Beyond One-hot Encoding: lower dimensional target embedding

Pau Rodríguez†, Miguel A. Bautista*, Jordi Gonzàlez†, Sergio Escalera$^{\dagger, \ddagger}$

†*Computer Vision Center, Universitat Autònoma de Barcelona, Spain*

**Heidelberg Collaboratory for Image Processing, Heidelberg University, Germany*

‡*University of Barcelona, Barcelona, Spain*

Abstract

Target encoding plays a central role when learning Convolutional Neural Networks. In this realm, One-hot encoding is the most prevalent strategy due to its simplicity. However, this so widespread encoding schema assumes a flat label space, thus ignoring rich relationships existing among labels that can be exploited during training. In large-scale datasets, data does not span the full label space, but instead lies in a low-dimensional output manifold. Following this observation, we embed the targets into a low-dimensional space, drastically improving convergence speed while preserving accuracy. Our contribution is two fold: *(i)* We show that random projections of the label space are a valid tool to find such lower dimensional embeddings, boosting dramatically convergence rates at zero computational cost; and *(ii)* we propose a normalized eigenrepresentation of the class manifold that encodes the targets with minimal information loss, improving the accuracy of random projections encoding while enjoying the same convergence rates. Experiments on CIFAR-100, CUB200-2011, Imagenet, and MIT Places demonstrate that the proposed approach drastically improves convergence speed while reaching very competitive accuracy rates.

Keywords: Error correcting output codes, output embeddings, deep learning,
computer vision

Email addresses: `pau.rodriguez@cvc.uab.cat` (Pau Rodríguez†), `miguel.bautista@iwr.uni-heidelberg.de` (Miguel A. Bautista*), `jordi.gonzalez@cvc.uab.cat` (Jordi Gonzàlez†), `sescalera@cvc.uab.cat` (Sergio Escalera$^{\dagger, \ddagger}$)

1. Introduction

Convolutional Neural Networks lie at the core of the latest breakthroughs in large-scale image recognition [1, 2], at present even surpassing human performance [3], applied to the classification of objects [4], faces [5], or scenes [6]. Due to its effectiveness and simplicity, one-hot encoding is still the most prevalent procedure for addressing such multi-class classification tasks: in essence, a function $f : \mathbb{R}^p \rightarrow \mathbb{Z}_2^n$ is modeled, that maps image samples to a probability distribution over a discrete set of the n labels of target categories.

Unfortunately, when the output space grows, class labels do not properly span the full label space, mainly due to existing label cross-correlations. Consequently, one-hot encoding might result inadequate for fine-grained classification tasks, since the projection of the outputs into a higher dimensional (orthogonal) space dramatically increases the parameter space of computed models. In addition, for datasets with a large number of labels, the ratio of samples per label is typically reduced. This constitutes an additional challenge for training CNN models in large output spaces, and the reason of slow convergence rates [7].

In order to address the aforementioned limitations, output embeddings have been proposed as an alternative to the one-hot encoding for training in large output spaces [8]: depending on the specific classification task at hand, using different output embeddings captures different aspects of the structure of the output space. Indeed, since embeddings use weight sharing during training for finding simpler (and more natural) partitions of classes, the latent relationships between categories are included in the modeling process.

According to Akata *et al.* [9], output embeddings can be categorized as:

- Data-independent embeddings, such as drawing rows or columns from a Hadamard matrix [10]: data-independent embeddings produce strong baselines [11], since embedded classes are equidistant due to the lack of prior knowledge;

- Embeddings based on a priori information, like attributes [12], or hierarchies

[13]: unfortunately, learning from attributes requires expert knowledge or extra labeling effort and hierarchies require a prior understanding of a taxonomy of classes, and in addition, approaches that use textual data as prior do not guarantee visual similarity [11]; and

- Learned embeddings, for capturing the semantic structure of word sequences (i.e. annotations) and images jointly [14]. The main drawbacks of learning output embeddings are the need of a high amount of data, and a slow training performance.

Thus, in cases where there exist high quality attributes, methods with prior information are preferred, while in cases of a known equidistant label space, data-independent embeddings are a more suitable alternative. Unfortunately, the architectural design of a model is bound to the particular choice among the above-mentioned embeddings. Thus, once a model is chosen and trained using an specific output embedding, it is hard to reuse it for another tasks requiring a different type of embedding.

In this paper, Error-Correcting Output Codes (ECOC) are proven to be a better alternative to one-hot encoding for image recognition, since ECOCs are a generalization of the three embedding categories [15], so a change in the ECOC matrix will not constitute a change in the chosen architecture. In addition, ECOCs naturally enable error-correction, low dimensional embedding spaces [16], and bias and variance error reduction [17].

Inspired by the latest advances on ECOCs, we circumvent one-hot encoding by integrating the Error-Correcting Output Codes into CNNs, as a generalization of output embedding. As a result, a best-of-both-worlds approach is indeed proposed: compact outputs, data-based hierarchies, and error correction. Using our approach, training models in low-dimensional spaces drastically improves convergence speed in comparison to one-hot encoding. Figure 1 shows an overview of the proposed model.

The rest of the paper is organized as follows: Section 2 reviews the existing work most closely related to this paper. Section 3 presents the contribution of the proposed embedding technique, which is two fold: *(i)* we show that random projections of the label space are suitable for finding useful lower dimensional embeddings, while boosting

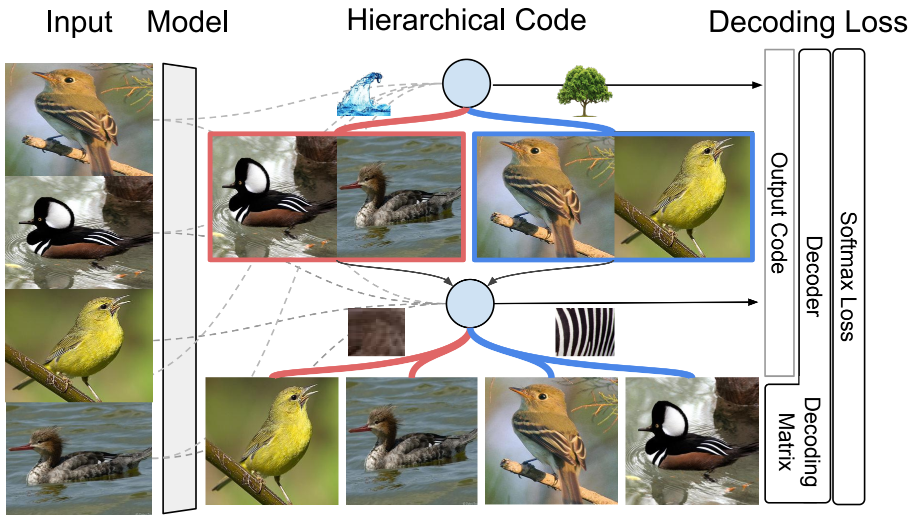


Figure 1: This paper proposes to replace the traditional one-hot output scheme of CNNs with a reduced scheme with at least $log_2(k)$ outputs. In addition, when using a hierarchical representation of the data labels, outputs show that the most discriminative attributes to split the target classes have been learned. In essence, a decoder computes the similarities of the ”predicted code” in a ”code-matrix”, and subsequently the output label is then obtained through a softmax layer. The internal code representation is depicted in a tree structure, where each bit of the code corresponds to the actual learned partition from the data, from lower partition cost (aquatic) to higher (stripped).

