@idc.ac.il

Leonid Sigal
Disney Research
lsigal@disneyresearch.com

Abstract

Most approaches for scene parsing, recognition or retrieval use detectors that are either (i) independently trained or (ii) jointly trained for conjunctions of object-object or object-attribute phrases. We posit that neither of these two extremes is uniformly optimal, in terms of performance, across all categories and conjunctions. The choice of whether one should train an independent or composite detector should be made for each possible conjunction separately, and depends on the statistics of the dataset as well. For example, person holding phone *may be more accurately modeled using a single composite detector, while* tall person *may be more accurately modeled as combination of two detectors. We extensively study this issue in the context of multiple problems and datasets. Further, for efficiency, we propose a predictor that is based on a number of category specific features (*e.g.*, sample size, entropy,* etc.*) for whether independent or joint composite detector may be more accurate for a given conjunction. We show that our prediction and selection mechanism generalizes and leads to improved performance on a number of large-scale datasets and vision tasks.*

1. Introduction

Object detection [3, 5, 7, 13], scene understanding/parsing [2, 11, 14, 26], and visual language grounding [11, 21] have been the cornerstones of computer vision research for the last 20+ years. Significant advances in recent years have been made using machine learning techniques that train detectors of various types on images. Such detectors have shown that both spatial and visual contextual reasoning are important. Some model the spatial layout of objects in the scene [2, 8, 11, 9, 14], while others model the visual appearance of the objects, and how it changes as those objects interact [10, 13, 20, 19].

*Work conducted while first author was an intern at Disney.

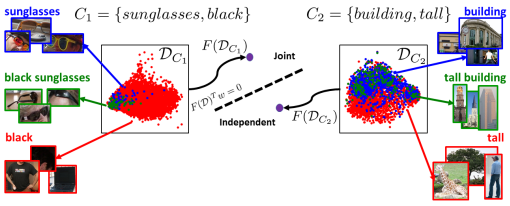

Figure 1: **Method overview:** given a visual composite C_i and the corresponding training data $\mathcal{D}$ (illustrated by tSNE plot of the CNN image features), statistical features $F(\mathcal{D})$ are extracted. The resulting vectors are used to determine which training strategy should be used for each specific composite, using a trained regressor. Our method chooses to train *black sunglasses*, $F(\mathcal{D}_{C_1})$, as a joint phrase and *tall building*, $F(\mathcal{D}_{C_2})$, as an independent product of *tall* and *building* classifiers.

Typical relationships that are considered include attribute-object [11], object-object [13] or object-relationship-object [11, 20, 19]. Learning detectors for these relationships is dealt with in one of two ways: (i) assuming conditional independence and simple geometric relationships, *e.g.*, similar to part-based models [11], or (ii) training of joint detectors, *e.g.*, for visual composite phrases [20, 19]. The issue with (i) is that while it accounts for geometric relationships among objects (or objects and attributes) it does not generally account for appearance variations induced by their combination. For example, a *sitting person* modeled as a product of two independent detectors, $P(sitting\ person) = P(sitting)P(person)$, would require to detect a *person* in all postures, which is difficult, and does not account for the fact that *sitting person* actually looks very different and is much easier to detect. In the second approach (ii), initially explored by Sadeghi and Farhardi *et al.* [20], this issue is resolved by training a single *sitting person* detector. While theoretically training such joint detectors is always better, in practice, this is not the case. The data available to train each individual phrase detector becomes scarce

(or even nonexistent) in many cases, which leads to overfitting (or inability to train a phrase detector).

The latest solution to this is to train both independent and various forms of composite detectors and combine their scores (*e.g.*, using a CRF [19]). However, this comes at a price, as an exponential number of detectors is needed for training and detection (*e.g.*, for a single object-relationship-object entity, 4 detectors are required in [19]). This makes scaling such methods to datasets containing thousands of objects and hundreds of relationships infeasible. Computational issues aside, various composite detectors may actually produce poorer performance that would degrade the overall performance, when the scores are combined, unless the contributions of each detector are well calibrated.

In this work, we first show that choosing the detector that results in the most accurate performance, either *joint* or *independent* for each composition, is clearly beneficial compared to the indiscriminate methods that do one or the other consistently across all categories. Further, we show that one can use a learned predictor, based on proxy category measures to predict which detector will be most beneficial. These features include the number of samples, separability, and entropy of image features. The predictor allows building the appropriate detector without training all possible options before selecting appropriate one (*e.g.*, using cross-validation). This is beneficial computationally and when number of samples is low. The illustration of the proposed selection process is depicted in Figure 1.

Contributions: Our contributions are three-fold:

- We study the effect of choosing between *independent* and *joint* visual phrase detectors for each conjunction relationship. We show that selecting appropriate detector can lead to large performance benefits.
- We introduce a novel method for predicting which of the two options is likely to be most beneficial in practice, based on statistical measures, without requiring pre-training of all possible detectors. To the best of our knowledge, this is the first method to show that this is possible.
- We demonstrate performance improvement using the proposed method on numerous computer vision tasks and large-scale datasets, including Scene Graph [11] and SUN [16] dataset.

