Equiangular Basis Vectors

Yang Shen   Xuhao Sun   Xiu-Shen Wei*

School of Computer Science and Engineering, Nanjing University of Science and Technology, China

{shenyang_98,sunxh,weixs}@njust.edu.cn

Abstract

We propose Equiangular Basis Vectors (EBVs) for classification tasks. In deep neural networks, models usually end with a k-way fully connected layer with softmax *to handle different classification tasks. The learning objective of these methods can be summarized as mapping the learned feature representations to the samples' label space. While in metric learning approaches, the main objective is to learn a transformation function that maps training data points from the original space to a new space where similar points are closer while dissimilar points become farther apart. Different from previous methods, our EBVs generate normalized vector embeddings as "predefined classifiers" which are required to not only be with the equal status between each other, but also be as orthogonal as possible. By minimizing the spherical distance of the embedding of an input between its categorical EBV in training, the predictions can be obtained by identifying the categorical EBV with the smallest distance during inference. Various experiments on the ImageNet-1K dataset and other downstream tasks demonstrate that our method outperforms the general fully connected classifier while it does not introduce huge additional computation compared with classical metric learning methods. Our EBVs won the first place in the 2022 DIGIX Global AI Challenge, and our code is open-source and available at* https://github.com/NJUST-VIPGroup/Equiangular-Basis-Vectors.

1. Introduction

The pattern classification field developed around the end of the twentieth century aims to deal with the specific problem of assigning input signals to two or more classes [56]. In recent years, deep learning models have brought breakthroughs in processing image, video, audio, text, and other

(a) Fully Connected Layers with Softmax

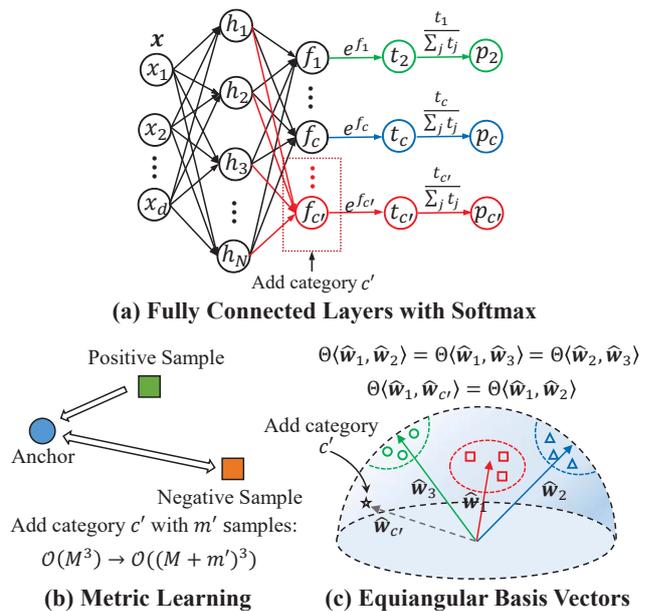**(b) Metric Learning**   **(c) Equiangular Basis Vectors**

Figure 1. Comparisons between typical classification paradigms and our proposed Equiangular Basis Vectors (EBVs). **(a)** A general classifier ends with k-way fully connected layers with softmax. When adding more categories, the trainable parameters of the classifier grow linearly. **(b)** Taking triplet embedding [60] as an example of classical metric learning methods, the complexity is $\mathcal{O}(M^3)$ when given M images and it will grows to $\mathcal{O}((M+m')^3)$ when adding a new category with m' images. **(c)** Our proposed EBVs. EBVs predefine fixed normalized vector embeddings for different categories and these embeddings will not be changed during the training stage. The trainable parameters of the network will not be changed with the growth of the number of categories while the complexity only grows from $\mathcal{O}(M)$ to $\mathcal{O}(M+m')$.

data [10, 19, 27, 58]. Aided by the rapid gains in hardware, deep learning methods today can easily overfit one million images [9] and easily overcomes the obstacle to the quality of handcrafted features in previous pattern classification tasks. Many approaches based on deep learning spring up like mushrooms and had been used to solve classification problems in various scenarios and settings such as remote sensing [38], few-shot [52], long-tailed [72], etc.

Figure 1 illustrates two typical classification paradigms. Nowadays, a large amount of deep learning methods [38, 72] adopt a trainable fully connected layer with softmax as the classifier. However, since the number of categories is fixed, the trainable parameters of the classifier rise as the number of categories becomes larger. For example, the memory consumption of a fully connected layer $\boldsymbol{W} \in \mathbb{R}^{d \times N}$ linearly scales up with the growth of the category number N and so is the cost to compute the matrix multiplication between the fully connected layer and the d-dimensional features. While some other methods based on classical metric learning [23, 30, 60–62] have to consider all the training samples and design positive/negative pairs then optimize a class center for each category, which requires a significant amount of extra computation for large-scale datasets, especially for those pre-training tasks.