dramatically convergence rates at zero computational cost; and *(ii)* In order to generate partitions of the label space that are more discriminative than the random encoding (which generates random partitions of the label space), we also propose a normalized eigenrepresentation of the class manifold to encode the targets with minimal information loss, thus improving the accuracy of random projections encoding while enjoying the same convergence rates. Subsequently, the experimental results on CIFAR-100 [18], CUB200-2011 [19], MIT Places [6], and ImageNet [1] presented in Section 4 show that our approach drastically improves convergence speed while maintaining a competitive accuracy. Lastly, Section 5 concludes the paper discussing how, when gradient sparsity on the output neurons is highly reduced, more robust gradient estimates and better representations can be found.

2. Related work

This section reviews those works on output embeddings most related to ours, in particular those using ECOC.

Output Embeddings Most of the related literature addresses the challenge of zero-shot learning, i.e. training a classifier in the absence of labels. Often, the proposed approaches take into account the attributes of objects [20, 21, 22, 9] related to the different classes through well-known, shared object features.

Due to their computing efficiency based on a divide-and-conquer strategy, output embeddings have been also proven useful for those multi-class classification problems in which testing all possible class labels and hierarchical structures is not feasible [23, 19, 14, 8]. Given a large output space, most labels are usually considered instances of a superior category e.g., sunflower and violet are flower plants. In this sense, the inherent hierarchical structure of the data makes divide-and-conquer hierarchical output spaces a suitable alternative to the traditionally flat 1-of-N classifiers. Likewise in the context of language processing, Mikolov *et al.* combine Huffman binary codes and hierarchical soft-max in order to map the most frequent codes to shorter paths in a tree [24].

Because output embeddings enforce weight sharing, they have been also used when the number of classes is rather large, with no clear inter-class boundaries, and a decaying ratio of the number of examples per class. In this context, in order to reduce the output space, Weston *et al.* proposed WSABIE, an online learning-to-rank algorithm to find an embedding for the labels based on images [25].

In the field of large-scale recognition, hierarchical approaches such as using tree-based priors [26], label relational graphs [27], CNN hierarchies [28], and HD-CNNs [29] have been proposed. For example in [30] binary hash codes are used for fast image retrieval. However, such hierarchical approaches need to be learned, and cannot be easily interchanged with other embeddings. In addition, for approaches learning codes as latent variables, to find the optimal ones in terms of class separability or error correction is not guaranteed [30]. Due to all this, ECOC constitute a better alternative for seamless integration with CNNs, as detailed next.

Error-Correcting Output Codes[1] ECOC have been applied in multiple fields such as medical imaging [31], face and facial-feature recognition [32, 33], and segmentation of human limbs citesanchez2015hupba8k+. ECOCs are a generic divide-and-conquer framework that combines binary partitions to achieve multi-class recognition [34]. Their core property is the capability to correct errors of binary classifiers using redundancy, while reducing the bias and variance of the ensemble [17]. Advanced approaches propose to use them as intermediate representations [35].

ECOC consist of two main steps: *coding* and *decoding*. The *coding* step consists in assigning a codeword of arbitrary length k to each of the n classes. Codewords are organized in a "code matrix" $\mathbf{M}_{k,n} \in \{-1, 1\}$, where each column is a binary partition on the label space in meta-classes. Since there are many possible bi-partitions, the design of the code is central for obtaining discriminative ones. Indeed there are several approaches for generating ECOCs: Exhaustive codes [34], BCH codes [36], random codes [37], and circular ECOC [38] are few examples of methods that generate codes independently from the inherent structure of the data.

Although ECOCs can be data-independent and even randomly generated, they can also be learnt from data: Pujol *et al.* propose a discriminant ECOC approach based on hierarchical partitions of the output space [39]. Subsequently, Escalera *et al.* [40] proposed to split complex problems into easier subclasses, embedded as binary dichotomizers in the ECOC framework, easier to optimize. In [41], it is also shown Optimal continuous ECOCs can be found by gradient descent. Griffin & Perona [42] use trees to efficiently handle multi-class problems, which posteriorly Zhang *et al.* improved by finding optimal partitions with spectral ECOCs [43].

In the decoding step, a sample x can be decoded as the output of k binary classifiers $\{f_1(x), f_2(x), ..., f_k(x)\}$. Given the predicted code, the class label y corresponds to the closest row in $M_{k,n}$. The most common decoding methods are the Hamming and Euclidean distances but there are more sophisticated approaches such as probabilistic-based decoding, especially with ternary codes [15].

[1] We use the standard notation in ECOCs: bold capital letters denote matrices (e.g. $\mathbf{X}$) and bold lower-case letters represent vectors (e.g., $\mathbf{z}$). All non-bold letters denote scalar variables.

Inspired from latest ECOC advances, we propose to integrate output codes in large-scale deep learning problems. In this context, few approaches in the literature have been presented: in [44, 27], CNNs are also used to directly predict the code bits for Optical Character recognition (OCR). We go a step further by: *(i)* showing that the convergence speed in large scale settings with millions of images can be dramatically improved; *(ii)* instead of directly predicting the code bits, we integrate the euclidean decoding with the cross-entropy loss, so that the network does not only optimize individual bits independently but also inter-code distances, which results in error-correction.

Our approach enhances the convergence of CNNs using random codes, i.e. when the inter-class relationships are not considered. We achieve even lower error rates with data-dependent codes, due to using more efficient data partitions. Similarly, Yang *et al.* also used CNNs to integrate data-independent Hadamard Codes with the Euclidean loss [45]. But due to the efficiency of data-dependent codes, our encoding proposal is shown more efficient than [45], by halving the required CNN output size, and eliminating the need of training multiple CNNs to predict code chunks.

3. Low dimensional target embedding

Figure 1 depicts our proposed model inspired by the ECOC framework [34] and applied for deep supervised learning. Given a set of n classes, an ECOC consists of a set of k binary partitions of the label space (groups of classes) representing each of the n classes in the dataset. The codes are usually arranged in a design matrix $\mathbf{M} \in \{-1, 1\}^{n \times k}$.