2. Related work

This work is related to a number of core topics in computer vision, including contextual object detection, scene understanding, scene parsing and visual language grounding. We review only most relevant literature.

Object detection: Most object detection algorithms treat each object category independently and build independent classifiers/detectors using standard supervised learning methods (*e.g.*, SVM, structured SVM, or latent SVM) based on hand-designed (*e.g.*, HOG [3]) or learned features. Until recently, discriminative part-based (DPM) models [5] were particularly popular, due to their ability to compactly model appearance variations of objects while maintaining geometric relationships among parts [24]. However, with recent advances in deep learning, there has been a shift to methods that either obtain a set of object proposals (*e.g.*, using selective search [23]) and then use a CNN for classification (*e.g.*, R-CNN [7]), or use CNN models that are trained to directly regress bounding box along with object category label [22]. Our approach builds upon the newer R-CNN formulation [6] combined with SVM and probabilistic score re-scaling (similar to [11]) as the basic detection model. However, our main observations and conclusions are independent of this choice.

Contextual object detection: Relatively early, in vision, it was hypothesized that contextual relationships among objects in scenes are very important for recognition. A number of works modeled co-occurrence between object categories by detecting individual objects and modeling their relative locations (and scales) using spatial distributions. Gupta *et al.* [8] used prepositions and adjectives to relate nouns (objects); Hoiem *et al.* used geometric relationships to reason about location and scale of objects in street scenes [9]. For a specific class of human-object context, Yao *et al.* [25] proposed a joint DPM model for a person and manipulated object.

Scene understanding: More generic multi-object relationships were explored in [14], where groups of objects that geometrically co-occurred were mined and modeled. A discriminative holistic model for scene understanding that combined segments, objects and scene labels was introduced in [26]. Particularly relevant is the recent work on semantic image retrieval [11], that introduced the concept of scene graphs – a construct, closely related to scene parsing, design to represent objects (*e.g.*, *man*, *boat*), attributes of objects (*e.g.*, *boat is white*) and relationships between objects (*e.g.*, *man standing on boat*).

However, all of these methods neglect to model the change in the appearance of an object due to interaction with another object and/or the attribute it possesses. For example, *person sitting* may look very different and potentially easier to detect than *person* and *sitting* that happen to co-occur in the same spatial location. In other words, above methods assume appearance independence, in order to express the object-

object or object-attribute relationships using an MRF or CRF [1, 11, 26]. Our approach does not make this assumption, and instead tries to determine if joint appearance variation is useful and model it.

Phrases and visual composites: In an attempt to model induced object-object appearance changes, Malisiewicz *et al*. [15] introduced a *visual memex* model that modeled visual similarity and spatial context between object exemplars using a graph. The concept of visual phrases was introduced by Sadeghi *et al*. [20]. In [20] it was shown that training joint detectors for phrases (*e.g*., *man riding horse*), as opposed to individual objects (*e.g*., *man*, *horse*), resulted in better performance, despite fewer training instances being available to train each joint phrase classifier. This idea was further extended in [13] by discovering visual composites and their spatial relations, through sub-categorization.

One important observation is that while these methods showed that performance *on average* increases by jointly modeling object-object appearance, this is not the case for *all* object category pairs considered. Building on this intuition, in [19], a model for visual knowledge extraction and visual verification of relational phrases was introduced. To verify relational predicates (*e.g*., *fish*(*bear*, *salmon*)), a model that considers all combinations of detectors is considered (*e.g*., *bear*, *salmon*, *bear fishing*, *fishing salmon* and *bear fishing salmon*); the scores of all of these detectors are combined using a form of CRF on a factor graph.

Our method builds on similar intuition as [19], however, instead of building all possible partial detectors and combining their scores (which is expensive and potentially sub-optimal), we attempt to choose which of the detectors for the given relational predicate would be most accurate and train only those. As a consequence, our model is no more expensive to evaluate and train than a model that assumes independence, yet allows us to jointly model induced appearance variations.

Attributes: We also apply our model to attribute-object and attribute-scene relationships. Attributes have received a lot of attention in vision [4, 12] and tend to refer to namable mid-level semantic concepts related to object (person *sitting*) or scenes (*man-made*). Our work is most closely related to [18], where authors propose a method for determining whether for multi-attribute queries one should train independent classifiers (one for each attribute) or conjunctions of attributes. Importantly, they identify conjunctions to train without explicitly training all combinations. We take a conceptually similar approach but learn how to determine which "conjunction" classifiers to train (as opposed to relaying on inter- and intra- class variances) and apply our method to a broader set of problems.

