From Red Wine to Red Tomato: Composition with Context

Ishan Misra Abhinav Gupta Martial Hebert
The Robotics Institute, Carnegie Mellon University

Abstract

Compositionality and contextuality are key building blocks of intelligence. They allow us to compose known concepts to generate new and complex ones. However, traditional learning methods do not model both these properties and require copious amounts of labeled data to learn new concepts. A large fraction of existing techniques, e.g. *using late fusion, compose concepts but fail to model contextuality. For example, red in red wine is different from red in red tomatoes. In this paper, we present a simple method that respects contextuality in order to compose classifiers of known visual concepts. Our method builds upon the intuition that classifiers lie in a smooth space where compositional transforms can be modeled. We show how it can generalize to unseen combinations of concepts. Our results on composing attributes, objects as well as composing subject, predicate, and objects demonstrate its strong generalization performance compared to baselines. Finally, we present detailed analysis of our method and highlight its properties.*

1. Introduction

Imagine a blue elephant. Having never seen a single example of such a creature, humans have no difficulty imagining it, or even recognizing it. Starting from Plato's Theaetetus to the early nineteenth century work of Gottlob Frege [20], compositionality is often regarded as a hallmark of intelligence. The core idea is that a complex concept can be developed by combining multiple simple concepts. In fact, the same idea has been explored in the field of computer vision as well: in the form of attributes [13, 54] or graphical models for SVOs (subject-object-verb triplets) [53]. While the idea of building complex concepts from simpler ones seems intuitive, current state-of-the-art methods for recognition or retrieval follow a more data-driven approach, where complex concepts are learned using hundreds and thousands of labeled examples instead of being composed. Why is that?

Interestingly, even in philosophy, there is clear tension between the idea of compositionality and the principle of contextuality. The principle of contextuality states that one cannot create a model of a simple concept without the context. This has often been stated as one of the main arguments against attributes: a red classifier in red wine is remarkably different from a red classifier in red tomato or even a red car. Figure 1 shows more such examples.

Figure 1: Visual concepts like objects and attributes are compositional. This compositionality depends on the context and the particular instances being composed. A small elephant is much larger than a small snake! Our surprisingly simple method models both compositionality and contextuality in order to learn visual classifiers. The results of our approach show that it composes while respecting context.

This direct tension between compositionality and contextuality leads to the basic exploration of this paper: do current vision algorithms have such a compositional nature? Can we some respect the principle of contextuality yet create compositional visual classifiers? One approach to capture context is to use the text itself to learn how the modifiers should behave. For example, a modifier like "red" should show similar visual modifications for related concepts like tomatoes and berries. Approaches such as [38] have tried to use text to capture this idea and compose visual classifiers. But do we really need taxonomy and knowledge from language to capture contextuality?

In this paper, we propose an approach that composes classifiers by directly reasoning about them in model space. Our intuition behind this is that the model space is smooth and captures visual similarity, *i.e.*, tomato classifiers are closer to berry classifiers than cars. Thus, modifiers should apply similarly to similar classifiers. One task we consider is composing attribute (adjective) and object (noun) visual classifiers to get classifiers for (attribute, object) pairs. As Figure 1 shows, the visual interpretation of attributes depends on the objects they are coupled with, *e.g.*, a small elephant is still much larger than a small snake. Our approach respects such contextuality because it is conditioned on *all* the visual concepts and models them together, rather than in isolation. We show that our compositional transform captures such relations between objects and attributes, and can create visual classifiers for them.

As our experiments show, our approach is able to generalize such compositionality and contextuality to *unseen combinations* of visual concepts (Section 4. Our approach naturally extends beyond composing two primitives. We show results on combining subject, object and verb classifiers to unseen combinations of subject-verb-object triples (Section 4.3). On all these tasks, our method shows generalization capability beyond existing methods. Section 5.5 also shows our method's generalization to unseen primitives. Finally, in Section 5, we analyze the various components of our method and its various properties.

2. Related Work

Our work is heavily influenced by the principle of compositionality. This principle has a long standing history in the fields of philosophy, theory of mind, neuroscience, language, mathematics, computer science, *etc*. As such a broad overview is beyond the scope of this paper, we focus on compositionality in the case of visual recognition. In its most basic form, the principle states that new concepts can be constructed from primitive elements. This principle is relevant for statistical learning as it paves the way for models that train with low sample complexity. Compositional models can learn primitives from large amounts of samples and then compose these primitives to learn new concepts with limited samples [18, 23, 63, 68].

One of the earliest examples of using compositonality for visual recognition is Biederman's Recognition-By-Components theory [6] and Hoffman's part theory [27]. Compositionality is an underlying principle for many modern visual recognition systems [34]. Convolutional Neural Networks [37] have been shown to capture feature representations at multiple semantic and part hierarchies [64]. The parts-based systems such as Deformable Part Models [15], grammars [25, 44, 59, 62, 67, 69], and AND-OR graphs [54, 61, 66] also rely on compositionality of objects to build recognition systems. Compositionality has also been a key building block for systems used for visual question answering [4, 5], handwritten digits [32, 33], zero-shot detection [31], segmentation and pose estimation [59, 62].