Let's define the output of the last layer of a neural network as z^l, with l the depth of the network. For the sake of clarity the identity non-linearity $\phi(\cdot)$ is used so that $\mathbf{z}^l = \phi(\mathbf{z}^l)$. Thus, given the weights of the previous layer $\boldsymbol{\Theta}^{(l-1)}$, and the corresponding bias $\mathbf{b}^{(l-1)}$, $\mathbf{z}^l$ can be computed as $\Theta^{(l-1)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l-1)}$.

In our case, we reduce the output dimensionality of a CNN, i.e. the dimensionality of $\mathbf{z}^l$, from n (the number of classes) to k, an arbitrary number of partitions. Then, given a design matrix $\mathbf{M}^{n \times k}$, where each row encodes a class label, the predicted class is obtained by finding the distance of the output with each row of the design matrix

$\mathbf{D} = \mathbf{M} - \mathbb{1}^\top \mathbf{z}^l$, with $\mathbb{1}^\top$ a column vector constituted by ones, and obtaining the label with $argmin(\mathbf{D})$. Then, we seamlessly integrate our proposal in the traditional log-likelihood and softmax loss layer.

3.1. Embedding output codes in CNNs

Given a training set $\{x_i, y_i\}$ $i = 1 : s$, of image-label pairs, CNNs constitute the state-of-the-art at finding good local minima by empirical risk minimization (ERM) using the cross-entropy as the loss function J by means of backpropagation [46]:

$$J(X, Y; \Theta) = -\frac{1}{s}\sum_{i=1}^{s}[y_i log(\hat{y})_i + (1 - y_i) log(1 - \hat{y}_i)],$$

where $\hat{y}_i = argmax(\mathbf{h}(\mathbf{z}^l(\mathbf{x}_i))) \in \mathbb{R}$ is the predicted label for the i^{th} example and $y_i \in \{0, 1\}$ the ground truth label. Since cross-entropy requires probability distributions, the output of the network $\mathbf{z}^l$ is fed to a softmax layer that assigns a probability score to each of the n possible classes:

$$h(\mathbf{z}^l)_j = \frac{e^{z_j^l}}{\sum_{i=1}^{N} e^{z_i^l}}, j \in \{1, 2, ..., n\}.$$

The derivative of the loss function J for gradient descent through backpropagation is known to be:

$$\frac{\delta J}{\delta z_i^l} = y_i - \hat{y}_i.$$

The decoder is introduced between the output $\mathbf{z}^l$ of the network and the softmax function $\mathbf{h}(\mathbf{z}^l)$. Concretely, the negative normalized Euclidean distance $\mathbf{D}(\mathbf{z}^l|\mathbf{M})$ between $\mathbf{z}^l$ and the rows in $\mathbf{M}$ is used, so that the output of the softmax represents the probability of the output of the CNN to be decoded as the $i^{th}, i \in \{1, 2, 3.., n\}$ output word.

We reformulate the softmax function $\mathbf{h}(\mathbf{z}^l)$ as $\mathbf{h}(\mathbf{D}(\frac{\mathbf{z}^l}{||\mathbf{z}^l||_2}))$, with the variable change of $\mathbf{D}(\frac{\mathbf{z}^l}{||\mathbf{z}^l||_2})$ by $\mathbf{D}(\mathbf{U})$ (with $\mathbf{U}(\mathbf{z})$ the normalized vector). The derivative of the loss can be computed using the chain rule:

$$\frac{\delta J(\mathbf{D}, Y; \Theta)}{\delta \mathbf{z}} = \frac{\delta J(\mathbf{D}, Y; \Theta)}{\delta \mathbf{D}} \frac{\delta \mathbf{D}}{\delta \mathbf{U}}^{(1)} \frac{\delta \mathbf{U}}{\delta \mathbf{z}^l}^{(2)}.$$

We now calculate:

$$\frac{\delta \mathbf{D}}{\delta \mathbf{U}} = \frac{\delta}{\delta \mathbf{U}} \frac{-1}{2} (\mathbf{M} - \mathbb{1}^\top \mathbf{U})(\mathbf{M} - \mathbb{1}^\top \mathbf{U})^\top = \mathbf{M} - \mathbb{1}^\top \mathbf{U}, \tag{1}$$

$$\frac{\delta \mathbf{U}}{\delta \mathbf{z}} = \frac{\delta}{\delta \mathbf{z}} \frac{\mathbf{z}}{||\mathbf{z}||_2} = \frac{\mathbf{I}||\mathbf{z}||_2 - \mathbf{z}\mathbf{U}^\top}{||\mathbf{z}||_2}. \tag{2}$$

Given eq. 1 and 2, it is possible to compute the derivative of the cross-entropy with the new decoding loss $\hat{J}$:

$$\frac{\delta \hat{J}}{\delta \mathbf{z}^l} = (\mathbf{Y} - \hat{\mathbf{Y}})[(\mathbf{M} - \mathbb{1}^\top \mathbf{U}) \frac{\mathbf{I}||\mathbf{z}^l||_2 - \mathbf{z}^l \mathbf{U}^\top}{||\mathbf{z}^l||_2}]^\top. \tag{3}$$

Provided the amount of computation that can be shared from the forward pass to the backward pass, this process does not slow-down the training phase. In fact, the cost is compensated by *(i)* the shrinkage of $\mathbf{z}$, which also results in a reduction of the number of network parameters, and *(ii)* the increase of convergence speed.

The convergence speed increases because reducing the output layer results in parameter sharing, which produces more robust gradient estimates. The explanation is that the softmax function distributes the probabilities among a high number of neurons. Thus, the the gradient $\delta J = y_i - \hat{y}_i$ is zero for most outputs because $y_i = 1$ only once in the ground truth vector, and $\mathbb{E}(\hat{y}_j) = \frac{1}{n}$. Given that the network is certain about the output i', the expected output for the rest of the outputs is even smaller $\mathbb{E}(\hat{y}_{j \neq i'}) = \frac{1-y_{i'}}{n-1}$.

In other words, output layers with huge number of outputs and smaller mini-batch size can only update the weights of few output units per iteration, since activation expected value is virtually zero. Thus, the gradients for these outputs are either zero or based on too few examples. This leads to noisy estimates to the real loss surface. As a result, reducing the output space with our method increases the ratio of activations per mini-batch, helping to obtain more robust gradient estimates and increasing convergence speed, reduces the mini-batch size, and thus the memory requirements.