3. Method

Given a visual composite (*e.g*., man holding a phone) we define the problem of choosing how to model and train the detector for this composite as *strategy selection*. We first make the empirical observation that non-uniform strategy selection can be beneficial (Section 3.2). Even though one strategy is dominant on *average*, some composites tend to perform better when modeled and trained using one of the alternative strategies. It is therefore intuitive that a careful selection of a training strategy *per composite* can boost performance, regardless of the task, as shown in Table 1. With this observation in mind, we aim to predict for each composite the training strategy that will result in optimal performance (Section 3.3). A reliable prediction is the performance measured on a validation set. However, pre-training all detectors for various strategies and cross-validating across them is computationally expensive. In order to avoid pre-training, we learn a proxy function using statistical features extracted from the training samples and validation results of previously observed composites. Given a new composite, we apply this function to select its training strategy.

3.1. Base model for detection

We formulate our base model following recent state-of-the-art methods for detection and classification. However, the exact form of the model is largely independent of our main findings. In particular, we represent each image (or an image patch) i using a feature vector $\mathbf{x}_i \in \mathbb{R}^{4096}$ from the last fully-connected layer (fc7) of a CNN network. We then train a detector using a linear SVM (similarly to the process described in [11]), and further calibrate the SVM scores to obtain probability estimates. The calibration is implemented using Platt scaling [17]: $P(y_c = 1|\mathbf{x}_i) = \frac{1}{1+e^{\alpha_c(\mathbf{w}_c^T\mathbf{x}_i+b_c)+\beta_c}}$, where α_c, β_c are the calibration coefficients; $\mathbf{w}_c$ and b_c are the learned SVM weights and bias, respectively, for class c.

For object detection, we use labeled bounding boxes containing object c as positive samples, and use CNN adapted for detection tasks [6] to compute $\mathbf{x}_i$. For scene classification, we use labeled full images containing scene c as positive samples, and a neural network fine-tuned for scene classification [27] compute $\mathbf{x}_i$ respectively. In both cases, negative patches/images are extracted from the training patches/images not containing c. For detection, we perform multiple rounds of retraining using hard negative mining for further learning refinement. When applying object detector, at test time, we use an object proposal scheme [23] to generate plausible hypotheses.

Note that if we want to build a detector for a relatively complex visual composite entity, *e.g.*, C = *man sitting*, we can do this using one of two ways: (1) assuming *independence*, i.e.,

$$P(y_C = 1|\mathbf{x}_i) = P(y_{man} = 1|\mathbf{x}_i)P(y_{sit} = 1|\mathbf{x}_i), \quad (1)$$

(2) by building a *joint* detector for the full visual phrase

$$P(y_C = 1|\mathbf{x}_i) = P(y_{man\ sitting} = 1|\mathbf{x}_i). \quad (2)$$

Detectors involved in (1) have the benefit of being trained from a larger set of samples, but may need to capture wider visual variances. Detector (2) has the benefit of modeling a presumably narrower visual variance, but could potentially lack sufficient number of samples to train a good model. As such, we posit that the performance of the two will be different at runtime, in general, and one will likely perform better than the other under certain conditions (see Table 1). Before studying the benefits of these strategies, we first more formally define the visual composite constructions that we use in the remainder of the paper.

3.2. Visual composites

We consider two types of visual composite $C \in \mathcal{C}$ consisting of up to 3 parts (or word tokens) of simple noun phrases: object-attribute or object-relationship-object phrases. Hence, each part/word token, $c_i \in \{\mathcal{O}, \mathcal{A}, \mathcal{R}\}$, can be a noun from a predefined set of noun object (or scene) categories $\mathcal{O}$ (*e.g.*, "man", "horse"), an adjective from a set of visual attributes describing objects (or scenes) $\mathcal{A}$ (*e.g.*, "tall", "bold", "open"), preposition and/or verb from a set of predefined object relationships $\mathcal{R}$ (*e.g.*, "in", "next to", "holding", "riding"). A visual composite is then either a pair of object and attribute (e.g., "red hat", where $\{c_1 = a =$ "*red*" $\in \mathcal{A}, c_2 = o =$ "*hat*" $\in \mathcal{O}\}$), or a triplet of object-relation-object (e.g., "man holding phone", where $\{c_1 = o_1 =$ "*man*" $\in \mathcal{O}, c_2 = r =$ "*holding*" $\in \mathcal{R}, c_3 = o_2 =$ "*phone*" $\in \mathcal{O}\}$).