In this paper, we propose Equiangular Basis Vectors (EBVs) to replace the fully connected layer associated with softmax within classification tasks in deep neural networks. EBVs predefine fixed normalized vector embeddings with equal status (equal angles) which will not be changed during the training stage. Specifically, EBVs pre-set a d-dimensional unit hypersphere, and for each category in the classification task, EBVs assign the category a d-dimensional normalized embedding on the surface of the hypersphere and we term these embedding as *basis vectors*. The spherical distance of each basis vector pair satisfies an artificially made rule to make the relationship between any two vectors as close to orthogonal as possible. In order to keep the trainable parameters of the deep neural networks constant with the growth of the category number N, we then propose the definition of EBVs based on Tammes Problem [55] and Equiangular Lines [57] in Section 3.2.

The learning objective of each category in our proposed EBVs is also different from previous classification methods. Compared with deep models that end with a fully connected layer to handle the classification tasks [19, 27], the meaning of the parameter weights within the fully connected layer in EBVs is not the relevance of a feature representation to a particular category but a fixed matrix which embed feature representations to a new space. Also, compared with regression methods [34, 51], EBVs do not need to learn the unified representations for different categories and optimize the distance between the representation of input images and category centers, which helps reduce the computational consumption for the extra unified representations learning. In contrast to classical metric learning approaches [14, 23, 50, 61], our EBVs do not need to measure the similarity among different training samples and constrain distance between each category, which will introduce a large amount of computational consumption for large-scale datasets. In our proposed method, the representations of different images learned by EBVs will be embedded into a normalized hypersphere and the learning

objective is altered to minimize the spherical distance of the learned representations with different predefined basis vectors. In addition, the spherical distance between each predefined basis vector is carefully constrained so that there is no need to spend extra cost in the optimization of these basis vectors. To quantitatively prove both the effectiveness and efficiency of our proposed EBVs, we evaluate EBVs on diverse computer vision tasks with large-scale datasets, including classification on ImageNet-1K, object detection on COCO, as well as semantic segmentation on ADE20K.

2. Related work

2.1. Deep networks for image classification

Image classification is the task of categorizing images into one of several predefined classes, which has been a fundamental problem in computer vision for a long time [4, 36]. It also forms the basis for many other computer vision tasks, *e.g.*, object detection [37], localization [28] and segmentation [39]. To solve the image classification problem, a dual-stage approach was used before the rise of deep learning. Specifically, handcrafted features were first extracted from images using feature descriptors. Then, a trainable classifier is adopted to perform the classification task with these input features [44]. The major hindrance of this approach was that the accuracy of the classification task was profoundly dependent on the design of the feature extraction stage, and this usually proved to be a formidable task [29].

In recent years, deep learning models that exploit multiple layers of nonlinear information processing, for feature extraction and transformation as well as for pattern analysis and classification, have been shown to overcome these challenges [44]. With the holding of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [49], a growing number of deep networks demonstrate superior classification performance [10, 15, 19, 27, 32, 33, 42]. Among them, Deep Convolutional Neural Networks (DCNNs) and Vision Transformers (ViTs) [10] have become the leading architectures for most image classification task in recent years. In addition, selected representative examples of other improvement attempts related to the following different aspects: network architecture, nonlinear activation functions, supervision components, regularization mechanisms and optimization techniques. Our proposed Equiangular Basis Vectors (EBVs) are different from these aspects as we do not change the overall architecture and training techniques but the optimization objectives for classification since we preliminary design fixed normalized vector embeddings for each predefined category.

2.2. Learning objectives

Besides traditional classifiers, machine learning or deep learning methods such as clustering [1, 45], regression [34, 51], metric learning [14, 50, 61], sparse learning [48, 67] can

be used to handle the classification tasks while the learning objectives of these methods vary considerably. In this section, we only discuss the learning objective of our EBVs with two prominent training paradigms of deep learning for the classification tasks, *i.e.*, general k-way classification layers [27] and classical deep metric learning [3, 66].

Regarding deep learning for classification, the k-way classification layer (always associated with `softmax`) is the most popular used approach for training deep models. It employs a single linear layer or multiple non-linear layers to map the learned deep representations to the semantic categories in the label space. The corresponding learning objective is minimizing the losses (*e.g.*, the cross-entropy loss [8]) between the mapped categorical signals (aka. predictions) with the ground truth categories. Compared with that, the learning objective of our EBVs is to make the vectorized embedding of input as close as possible to its categorical equiangular basis vector correspondingly.

The learning objective of classical deep metric learning seems similar to ours. However, there are several crucial and fundamental differences. Firstly, although deep metric learning minimizes the distances between sample vectors belonging to the same category and meanwhile maximizes the distances between samples from different categories, all these sample vectors are constantly changing during the model training process. But, in our method, the equiangular basis vectors corresponding to the categories are predefined, *i.e.*, they are fixed. More importantly, these equiangular basis vectors in the spherical space are forced to be with equal status and be as orthogonal as possible to each other, which could contribute to strong model discriminative ability and good classification accuracy. In practice, both with equal status and being orthogonal of vectors are strict conditions w.r.t. optimization. Thus, our EBVs predefine the satisfied categorical basis vectors and our learning objective focuses on only optimizing the learned feature representations to prevent the model from being unable to converge (when EBVs can also change under such strict conditions). Secondly, the learning objective of deep metric learning is upon massive training samples, while our learning objective is for the training sample and its fixed categorical vectorized embedding, *i.e.*, EBV. It could bring the computational economy, especially for the large-scale training data scenario, cf. Section 3.5. In addition, compared with several specific metric learning approaches, *e.g.*, the center loss [62], the prototypical network [52] and the nearest class mean approach [40], our method is still quite distant from them. Specifically, the basic idea of these works is to construct the "center" of samples belonging to a category to represent its semantics and then leverage the centers to optimize sample distances for classification. Similarly, since these categorical centers are always changing during training, these approaches will also encounter the same problems as aforementioned.