3.2. Connections with Normalized Cuts

CNNs trained with our approach are robust and fast even when drawing codes from a normal distribution. The reason is the fact that random gaussian matrices tend to follow the coding properties described in the literature [34, 47], such as row and column orthogonality. For most large datasets the label space follows a hierarchical structure and defining random partitions of the label space is rather unnatural. In order to find the most simple partitions we use an eigenrepresentation of the class manifold based on the class similarities found in the dataset. Concretely, solving the normalized cut (Ncut) problem on the class similarity graph is a way of obtaining n uncorrelated low-cost partitions, with n the number of classes [48]. The NCut can be approximated by solving the eigendecomposition of the normalized Laplacian of the class similarity matrix $\mathbf{L_M}$:

$$\mathbf{L_G} = \mathbf{D}^{\frac{1}{2}}(\mathbf{D} - \mathbf{M})\mathbf{D}^{\frac{-1}{2}} = \lambda\mathbf{V},$$

where $\mathbf{M}$ is the class similarity matrix, $\mathbf{D}$ is the degree matrix, $\lambda_\mathbf{i}$ are the eigenvalues in ascending order and $\mathbf{v_i}$, the corresponding eigenvectors $i \in \{0, 1, 2, ..., k\}$. Given that $\lambda_0 = 0$, the eigenvectors $\mathbf{v}_i, i \in \{1, ..., k\}$ constitute the partitions ordered by the Ncut cost. As explained in [6], this kind of codes have desirable properties such as balancing, orthogonality, lower error bounds due to the separability maximization, and similarity preserving, i.e. similar classes have similar codes. We show that training CNNs to predict the embedded target, together with this data-based codes, exhibit lower error rates than using random codes. Contrary to [43], we do not threshold the eigenvectors so as to obtain a binary code but we interpret the values as likelihoods.

In the following section, we provide empirical evidence confirming that CNNs trained with our proposed methodology on CIFAR-100, CUB-200, MIT Places, and Imagenet have faster convergence rates (with comparable or better recognition rates), even with smaller mini-batch size, than their one-hot counterparts.

4. Experiments

To validate our approach, we perform a thorough analysis of the advantages of embedding output codes in CNN models over different state-of-the-art datasets. First, we describe the considered datasets, methods and evaluation.

4.1. Datasets

We first experiment the ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012) [1] and the MIT Places-205 [6] datasets. ImageNet consists of 1.2M images, and 50K validation images with 10K object classes. MIT Places is constituted by 2.5M images from 205 scene categories for training, and 100 images for category for testing.

Subsequently we experiment on the CIFAR-100 [18] and the Caltech-UCSD Birds-200-2011 [49]. CIFAR-100 consists of 50K 32×32 images for training, and 10K 32×32 images for testing belonging to 10 coarse categories and 100 fine-grained categories. CUB-200 contains 11,788 images (5,994 images for training and 5,794 for test) of 200 bird species, each image annotated with 15 part locations, 312 binary attributes, and 1 Bounding Box.

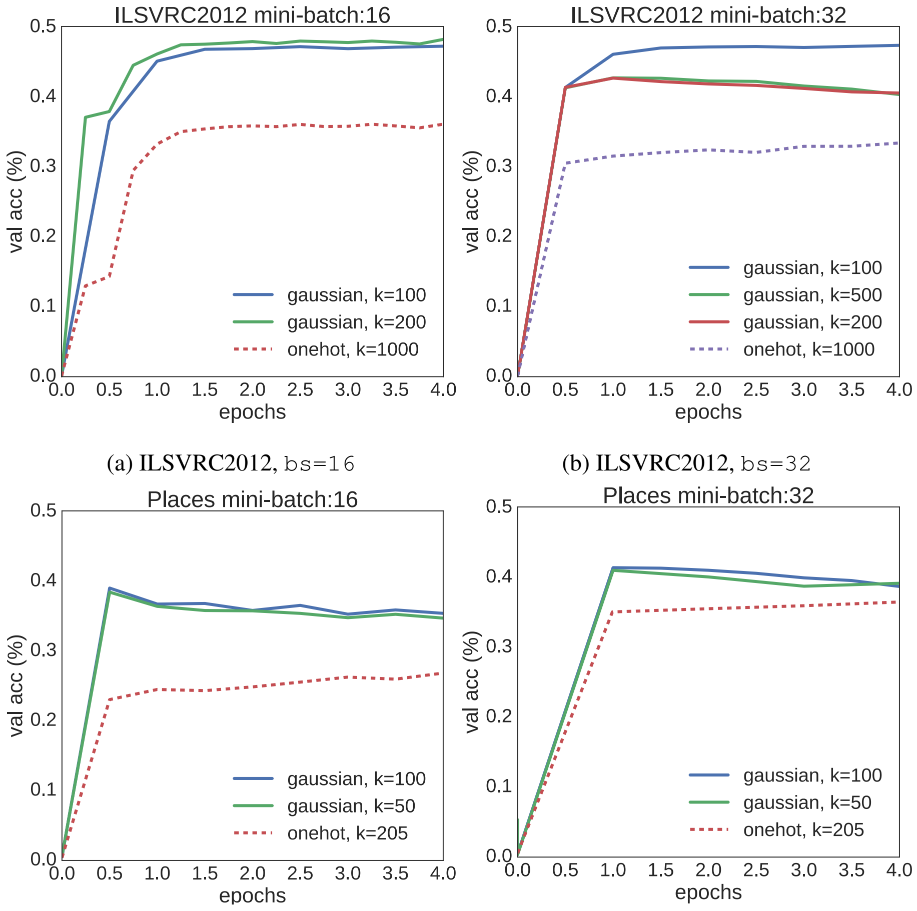
4.2. Methods and evaluation

We use standard state-of-the-art models to evaluate the contribution of the proposed target embedding procedure instead of comparing with state-of-the-art results on the considered datasets. Note that any model, including more recent and powerful state-of-the-art architectures, can benefit from our target embedding methodology.

As a proof of concept, we first validate data-independent codes on the Imagenet and MIT Places datasets. Concretely, we retrain with our approach the `fc7` and `fc8` layers of an Alexnet model [50] pre-trained on the respective datasets. Concretely, we randomly reinitialize their weights and train them using SGD with a global learning rate (`lr`) of 0.001, and the specific `lr` of the reinitialized layers is multiplied by 10.

Then, we demonstrate the advantages of data-dependent codes on the fine-grained CIFAR-100 and CUB-200 2011. For CIFAR-100, we use the `cifar_quick` models found in the Caffe framework [51]. The network is initialized with noise sampled from a gaussian distribution, and the model is trained for 100 epochs. Fine-tuning on CUB-200 is performed with the same pre-trained model of the Imagenet experiments for 30 epochs, and the `lr` is divided by 10 after 15 epochs.

