Choose Your Neuron: Incorporating Domain Knowledge through Neuron-Importance

Ramprasaath R. Selvaraju[1*] **Prithvijit Chattopadhyay**[1*] **Mohamed Elhoseiny**[2]
Tilak Sharma[2] **Dhruv Batra**[1,2] **Devi Parikh**[1,2] **Stefan Lee**[1]
[1]Georgia Institute of Technology [2]Facebook
{ramprs,prithvijit3,dbatra,parikh,steflee}@gatech.edu
{elhoseiny,tilaksharma,dbatra,parikh}@fb.com

Abstract

Individual neurons in convolutional neural networks supervised for image-level classification tasks have been shown to implicitly learn semantically meaningful concepts ranging from simple textures and shapes to whole or partial objects – forming a "dictionary" of concepts acquired through the learning process. In this work we introduce a simple, efficient zero-shot learning approach based on this observation. Our approach, which we call Neuron Importance-Aware Weight Transfer (NIWT), learns to map domain knowledge about novel classes onto this dictionary of learned concepts and then optimizes for network parameters that can effectively combine these concepts – essentially learning classifiers by discovering and composing learned semantic concepts in deep networks. In addition to demonstrating improvements on the generalized zero-shot learning benchmark, we show that by having an additional component which requires grounding neuron-level concepts in human-interpretable semantics, we can also interpret the decisions made by the learned classifiers at a fine-grained level of neurons.
Our code is available at https://github.com/ramprs/neuron-importance-zsl.

1 Introduction

Deep neural networks have pushed the boundaries of standard classification tasks in the past few years, with performance on many challenging benchmarks reaching near human-level accuracies. One caveat however is that these deep models require massive labeled datasets – failing to generalize from few examples or descriptions of unseen classes like humans can. To close this gap, the task of learning deep classifiers for unseen classes from external domain knowledge alone – termed zero-shot learning (ZSL) – has been the topic of increased interest within the community [17, 16, 10, 21, 29, 36, 31, 2, 11, 3, 25, 5, 14].

As humans, much of the way we acquire and transfer knowledge about novel concepts is in reference to or via composition of concepts which are already known. For instance, upon hearing that *"A Red Bellied Woodpecker is a small, round bird with a white breast, red crown, and spotted wings."*, we can compose our understanding of colors and birds to imagine how we might distinguish such an animal from other birds. However, applying a similar compositional learning strategy for deep neural networks has proven challenging.

While individual neurons in deep networks have been shown to learn localized, semantic concepts, these units lack referable groundings – *i.e.* even if a network contains units sensitive to *"white breast"* and *"red crown"*, there is no explicit mapping of these neurons to the relevant language name or description. This observation encouraged prior work in interpretability to crowd-source "neuron names" to discover these groundings [4]. However, this annotation process is model dependent and

[*]The first two authors contributed equally. Published as a conference paper at ECCV 2018.

Preprint. Work in progress.

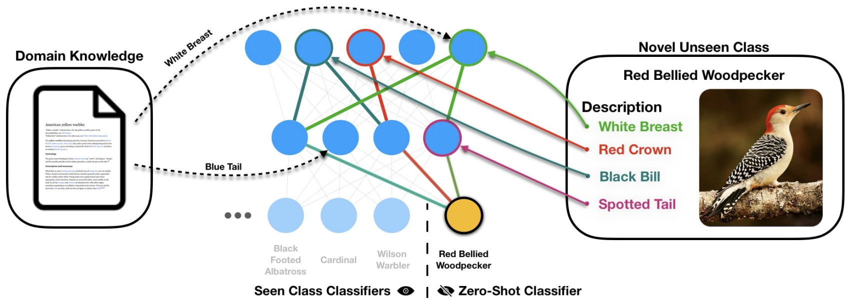


Figure 1: We present our Neuron Importance-Aware Weight Transfer (NIWT) approach which maps free-form domain knowledge about unseen classes to relevant concept-sensitive neurons within a pretrained deep network. We then optimize the weights of a novel classifier such that the activation of this set of neurons results in high output scores for the unseen classes in the generalized zero-shot learning setting.

needs to be re-executed for each model trained, which makes it expensive and impractical. Moreover, even if given perfect "neuron names", it is an open question how to leverage this neuron-level descriptive supervision to train novel classifiers. This question is at the heart of our approach.

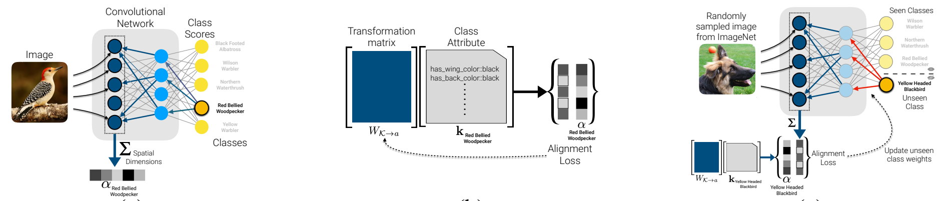
Many existing zero-shot learning approaches make use of deep features (*i.e.* vectors of activations from some late layer in a network pretrained on some large-scale task) to learn joint embeddings with class descriptions [32, 1, 3, 5, 23, 8, 9, 7]. These higher-level features collapse many underlying concepts in the pursuit of class discrimination; consequentially, accessing lower-level concepts and recombining them in new ways to represent novel classes is difficult with these features. Mapping class descriptions to lower-level activations directly on the other hand is complicated by the high intra-class variance of activations due to both spatial and visual differences within instances of a class. Our goal is to address these challenges by grounding class descriptions (including attributes and free-form text) to the *importance* of lower-layer neurons to final network decisions [26].