3. Methodology

3.1. Preliminaries

The proposed Equiangular Basis Vectors (EBVs) are based on the study with regard to Equiangular Lines [21, 22, 57] and the Tammes Problem [55].

A set of lines passing through the origin in $\mathbb{R}^d (d \in \{2, 3, 4, 5, \ldots\})$ is called *equiangular* if they are pairwise separated by the same angle. The study of equiangular lines was initiated by Haantjes [22] and it plays an important role in the coding theory [53] and quantum information theory [46].

The problem of how to determine the maximum number $N(d)$ of equiangular lines in a given dimension d was formally posed by Van Lint and Seidel [57] and has gained a major breakthrough by Jiang et al. [21] last year — Fix $0 < \alpha < 1$, let $N_\alpha(d)$ denote the maximum number of lines through the origin in $\mathbb{R}^d$ with pairwise common angle $\arccos \alpha$, let k denote the minimum number (if it exists) of vertices in a graph whose adjacency matrix has spectral radius exactly $(1 - \alpha)/(2\alpha)$. If $k < \infty$, then $N_\alpha(d) = \lfloor k(d-1)/(k-1) \rfloor$ for all sufficiently large d, and otherwise $N_\alpha(d) = d + o(d)$. In particular, $N_{1/(2k-1)}(d) = \lfloor k(d-1)/(k-1) \rfloor$ for every integer $k \geq 2$ and all sufficiently large d [21].

In simple terms, with a fixed common angle, the maximum number $N(d)$ of equiangular lines is linearly correlated with the dimension d as $d \to \infty$ while there is still no precise lower bound for $N(d)$ with smaller values of d [13]. Therefore, we further refer to the Tammes Problem. Let a_n be the maximal number with the property that one can place on a unit hypersphere $S^d \in \mathbb{R}^d (d \in \{3, 4, 5, \ldots\})$ n points so that the spherical distance of any two points is at least a_n. The problem of finding a_n together with the corresponding arrangement of each point is known as the Tammes Problem [55] or the optimal spherical code [12]. It is easy to find that the number of these points will be close to infinity as a_n tends to zero.

3.2. Definition of Equiangular Basis Vectors

In order to predefine fixed d-dimensional embeddings for as many categories as possible but still keep these embeddings at a distance from each other on a unit hypersphere $S^d \in \mathbb{R}^d$, we therefore propose Equiangular Basis Vectors (EBVs). Specifically, we fix $0 \leq \alpha < 1$ and let $\mathcal{N}_\alpha(d)$ denote the range of values for the number of coordinate vectors in $\mathbb{R}^d$ with the pairwise common angle between $\arccos \alpha$ and $\arccos -\alpha$. The problem of our proposed EBVs is to calculate the coordinates of each vector in the vector set $\mathcal{W}$ when given fixed α, d and $N \in \mathcal{N}_\alpha(d)$ if possible, *i.e.*, solve the set $\mathcal{W}$ which satisfies:

$$\forall \boldsymbol{w}_i, \boldsymbol{w}_j \in \mathcal{W}, i \neq j, \quad -\alpha \leq \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|} \leq \alpha, \quad (1)$$

where $\boldsymbol{w}_i \in \mathbb{R}^d$, $\mathrm{card}(\mathcal{W}) = N$ and $\|\cdot\|$ denotes the Euclidean norm. Let ϕ denote the spherical distance function, which can also be replaced by the cosine similarity function. EBVs produce a distribution over classes for a query point $\boldsymbol{v} \in \mathbb{R}^d$ based on softmax over cosine similarity to the N fixed coordinate vectors in the embedding space:

$$p(y = k|\boldsymbol{v}) = \frac{\exp(-\phi(\boldsymbol{v}, \boldsymbol{w}_k))}{\sum_{k'} \exp(-\phi(\boldsymbol{v}, \boldsymbol{w}_{k'}))}, \quad (2)$$

where $y = k \in \{1, 2, \ldots, N\}$ denotes the corresponding coordinate vector, which can also be seen as the corresponding label. While k' represents the associated basis vectors in $\mathcal{W}$. Relations between α, d and N are described in the supplementary materials. With such a set $\mathcal{W}$, the maximum number of categories it can handle is N. Additionally, training with any category number less than N, it is sufficient to randomly select any of the same numbers of coordinate vectors in $\mathcal{W}$, since these vectors are exactly equivalent.

3.3. How to generate EBVs?

The basic idea of our Equiangular Basis Vectors (EBVs) is to generate fixed normalized vector embeddings with equal status (equal angles) as "predefined classifiers" for all the relevant categories. The question then comes to how to calculate the predefined EBVs which satisfy Eq. (1). Therefore, we will discuss how to generate the proposed EBVs when given fixed α, d and $N \in \mathcal{N}_\alpha(d)$ in this section.