Experiments with the standard Alexnet CNN [50] (caffe version [51]) on Imagenet, and MIT Places, prove that CNNs trained with random codes and our approach show



(a) ILSVRC2012, `bs=16`

(b) ILSVRC2012, `bs=32`

(c) MIT Places, `bs=16`

(d) MIT Places, `bs=32`

Figure 2: **Validation accuracy on ILSVRC2012 and MIT Places.** Using output codes randomly sampled from a normal distribution results in faster convergence, especially for small mini-batch sizes (a,c)

faster convergence rates than using one-hot encoding, especially for small mini-batch sizes, while matching one-hot in performance for bigger mini-batch sizes. Thus, the proposed data-dependent encoding approach performs better than using random codes for fine-grained datasets, with fuzzy inter-class boundaries, essentially because random codes alone do not take into account the correlation of attributes.

4.3. Random codes for faster convergence

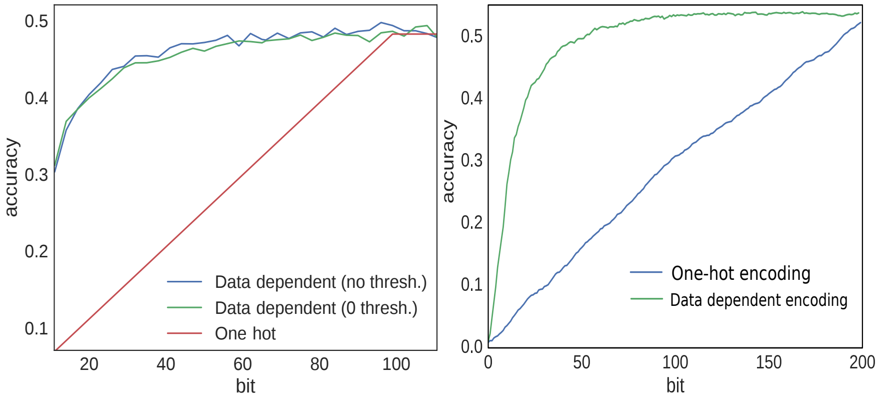
Output encodings allow to embed sparse output spaces into compact representations. For instance, codes generated with the dense random strategy only need $k = 10log(n)$ bits [37] to encode n classes. An inherent property of one-hot encoding is the output activation sparsity for huge output spaces. Given a randomly initialized CNN with one-hot encoding, provided that the output neurons follow a uniform distribution, the probability assigned to each class will be $\frac{1}{n},\ n = \#Classes$, which tends to 0 for $n \to \infty$. In the final stages of training, the situation will persist since just an extremely small ratio of the neurons activate, i.e. a small subset of the neurons show high probability for the predicted class while the residual probability mass is spread over a much larger number of neurons.

Thus, it can be coarsely estimated that the update probability of the parameters associated to an output neuron during an SGD step is related to the ratio $\rho = \frac{bs}{n}$, with mini-batch size bs, being $\rho = 256 \cdot 10^3$ for Alexnet trained on Imagenet, provided that $p(Y = n_i) = p(Y = n_j),\ i \neq j$. In other words, given a label, sampling more images increases the probability of that label being in the set of samples, and drawing less samples than the number of labels ensures that at least $n - s$ labels will not be seen during the update.

Figure 2 shows the resulting validation accuracy when training Alexnet on the ILSVRC2012 and MIT Places for different mini-batches and a random code sampled from $\mathcal{N}(0, 1)$. As it can be seen, models trained with our approach converge faster than those trained with one-hot encoding.

4.4. Using data-based encodings

In order to adapt to fine-grained settings, i.e. with high inter-class correlations, and few examples per class, we propose to generate the output codes using the eigenvectors of the normalized Laplacian of the class similarity matrix. Since this eigendecomposition generates the most discriminating, hierarchical partitions based on the data, models trained with this data-dependent codes result in higher accuracy bounds than the random counterparts.



(a) CIFAR100

(b) CUB200

Figure 3: Classification accuracy based on the number of the code bits. As expected, the same amount of information is encoded for each of the one-hot bits while the same results are obtained with just the 25% of the data-based codes.

To confirm the aforementioned advantages of using data-dependent codes we choose to experiment on the well-established CIFAR-100 and CUB-200 2011 fine-grained datasets. see Fig. 3. We use CIFAR-100 for fast experimentation, and then we apply the best setting to CUB-200.

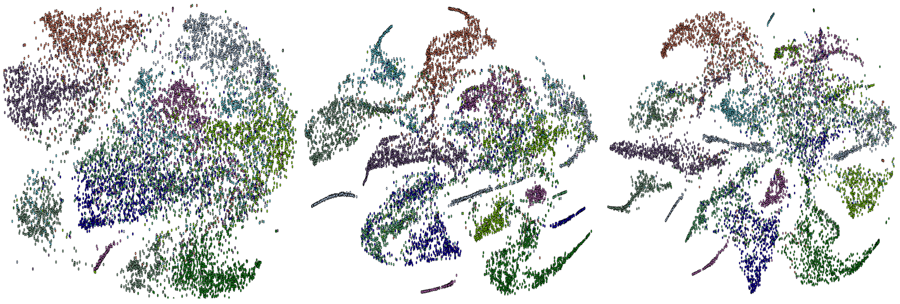
CIFAR-100. First, we evaluate different procedures for generating the codes:

1. One-hot. A vector of $n-1$ zeros and a one at the target position (with n the number of classes).
2. Dense random [37]. Sampling the matrix with the most uncorrelated rows and columns from $\mathcal{U}(0,1)$.
3. Gaussian. Sampling matrices from a normal distribution.
4. Data-based. Constructing the code matrix from the eigenvalues of the class similarity Laplacian.

Note that Gaussian and Data-based codes are composed of real numbers and a thresholding function should be applied for obtaining binary partitions. We test thresholding at zero and the median of the rows of the code matrix. Additionally, we test the raw values, interpreting them as the likelihood of the k^{th} metaclass to be present in the n^{th} class.

Code	One-hot			Gaussian								
Binarization	-			-			Zero			Median		
Length	66	100	200	66	100	200	66	100	200	66	100	200
Accuracy (%)	32.4	49.2	-	44.9	44.8	44.8	45.0	47.1	49.1	45.6	47.8	48.4
	Dense Random			Data-dependent								
	-	-	-	-			Zero			Median		
	66	100	200	66	99	200	66	99	200	66	99	200
	43.9	44.5	44.3	48.0	50.0	49.7	46.7	49.0	48.9	47.4	47.8	49.7

Table 1: **Influence of code designs on CIFAR-100.** Dense output encodings are more robust than One-hot to the loss of bits. As expected, data-based codes outperform the rest of encodings (50%), especially when no threshold is applied to binarize the code.



(a) One-hot. (b) Dense random. (c) Data-based.