In our approach, which we call Neuron Importance-based Weight Transfer (NIWT), we learn a mapping between class-specific domain knowledge and the importances of individual neurons within a deep network. This mapping is learnt using images (to compute neuron-importance) and corresponding domain knowledge representation(s) of training classes. We then use this learned mapping to predict neuron importances from knowledge about unseen classes and optimize classification weights such that the resulting network aligns with the predicted importances. In other words, based on domain-knowledge of the unseen categories, we can predict which low-level neurons should matter in the final classification decision. We can then learn network weights such that the neurons predicted to matter actually do contribute to the final decision. In this way, we connect the description of a previous unseen category to weights of a classifier that can predict this category at test time – all without having seen a single image from this category. To the best of our knowledge, this is the first zero-shot learning approach to align domain knowledge to intermediate neurons within a deep network. As an additional benefit, the learned mapping from domain knowledge to neuron importances grounds the neurons in interpretable semantics; automatically performing neuron naming.

We focus on the challenging generalized zero-shot (GZSL) learning setting. Unlike standard ZSL settings which evaluate performance only on unseen classes, GZSL considers both unseen and seen classes to measure the performance. In effect, GZSL is made more challenging by dropping the unrealistic assumption that test instances are known a priori to be from unseen classes in standard ZSL. We validate our approach across two standard datasets - Caltech-UCSD Birds (CUB) [30] and Animals with Attributes 2 (AWA2) [32] - showing improved performance over existing methods. Moreover, we examine the quality of our grounded explanations for classifier decisions through textual and visual examples.

Contributions. Concretely, we make the following contributions in this work:

- We introduce a zero-short learning approach based on mapping unseen class descriptions to neuron importance within a deep network and then optimizing unseen classifier weights to effectively combine these concepts. We demonstrate the effectiveness of our approach by reporting improvements on the generalized zero-shot benchmark on CUB and AWA2. We also show our approach can handle arbitrary forms of domain knowledge including attributes and captions.
- In contrast to existing approaches, our method is capable of explaining its zero-shot predictions with human-interpretable semantics from attributes. We show how inverse mappings from neuron importance to domain knowledge can also be learned to provide interpretable visual and textual explanations for the decisions made by newly learned classifiers for seen and unseen classes.



(a) (b) (c)

Figure 2: Our Neuron Importance-Aware Weight Transfer (NIWT) approach can be broken down in to three stages. a) class-specific neuron importances are extracted for seen classes at a fixed layer, b) a linear transform is learned to project free-form domain knowledge to these extracted importances, and c) weights for new classifiers are optimized such that neuron importances match those predicted by this mapping for unseen classes.

2 Related Work

Model Interpretability. Our method aligns human interpretable domain knowledge to neurons within deep neural networks, instilling these neurons with understandable semantic meanings. There has been significant recent interest in building machine learning models that are transparent and interpretable in their decision making process. For deep networks, several works propose explanations based on internal states or structures of the network [34, 12, 37, 26]. Most related to our work is the approach of Selvaraju *et al*. [26] which computes neuron importance as part of a visual explanation pipeline. In this work, we leverage these importance scores to embed free-form domain knowledge to individual neurons in a deep network and train new classifiers based on this information. In contrast, Grad-CAM [26] simply visualizes the importance of input regions.

Attribute or Text-based Zero Shot Learning. One long-pursued approach for zero-shot learning is to leverage knowledge about common attributes and shared parts (e.g., furry, in addition to being simpler and more efficient [25, 3, 2, 32]. Leveraging pure textual descriptions instead of attributes to design zero-shot classifiers for novel classes has also been a significantly popular approach in the computer vision community [8, 25, 18, 7]. The description of a new category in such situations could be extracted easily by just mining article(s) from the web (e.g., Wikipedia) or crowdsourcing natural language descriptions [24]. These approaches primarily rely on learning a similarity function between class-level descriptions (attributes or otherwise) and images (either linearly [8, 25] or non-linearly – deep neural networks [18], kernels [7]).

In contrast to these approaches, we directly map external domain knowledge (text-based or otherwise) to internal components (neurons) of deep neural networks rather than learning associative mappings between images and text – inherently providing interpretability for our novel classifiers.

3 Neuron Importance-Aware Weight Transfer (NIWT)

In this section, we describe our proposed approach – Neuron Importance-Aware Weight Transfer (NIWT). At a high level, NIWT maps free-form domain knowledge to neurons within a deep network and then learns classifiers based on novel class descriptions which respect these groundings. Concretely, NIWT consists of three steps: (1) estimating the importance of individual neuron(s) at a fixed layer w.r.t. the decisions made by the network for the seen classes (see Figure 2a), (2) learning a mapping between domain knowledge and these neuron-importances (see Figure 2b), and (3) optimizing classifier weights with respect to predicted neuron-importances for unseen classes (see Figure 2c). We discuss each of these stages in the following sections.