Assuming each $\boldsymbol{w}_i \in \mathcal{W}$ ($i = \{1, 2, \ldots, N\}$) as a line, we can construct the *Grassmannian Matrices* to solve the set $\mathcal{W}$ [11]. Specifically, we assemble the vectors in $\mathcal{W}$ into a matrix $\boldsymbol{W} \in \mathbb{R}^{d \times N}$, where $\boldsymbol{W} = [\boldsymbol{w}_1; \boldsymbol{w}_2; \ldots; \boldsymbol{w}_N]$. Then, the mutual-coherence of $\boldsymbol{W}$ is defined by:

$$\mu(\boldsymbol{W}) = \max_{1 \leq i,j \leq N, i \neq j} \frac{|\boldsymbol{w}_i^\top \cdot \boldsymbol{w}_j|}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|} = \max_{1 \leq i,j \leq N, i \neq j} \cos\theta_{ij}, \quad (3)$$

which is transformed to the smallest possible mutual-coherence possible [11]. Therefore, the lower bound for α is $\sqrt{\frac{N-d}{d(N-1)}}$ and the upper bound for $\mathcal{N}_\alpha(d)$ is $1 + \frac{d-1}{1-\alpha^2 d}$ ($d \leq N$, $1 - \alpha^2 d > 0$). In addition, such matrix is possible only if $N < \min(d(d+1)/2, (N-d)(N-d+1)/2)$. Construction of such a matrix has a strong connection with the packing of vectors/subspaces in the $\mathbb{R}^d$-space. In the case of $N = d$, we can simply construct a unitary matrix, while in other cases, it will be very hard to construct such a general Grassmannian matrix [11].

In the definition of our proposed EBVs, whether the angle between any two basis vectors $\boldsymbol{w}_i, \boldsymbol{w}_j \in \mathcal{W}$ ($i \neq j$) is $\arccos\alpha$ or $\arccos-\alpha$ is equivalent. It is clear that we can not construct the Grassmannian matrix in a situation such as $N = 30,000$ and $d = 200$. However, it is still possible for us to construct $\mathcal{W}$ which satisfies Eq. (1). Therefore, as

Algorithm 1 Generation of EBVs in a PyTorch-like style

```
# d: Dim for each coordinate vectors
# N: The number of coordinate vectors
# α: Threshold of ŵ_i · ŵ_j, i ≠ j
# W: The EBVs matrix W ∈ ℝ^{d×N}
# slice: In the case of N ≫ d, optimize W by slicing
# λ: Learning rate.

Initialize W randomly;
while True:
    #Normalize each row in W
    W = Normalize(W)
    for i in ⌈N/slice⌉:
        start = i * slice
        end = min(N, (i + 1) * slice)
        E = F.onehot(arange(start, end), N)
        C = (W[start:end]@W.T).abs()-E
        loss = ReLU(C − α).sum() # Cutout the
        gradient of vector pairs which satisfies
        −α ≤ ŵ_i · ŵ_j ≤ α
        loss.backward()
    if max(α, C.max()) < α + o(α):
        Save(W)
        break
    W = W − λ * W.grad # Update W
```

an alternative, we adopt Stochastic Gradient Descent [47] to search the set $\mathcal{W}$ that satisfies the definition of EBVs when given fixed α, d and N. Specifically, we random initialize a matrix $\boldsymbol{W} \in \mathbb{R}^{d \times N}$ with normalized rows such that the angle between any two vectors $\hat{\boldsymbol{w}}_i, \hat{\boldsymbol{w}}_j \in \mathbb{R}^d$, $i, j \in \{1, 2, \ldots, N\}$, $i \neq j$, $\hat{\boldsymbol{w}}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}$ can be represented as $\arccos(\hat{\boldsymbol{w}}_i \cdot \hat{\boldsymbol{w}}_j)$. Then, we cut out the gradient of those vector pairs which satisfies $-\alpha \leq \hat{\boldsymbol{w}}_i \cdot \hat{\boldsymbol{w}}_j \leq \alpha$ and optimize the remaining vector pairs. The optimization function of the generation of EBVs can be formulated by:

$$\arg\min_{\boldsymbol{W}} \sum_{i=1}^{N-1} \sum_{j>i}^{N} \max(\hat{\boldsymbol{w}}_i \cdot \hat{\boldsymbol{w}}_j - \alpha, 0). \quad (4)$$

Algorithm 1 provides the code of a simple generation method of the proposed EBVs in a PyTorch-like style. It is also worth mentioning that the EBVs matrix $\boldsymbol{W}$ will not be changed in the following training stage within all the tasks.