In this paper, we focus on compositionality to compose unseen combinations of primitive visual concepts. This has been classically studied under the zero-shot Learning paradigm [35]. Zero-shot learning tries to generalize to new visual concepts without seeing any training examples. Generally, these methods [2, 3, 9, 35, 41, 50, 65] rely on an underlying embedding space, such as attributes, in order to recognize unseen categories. It is assumed that the description of the unseen categories is explicitly known in the underlying embedding space. As such explicit knowledge is not always available, another line of work [10, 21, 38–40, 49] relies on the finding such similarity in the linguistic space. Specifically, they leverage distributional word representations to capture some notion of taxonomy and similarity. However, in this paper, we do not make assumptions about the availability of such a common underlying embedding or an external corpus of knowledge. Our aim is to explore compositionality purely in the visual domain.

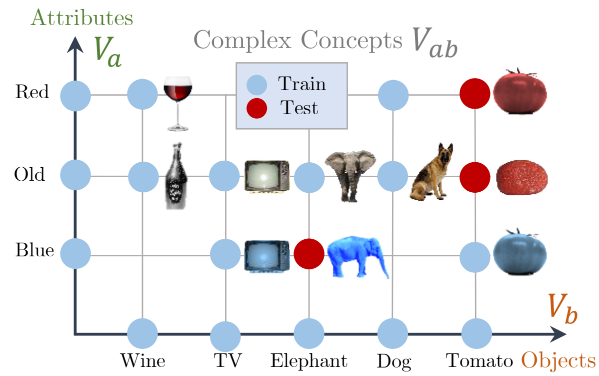

Figure 2: We assume that complex visual concepts can be composed using primitive visual concepts. By observing a few such complex concepts and their constituent visual primitives, we aim to learn a compositionality transform that can compose unseen combinations of primitives.

Another area related to our work is that of transfer learning [7–9, 45, 47, 56], feature embeddings [30, 52] and low-shot recognition [14, 16, 17, 58]. These methods generalize to new categories by utilizing knowledge gained from familiar categories. Like our method, they rely on the visual similarity of unseen classes in order to generalize existing classifiers or features. However, unlike our method, these approaches need training examples of the 'unseen' classes. We build upon the insight from [60] that meaningful transformations can be learned directly in the model space without external sources of knowledge.

We study compositionality in visual recognition in the context of two well known problems - objects and attributes [13, 28, 29, 46, 48], and subject-verb-object (SVO) phrases [24, 39, 53, 57]. Both these problems capture compositionality of primitive visual concepts. Contextuality is an important aspect of composing primitives in both these problems and leads to varied visual appearance: *e.g.*, small elephant vs. small snake or person sitting on chair vs. person sitting on sofa. As has been noted in [39, 53], annotations for composite or complex visual concepts are far fewer in number than for primitive concepts. Thus, our work has important practical applications as it can compose visual

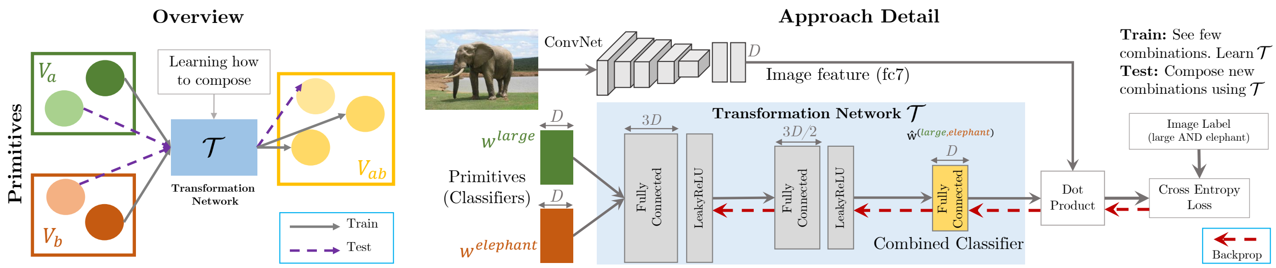

Figure 3: Our method composes classifiers from different types of primitive visual concepts. At training time, we assume access to a limited set of combinations, *e.g.*, `(large, elephant)` of these primitives. We model each of these primitives by learning linear classifiers (w) for them. We then learn a transformation network that takes these classifiers as input and composes them to produce a classifier for their combination. At test time, we show that this transformation can generalize to unseen combinations of primitives (Sections 4.2 and 4.3) and even unseen primitives (Section 5.5).

primitives to recognize unseen complex concepts.

3. Approach

Our goal is to compose a set of visual concept primitives to get a complex visual concept as output. As a simple example, shown in Figure 2, consider the complex concepts spanned by combinations of attributes and objects. Given existing classifiers for an attribute `large` and an object `elephant`, we want to learn to compose them to get a classifier for the (attribute, object) pair of `large elephant`.

We represent our visual primitives by classifiers trained to recognize them. We then learn a transformation of these input visual classifiers to get a classifier that represents the complex visual concept. As shown in Figure 3, we parametrize our transformation by a deep network which accepts linear classifiers of the primitives as inputs and produces a linear classifier of the complex visual concept as output. As multi-layered networks can capture complex non-linear functions, we hope that such a network can learn to compose visual primitives while capturing contextuality. We show that such a network can generalize to unseen combinations of visual primitives and compose them.

3.1. Intuition

Evidence for visual compositionality exists in neuroscience and has been widely studied [1, 19]. Our intuition behind composing directly in the classifier space is that classifiers themselves represent visual similarity, *e.g.*, a classifier of an elephant is closer to an animal classifier rather than a plate classifier. Thus, classifiers of 'unseen' combinations of classes can be composed by looking at 'seen' combinations using this visual similarity in classifier space.

3.2. Approach Details

We now describe our approach on how to compose complex visual classifiers from two or more simple visual classifiers. Without loss of generality, we will explain the details of our approach for the case of combining two classifiers but our approach can generalize to combine more types of primitives as demonstrated in our experiments.