3.1 Preliminaries: Generalized Zero-Shot Learning (GZSL)

Consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ comprised of example input-output pairs from a set of *seen classes* $\mathcal{S} = \{1, \dots, s\}$ and *unseen classes* $\mathcal{U} = \{s{+}1, \dots, s{+}u\}$. For convenience, we use the subscripts $\mathcal{S}$ and $\mathcal{U}$ to indicate subsets corresponding to seen and unseen classes respectively, *e.g.* $\mathcal{D}_\mathcal{S} = \{(x_i, y_i) \mid y_i \in \mathcal{S}\}$. Further, assume there exists domain knowledge $\mathcal{K} = \{k_1, ..., k_{s+u}\}$ corresponding to each class (*e.g.* class level attributes or natural language descriptions). Concisely, the goal of generalized zero-shot learning is then to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{S} \cup \mathcal{U}$ from the input space $\mathcal{X}$ to the combined set of seen and unseen class labels using only the domain knowledge $\mathcal{K}$ and instances $\mathcal{D}_\mathcal{S}$ belonging to the seen classes.

3.2 Class-dependent Neuron Importance

Class descriptions capture salient concepts about the content of corresponding images – for example, descriptions regarding the coloration and shape of a bird's head. Similarly, a trained classifier must also learn discriminative visual concepts in order to succeed; however, these concepts are not grounded in human interpretable language. In this stage, we identify neurons corresponding to these discriminative concepts before aligning them with domain knowledge in Section 3.3.

Consider a deep neural network $\texttt{NET}_{\mathcal{S}}(\cdot)$ trained for classification which predicts scores $\{o_c \mid c \in \mathcal{S}\}$ for seen classes $\mathcal{S}$. One intuitive measure of a neuron n's importance to the final score o_c is simply the gradient of o_c with respect to the neuron's activation a^n (where n indexes the channel dimension). For networks containing convolutional units (which are replicated spatially), we follow [26] and simply compute importance as the mean gradient (along spatial dimensions), writing the neuron importance α_c^n as

$$\alpha_c^n = \overbrace{\frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W}}^{\text{global average pooling}} \underbrace{\frac{\partial o_c}{\partial a_{ij}^n}}_{\text{gradients via backprop}} \tag{1}$$

where $a_{i,j}^n$ is the activation of neuron n at spatial position i, j. For a given input, the importance of every neuron in the network can be computed for a given class via a single backward pass followed by a global average pooling operation for convolutional units. In practice, we focus on α's from single layers in the network in our experiments. We note that other measures of neuron importance have been proposed [33, 15] in various contexts; however, this simple gradient-based importance measure has some notable properties which we leverage.

Firstly, we find gradient-based importance scores to be quite consistent across images of the same class despite the visual variation between instances, and likewise to correlate poorly across classes. To assess this quantitatively, we computed α's for neurons in the final convolutional layer of a convolutional neural network trained on a fine-grained multi-class task (`conv5-3` of VGG-16 [27] trained on AWA2 [32]) for 10,000 randomly selected images. We observed an average rank correlation of 0.817 for instances within the same class and 0.076 across pairs of classes. This relative invariance of α's to intra-class input variation may be due in part to the piece-wise linear decision boundaries in networks using ReLU [20] activations. As shown in [22], transitions between these linear regions are much less frequent between same-class inputs than across classes. Within the same linear region, activation gradients (and hence α's) are trivially identical.

Secondly, this measure is fully differentiable with respect to model parameters which we leverage to learn novel classifiers with gradient methods (see Section 3.4).

3.3 Mapping Domain Knowledge to Neurons

Without loss of generality, consider a single layer L within $\texttt{NET}_{\mathcal{S}}(\cdot)$. Given an instance $(x_i, y_i) \in \mathcal{D}_{\mathcal{S}}$, let $\mathbf{a}_c = \{\alpha_c^n \mid n \in L\}$ be a vector of importances computed for neurons in L with respect to class c when x_i is passed through the network. In this section, we learn a simple linear mapping from domain knowledge to these importance vectors – aligning interpretable semantics with individual neurons.

We first compute the importance vector $\mathbf{a}_{y_i}$ for each seen class instance (x_i, y_i) and match it with the domain knowledge representation k_{y_i} of the corresponding class. Given this dataset of $(\mathbf{a}_{y_i}, k_{y_i})$ pairs, we learn a linear transform $W_{\mathcal{K}\to a}$ to map domain knowledge to importances. As importances are gradient based, we penalize errors in the predicted importances based on cosine distance – emphasizing alignment over magnitude. We minimize the cosine distance loss as

$$\mathcal{L}(\mathbf{a}_{y_i}, \mathbf{k}_{y_i}) = 1 - \frac{(W_{\mathcal{K}\to a} \cdot \mathbf{k}_{y_i}) \cdot \mathbf{a}_{y_i}}{\|W_{\mathcal{K}\to a} \cdot \mathbf{k}_{y_i}\| \, \|\mathbf{a}_{y_i}\|}, \tag{2}$$

via gradient descent to estimate $W_{\mathcal{K}\to a}$. We stop training when average rank-correlation of predicted and true importance vectors stabilizes for a set of held out validation classes from $\mathcal{S}$.