3.4. How to achieve the learning objective of EBVs?

Equiangular Basis Vectors (EBVs) provide fixed learning targets for each independent optimization objective, *i.e.*, semantic categories. In the following, we introduce how to achieve the learning objective of EBVs for all the training samples. Generally, a deep network is performed to extract the high-dimensional features, and a fully connected classification layer is then deployed to map the features to semantic categories. While in our proposed EBVs, each category will be bound to a unique normalized d-dimensional basis vector in $\mathcal{W}$. Thus, for a training sample x, we directly use a unified deep model to generate a d-dimensional embedding $\boldsymbol{v}$, as well as optimizing the cosine distance between $\boldsymbol{v}$ and

the relevant basis vector. Below we analyze the underlying distance/loss function used in the training stage.

For our EBVs, many existing distance functions including squared Euclidean distance, Mahalanobis distance, or cosine distance are permissible. However, a particular class of distance functions, *e.g.*, *regular Bregman divergences* [2], seems hard to explain and optimize in the proposed EBVs settings. In addition, intuitively, the straightforward way is to optimize the spherical distance between any two vectors on the surface of the hyper-sphere. Therefore, we adopt the cosine distance as the distance metric, which is widely used for measuring whether two inputs are similar or dissimilar and have been widely used in the Tammes problem [55].

Implementations Suppose we have M sample-label pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\}$ in N classes and their d-dimensional features $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M$ with $\boldsymbol{v}_i = f_{\boldsymbol{\theta}}(x_i)$, where $f_{\boldsymbol{\theta}}(\cdot)$ represents a feature extractor. A straightforward way to achieve the learning objective of EBVs is to directly optimize the cosine similarity between each $\boldsymbol{v}_i$ and the corresponding basis vector $\hat{\boldsymbol{w}}_{y_i} \in \mathcal{W}$.

However, the basis vector itself is not correlated with the training data since it is predefined. Fitting the training samples directly to the corresponding basis vectors might disrupt the representation learning. Therefore, we consider the conventional parametric `softmax` formulation, of which for image x_i with embedding $\boldsymbol{v}_i$, the probability of it being recognized as the y_i category can be formulated as :

$$P(y = y_i|\boldsymbol{v}_i) = \frac{\exp(\boldsymbol{m}_i^\top \boldsymbol{v}_i)}{\sum_{j=1}^{N} \exp(\boldsymbol{m}_j^\top \boldsymbol{v}_i)}, \qquad (5)$$

where $\boldsymbol{m}_j$ is a weight vector for the j-th class. Thus, according to Eq. (2) and Eq. (5), the probability of the embedding $\boldsymbol{v}_i$ being recognized as category y_i in our proposed EBVs can be formulated as:

$$P_{\text{EBVs}}(y = y_i|\boldsymbol{v}_i) = \frac{\exp(\hat{\boldsymbol{w}}_i \hat{\boldsymbol{v}}_i/\tau)}{\sum_{j=1}^{N} \exp(\hat{\boldsymbol{w}}_j \hat{\boldsymbol{v}}_i/\tau)}, \qquad (6)$$

where $\hat{\boldsymbol{v}}_i$ denotes the ℓ_2-normalized of the embedding $\boldsymbol{v}_i$ and τ is a hyper-parameter of temperature [17, 20, 64] which usually used in unsupervised learning. The learning objective is then to maximize the joint probability $\prod_{i=1}^{M} P_{\boldsymbol{\theta}}(y_i|f_{\boldsymbol{\theta}}(x_i))$, or equivalently to minimize the negative log-likelihood over the training set which can be formulated as:

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{M} \log P_{\text{EBVs}}(y = y_i|f_{\boldsymbol{\theta}}(x_i)). \qquad (7)$$

With this optimization approach, we relax the learning objective to make the angle between the embedding $\boldsymbol{v}$ of a training sample and its corresponding basis vector smaller than the angle between $\boldsymbol{v}$ and the other basis vectors.

3.5. Merits of our EBVs

We briefly summarize the merits of our proposed EBVs in this section. First of all, the embedding dimension of EBVs can be manually altered and the trainable parameters of the classifier will not grow linearly when the number of categories increases. Specifically, if the embedding dimension of each image is d' and the number of categories is N, the trainable parameters of a general classifier are $d' \times N$. However, with our proposed EBVs, as the dimension of the embedding for each category is fixed as d, then the trainable parameters of the classifier are $d' \times d$. What's more, d can be set at least close to $\sqrt{2N}$ when N is very large according to Section 3.3. Experimental results can be found in Table 2 and the supplementary materials.

Secondly, as the proposed EBVs are generated before the training step and the fixed d-dimensional embedding of each category will not be changed throughout the optimization process, EBVs will not introduce a large amount of computation during the training stage like the metric learning. In particular, metric learning methods such as pairwise/triplet embedding [60] are time-consuming, where the complexity can be $\mathcal{O}(K^2)$ and $\mathcal{O}(K^3)$ when given K images from N categories. However, the complexity of our proposed EBVs equals $\mathcal{O}(K)$. In addition, EBVs are not sensitive to the optimizers and previous training tricks while they can still achieve state-of-the-art performance as the results shown in Table 1 and the supplementary materials. Also, further results on the downstream tasks can be found in Section 4 and the supplementary materials.

4. Experiments

In this section, quantitative and qualitative experiments are exhibited on models that end with a k-way fully connected layer with `softmax` and our proposed method to demonstrate the effectiveness of the proposed Equiangular Basis Vectors (EBVs). All experiments are conducted with 8 RTX 3090 GPUs.