Let us assume that we want to combine two different types of primitives. We represent these sets of primitives by $(\mathcal{V}_a, \mathcal{V}_b)$. These primitives are composed to form a complex primitive represented as $\mathcal{V}_{ab}$. As an example, consider $\mathcal{V}_a$ as the set of attributes and $\mathcal{V}_b$ as objects and thus $\mathcal{V}_{ab}$ consists of complex concepts formed by attribute, object pairs. We use $a, b, (a, b)$ to denote elements in $\mathcal{V}_a, \mathcal{V}_b, \mathcal{V}_{ab}$ respectively. Continuing our analogy of attributes and objects, $a, b, (a, b)$ can represent `large`, `elephant`, and `(large, elephant)`. We assume our vocabulary consists of M primitives of first type ($\mathcal{V}_a$) and N primitives of second type ($\mathcal{V}_b$). We also assume we have training data for some K complex concepts which combine one of M and N primitives.

We first train a linear classifier (SVM) for every type of primitive. Therefore, the primitive is parametrized by the weight vector for the linear classifier. Using the training data, we obtain weight vectors of $M + N$ primitives. Let us represent the weight vector for primitives $a \in \mathcal{V}_a, b \in \mathcal{V}_b$ as w_a, w_b. We can also train SVMs w_{ab} for complex concepts $(a, b) \in \mathcal{V}_{ab}$ using the training data available. However, since the training data for (a, b) pairs is limited compared to training data for a and b individually, directly training individual w_{ab} classifiers is difficult (see Section 4 for experiments). Instead, we want to use w_a and w_b to directly learn about the complex concept (a, b) *without* looking at w_{ab}.

As Figure 3 shows, we want to learn a function $\mathcal{T}$ such that it transforms the weights of two primitives (w_a, w_b) and outputs the weight of the complex concept (a, b):

$$\hat{w}_{ab} = \mathcal{T}(w_a, w_b). \tag{1}$$

Our training data contains the pairs (w_a, w_b). However, at training time, we do not have all possible combinations of (a, b) but very few combinations ($K << MN$). In order to detect unseen combinations at test time, we want to use compositionality and learn to combine two different primitives a and b to get the combinations (a, b).

We use a multi-linear perceptron to parameterize the function $\mathcal{T}$ and describe the architecture and the loss func-

tion.
Architecture: The transformation network $\mathcal{T}$ is a feedforward network with three fully connected layers. We use the LeakyReLU [26] non-linearity in between the layers. Given n SVMs (for n primitive concepts) each of dimensionality D as input, the output sizes of the three layers are $(n+1)D$, $(n+\frac{1}{2})D$ and D.
Loss Function: We compute the score between the output of the transformation $\mathcal{T}$ and the input image features $\phi(I)$.

$$p = \text{sigmoid}\left(\mathcal{T}(w_a, w_b)^\top \phi(I)\right)$$

This score reflects the compatibility between the model transformation and the image. We want this score to be high only if the image contains the complex concept (a, b) and low otherwise. As an example, we want the score to be high only for `(large, elephant)` and want it to be low for an image containing either `elephant` or `large` (not both), or neither of the two concepts.

We train the parameters of the transformation network $\mathcal{T}$ to minimize the binary cross entropy loss

$$L(I, w_a, w_b) = y\log(p) + (1-y)\log(1-p), \quad (2)$$

where the image label y is 1 only if the image has the complex concept (a, b) present. During training, we train a single transformation network using positive/negative images from various combinations of the primitives.

3.3. Implementation Details

We use linear SVMs, *e.g.*, w_a, w_b trained on `fc7` layer representations from the VGG-M-1024 [55] ConvNet. This ConvNet was pre-trained on the ImageNet dataset [51]. The input classifiers are 1024 dimensional each. The transformation network $\mathcal{T}$ consists of 3 fully connected layers with LeakyReLU [26] non-linearities. We set the slope of LeakyReLU to 0.1. We do *not* update the weights of the ConvNet to ensure a fair comparison to the baselines.

At test time, we first feed in the primitive tuples and cache the classifiers of the complex concepts. Given an image, we then run a single forward pass to get the image features and compute the scores using the cached classifiers.

4. Experiments

We now quantify the performance of our method on benchmark datasets. We do so in two settings - 1) compose object and attribute classifiers on the MITStates dataset [28]; and 2) compose three primitives subject, predicate and object classifiers on the Stanford VRD dataset [39].

4.1. Common Setup

We first describe the common experimental setup used for these set of experiments.
Metrics: Following [35], we measure the multi-class classification accuracy over the classes in the test split. Existing datasets are not exhaustively labeled [39, 43] in terms of combinations of visual concepts, *i.e.*, a `modern city` can also have `narrow streets` and one or both of these labels can be missing. To account for this, we follow [51] and use the top-k classification accuracy metric. We also report mean Average Precision [11] (mAP) by computing average precision for each of the classes and taking the mean.
Features and Classifiers: We use the `fc7` representation from the VGG-M-1024 network [55] pre-trained on ImageNet [51]. We learn our base visual classifiers (w_a, w_b) as linear SVMs on these `fc7` features and choose SVM parameters by 4 fold cross-validation using liblinear [12].
Training Details: We describe the architecture for the transform network $\mathcal{T}$ in Section 3.3. We train it for $220k$ iterations with a mini-batch size of 256 and momentum 0.9, with a learning rate of 0.01 dropping by a factor of 10 after $200k$ iterations. We form each minibatch with a ratio of 25% positive examples sampled uniformly in the space of the complex visual concepts. The ConvNet weights are not updated for fair comparison. The supplementary material contains additional experiments with end-to-end learning.
Evaluation Setting: As our method does not assume prior knowledge about unseen primitives or complex objects, it is not possible to compare against traditional zero-shot learning methods similar to [35]. Instead, we compare against methods that can directly 'compose' in zero-shot settings without knowing relations to unseen classes at training time.
Baselines without compositionality or contextuality: These baselines do not model compositionality or contextuality explicitly and work directly on the predictions of the base classifiers w_a, w_b. We denote them as:

- **Individual**: This set of baselines does not use compositionality. The probability of the complex concept (a, b) being present in an image is considered to be the probability of only one of the primitives a or b, *i.e.* $p(a, b) = p(a)$ or $p(a, b) = p(b)$. For three primitives a, b, c we can also consider pairs formed by leaving one primitive out, *e.g.*, $p(a, b, c) = p(a)p(b)$ *etc.*
- **Visual Product**: This baseline is inspired from the VisualOnly method from [39]. It does not model contextuality and just 'composes' outputs of classifiers for the primitives, by computing their product, *i.e.*, $p(a, b) = p(a)p(b)$. It can be thought of as late fusion. Unfortunately, since the detectors or training code for [39] were not available at the time of submission, we are unable to directly compare against their implementation/results.

Baselines composing without visual classifiers: These baselines use word embeddings to capture visual similarity, *e.g.*, word embedding of `animal` is closer to `elephant` than `paper`. They compose using word embeddings of labels, rather than visual classifiers.

- **Label Embeddings (LE)**: This baseline is inspired from the work of [10, 38]. To implement this method, we modify our approach to compute the transform $\mathcal{T}$ on embed-

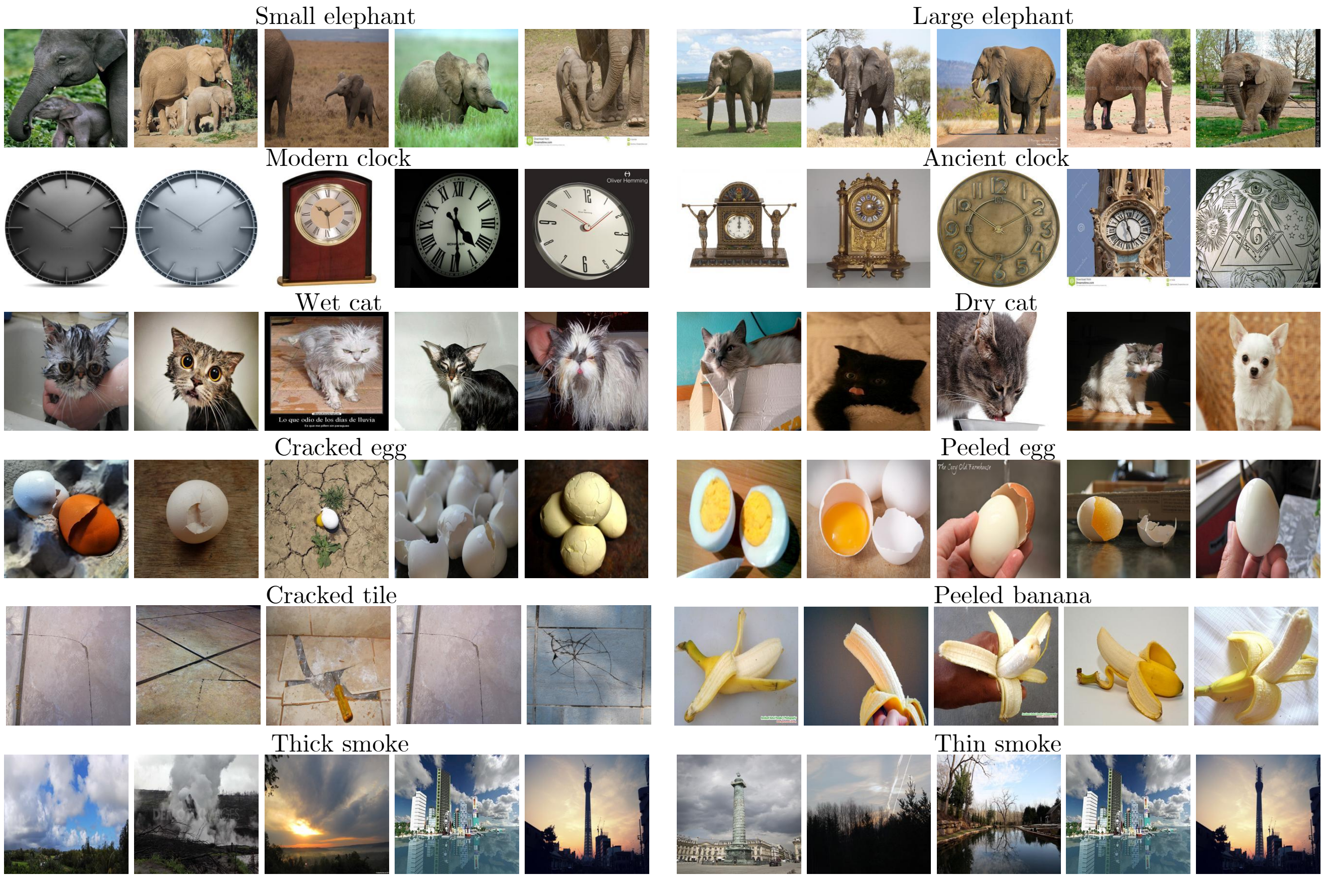

Figure 4: We show the top retrievals on the MITStates dataset [28]. These retrievals are computed on unseen combinations of (attribute, object) pairs. We see that our method learns to compose attributes and objects while respecting their context. The last row shows some failure cases of our method.

dings of the visual primitives, rather than the classifiers. We use the exact same network to compute $\mathcal{T}(e_a, e_b)$, where e_a is an embedding of the primitive a. We use a 300 dimensional word embedding [42] learned using an external corpus (Google News).