In general, if we want to detect or ground C, given an image $\mathbf{x}$, we have a number of options available to us, as alluded to in the previous section. Specifically, for "man holding phone", if we treat each part/token *independently*, we obtain traditional formulation [11]:

$$\begin{aligned}\mathbf{b}_1^*, \mathbf{b}_2^* &= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_C = 1|\mathbf{x}) \\ &= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_{o_1} = 1|\mathbf{x}_{\mathbf{b}_1})P(y_{o_2} = 1|\mathbf{x}_{\mathbf{b}_2})P(\mathbf{b}_1, \mathbf{b}_2|r), \quad (3)\end{aligned}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bounding boxes for o_1 and o_2, respectively, $\mathbf{x}_{\mathbf{b}_1}$ and $\mathbf{x}_{\mathbf{b}_2}$ are corresponding CNN features of the image patches enclosed by these bounding boxes and $P(\mathbf{b}_1, \mathbf{b}_2|r)$ is a spatial distribution for relationship r (*e.g.*, a Gaussian mixture model [11]), designed to encode spatial consistency between two object patches. Alternatively, the problem can also be expressed using *joint* information in the following ways:

$$= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_{o_1} = 1|\mathbf{x}_{\mathbf{b}_1})P(y_{o_2+r} = 1|\mathbf{x}_{\mathbf{b}_2})P(\mathbf{b}_1, \mathbf{b}_2|r), \quad (4)$$

$$= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_{o_1+r} = 1|\mathbf{x}_{\mathbf{b}_1})P(y_{o_2} = 1|\mathbf{x}_{\mathbf{b}_2})P(\mathbf{b}_1, \mathbf{b}_2|r), \quad (5)$$

$$= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_{o_1+r} = 1|\mathbf{x}_{\mathbf{b}_1})P(y_{o_2+r} = 1|\mathbf{x}_{\mathbf{b}_2})P(\mathbf{b}_1, \mathbf{b}_2|r), \quad (6)$$

$$= \underset{\mathbf{b}_1,\mathbf{b}_2}{\operatorname{argmax}} P(y_{o_1+r+o_2} = 1|\mathbf{x}_{\mathbf{b}_1 \cup \mathbf{b}_2}), \quad (7)$$

where, for example, $P(y_{o_1+r} = 1|\mathbf{x}_{\mathbf{b}_1})$ and $P(y_{o_2+r} = 1|\mathbf{x}_{\mathbf{b}_2})$ are classifiers trained to detect *"man holding"* and *"holding phone"* phrases respectively. Note that Eq.(3)–(7) illustrate different, but equally valid, factorizations of the conditional distribution $P(y_C = 1|\mathbf{x})$, resulting in potentially different solutions for $\mathbf{b}_1$ and $\mathbf{b}_2$. The main difference among these factorization is the amount of assumed appearance independence (from complete – Eq.(3), to no independence – Eq.(7)).

Our overall goal is to determine which of the Eq. (3)–(7), that determine the learning strategy, would result in the most accurate model at testing time. Even more fundamentally, if this choice matters and when?

Training: To train the corresponding classifiers, we assume that we are working with a fully annotated dataset $\mathcal{D}_{\mathcal{OA}}$ of N training images, each image includes labeled B_i bounding boxes corresponding to objects. $\mathcal{D}_{\mathcal{OA}} = \{(\mathbf{b}_{i,j}, (o_{i,j}, \mathbf{a}_{i,j}))\}$, where $i \in [1, N]$ is the image index and $j \in [1, B_i]$ is one of the B_i annotated regions in image i. The variable $\mathbf{b}_{i,j} \in \mathbb{R}^4$ then denotes the bounding box of the corresponding image region and the pair $(o_{i,j}, \mathbf{a}_{i,j})$ denotes the object label $o_{i,j} \in \mathcal{O}$ and its (possible empty) set of attributes $a_{i,j,k} \in \mathcal{A}$ taken from all possible attributes $\mathcal{A}$. $k \in [1, K_{i,j}]$ is the index of the attribute from all $K_{i,j}$ attributes assigned to region j in image i. We use $\mathbf{l}_{i,j}$ to denote the object label- attributes pair. For example, for a region j in image i labeled "tall old person", the number of attributes $K_{i,j} = 2$ and $\mathbf{l}_{i,j} = (person, \{tall, old\})$.

In addition, each pair of bounding box annotations in a given image i can be associated with a set of relationship, such that $\mathcal{D}_{\mathcal{R}} = \{(\mathbf{b}_{i,j}, \mathbf{b}_{i,k}, \mathbf{r}_{i,j,k})\}$. For example, annotation "person holding and swinging the racket", would correspond to $\mathbf{r}_{i,j,k} = \{holding, swinging\}$. For scene-attribute scenario, the setting is somewhat simplified by effectively setting $\mathbf{b}_{i,j}$ to full images, leading to $\mathcal{D}_{\mathcal{SA}}$. All images (or image regions) that are not part of the positive training set are used as negatives for training a specific classifier. Overall, we consider the following choices:

– **scene-attribute**: we choose between $P(y_s = 1|\mathbf{x})$ $P(y_a = 1|\mathbf{x})$ and $P(y_{s+a} = 1|\mathbf{x})$ trained with samples of scene s with attributes a from $\mathcal{D}_{\mathcal{SA}}$.
– **object-attribute**: we choose between a product $P(y_o = 1|\mathbf{x_b})$ $P(y_a = 1|\mathbf{x_b})$ and $P(y_{o+a} = 1|\mathbf{x_b})$ trained with samples of object o with attribute a.
– **object-relationship-object**: where we choose among the choices denoted in Eq.(3)–(6) trained with respective data subsets from $\mathcal{D}_{\mathcal{SA}}$ and $\mathcal{D}_{\mathcal{R}}$.