4.1. Quantitative results on image classification

4.1.1 Dataset and settings

We conduct the general image classification task on the ImageNet-1K [9] dataset, which contains 1.28M training images and 50K validation images from 1,000 different object classes. Then, we report ImageNet-1K top-1 accuracy on the validation set under a single crop setting.

In order to demonstrate the effectiveness of the proposed EBVs, we follow the state-of-the-art training methods provided by TorchVision [59] and timm [63]. For fair comparisons, we offer the following diverse training settings for the general classification task.

Table 1. Comparisons on the ImageNet-1K validation set. "FC" denotes models ending with a 1000-way fully connected layer with `softmax`. The test size for each image is set as "224^2" if there is one result while it is set as "$224^2/256^2$" if there exist two results.

Backbone	Method	Optimizer	LR	Epoch	Setting	Params.	GFLOPs	Test size	#Forward pass	Top-1 Acc (%)
ResNet-50	FC	SGD	0.1	90	A0	25.6M	4.1	224^2	600k	77.15
ResNet-50	FC EBVs	SGD	0.5	5/100	A1	25.6M	4.1	$224^2/256^2$	131k	76.88/77.92 **77.55/78.73**
ResNet-50	FC EBVs	SGD	0.5	5/100	A2	25.6M	4.1	$224^2/256^2$	131k	77.71/78.59 **78.14/78.99**
ResNet-50*	FC EBVs	SGD	0.5	5/600	A2	25.6M	4.1	$224^2/256^2$	755k	79.51/ – **79.73/80.45**
ResNet-50	FC FC EBVs	AdamW	0.001 0.01 0.01	5/100	A1	25.6M	4.1	$224^2/256^2$	131k	72.57/73.79 72.51/74.03 **75.62/77.14**
ResNet-50	FC FC EBVs	AdamW	0.001 0.01 0.01	5/100	A2	25.6M	4.1	$224^2/256^2$	131k	75.42/76.48 NaN **76.46/77.52**
Swin-T	FC EBVs	AdamW	0.001	5/100	A1	28.3M	4.5	224^2	131k	75.64 **78.37**
Swin-T	FC EBVs	AdamW	0.001	5/100	A2	28.3M	4.5	224^2	131k	79.12 **79.34**

* represents the result of the model with a general fully connected layer with `softmax` is provided by TorchVision [59].

Setting A0 The offical training setting provided in ResNet [19]. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [27]. It trains the backbone by using SGD optimizer with momentum as 0.9, weight decay as 1×10^{-4} and batch size as 256. The training iteration is up to 60×10^4. The standard color augmentation in [27] is used. It adopts the standard **10-crop testing** [27] in the validation stage.

Setting A1 We employ an AdamW [35] or an SGD [47] optimizer for 100 epochs using a cosine decay learning rate scheduler and 5 epochs of linear warm-up. The batch size is set as 1024 and the initial learning rate for the AdamW optimizer is 0.01 or 0.001 while it is set as 0.5 for the SGD optimizer. A 224×224 crop is randomly sampled from each image and the weight decay is set as 2×10^{-5}. We perform TrivialAugment [41], which is extremely simple and can be considered "parameter-free". We also adopt random erasing [71] and the probability is set as 0.1.

Setting A2 On the basis of Setting A1, we add label-smoothing [54] and the value is set as 0.1. We also perform mixup [69] and cutmix [68]. The setting of hyper-parameters of the two techniques is the same with TorchVision [59].

We adopt ResNet-50 [19] and Swin-T [32] as the typical backbone for Convolutional Neural Networks (CNNs) and Vision Transformers [10]. The hyper-parameters τ for the proposed EBVs are set as 0.07 for all the following experiments unless otherwise specified.

4.1.2 Main results

Table 1 presents comparisons between deep networks ending with FC, *i.e.*, a 1000-way classification layer with `softmax` and our proposed EBVs. The fixed α, d and N of EBVs are set as 0.004, 1000 and 1000, respectively. When adopting ResNet-50 as the backbone, EBVs outperform FC among all the settings. In addition, when adopting AdamW as the optimizer, EBVs gain 3.11%/1.04% improvement over FC under Setting A1 and Setting A2 when using $224^2/256^2$ testing. EBVs still gain around 2.7%/0.2% improvement on Swin-T under Setting A1 and Setting A2. As previous training techniques summarised by both TorchVision [59] and timm [63] are under the setting of FC, we suspect it will have some impact on the performance of EBVs. We further conduct ablation studies in the supplementary materials.

4.1.3 Ablation studies

We conduct ablation studies on the dimension d of each basis vector to demonstrate the merits of our proposed EBVs. Table 2 reports ImageNet-1K top-1 accuracy on the validation set with different dimensions d. Taking $d = 100$ as an example, the dimension of the embedding of an image is set as 100. We adopt ResNet-50 as the backbone and the setting of all the experiments follows Setting A2. We use SGD [47] as the optimizer and the initial learning rate is set as 0.01. The test size of each image is set as 224^2 and 256^2. The Min. Ang. which represents the minimum angle between each two basis vectors is defined as $\min_{1 \leq i,j \leq N, i \neq j} \min(\pi - \arccos(\hat{\boldsymbol{w}}_i \cdot$

Table 2. Ablation studies on the dimension of our proposed EBVs. "EBVs Dim." denotes the embedding dimension for each category. The test size for each image is set as 224^2 and 256^2.