- **Label Embeddings Only Regression (LEOR)**: This baseline is inspired from the work of [10, 53, 60]. It is implemented similar to Label Embeddings, except for the loss function. We implement the loss function as a regression to the classifier for the complex visual concept, *e.g.*, a SVM trained on (attribute, object) pairs. Thus transform $\mathcal{T}(e_a, e_b)$ is trained to minimize the euclidean distance to the classifier w_{ab} where w_{ab} is the SVM trained directly on (a, b) pairs.
- **Label Embeddings With Regression (LE+R)**: This baseline combines the loss function from the LE and LEOR baselines and can be viewed as a variation of [10].

4.2. Composing objects and attributes

In this section, we learn the transformation on two sets of visual primitives: objects and attributes on the MITStates dataset [28]. We first describe more details about the experimental setup and then present results.

Task: We consider the task of predicting a relevant (attribute, object) pair for a given image in the test set, in an image classification setting. Our test set has (attribute, object) pairs that have never been seen together in the training set. We ensure that all objects and attributes appear *individually* in the training split. We use the unseen (attribute, object) pairs for evaluation using Average Precision and top-k accuracy as described in Section 4.1.

Dataset: We use the MITStates Dataset [28] which has pairs of (attribute, object) labels for images (one label per image). It has 245 object classes, 115 attribute classes and about 53k images. We randomly split the dataset into train and test splits such that both splits have *non-overlapping* (attribute, object) pairs. The training split consists of 1292 pairs with 34k images, and the test set has 700 pairs with 19k images. Thus, the training and test set have non-overlapping combinations of visual concepts ($\sim 35\%$ unseen concepts) and are suitable for 'zero-shot' learning. We

Table 1: Evaluating on unseen (attribute, object) pairs on the MITStates Dataset [28]. We evaluate on 700 unseen (attribute, object) pairs on 19k images.

	AP		Top-k Accuracy		
		$k \rightarrow$	1	2	3
Chance	-		0.14	0.28	0.42
Indiv. Att.	2.2		-	-	-
Indiv. Obj.	9.2		-	-	-
Visual Product	8.8		9.8	16.1	20.6
Label Embed (LE)	7.9		11.2	17.6	22.4
LE Only Reg. (LEOR)	4.1		4.5	6.2	11.8
LE+Reg. (LE+R)	6.7		9.3	16.3	20.8
Ours	**10.4**		**13.1**	**21.2**	**27.6**

provide more details in the supplementary material.

Baselines: We use the baselines described in Section 4.1. We denote the 'Individual' baselines for attributes and objects as 'Indiv Att.' and 'Indiv Obj.' respectively.

Quantitative Results: We summarize the results for our method and the baselines on the MITStates dataset in Table 1. We use the *unseen* attribute, object pairs for evaluation. The 'Indiv' baseline methods that do not model both compositionality or contextuality show poor performance. This is to be expected as predicting a `(large, elephant)` using only one of `large` or `elephant` is rather ill-posed. The Indiv Att. baseline performs the worst. We believe the reason is that attribute images are visually very diverse compared to objects (also noted in [48]).

Table 1 also shows that the Visual Product baseline gives strong performance in AP, but does not perform well on top-k accuracy. The LE baseline, has the opposite behavior, which suggests that using multiple metrics for evaluation is helpful. We observed that methods with high AP/low accuracy tend to get the object correct while methods with high accuracy/low AP tend to get (attribute, object) pairs correct but generally get objects wrong. Our method shows significant improvement over baselines across all metrics.

We also see that the LEOR and LE+R baselines both have worse performance than the LE baseline. This suggests that using regression to w_{ab} in the loss function is not optimal. On further inspection, we found that the $w_{\text{attribute,object}}$ are poorly trained because of the very few positive examples for (attribute, object) pairs, compared to the larger number examples available for attributes and objects individually. Thus, regressing to these poorly trained classifiers hurts performance. We further explore this in Section 5.1.

Qualitative Results: Figure 4 shows some qualitative results of our method. For the *unseen* pairs of attributes and objects, we use our transformation $\mathcal{T}$ to predict a classifier and retrieve the top results on the test set. Our model shows both compositionality and contextuality for these concepts. It also shows that our model understands the different 'modes' of appearances for objects. Additional qualitative results are presented in the supplementary material.

We present further results (combining unseen primitives *etc*.) and analysis of our method on this dataset in Section 5.

Table 2: Evaluating subject-predicate-object predictions on unseen tuples. We use the StanfordVRD Dataset [39] with 1029 unseen tuples over 1000 images for evaluation.