Evaluation: Table 1 (left two rows) illustrate example performance results of the *independent* and the various *joint* strategy(ies) for individual object-attribute and object-relationship-object[1] composites. One key observation is that for certain composites one strategy performs significantly better than the other. Further, notice that both *independent* and *joint* training strategies are useful. In particular note that the optimal stategy is not necessarily the function of the object, but rather of the composite as a whole (*e.g.*, for "street-black" joint detector is better, where as for "street-paved" independent detector leads to 22% improvement). This raises the issue of how one should select the best strategy in each case, which we will address next.

3.3. Predicting Learning Strategy

As no single strategy for training detectors is better for all cases, there is a need to choose the optimal strategy for each composite phrase. One possibility for determining this is to use a subset of the data for *cross validation*. In this case all types of detectors are tested on the cross validation data and the best strategy is chosen as the detector. Table 1 shows the results of the cross validation strategy in (Validation) column, where "+" designates preference of the cross validation for the *joint* strategy and "-" designates preference for the independent strategy (the value itself measures strength of preference, which corresponds to the maximum absolute difference between the cross validation performance of the independent and the best joint strategy).

Using cross-validation, however, may be extremely time consuming and error prone. Specifically, for the case of object-relationship-object phrases we need to train 4 separate classifiers and evaluate them on a validation set to choose an appropriate strategy. Further, from a practical point of view, in many cases there may not be enough data for cross validation which may result in high variance in strategy selection. To address these issues, we propose a method for learning how to choose an optimal strategy without explicitly training all the detectors. To this end, we propose a simple regression scheme defined on a set of features extracted from the training sample sets.

Regression: Our key idea is simple, but surprisingly effective. We use cross validation on $\mathcal{C}_{tr}$ - a small fraction of composites (20%–30%), to learn to predict the preference between the two strategies. We do this by first training and measuring cross-validation performance for this small subset of composites. We then use a set of features of the corresponding training samples to regress the difference in cross-validation performance (the values in the (Validation) column of the Table 1). The learned regressor can then serve as a predictor on new composites to choose which strategy to use for the remainder of (80%–70%) the composites.

Let $S = \{I, J_1, J_2, .., J_M\}$ denote the set of $M + 1$ possible training strategies[2], where I and $\{J_m\}_{m=1}^{M}$ are the independent and (possibly multiple) joint strategies, respectively. Let $P_s(y_C|\mathbf{x})$ be the resulting trained classifier for composite C with strategy s. By applying $P_s(y_C|\mathbf{x})$ on a validation set we can obtain validation accuracy, which we denote $V_{s,C}$. Our goal is to train a regressor to predict $V_{s,C}$ from the features of the data sub-set, lets call it $\mathcal{D}_C$, used for training $P_s(y_C|\mathbf{x})$ directly. For this we define a feature mapping $\mathbf{f}_C = F(\mathcal{D}_C)$, discussed in detail in the next section. In practice, we predict the differences $V_{J_m,C} - V_{I,C}$, instead of each $V_{s,C}$, learning a linear prediction models, using Support Vectore Regression (SVR), as follows:

$$\mathbf{w}_m = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{C=1}^{|\mathcal{C}_{tr}|} (\mathbf{f}_C^T \mathbf{w} - [V_{J_m,C} - V_{I,C}])^2 + \lambda ||\mathbf{w}||_2^2, \quad (8)$$

where the second term is is a regularizer with a weight λ (we experimentally set $\lambda = 0.01$). Simply put, we are regressing the difference in performance between an independent and each available joint strategy, measured on a validation set.

Given the learned regression, when observing a new composite C_{new}, our model makes a selection of the strategy, $s_{new} \in S$ according to the following prediction rule, which no longer requires training of different strategies for C_{new} or evaluation of resulting classifiers on validation data:

$$s_{new} = \begin{cases} J_{m^*} & \text{if } \mathbf{f}_{C_{new}}^T \mathbf{w}_{m^*} > 0 \\ I, & \text{otherwise} \end{cases}$$
$$\text{where } \; m^* = \underset{m}{\operatorname{argmax}} \, \mathbf{f}_{C_{new}}^T \mathbf{w}_m. \quad (9)$$

The above formulation predicts *independent* strategy (I) when it is predicted to outperform all *joint* strategies, otherwise, joint strategy with largest predicted margin of improvement (J_m^*) is returned. Now we turn our attention to formalizing the features $\mathbf{f}_C$.

[1] For object-relationship-object joint strategy we report the best performance among 3 joint strategies considered.

[2] *e.g.*, object-attribute pairs share the same set of strategies.