EBVs Dim.	Epoch	Min. Ang.	Ave. Del. Ang.	Top-1 Acc (%)
100	5/100	82.24°	4.84°	76.10/76.83
200	5/100	85.41°	3.37°	77.04/77.83
300	5/100	86.67°	2.64°	77.25/78.08
400	5/100	87.48°	2.18°	77.20/78.38
500	5/100	87.99°	1.82°	77.39/78.34
1000	5/100	89.98°	0.01°	**78.14/78.99**
2000	5/100	89.98°	0.01°	77.87/78.58
100	5/300	82.24°	4.84°	78.33/79.25
1000	5/300	89.98°	0.01°	**79.10/79.96**

Table 3. Object detection and instance segmentation on the COCO 2017 dataset. Models are based on Mask R-CNN [18] and "1×" denotes we train models for 12 epochs while "3×" denotes 36 epochs. "AP^b" and "AP^m" refer to bounding box AP and mask AP, respectively. Results shaded in gray denote models ending with the general fully connected classifier while others denote models ending with our proposed EBVs.

Backbone	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-50	1×	38.2	58.8	41.4	34.7	55.7	37.2
ResNet-50 (EBVs)	1×	**38.3**	**59.0**	**42.0**	**35.2**	**56.2**	**37.7**
ResNet-50	3×	40.9	61.3	44.8	37.1	58.3	39.9
ResNet-50 (EBVs)	3×	**41.1**	**61.7**	**44.9**	**37.7**	**58.9**	**40.5**
Swin-T	1×	42.7	65.2	46.8	39.3	**62.2**	42.2
Swin-T (EBVs)	1×	**42.8**	**65.3**	**47.2**	**39.4**	62.2	**42.6**

$\hat{\boldsymbol{w}}_j), \arccos(\hat{\boldsymbol{w}}_i \cdot \hat{\boldsymbol{w}}_j))$, where N represents the number of categories. While the Ave. Del. Ang. which represents the average angle between each of two basis vectors is defined as $\frac{N(N-1)}{2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left| \arccos(\hat{\boldsymbol{w}}_i \cdot \hat{\boldsymbol{w}}_j) - \frac{\pi}{2} \right|$. For a clearer representation, we adopt an angle instead of a radian. It can be seen from Table 2 that dimension has an impact on the performance of a model when training with limited cycles. However, this gap decreases a lot when adopting longer training cycles. Taking $d = 100$ and $d = 1000$ as an example, the top-1 accuracy under $d = 1000$ gains 2.04%/2.16% improvement over $d = 100$ when training with 105 epochs while this gap is reduced to 0.77%/0.71% when training with 305 epochs.

4.2. Empirical evaluations on object detection

Settings Object detection and instance segmentation experiments are conducted on the COCO 2017 benchmark [31], which contains 118K images in the training set and 5K images in the validation set. We consider Mask R-CNN [18] in MMdetection [5] as the detection framework and adopt

Table 4. Results of semantic segmentation on the ADE20K validation set. Results shaded in gray denote models ending with the general fully connected classifier while others denote models ending with our proposed EBVs. "1×" denotes we train models for 80,000 steps while "2×" denotes 160,000 steps.

Backbone	Schedule	mIoU	mIoU(ms+flip)
ResNet-18	1×	38.76	39.81
ResNet-18 (EBVs)	1×	**38.98**	**40.09**
ResNet-18	2×	39.23	39.97
ResNet-18 (EBVs)	2×	**39.75**	**40.91**
ResNet-50	1×	40.70	41.81
ResNet-50 (EBVs)	1×	**42.73**	**44.19**
ResNet-50	2×	42.05	42.78
ResNet-50 (EBVs)	2×	**42.44**	**43.94**
Swin-T	2×	**44.41**	45.79
Swin-T (EBVs)	2×	44.30	**45.88**

ResNet [19] and Swin Transformer [32] as backbones. For fair comparisons, we utilize the same settings according to MMdetection and previous work. We scaled the image to a maximum of 800 on the short side and 1333 on the long side. We adopt 50% scale horizontal flip and use SGD [47] with momentum= 0.9 as the optimizer, the initial learning rate is 0.02. The weight decay equals 0.0001 while the batch size is 16 for experiments under the backbone of ResNet. For experiments on Swin Transformer, we use Adam [24] as the optimizer and the initial learning rate is 0.0001. The weight decay is changed to 0.05. To demonstrate the effectiveness of the proposed EBVs, we replace both instance-level and pixel-level classifiers in the detection framework with EBVs. In order to control the parameters of each backbone for fair comparisons, the dimension d of the basis vectors is kept consistent with the number of categories. All backbone models are pre-trained on the ImageNet-1K training set.