Figure 4: **T-sne visualization on CIFAR-100** on the ten coarse categories for the hidden fc layer of a CNN trained with (a) one-hot encoding, (b) an output code generated with the dense random strategy, and (c) a data-based code.

As it can be seen in table 1, output encodings are more robust, losing a smaller percentage of the accuracy when the number of code-bits are halved, while one-hot scales linearly with the number of bits, see 3a for a detailed analysis. In addition, data-based codes find the more discriminative partitions, resulting in better accuracy than the rest of the encodings. Moreover, keeping the raw values of the eigenvectors provides additional information about the likelihood of a metaclass to be present in a certain class, resulting in more robust predictions. Since output codes are based on

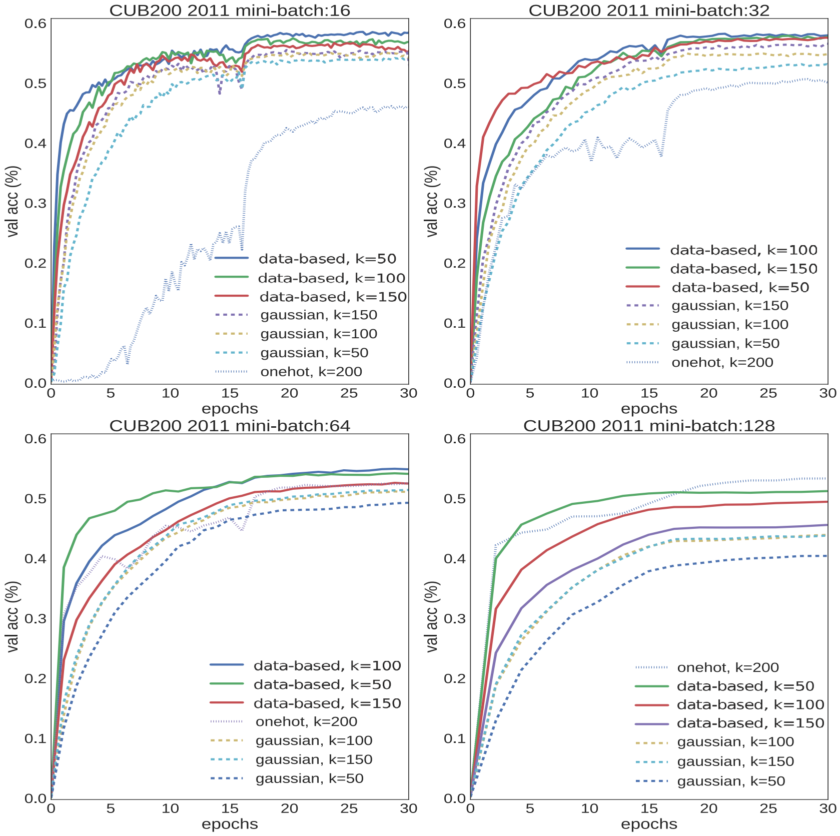
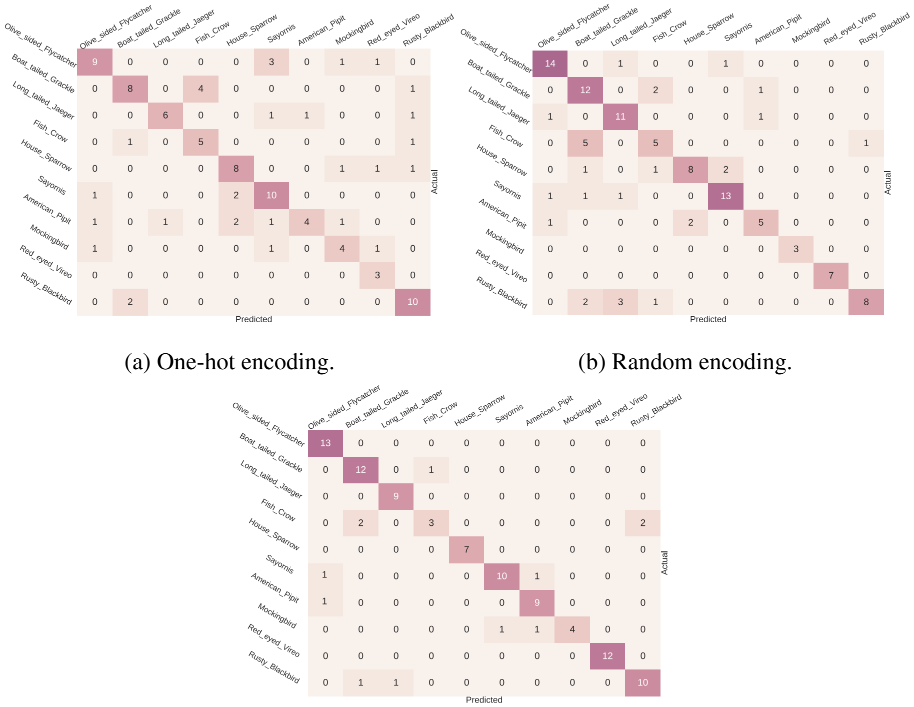


Figure 5: **Validation accuracy on CUB200.** Plots have been generated for different mini-batch sizes. (a) when mini-batch size is 16, the performance of one-hot encoding is dramatically reduced

binary partitions, they constrain the learning so that features are encoded to fall into hyperplanes.

In figure 4 we show the 2D projection of those hyperplanes using t-sne. Note the higher overlapping of samples from different classes displayed on the target embedding space of 1-hot in comparison to dense and data-dependent alternatives. In particular, the proposed eigendecomposition of the output space shows a more discriminative splitting of the data samples according to their labels.

CUB-200. Figure 5 shows that using small mini-batch sizes with data-based encodings largely outperforms the one-hot baseline for different code lengths when training a CNN on CUB-200 with data-dependent codes based on the raw eigenvalues of the class similarity matrix (best setting on CIFAR100). Moreover, in figure 3b, it can be



(a) One-hot encoding.

(b) Random encoding.

(c) Data-dependent code.

Figure 6: **Confusion matrices on CUB200-2011.** Alexnet trained with random codes sampled from a normal distribution (b) already advantage those trained with one-hot encoding (a) e.g., reducing the number of confusions of ”Olive sided Flycatcher” with the rest of the classes. Moreover, data-dependent codes based on eigenrepresentations of the output space (d), can better discriminate even more classes, like ”boat tailed Grackle” from ”Fish crow”. Samples for the classes in the confusion matrices are shown in figure 7.

seen that the data-based code matches the one-hot encoding with just the 25% of the bits. As expected, the first bits correspond to the most discriminative partitions ordered by cut cost. The class similarity matrix was built with the `fc7` outputs of a pre-trained network, but any other would also work if it reflects the inter-class relationships.