	Performance		Prediction		
Composite Phrase	Independent	Joint	Validation	Our Model	Improvement
laptop-white	0.353	**0.409**	+0.173	-0.064	-15.8%
street-black	0.375	**0.608**	+0.133	+0.005	+62.1%
street-paved	**0.594**	0.486	-0.091	-0.092	+22.2%
hair-black	0.209	**0.234**	+0.030	+0.022	+11.9%
tracks-metal	0.383	**0.414**	+0.054	+0.001	+8.1%
bench-wood	**0.298**	0.196	-0.087	-0.025	+52.0%
man-holding-phone	0.241	**0.265**	+0.011	+0.018	+9.9%
man-behind-man	0.203	**0.230**	+0.047	-0.032	-13.3%
woman-next to-bus	**0.439**	0.328	-0.046	-0.022	+14.0%

Table 1: Examples of performance of the two learning strategies, joint or independent, for detectors on various visual phrases in SceneGraph dataset. Note that there is no clear strategy that outperforms the other on all phrases. Third column shows cross validation prediction and our method's predictions are in the fourth.

Feature extraction: Naively, one may think that number of samples is a sufficient indicator of which training strategy maybe useful (*i.e.*, few available samples implies independence, many samples are indicative of the joint preference). While number of samples is indeed a useful feature, it, by itself, is not sufficient as we will show in experiments (see Fig. 2 (Threshold)). One simple counter-example to explain why, is duplication of sample data (*e.g.*, having large number of duplicate or nearly duplicate training samples is no more informative than having a single sample).

We hypothesise that the *topology* of training examples for a given composite contains informative cues for the strategy selection. Our feature selection rises from the intuition that it should capture the trade-off between cardinality and compactness of the samples, as suggested by [18]. Recall that $\mathcal{D}_c$ is positive data subset for a single composite part. We use $\mathcal{D}_c$ to extract a feature vector $\mathbf{f}_c = [f_{c,1}, f_{c,2}, ..., f_{c,6}] \in \mathbb{R}^6$ comprising:

- Number of samples: $f_{c,1} = |\mathcal{D}_c|$
- Compactness of samples, represented as statistics extracted from pairwise cosine distances:

  $$\mathbf{r} = \{|\mathbf{x}_i - \mathbf{x}_j|\}, i \neq j, \mathbf{x} \in \mathcal{D}_c$$

  encoded by $f_{c,2} = max(\mathbf{r}), f_{c,3} = min(\mathbf{r}), f_{c,4} = med(\mathbf{r}), f_{c,5} = mean(\mathbf{r})$.
- Sample entropy, estimated using Nearest neighbour distance approximation:

  $$f_{c,6} = \frac{1}{N}\sum_{i=1}^{N} ln(N\rho_i) + ln2 + \gamma,$$

  where $\rho_i = min_{j \neq i}|\mathbf{x}_i - \mathbf{x}_j|$; γ is Euler const.

We repeat this process for each of the parts and corresponding pairwise composites and concatenate the resulting features. For example, for $C = \{white, boat\}$ we have $F(\mathcal{D}_C) = [\mathbf{f}_{boat}, \mathbf{f}_{white}, \mathbf{f}_{white+boat}]$.

The last column of Table 1 shows the predictions made by the learned regressor. Note, that the regressor was trained on composites other than the ones listed in the table, so it has not seen these composites during training. Even so, the regressor is able to predict the sign (which corresponds to the preference of strategy) in 7 out of the 9 cases and in the two cases where it didn't, the difference between the strategies was small.

4. Experiments

Datasets: We evaluate our method for variety of detection and classification tasks. We use two public datasets. For detection and grounding, we use SceneGraph dataset [11]; for scene classification, SUN [16].

SceneGraph dataset [11] consists of $5,000$ images containing large number of object, attribute and relationship annotations (see [11] for statistics). We use $4,000$ images for training and $1,000$ for testing using the split prescribed in [11]. We test our model in object localization task on object-attribute, object-attribute-relationship and object-relationship-object composite queries. To quantitatively compare performance we report both median and mean Intersection over Union (med IoU and mean IoU) as well as the fraction of instances with IoU above various thresholds (IoU@t).

SUN dataset: We use SUN Attribute dataset [16] comprising $14,340$ images from 707 scene categories and annotated with 102 discriminative attributes. In addition, to get more samples for each scene category, we augment each scene class in SUN Attribute with up to additional 80 images from the full SUN dataset (less if 80 is unavailable). We test our model in the context of scene-attribute retrieval task, where given a scene-attribute classifier we rank all test images in terms of their relevance to this query. We report performance using Mean Average Precision (mAP) computed on 100 top-ranked images. We use 5 images from each scene-attribute composite pair for testing, rest for training.