Notably, this is a one-to-many mapping – the domain knowledge of one class needing to predict many different importance vectors. Despite this, this mapping achieves average rank correlations of 0.2 to 0.5 for validation class instances. We explore the impact of error in importance vector prediction on weight optimization in Section 3.4. We also note that this simple linear mapping can also be learned in an inverse fashion, mapping neuron importances back to semantic concepts within the domain knowledge (which we explore in Section 6).

3.4 Neuron Importance to Classifier Weights

Here we use predicted importances to learn classifiers for the unseen classes. As these new classifiers will be built atop the trained seen-class network $\texttt{NET}_{\mathcal{S}}$, we modify $\texttt{NET}_{\mathcal{S}}$ to extend the output space to include the unseen class – expanding the final fully-connected layer to include additional neurons with weight vectors $\mathbf{w}^1, \ldots, \mathbf{w}^u$ for the unseen classes such that the network now additionally outputs scores $\{o_c \mid c \in \mathcal{U}\}$. We refer to this expanded network as $\texttt{NET}_{\mathcal{S}\cup\mathcal{U}}$. At this stage, the weights for the unseen classes are sampled randomly from a multivariate normal distribution with parameters estimated from the seen class weights and as such the output scores are uncalibrated and uninformative.

Given the learned mapping $W_{\mathcal{K}\to A}$ and unseen class domain knowledge $\mathcal{K}_{\mathcal{U}}$, we can predict unseen class importances $A_{\mathcal{U}} = \{\mathbf{a}_1, ..., \mathbf{a}_u\}$ with the importance vector for unseen class c predicted as $\mathbf{a}_c = W_{\mathcal{K}\to a}\mathbf{k}_c$. For a given input, we can compute importance vectors $\hat{a}_c$ for each unseen class c. As $\hat{\mathbf{a}}^c$ is a function of the weight parameters $\mathbf{w}_c$, we can simply supervise $\hat{\mathbf{a}}_c$ with the predicted importances $\mathbf{a}_c$ and optimize w^c with gradient descent – minimizing the cosine distance loss between predicted and observed importance vectors. However, the cosine distance loss does not account for scale and without regularization the scale of weights (consequentially the outputs) of seen and unseen classes might vary drastically, resulting in bias towards one set or the other.

To address this problem, we introduce a L_2 regularization term which constrains the learned unseen weights to be a similar scale as the mean of seen weights $\overline{\mathbf{w}}_{\mathcal{S}}$. We write the final objective as

$$\mathcal{L}(\hat{\mathbf{a}}_c, \mathbf{a}_c) = 1 - \frac{\hat{\mathbf{a}}_c \cdot \mathbf{a}_c}{\|\hat{\mathbf{a}}_c\| \, \|\mathbf{a}_c\|} + \lambda \|\mathbf{w}_c - \overline{\mathbf{w}}_{\mathcal{S}}\|, \tag{3}$$

where λ is controls the strength of this regularization. We examine the effect of this trade-off in Section 5.1, finding training to be robust to a wide range of λ values. We note that as observed importances $\mathbf{a}^c$ are themselves computed from network gradients, updating weights based on this loss requires computing a Hessian-vector product; however, this is relatively efficient as the number of weights for each unseen class is small and independent of those for other classes.

Training Images. Note that to perform the optimization described above, we need to pass images through the network to compute importance vectors. We observe importances to be only weakly correlated with image features and find they can be computed for any of the unseen classes irrespective of the input image class – as such, we find simply inputing images with natural statistics to be sufficient. Specifically, we pair random images from ImageNet [6] with random tuples $(\hat{\mathbf{a}}_c, \mathbf{k}_c)$ to perform the importance to weight optimization.

4 Experiments

In this section, we evaluate our approach on generalized zero-shot learning (GZSL) (Section 4.1) and present analyses for different stages of NIWT (Section 5).

4.1 Experimental Setting

Datasets and Metrics. We conduct our GZSL experiments on the following datasets.

- **Animals with Attributes 2 (AWA2) [32]** – The AWA2 dataset consists of 37,322 images of 50 animal species (on average 764 per class but with a wide range). Each class is labeled with 85 binary and continuous attributes.
- **Caltech-UCSD Birds 200 (CUB) [30]** – The CUB dataset consists of 11788 images corresponding to 200 species of birds. Each image and each species has been annotated with 312 binary and continuous attribute labels respectively. These attributes describe fine-grained physical bird features such as the color and shape of specific body parts. Additionally, each image is associated with 10 human captions [24].

For both datasets, we use the GZSL splits proposed in [32] which ensure that no unseen class occurs within the ImageNet [6] dataset (commonly used for training classification networks for feature extraction). As in [31], we evaluate our approach using class-normalized accuracy computed over both seen and unseen classes (*i.e.* 200-way for CUB) – breaking the results down into unseen accuracy $\texttt{Acc}_{\mathcal{U}}$, seen accuracy $\texttt{Acc}_{\mathcal{S}}$, and the harmonic mean of the two, $\texttt{H}$.