Figure 6 contains the confusion matrices for ten of the CUB-200 classes. Note that data-dependent encodings find low cost partitions, discriminating classes prone to be confused in the first stages of the hierarchy (the first encoding bits), and keeping those harder classification problems to the leafs. A comparison of one-hot, random and data-

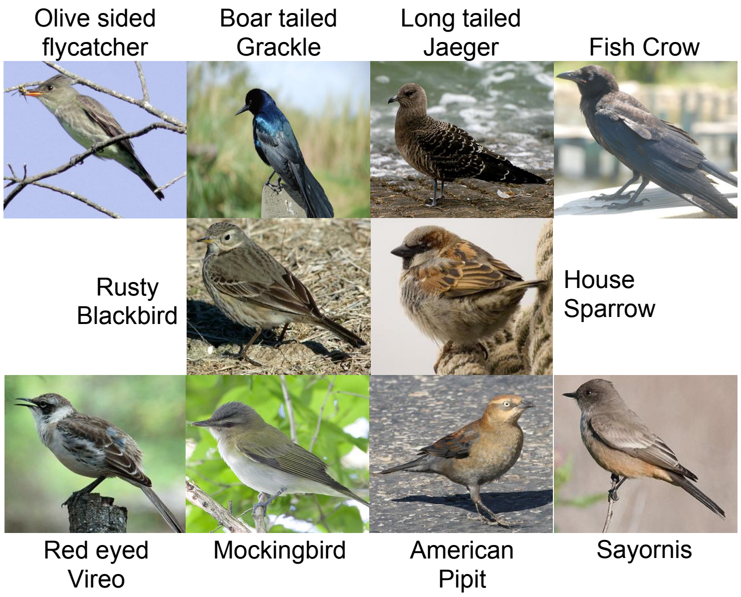


Figure 7: Confusion matrix classes.

dependent encodings for the classification of ”Fish crow” and ”Grackle” is shown in figure 8.

We lastly verify the correspondence of the metaclasses found with data-dependent encodings by computing the Pearson Correlation Coefficient (CCP) between the columns of the code-matrix and the attributes associated to each of the CUB-200 classes, see table 2.

As expected, the data-dependent code finds a high-level partition that already discriminates both classes. One-hot, instead acts directly at the class level, without being explicitly based on shared attributes. On the other hand, random codes, although also based on metaclasses (attributes), do not guarantee that those metaclasses are the most discriminative ones.

5. Conclusion

In this work, output codes are integrated with the training of deep CNNs on large-scale datasets. We found that CNNs trained on CIFAR-100, CUB200, Imagenet, and MIT Places using our approach show less sparsity at the output neurons. As a result,

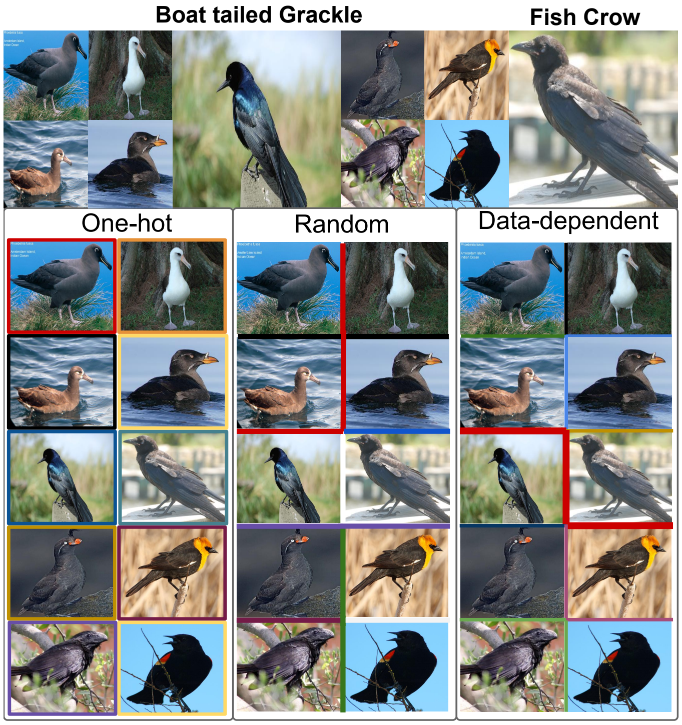


Figure 8: **Classifying Boat tailed Grackle and Fish Crow.** One-hot encoding directly assigns labels to each of the examples. Random encoding partitions groups of classes into meta-classes systematically. Data-depending codes first group aquatic and non-aquatic birds, eliminating posterior confusions.

models trained with our approach showed more robust gradient estimates and faster convergence rates than those trained with the prevalent one-hot encoding at a small cost, especially for huge label spaces. As a side effect, CNNs trained with our approach can use smaller minibatch sizes, lowering the memory consumption. Moreover, we showed that training with data-dependent codes based on eigenrepresentations of the class space allows for more efficient, hierarchical representations, achieving lower error rates than those trained with data-independent output codes.

Acknowledgements

Authors acknowledge the support of the Spanish project TIN2015-65464-R (MINECO FEDER), the 2016FI_B 01163 grant (Secretaria d'Universitats i Recerca del Departa-

Code bit	**1**	**2**	**3**	**4**	**5**	**6**
Attribute	Belly-color red	Head-pattern eyeline	breast-color blue	bill-color green	head-pattern unique	crown-color yellow
PCC	0.18	0.16	0.15	0.15	0.14	0.14
Attribute	Tail-shape rounded	Under-tail-color iridiscent	biill-color brown	belly-color pink	bill-shape all-purpose	tail-shape forked
PCC	-0.22	-0.17	-0.17	-0.16	-0.18	-0.18

(a) Random Code.

Code bit	**1**	**2**	**3**	**4**	**5**	**6**
Attribute	shape perching-like	primary-color yellow	back-color black	bill-color black	throat-color yellow	upperpart-color white
PCC	0.79	0.64	0.50	0.44	0.53	0.55
Attribute	size:medium	upper-tail-color brown	wing-color grey	primary-color red	primary-color rufous	belly-color black
PCC	-0.73	-0.56	-0.58	-0.38	-0.42	-0.48

(b) Data-dependent code.