Baselines: We implemented a number of baseline selection strategies in order to illustrate the benefits of our regression-based selection approach:
– *Optimal*: An oracle selection that results in the highest performance at run-time. This is an upper bound on improvement in performance that can be achieved by choosing best strategy per composite.
– *Cross-validation*: Selection that consists of pre-training all strategies per composite, then selecting one that scores best on the validation set.
– *Independent*: Using an independent detector strategy for each and every composite. For object localization this is equivalent to model in [11].
– *Joint*: Using a joint phrase [20] detector for each and every composite.

Where appropriate, we also use additional baselines consisting of threshold selection strategy on the number of joint samples and strategy proposed in [18].

Implementation and setup: For all baselines we use the same image features, form of the detector, and the training procedure as we do for our regression-based method. We evaluate our method using 3-fold cross-validation, where regressor is trained on 30% of all composites, then evaluated on the rest 70% to choose among the independent and joint variants. The chosen variants are then trained and tested[3]; reported results are averaged across the 3-folds. For stability we repeating the experiments multiple times and average. Since it doesn't make sense to train a joint detector with very few samples, we prune composites with less training examples than a certain threshold. We plot performance as a function of this threshold (otherwise we use all composites with more than 3 joint examples).

4.1. Detection and Retrieval on SceneGraphs

Object-Attribute Detection: We extract a total of $2,295$ ($|\mathcal{C}| = 2,295$) object-attribute composites, most frequently appearing in the SceneGraphs data-set [11]. After choosing a joint/independent strategy for each composite, we train a classifier accordingly. The detector works by first computing a set of object bounding box proposals, using [23], and then evaluating the probability of each proposal containing an object-attribute pair using the trained classifier. We report Average IoU over top 5 most confident detections in Fig. 2 (left).

From Figure 2, we can see that when there are many composites with low number of samples, the independent strategy performs better on average. This trend changes as the number of joint samples increases from left to right. Note that the potential gain from selecting the correct strategy is significant, as shown by the optimal curve. Learning from 0.3 of the data, we reach performance of cross-validation at a fraction of computational cost. Using a bigger portion of data for training does not lead to significant improvement (see Supplementary Material). While average improvement may appear small, we do get large improvements for certain individual composites. For example, in Table 1 we show that improvement for "street-black" is 62.1%; for "bench-wood" 52%. We can see a degradation in performance as we focus on composites with larger number of samples (a more atypical case), due to the decrease in the number of training composites, which results in regressor overfitting.

[3]Only final testing is done on the test split, rest on training.

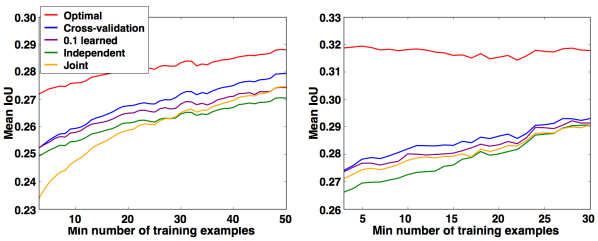

Figure 2: **Detection Performance**: for composite Object-Attributes (left) and Object-Relationship-Object (right). Performance is measured in Mean IoU. Our regression-based selection method learned from 30% of composites is denoted by "0.3 learned". Performance is reported as a function of threshold used to filter out composites with fewer samples than the threshold shown along the x-axis.

Object-Relationship-Object Detection: We consider $4,030$ composites $C = \{o_1, r, o_2\}$ appearing both in train and test set. This results in 303 $\{o_1 + r\}$ pairs (*e.g.*, man-holding) and 317 $\{r + o_2\}$ pairs (*e.g.*, holding-phone). In addition to an independent strategy in Eq.(3), we consider 3 joint strategies, listed in Eq.(4)–(6), thus training 3 different regressors. We also train a spatial relationship term $P(\mathbf{b}_1, \mathbf{b}_2|r)$ for each relationship r as discussed in Sec. 3.2. We again use selective search to obtain object proposals. We evaluate pairs of object proposals by evaluating a product of object classification terms and the spatial term. We measure performance using average IoU for the object pair. We weigh top 5 pairs using corresponding probabilities. Results are in Fig. 2 (right).

We see a similar trend to the previous experiment, where our model is comparable to the cross-validation baseline. Notable difference is that for object-object relationships joint strategy seems to be preferred; this is consistent with results from Sadeghi and Farhardi [20]. We can observe that cross-validation fails to predict test accuracy well (optimal selection is much higher).

Experiment	Joint	Threshold @0.25	@0.5	@0.75	Independent	Multi-Att[18]	Our Method
obj-att	0.234	0.238	0.243	0.247	0.249	0.234	**0.252**
obj-rel	0.271	0.270	0.270	0.269	0.266	-	**0.274**
scene-att	0.155	0.156	0.157	0.159	0.161	0.155	**0.167**

Table 2: **Comparison of selection strategies:** Our approach consistently outperforms all baselines in all experiments. For obj-att and obj-rel the performance is reported in Mean IoU and for scene-att in mAP.