Models. We experiment with ResNet101 [13] and VGG16 [28] models pretrained on ImageNet [6] and fine-tuned on the seen classes. For each, we train a version by finetuning all layers and another by updating only the final classification weights. Compared to ResNet, where we see sharp declines for fixed models (60.6% finetuned vs 28.26% fixed for CUB and 90.10% vs 70.7% for AWA2), VGG achieves similar accuracies for both finetuned and fixed settings (74.84% finetuned vs 66.8% fixed for CUB and 92.32% vs 91.44% for AWA2).

NIWT Settings. To learn the mapping $W_{\mathcal{K}\to a}$, we hold out five seen classes and stop optimization when rank correlation between observed and predicted importances is highest. For attribute vectors, we use the class level attributes directly and for captions on CUB we use average word2vec embeddings[19] for each class. When optimizing for weights given importances, we stop when the loss fails to improve by 1% over 40 iterations. We choose values of λ (between $1e^{-5}$ to $1e^{-2}$),

		Method	AWA2 [32]			CUB [30]		
			$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	$\texttt{H}$	$\texttt{Acc}_{\mathcal{U}}$	$\texttt{Acc}_{\mathcal{S}}$	$\texttt{H}$
ResNet101 [13]	Fixed	ALE [2][1]	20.9	**88.8**	33.8	**24.7**	62.3	**34.4**
		Deep Embed. [35][1]	**28.5**	82.3	**42.3**	22.3	45.1	29.9
		NIWT-Attributes	21.6	37.8	27.5	10.2	57.7	17.3
	FT	ALE [2][1]	22.7	**75.1**	34.9	24.1	**60.8**	**34.5**
		Deep Embed. [35][1]	21.5	59.6	31.6	**24.7**	57.4	**34.5**
		NIWT-Attributes	**42.3**	38.8	**40.5**	20.7	41.8	27.7
		NIWT-Caption		N/A		22.1	25.7	23.8
VGG16 [28]	Fixed	ALE [2][1]	17.9	**84.3**	29.5	22.2	**54.8**	**31.6**
		Deep Embed. [35][1]	**28.8**	81.7	**42.6**	**24.1**	45.2	31.5
		NIWT-Attributes	43.8	30.7	36.1	17.0	54.6	26.7
	FT	ALE [2][1]	16.9	**91.5**	28.5	25.3	**62.6**	36.0
		Deep Embed. [35][1]	26.6	83.3	38.2	27.0	49.7	35.0
		NIWT-Attributes	**35.3**	75.5	**48.1**	**31.5**	44.9	**37.0**
		NIWT-Caption		N/A		15.9	46.5	23.6

Table 1: Generalized Zero-Shot Learning performances on the proposed splits [32] for AWA2 and CUB. We report class-normalized accuracies on seen and unseen classes and harmonic mean. [1] based on code provided by the authors by tuning hyper-parameters on the test-set to convey an upper-bound of performance.

learning rate ($1e^{-5}$ to $1e^{-2}$) and the batch size ($\{16, 32, 64\}$) by grid search on $\texttt{H}$ for a disjoint set of validation classes sampled from the seen classes of the proposed splits [32] (see Table. 1).

Baselines. We compare NIWT with two well-performing zero-shot learning approaches – ALE [2] and Deep Embed. [35]. While the former relies on learning compatibility functions for class labels and visual features the latter leverages deep networks, jointly aligning domain knowledge with deep features end-to-end. For the mentioned baselines, we utilize code provided by the authors and report results by directly tuning hyper-parameters on the test-set to convey an upper-bound of performance.

4.2 Results

Our results are summarized in Table 1. Some notable trends are,

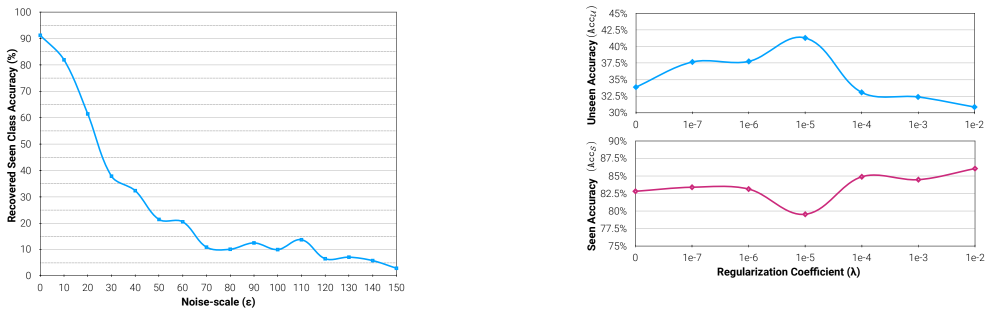
1. **NIWT shows improvements on the generalized zero-shot learning benchmark.** For both datasets, NIWT-Attributes based on VGG establishes a new state of the art for harmonic mean (48.1% for AWA2 and 37.0% for CUB). For AWA2, this corresponds to a $\sim$ 10% improvement over prior state-of-the-art which is based on deep feature embeddings – implying that grounding domain knowledge to internal neurons of a network can indeed lead to improved results.
2. **Seen-class finetuning yields improved harmonic mean $\texttt{H}$.** For the CUB and AWA2 datasets, finetuning the VGG network on seen class images offers significant gains for NIWT (26.7%$\to$37.0% $\texttt{H}$ and 36.1%$\to$48.1% $\texttt{H}$ respectively); finetuning ResNet sees similar gains (17.3%$\to$27.7% $\texttt{H}$ on CUB and 27.5%$\to$40.5 %$\texttt{H}$ on AWA2). Notably, these trends seem inconsistent for the compared methods.
3. **NIWT effectively grounds both attributes and free-form language.** We see reasonably strong performance for captions in addition to attributes across both networks (37.0% and 23.6% $\texttt{H}$ for VGG and 27.7% and 23.8% $\texttt{H}$ for ResNet). We note that we use relatively simple, class-averaged representations for captioning which may contribute to the lower absolute performance.