Table 2: **Top CUB200 attributes by correlation with the code.** Random codes do not show relevant correlations with the data attributes, while data-dependent codes are visibly correlated with the attributes. Concretely, the first bit of the code, i.e. the partition with the lowest cut cost, is highly correlated with shape and size attributes (0.79). The sign of the PPC indicates the expected side of the bi-partition associated for the attribute. As expected, the PPC coefficient decreases in absolute value as the cut cost increases, since higher bits correspond to increasingly difficult partitions.

ment d'Economia i Coneixement de la Generalitat de Catalunya), and the COST Action IC1307 iV&L Net (European Network on Integrating Vision and Language) supported by COST (European Cooperation in Science and Technology). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU and a GTX TITAN GPU, used for this research.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. doi:10.1007/s11263-015-0816-y. 2, 4, 11

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755. 2

[3] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034. 2

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision 111 (1) (2015) 98–136. 2

[5] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015. 2

[6] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Advances in neural information processing systems, 2014, pp. 487–495. 2, 4, 10, 11

[7] S. Vijayanarasimhan, J. Shlens, R. Monga, J. Yagnik, Deep networks with large output spaces, arXiv preprint arXiv:1412.7479. 2

[8] S. Bengio, J. Weston, D. Grangier, Label embedding trees for large multi-class tasks, in: Advances in Neural Information Processing Systems, 2010, pp. 163–171. 2, 5

[9] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE transactions on pattern analysis and machine intelligence 38 (7) (2016) 1425–1438. 2, 5

[10] D. Hsu, S. Kakade, J. Langford, T. Zhang, Multi-label prediction via compressed sensing., in: NIPS, Vol. 22, 2009, pp. 772–780. 2

[11] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: Advances in neural information processing systems, 2013, pp. 2121–2129. 2, 3

[12] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for attribute-based classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 819–826. 2

[13] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, Journal of Machine Learning Research 6 (Sep) (2005) 1453–1484. 3

[14] J. Weston, S. Bengio, N. Usunier, Large scale image annotation: learning to rank with joint word-image embeddings, Machine learning 81 (1) (2010) 21–35. 3, 5

[15] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 120–134. 3, 6

[16] M. Á. Bautista, S. Escalera, X. Baró, P. Radeva, J. Vitriá, O. Pujol, Minimal design of error-correcting output codes, Pattern Recognition Letters 33 (6) (2012) 693–702. 3

[17] E. B. Kong, T. G. Dietterich, Error-correcting output coding corrects bias and variance., in: ICML, 1995, pp. 313–321. 3, 6

[18] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images. 4, 11

[19] K. Q. Weinberger, O. Chapelle, Large margin taxonomy embedding for document categorization, in: Advances in Neural Information Processing Systems, 2009, pp. 1737–1744. 4, 5

[20] X. Yu, Y. Aloimonos, Attribute-based transfer learning for object categorization with zero/one training example, in: European conference on computer vision, Springer, 2010, pp. 127–140. 5

[21] M. Rohrbach, M. Stark, B. Schiele, Evaluating knowledge transfer and zero-shot learning in a large-scale setting, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1641–1648. 5

[22] P. Kankuekul, A. Kawewong, S. Tangruamsub, O. Hasegawa, Online incremental attribute-based zero-shot learning, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3657–3664. 5

[23] Y. Amit, M. Fink, N. Srebro, S. Ullman, Uncovering shared structures in multiclass classification, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 17–24. 5

[24] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781. 5

[25] J. Weston, S. Bengio, N. Usunier, Wsabie: Scaling up to large vocabulary image annotation. 5

[26] N. Srivastava, R. R. Salakhutdinov, Discriminative transfer learning with tree-based priors, in: Advances in Neural Information Processing Systems, 2013, pp. 2094–2102. 5

[27] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, H. Adam, Large-scale object classification using label relation graphs, in: European Conference on Computer Vision, Springer, 2014, pp. 48–64. 5, 7

[28] T. Xiao, J. Zhang, K. Yang, Y. Peng, Z. Zhang, Error-driven incremental learning in deep convolutional neural network for large-scale image classification, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 177–186. 5

[29] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, Y. Yu, Hd-cnn: Hierarchical deep convolutional neural network for large scale visual recognition, in: ICCV'15: Proc. IEEE 15th International Conf. on Computer Vision, 2015. 5

[30] K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen, Deep learning of binary hash codes for fast image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 27–35. 5

[31] X. Bai, S. I. Niwas, W. Lin, B.-F. Ju, C. K. Kwoh, L. Wang, C. C. Sng, M. C. Aquino, P. T. Chew, Learning ecoc code matrix for multiclass classification with application to glaucoma diagnosis, Journal of medical systems 40 (4) (2016) 1–10. 6

[32] T. Windeatt, G. Ardeshir, Boosted ecoc ensembles for face recognition, in: IEE conference publication, Institution of Electrical Engineers, 2003, pp. 165–168. 6

[33] R. S. Smith, T. Windeatt, Facial action unit recognition using multi-class classification, Neurocomputing 150 (2015) 440–448. 6

[34] T. G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of artificial intelligence research 2 (1995) 263–286. 6, 7, 10

[35] Z. Jiang, Y. Wang, L. Davis, W. Andrews, V. Rozgic, Learning discriminative features via label consistent neural network, arXiv preprint arXiv:1602.01168. 6

[36] R. C. Bose, D. K. Ray-Chaudhuri, On a class of error correcting binary group codes, Information and control 3 (1) (1960) 68–79. 6

[37] E. L. Allwein, R. E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of machine learning research 1 (Dec) (2000) 113–141. 6, 13, 14

[38] R. Ghaderi, T. Windeau, Circular ecoc. a theoretical and experimental analysis, in: Pattern Recognition, 2000. Proceedings. 15th International Conference on, Vol. 2, IEEE, 2000, pp. 203–206. 6

[39] O. Pujol, P. Radeva, J. Vitria, Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1007–1012. 6

[40] S. Escalera, D. M. Tax, O. Pujol, P. Radeva, R. P. Duin, Subclass problem-dependent design for error-correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 1041–1054. 6

[41] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Machine learning 47 (2-3) (2002) 201–233. 6

[42] G. Griffin, P. Perona, Learning and using taxonomies for fast visual categorization, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8. 6

[43] X. Zhang, L. Liang, H.-Y. Shum, Spectral error correcting output codes for efficient multiclass recognition, sign (M (r, i) 1 (2009) 1. 6, 10

[44] H. Deng, G. Stathopoulos, C. Y. Suen, Applying error-correcting output coding to enhance convolutional neural network for target detection and pattern recognition, in: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE, 2010, pp. 4291–4294. 7

[45] S. Yang, P. Luo, C. C. Loy, K. W. Shum, X. Tang, Deep representation learning with target coding., in: AAAI, 2015, pp. 3848–3854. 7

[46] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, The handbook of brain theory and neural networks 3361 (10) (1995) 1995. 8

[47] T. Hastie, R. Tibshirani, et al., Classification by pairwise coupling, The annals of statistics 26 (2) (1998) 451–471. 10

[48] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on pattern analysis and machine intelligence 22 (8) (2000) 888–905. 10

[49] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-ucsd birds 200. 11

[50] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105. 11

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093. 11