	AP		Top-k Accuracy		
		$k \rightarrow$	1	2	3
Chance	-		0.09	0.18	0.27
Indiv. Sub.	2.9		-	-	-
Indiv Pred.	0.4		-	-	-
Indiv. Ob.	3.7		-	-	-
Indiv Sub. Pred.	2.9		-	-	-
Indiv Pred. Ob.	3.6		-	-	-
Indiv Sub. Ob.	4.9		-	-	-
Visual Product	4.9		3.2	5.6	7.6
Label Embed (LE)	4.3		4.1	7.2	10.6
LE Only Reg.(LEOR)	0.9		1.1	1.3	1.3
LE+Reg. (LE+R)	3.9		3.9	7.1	10.4
Ours	**5.7**		**6.3**	**9.2**	**12.7**

4.3. Beyond two primitives: Composing subject, predicate and objects

In this section, we learn our transformation on three sets of visual primitives: subject, predicate and object. We first present additional details on the experimental setup.

Task: We predict a relevant (subject, predicate, object) tuple for a given (ground truth) bounding box from an image. The test set has *unseen tuples* which are used for evaluation. We use the metrics of Average Precision and top-k accuracy as described in Section 4.1.

Dataset: We use the recently published StanfordVRD [39] dataset. We use their provided train/test splits of 4k/1k images. The dataset contains SPO subject-predicate-object (generalization of subject-verb-object, SVO) annotations, *e.g.*, `man sitting on a chair` in which the subject-predicate-object tuple is `(man, sitting on, chair)`. In our notation, this dataset consists of three types of visual primitives $(\mathcal{V}_a, \mathcal{V}_b, \mathcal{V}_c)$ as (subject, predicate, object) respectively. The dataset has 7701 such tuples of which 1029 occur only in the test set. The dataset has 100 subjects and objects, and 70 predicates. We use the ground-truth bounding boxes and treat the problem as classification into SPO tuples rather than detection.

Baselines: We use the baselines described in Section 4.1. For the 'Individual' baselines we explicitly mention which primitives were used, *e.g.*, 'Indiv Predicate' denotes that only predicate was used, or 'Indiv Pred. Ob.' denotes that product of predicate and object was used.

Quantitative Results: The results for our method and baselines are summarized in Table 2. We evaluate all methods on the unseen subject, predicate, object tuples on the test set. Following the trend observed in Section 4.2, the 'Indiv' baseline methods show poor performance. Unsurprisingly, predicting a (subject, predicate, object) tuple by considering predictions of only one or two of the primitives does not perform well. We also see that the Indiv. Sub. Ob. baseline shows strong performance, while Indiv. Pred. shows very weak performance. Predicates show higher visual diversity than either subjects or objects and are much more difficult to capture in visual models [39, 53].

Additionally, in Table 2, the Indiv Sub. Ob. and Visual Product baselines show similar performance. It again suggests that predicate classifiers do not generalize. Similar to Section 4.2, we see that the LEOR and LE+R baselines both have worse performance than the LE baseline. The regression loss used in both these methods regresses to a $w_{\text{subject,predicate,object}}$ classifier. As there are limited examples available for (subject, predicate, object) tuples (compared to examples available individually for the concepts), these classifiers show poor performance (as also noted by [39]). Our method shows improvement over all baseline methods across all metrics, suggesting that the transformation $\mathcal{T}$ has some generalization.

5. Detailed Analysis

We now present detailed analysis of our approach and quantify our architectural design decisions. We also analyze other interesting properties of our learned transform $\mathcal{T}$. For all these experiments, we use the MITStates [28] dataset and follow the experimental setup from Section 4.2. We report the results on the 700 unseen pairs from the test set.

5.1. Architectural decisions

We analyze the impact of our design decisions on performance of the transformation network $\mathcal{T}$.

Choice of Loss Function: In Section 3, we described the Cross Entropy (CE) loss function (Equation 2). Here, we explore a few more choices for training our method:

- **Regression:** This loss function is inspired from the work of [10, 60] (also used in Section 4). The transform $T(w_{\text{large}}, w_{\text{elephant}})$ is trained to minimize the euclidean distance to the classifier $w_{(\text{large, elephant})}$.
- **Regression+CE:** We combine the Cross Entropy and Regression Loss functions (with a loss weight of 1 each).

Table 3: We analyze the effect of varying the loss function and initialization used to train the transform network $\mathcal{T}$. We test the network on the 700 unseen combinations

Loss	Init	Performance				
		AP		Top-k Accuracy		
			$k \rightarrow$	1	2	3
Cross Entropy	Gauss.	9.8		10.5	17.4	23.3
Regression	Gauss.	3.1		2.4	3.8	5.1
Cross Ent.+Reg.	Gauss.	7.6		10.2	17.0	22.1
Cross Entropy	Xavier	9.9		10.1	17.2	22.3
Cross Entropy	Identity	**10.4**		**13.2**	**21.2**	**27.6**

Table 4: Evaluating on unseen (attribute, object) pairs on the MITStates Dataset [28]. We vary the ratio of unseen pairs to seen pairs and evaluate our method.

Unseen Ratio		AP	Top-k Accuracy			
			$k \rightarrow$	1	2	3
0.1	Chance	-		1.5	3.0	4.5
0.1	Visual Product	28.7		48.6	58.1	66.2
0.1	Label Embed (LE)	29.2		49.7	59.2	**69.1**
0.1	Ours	**29.8**		**51.4**	**59.6**	68.9
0.3	Chance	-		0.1	0.3	0.4
0.3	Visual Product	8.8		9.8	16.1	20.6
0.3	Label Embed (LE)	7.9		11.1	17.6	22.4
0.3	Ours	**10.4**		**13.2**	**21.2**	**27.6**
0.5	Chance	-		0.1	0.2	0.3
0.5	Visual Product	5.9		6.2	8.8	10.5
0.5	Label Embed (LE)	5.9		7.8	12.6	16.9
0.5	Ours	**8.2**		**10.4**	**17.8**	**23.1**

Initialization: We found that using standard initialization methods like random gaussian or xavier [22] gave low performance for our transformation network $\mathcal{T}$.