5 Analysis

To better understand the different stages of NIWT, we perform a series of experiments to analyze isolated components in our approach.

5.1 Effect of Regularization Coefficient λ.

One key component to our importance to weight optimization is the regularizer which enforces that learned unseen weights be close to the mean seen class weight – avoiding arbitrary scaling of the learned weights and the bias this could introduce. To explore the effect of the regularizer, we vary the coefficient λ (0 to $1e^{-2}$) and observe variations in $\texttt{Acc}_{\mathcal{S}}$ and $\texttt{Acc}_{\mathcal{U}}$ (see Fig. 3b).



(a) Noise Tolerance (ϵ) (b) Regularizer Sensitivity (λ)

Figure 3: Analysis of the importance vector to weight optimization for VGG-16 trained on AWA2 (a). We find that ground-truth weights can be recovered for a pre-trained network even in the face of high magnitude noise. (b) We also show the importance of the regularization term to final model performance.

Without regularization (λ=0) the unseen weights tend to be a bit too small and achieve an unseen accuracy of only 33.9% on AWA2. As λ is increased the unseen accuracy grows until peaking at λ=$1e^{-5}$ with an unseen accuracy of 41.3% – an improvement of over 8% from the unregularized version! Of course, this improvement comes with a trade-off in seen accuracy of about 3% over the same interval. As λ grows larger $>1e^{-4}$, the regularization constraint becomes too strong and NIWT has trouble learning anything substantial for the unseen classes.

5.2 Noise Tolerance in Neuron Importance to weight optimization

Recall that NIWT inherently relies on grounding human-interpretable semantic concepts in the neurons of a deep network (see Sec. 3.3) – allowing us to have a representation of the sub-level concepts in a network in the corresponding referable domain. Due to the inherent noise involved in the mapping $W_{\mathcal{K}\rightarrow A}$, the classifiers obtained for unseen classes in the expanded network $\texttt{NET}_{\mathcal{S}\cup\mathcal{U}}$ are not perfect. In order to judge the capacity of the optimization procedure, we experiment with a toy setting where we initialize an unseen classifier head with the same dimensionality as the seen classes and try to explicitly recover the seen class weights with supervision only from the *oracle* $\mathbf{a}_c$ obtained from the seen classifier head for the seen classes. To simulate for the error involved in estimating $\mathbf{a}_c$, we add increasing levels of zero-centered gaussian noise and study recovery performance in terms of accuracy of the recovered classifier head on the seen-test split. That is, the supervision from importance vectors is constructed as follows:

$$\tilde{\mathbf{a}}_c = \mathbf{a}_c + \epsilon\overline{||\mathbf{a}_c||}_1\mathbf{z} \quad \text{such that} \quad \mathbf{z} \sim \mathcal{N}(0, \mathtt{I}) \tag{4}$$

We operate at different values of ϵ, characterizing different levels of corruption of the supervision from $\mathbf{a}_c$ and observe recovery performance in terms of accuracy of the recovered classifier head. 3a shows the effect of noise on the ability to recover seen classifier weights (`fc7`) for a VGG-16 network trained on 40 seen classes of AWA2 dataset with the same objective as the one used for unseen classes.

In the absence of noise over $\mathbf{a}_c$ supervision, we find that we are exactly able to recover the seen class weights and are able to preserve the pre-trained accuracy on seen classes. Even with a noise-level of ϵ=10 (or adding noise with a magnitude 10x the average norm of $\mathbf{a}_c$), we observe only minor reduction in the accuracy of the recovered seen class weights – implying the importance vector to weight optimization process is quite robust to fairly extreme noise. As expected, this downward trend continues as we increase the noise-level until we reach almost chance-level performance on the recovered classifier head.

5.3 Network Depth of Importance Extraction.

In this section, we explore the sensitivity of NIWT with respect to the layer from which we extract importance vectors in the convolutional network. As an experiment (in addition to Table 1) we evaluate NIWT on AWA2 with importance vectors extracted at different convolutional layers of VGG-16. We observe that out of those we experimented with `conv5_3` performs the best with $\mathtt{H} = 48.1$ followed by `conv4_3` ($\mathtt{H} = 39.3$), `conv3_3` ($\mathtt{H} = 35.5$), `conv2_2` ($\mathtt{H} = 23.8$) and `conv2_1` ($\mathtt{H} = 20.8$). We also experimented with the fully-connected layers `fc6` and `fc7` resulting in values of $\mathtt{H}$ being 40.2 and 1 respectively.