	Obj-attr		
	Independent/[11]	Joint	Our method
Med IoU	0.059/0.026	0.054	**0.064**
R@0.1	0.466/0.447	0.463	**0.471**
R@0.3	0.315/0.341	0.315	**0.322**
R@0.5	0.188/0.234	0.190	**0.193**
	Obj-attr-rel		
	Independent/[11]	Joint	Our method
Med IoU	0.075/0.067	0.066	**0.082**
R@0.1	0.476/0.476	0.473	**0.483**
R@0.3	0.321/0.357	0.321	**0.328**
R@0.5	0.188/0.239	0.194	**0.196**

Table 3: **SceneGraph retrieval:** Illustrated using only object and attributes (top) and including relationships (bottom). For independent baseline we report our re-implementation and original results [11].

This, in turn, has a big negative effect on our model.

Retrieval and Comparison to [11]: For more direct comparison to [11] we reproduce the object localization evaluation setup in [11][4]. We are given a test image, a set of object proposals and a corresponding query represented as a *Scene Graph*, *i.e.*, object nodes, possibly described by attributes, connected by relationship edges. Our task is one of grounding object nodes with respect to bounding box proposals, which is formulated as inference in the *Scene Graph* induced CRF. We train each $\{o, a\}$ and $\{o_1, r, o_2\}$ using the strategy selected by our model, and compare to Independent (which is equivalent to [11]), and Joint strategies applied to all composites. Results, using the metrics in [11], are reported in Table 3 for both object-attribute and object-attribute-relationship queries. We note that while out results for performance of [11] are slightly lower at higher IoU precision, they are overall (in terms of Median IoU) much better than what was reported in [11] (by as much as 125% for obj-attr).

We observe a consistent improvement when using our model, compared to both uniform joint and independent strategies. As shown in Table 3, we improve median IoU by 8% when using only objects and attributes, and by 9% when using full scene graphs. Some visual examples of improved localization are shown in Supplemental Material.

[4]We reimplemented using guidelines of the author.

4.2. Scene classification on SUN dataset

We consider $5,071$ ($|\mathcal{C}| = 5,071$) scene-attribute composites, $C = \{s, a\}$, found in the SUN Attribute dataset. Results of scene-attribute query retrieval are illustrated in Table 2 (bottom row). A more thorough report is given in the supplemental material, where we also show performance of regressors trained with 0.1, 0.3, and 0.5 fraction of the composites (tested on the rest). These experiments illustrate that we can outperform baseline strategies (and nearly match cross-validation performance) with as little as 10% training.

Learned Selection Strategy Evaluation: We compare our selection strategy to a number of alternatives, and illustrate performance across datasets and all tasks in Table 2. As can be seen, our method consistently outperforms uniform independent and joint selection, as well as multi-attribute selection approach of [18].

One may argue that simply using a threshold on the number of samples and choosing to train the joint detector when enough samples are available is a simple and effective strategy. We compare to such strategies using different thresholds ("Threshold" columns in Table 2) and show that our method outperforms these as well. These results illustrate that number of samples by itself is an insufficient criterion, and there is a need to weigh in the sample statistics. This is precisely what our regression model learns.

It should be noted that our performance, although achieves small absolute gains, is very close to cross-validation result (which is effectively our upper bound) across different datasets and experiments.

Computational efficiency: Our method has a negligible cost for training and selection prediction. For example, it took us 5 minutes on average to cross-validate performance of a single composite detector, and less than 2 seconds to extract features from a single composite and do prediction (a saving of 99.3% of time).

Conclusions: In this paper, we observed that a individual selection of training strategy for visual composites can be highly beneficial. We showed that an effective selection proxy function can be learned with small amount of data, achieving high (similar to cross-validation) performance at low computational cost. We note that more sophisticated variants of Eq.(8) may further improve performance and should be explored.

References

[1] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

[2] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010.

[6] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[8] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008.

[9] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[10] H. Izadinia, F. Sadeghi, and A. Farhadi. Incorporating scene context and object layout into appearance modeling. In *CVPR*, 2014.

[11] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015.

[12] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot learning of object categories. *PAMI*, 2013.

[13] T. Lan, M. Raptis, L. Sigal, and G. Mori. From subcategories to visual composites: A multi-level framework for object detection. In *ICCV*, 2013.

[14] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *CVPR*, 2012.

[15] T. Malisiewicz and A. A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, 2009.

[16] G. Patterson, C. Xu, H. Su, and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014.

[17] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74, 1999.

[18] M. Rastegari, A. Diba, D. Parikh, and A. Farhadi. Multi-attribute queries: To merge or not to merge? In *CVPR*, 2013.

[19] F. Sadeghi, S. K. Divvala, and A. Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *CVPR*, 2015.

[20] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

[21] J. M. Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371–391, 1995.

[22] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013.

[23] J. R. R. Uijlings, K. E. A. Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[24] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *CVPR*, 2011.

[25] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[26] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[27] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems 27*, pages 487–495. 2014.