Inspired from [36], we initialize the weights of our network as block diagonal identity matrices. This has the desirable property that immediately from initialization, the network can 'copy' its inputs to produce a reasonable output.

Table 3 summarizes the results for both these choices. We notice that the Regression loss performs poorly by itself. As noted in Sections 4.2 and 4.3, this is because it tries to mimic individual classifiers trained for each complex concept, *e.g.*, `(large elephant)`. These classifiers have little data available to train. Among initialization methods, our identity initialization improves performance.

Depth of network: We found that increasing the number of layers in the transformation network $\mathcal{T}$ did not give significant improvements. We opted for the minimal design that gave the best results.

5.2. Does the transform 'copy' the inputs?

We compute the distance between the inputs to the transformation network $\mathcal{T}$ and the produced output, *i.e.*, $d(w_a, \mathcal{T}(w_a, w_b))$. We compute this distance over the unseen combinations for the MITStates dataset and show it (after sorting) in Figure 6. We see that the transformation changes both inputs and does not just 'copy' them. In the supplementary material, we show that the predicted 'unseen' pairs are different from the 'seen' pairs.

5.3. Which classes gain the most?

Figure 5 shows the top classes for which our method improves over the Visual Product baseline. We see that the improvement for objects is across both man-made and natural objects. Similarly, the attributes improved by our method

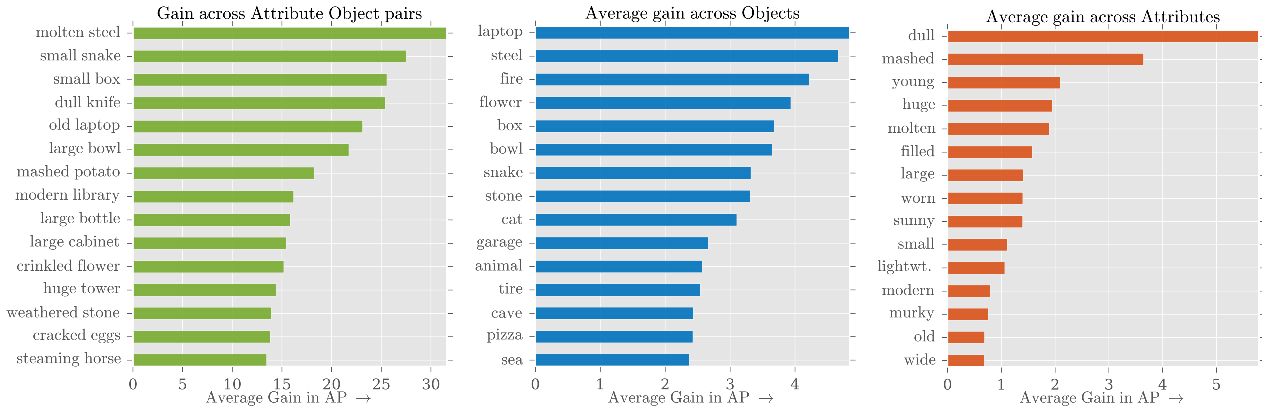

Figure 5: We show the top classes and the gain in AP over the Visual Product baseline on the MITStates dataset. We show the gain for (attribute, object) pairs, and for individual objects and attributes (after averaging across pairs).

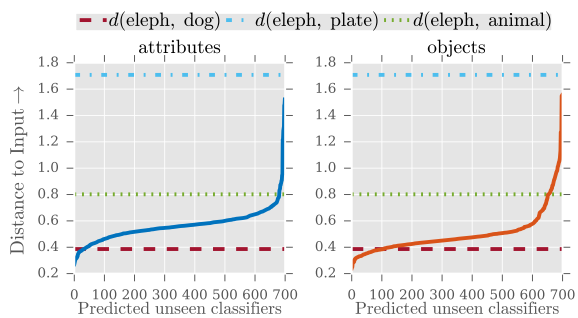

Figure 6: We show the distance between the inputs of the transformation network and its output for the unseen pairs. For visualization, we sort this distance individually for both attribute and object inputs. We provide distance between classifiers across 3 pairs of known classes for reference. We see that the transformation modifies all the inputs.

have diverse visual interpretations. The pairs for which the baseline is better are generally those where just predicting the object for the (attribute, object) pair gives the best performance, *i.e.*, attributes do not model much information about the object appearance.

5.4. Varying the ratio of seen/unseen concepts

We evaluate the effect of decreasing the training data for our transformation network. We vary the ratio of seen/unseen (attribute, object) pairs on MITStates from $[0.1, 0.3, 0.5]$, and train our network for each setting. We compare to the Visual Product and LE baselines from Section 4.1. Table 4 summarizes the results. We see that our algorithm is sensitive to the amount of training data available. It also shows improvement over baseline methods in all these settings. Comparing the performance for unseen ratios of 0.1 to 0.5, we see that our method's gain over baselines increases as we reduce the training data.

Table 5: We evaluate our method by composing unseen attributes and objects to form unseen combinations (attributes, objects). We use the MITStates dataset.