Note that performing NIWT on importance vectors extracted from the penultimate layer `fc7` is equivalent to learning the unseen head classifier weights directly from the domain space representation ($\mathbf{k}_c$). Consistent with our hypothesis, this performs very poorly across all the metrics with almost no learning involved for the unseen classes at all. Though we note that to some extent this could also be attributed to the restricted capacity of the linear transformation $W_{\mathcal{K}\rightarrow A}$ involved in the process.

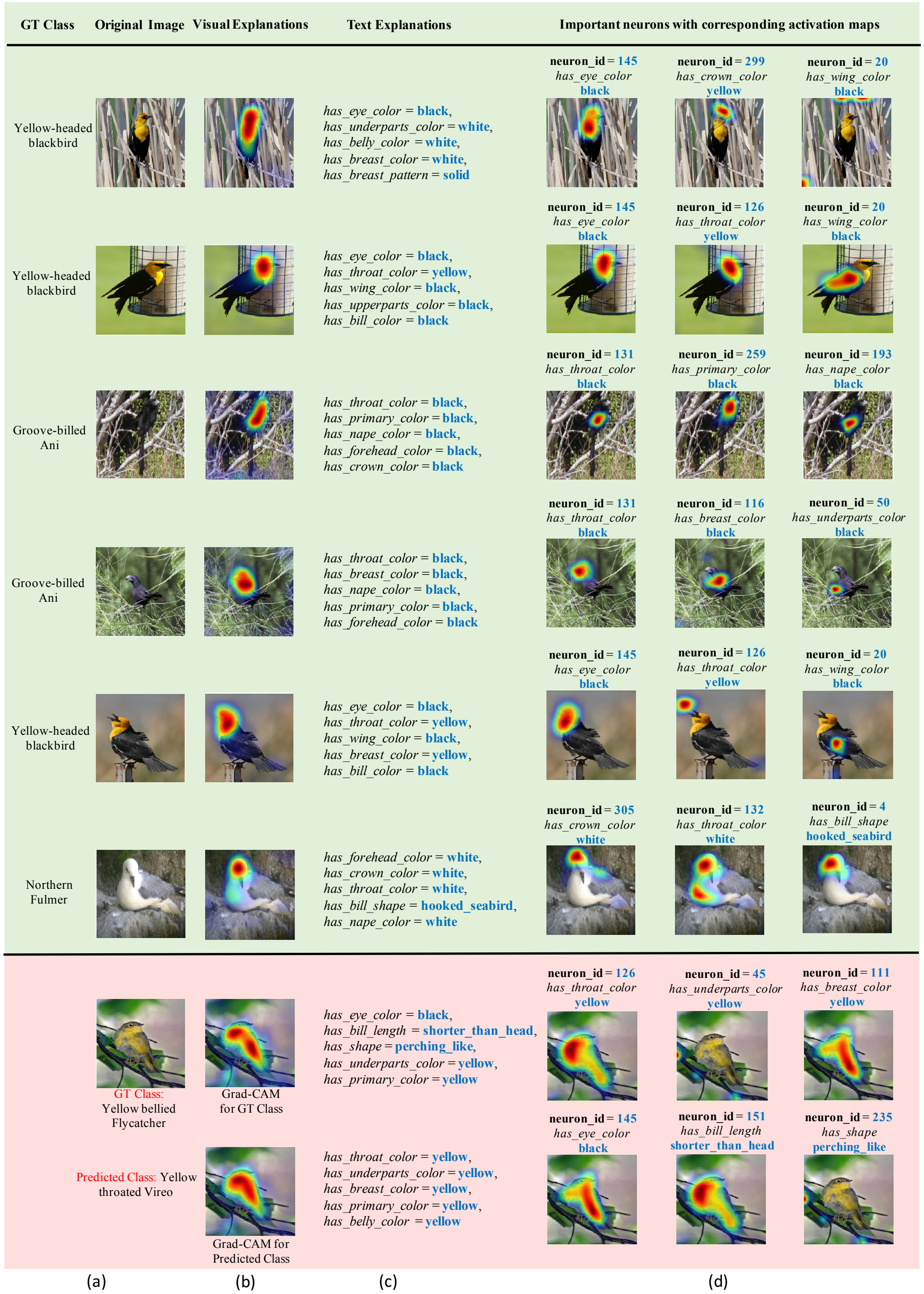


Figure 4: Explanations corresponding to the decisions made by the learned classifier for instances of the unseen classes on CUB.(a) the ground truth class and image, (b) visual explanations for the GT category, (c) textual explanations obtained using the inverse mapping $W_{a \to \mathcal{K}}$, (d) most important neurons for this decision, associated names and activation maps. The last 2 rows show a failure case, where the model misclassified the given image. Explicitly grounding network concepts in referable domains, we can expose the relative focus across fine-grained concepts composing a class – for the image correctly classified as a yellow-headed blackbird above (row 2), the visualizations for the class focuses specifically at the union of attributes that comprise the class - *black eye*, *yellow throat*, and *black wing*.

5.4 Importance to Weight Input Images

We show performance with differing input images during weight optimization (random noise, ImageNet, and seen class images) in Table. 2. As expected, performance improves as input images more closely resemble the unseen classes; however, we note that learning occurs even with random noise images.