	AP		Top-k Accuracy		
		$k \rightarrow$	1	2	3
Chance	-		0.7	1.3	2.0
Visual Product	6.4		7.1	8.6	9.1
Label Embed (LE)	8.4		8.2	12.3	17.4
Ours	**9.6**		**10.1**	**18.3**	**22.9**

5.5. Moving from unseen combinations of primitives to unseen primitives

In this set of experiments, we randomly drop a set of object and attribute primitives from the training set of our transform network $\mathcal{T}$. The network never sees these classifiers at training time. At test time, we evaluate on the attribute, object pairs formed by these 'dropped' primitives. Concretely, we randomly drop 20% of objects and attributes: 49 of the 245 objects and 23 of the 115 attributes. We evaluate on 142 (attribute, object) pairs formed by these dropped primitives. We report these results in Table 5. Our method is able to generalize to these unseen input primitives and combine them to form the unseen pairs of concepts.

6. Conclusion

We presented a simple approach to compose classifiers to generate classifiers for new complex concepts. Our experiments on composing attributes and objects show that our method respects contextuality. We also show that our method can compose multiple primitives, and can generalize not only to unseen combinations of primitives, but also unseen primitives. It consistently gives better results than the baselines across different metrics and datasets.

Acknowledgements: We thank Abhinav Shrivastava, David Fouhey, Allison Del Giorno, and Saloni Potdar for helpful feedback. This work was supported by ONR MURI N000141612007 and the US Army Research Laboratory (ARL) under the CTA program (Agreement W911NF-10-2-0016). We thank NVIDIA for hardware donations, and Ed Walter for his help with the hardware.

References

[1] M. Abeles, M. Diesmann, T. Flash, T. Geisel, M. Herrmann, and M. Teicher. Compositionality in neural control: an interdisciplinary study of scribbling movements in primates. *Frontiers in computational neuroscience*, 7, 2013. 3

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 2

[3] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *TPAMI*, 36(12), 2014. 2

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016. 2

[5] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016. 2

[6] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94, 1987. 2

[7] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998. 2

[8] J. Choi, M. Rastegari, A. Farhadi, and L. S. Davis. Adding unlabeled samples to categories by learned attributes. In *CVPR*, 2013.

[9] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*. Springer, 2014. 2

[10] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013. 2, 4, 5, 7

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). 4

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 2008. 4

[13] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2

[14] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 28, 2006. 2

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[16] A. Ferencz, E. G. Learned-Miller, and J. Malik. Building a classification cascade for visual identification from one example. In *ICCV*, volume 1, 2005. 2

[17] M. Fink. Object classification from a single example utilizing class relevance metrics. *NIPS*, 17, 2005. 2

[18] J. A. Fodor. *The language of thought*, volume 5. Harvard University Press, 1975. 2

[19] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988. 3

[20] G. Frege. Sense and reference. *The philosophical review*, 57(3):209–230, 1948. 1

[21] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2

[22] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010. 7

[23] U. Grenander. *General pattern theory-A mathematical study of regular structures*. Clarendon Press, 1993. 2

[24] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013. 2

[25] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing with attribute grammar. *TPAMI*, 31, 2009. 2

[26] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 4

[27] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18, 1984. 2

[28] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2, 4, 5, 6, 7

[29] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*, 2014. 2

[30] A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. 2

[31] V. Krishnan and D. Ramanan. Tinkering under the hood: Interactive zero-shot learning with net surgery. *arXiv preprint arXiv:1612.04901*, 2016. 2

[32] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, volume 172, page 2, 2011. 2

[33] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Concept learning as motor program induction: A large-scale empirical study. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, 2012. 2

[34] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *arXiv preprint arXiv:1604.00289*, 2016. 2

[35] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2, 4

[36] Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015. 7

[37] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1, 1989. 2

[38] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *CVPR*, 2015. 1, 2, 4

[39] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 2, 4, 6, 7

[40] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. In *ICCV*, 2015. 2

[41] T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 2

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 5

[43] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick. Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels. In *CVPR*, 2016. 4

[44] E. Mjolsness. Connectionist grammars for high-level vision. *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*, pages 423–451, 1994. 2

[45] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22, 2010. 2

[46] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2

[47] N. Patricia and B. Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *CVPR*, 2014. 2

[48] G. Patterson and J. Hays. Coco attributes: Attributes for people, animals, and objects. In *ECCV*, 2016. 2, 6

[49] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. *arXiv preprint arXiv:1605.05395*, 2016. 2

[50] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015. 4

[52] F. Sadeghi, C. L. Zitnick, and A. Farhadi. Visalogy: Answering visual analogy questions. In *NIPS*, 2015. 2

[53] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 1, 2, 5, 7

[54] Z. Si and S.-C. Zhu. Learning and-or templates for object recognition and detection. *TPAMI*, 35, 2013. 1, 2

[55] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[56] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, 2012. 2

[57] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, volume 2, page 9, 2014. 2

[58] S. Thrun and T. M. Mitchell. Learning one more thing. Technical report, DTIC Document, 1994. 2

[59] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005. 2

[60] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 2, 5, 7

[61] T. Wu and S.-C. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *IJCV*, 93, 2011. 2

[62] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2

[63] A. Yuille and R. Mottaghi. Complexity of representation and inference in compositional models with part sharing. *JMLR*, 2016. 2

[64] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[65] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 2

[66] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille. Max margin and/or graph learning for parsing the human body. In *CVPR*, 2008. 2

[67] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 2

[68] L. L. Zhu, Y. Chen, and A. Yuille. Recursive compositional models for vision: Description and review of recent work. *Journal of Mathematical Imaging and Vision*, 41, 2011. 2

[69] S.-C. Zhu and D. Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007. 2