Sampling Mode	$\mathtt{Acc}_{\mathcal{U}}$	$\mathtt{Acc}_{\mathcal{S}}$	H
Random Normal	23.9	41.0	30.2
ImageNet	31.5	44.9	37.0
Seen-Classes	36.4	40.0	38.1

Table 2: Results by sampling images from different sets for NIWT-Attributes on VGG-CUB.

6 Explaining NIWT

Recall that NIWT involves an explicit learning component that requires us to ground the salient concepts for a class of interest in the important neurons. Here we explore how a similar grounding framework would allow us to expose the decision making process of the network for a given instance at a fine-grained level of neurons – where in addition to grounding the important neurons with respect to a prediction in human-interpretable semantics, we can also express the relative *visual* focus across said concepts. Fig. 4 demonstrates explanations for decisions made by the learned classifiers.

Visual Explanations. Since learning classifiers for the novel classes via NIWT preserves the end-to-end differentiable nature of the network as a whole, any gradient-based interpretability technique (or otherwise) is applicable to provide support for decisions made at inference. We use Grad-CAM [26] on instances of unseen classes to provide explanations for the novel classifier learned via NIWT. Quantitative results on CUB – characterized by the mean fraction of Grad-CAM activation present inside the bounding box of the object of interest – indicate that the learned classifier is indeed capable of focusing on the relevant regions (0.80 ± 0.008 for seen and 0.79 ± 0.005 for unseen classes).

Textual Explanations. In our setup, we frame textual explanations as the problem of retrieving relevant attributes given the neurons important for a decision made by the network. We instantiate this as learning an inverse mapping $W_{a \rightarrow \mathcal{K}}$ – from importance scores $\mathbf{a}_c$ to associated domain-knowledge $\mathbf{k}_c$ – in a manner similar to Sec. 3.3. At inference, for a decision made by the learned classifier, we retrive the top-5 scoring attributes under $W_{a \rightarrow \mathcal{K}}$. Intuitively, a high scoring $\mathbf{k}_c$ retrieved via $W_{a \rightarrow \mathcal{K}}$ from a certain $\mathbf{a}_c$ emphasizes the relevance of that attribute for the corresponding class c. Quantitatively, we evaluate the fidelity of the retrieved explanations as the percentage of associated ground truth attributes for an instance in the retrieved top-k ones – 83.9% for CUB. Qualitatively, the retrieved explanations correlate well with the associated visual explanations as described above.

Neuron Names and Focus. Treating neuron-names as referable groundings of concepts captured by a deep convlutional network – we characterize the same as the top-1 textual explanation retrieved for a single-neuron under $W_{a \rightarrow \mathcal{K}}$. We instantiate this by feeding a one-hot encoded vector corresponding to the important neurons one at a time to $W_{a \rightarrow \mathcal{K}}$ and retrieving the top-scoring $\mathbf{k}_c$. In contrast to prior work, this one-shot process circumvents issues surrounding the collection of expensive annotations or performing any additional optimization on top of the same. In addition, observing the activation map of the associated neurons allows us to thereby characterize the *focus* of the neuron of interest.

7 Conclusion

To summarize, we propose an approach we refer to as Neuron Importance-aware Weight Transfer (NIWT), that learns to map domain knowledge about novel classes directly to classifier weights by grounding it into the importance of network neurons. Our weight optimization approach on this grounding results in classifiers for unseen classes which outperform existing approaches on the generalized zero-shot learning benchmark. We further demonstrate that this grounding between language and neurons can also be learned in reverse, linking neurons to human interpretable semantic concepts, providing visual and textual explanations.

Acknowledgements. We thank Yash Goyal and Nirbhay Modhe for help with figures; Peter Vajda and Manohar Paluri for helpful discussions. This work was supported in part by NSF, AFRL, DARPA, Siemens, Google, Amazon, ONR YIPs and ONR Grants N00014-16-1-{2713,2793}. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013. 2

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2016. 1, 3, 6

[3] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015. 1, 2, 3

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017. 1

[5] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 1, 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[7] Mohamed Elhoseiny, Ahmed Elgammal, and Babak Saleh. Write a classifier: Predicting visual classifiers from unstructured text. *Ieee T Pattern Anal*, PP(99):1–1, 2017. 2, 3

[8] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2584–2591, 2013. 2, 3

[9] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the "beak": Zero shot learning from noisy text description at part precision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[10] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009. 1

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, 2013. 1

[12] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Interpreting visual question answering models. *CoRR*, abs/1608.08974, 2016. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[14] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[15] Sandeep Konam, Ian Quah, Stephanie Rosenthal, and Manuela Veloso. Understanding convolutional networks with apple : Automatic patch pattern labeling for explanation. *First AAAI/ACM Conference on AI, Ethics, and Society*, 2018. 4

[16] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 1

[17] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. 1

[18] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015. 3

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 5

[20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 4

[21] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014. 1

[22] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018. 4

[23] Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, year=2016*. 2

[24] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. 3, 5

[25] Bernardino Romera-Paredes and PHS Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015. 1, 3

[26] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *ICCV*, 2017. 2, 3, 4, 9

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[28] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 5, 6

[29] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, 2013. 1

[30] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5, 6

[31] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 5

[32] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3, 4, 5, 6

[33] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I. Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S. Davis. NISP: pruning networks using neuron importance score propagation. *CVPR*, 2018. 4

[34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 3

[35] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 6

[36] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1

[37] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014. 3