Results Table 3 presents comparisons between the general framework and our proposed EBVs. When adopting ResNet-50 as the backbone, the framework ending with EBVs outperforms those ending with fully connected layers among six evaluation indicators. While adopting Swin-T as the backbone, the framework ending with EBVs also achieves comparative results with state-of-the-art performance.

4.3. Empirical evaluations on semantic segmentation

Settings Semantic segmentation experiments are conducted on the ADE20K [73], which contains 20,000 images in the training set and 2,000 images in the validation set. We adopt FPN [25] and UperNet [65] in MMSEG [6] as segmentation framework and adopt ResNet [19] and Swin Trans-
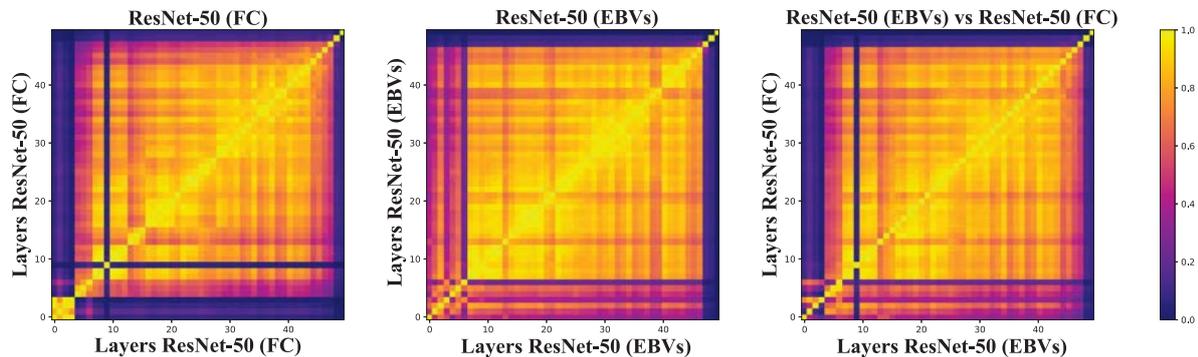
Figure 2. Representation structure of ResNet-50. **Left:** Similarity between layers within ResNet-50 ends with a general fully connected classifier (FC). It shows that the last few layers share minimal similarity with the shallow layers while a few shallow layers also share minimal similarity with all the other layers. **Middle:** Similarity between layers within ResNet-50 ends with EBVs. Only the last few layers share minimal similarity with other layers. **Right:** Similarity between layers across ResNet-50 ends with a general fully connected layer with `softmax` and our proposed EBVs. Around 10% initial layers share a little similarity while the last few layers share the least similarity.

former [32] as backbones. For fair comparisons, we utilize the same settings in MMSEG. For experiments on ResNet, we use SGD [47] as the optimizer. The initial learning rate equals 0.01, weight decay equals 0.0005, momentum equals 0.9 and batch size is set as 16. For experiments on Swin Transformer, we use AdamW [35] as the optimizer and the initial learning rate is 0.00006 while the weight decay is set as 0.01. To perform our proposed method, We modify pixel-level classifiers in both the decode head and auxiliary head to EBVs. All backbone models are pre-trained on the ImageNet-1K training set.

Results According to Table 4, we find that EBVs surpass the framework ending with fully connected layers as the classifier under the backbone of ResNet. Additionally, when the training step is set as 80,000, EBVs gain a higher mIoU score than the general framework trained for 160,000 steps. While adopting Swin-T as the backbone, the framework ending with EBVs still achieving comparative performance.

5. Discussions

We explore whether there are differences in the way EBVs represent and solve image tasks compared to a fully connected layer with `softmax` in this section. In order to answer this question, we have to analyze the features in the hidden layers as features are usually spread across neurons. However, different layers generally have different numbers of neurons. Recently, Raghu et al. [43] and Zhen et al. [70] applied the Centered Kernel Alignment (CKA) [7, 26] to solve the above task. CKA is effective because it involves no constraint on the number of neurons. It is also independent of the orthogonal transformations of representations [70]. Therefore, we adopt CKA to analyze the above question.

We pick ResNet-50 as the backbone while the experi-

ments on Swin-T can be found in the supplementary materials. One model ends with a general fully connected (FC) classifier and another model ends with our proposed EBVs. Both of the models are pre-trained on the ImageNet-1K [9] training dataset. We use 49 convolutional layers and the last fully connected layer. We plot CKA similarities between all pairs of layers within the whole ImageNet-1K validation dataset in Figure 2. As shown in Figure 2, for the ResNet-50 model ending with a general FC classifier, the feature similarity shows that around 10% initial convolutional layers almost share no similarity with all the other layers while they themselves share a high similarity. If we change the last layer and the learning objective with our proposed EBVs, these shallow layers then share a relatively higher similarity.

6. Conclusion

In this paper, we proposed Equiangular Basis Vectors (EBVs) for classification tasks. Different from previous classifiers and classical metric learning methods, EBVs predefined a fixed embedding for all the semantic categories. The learning objective of EBVs then changed to minimize the spherical distance of the embedding of input and each basis vector. Various experiments on the ImageNet-1K [9], COCO 2017 [31] and ADE20K [73] datasets and ablation studies also demonstrated the effectiveness of our proposed EBVs. In the future, we would like to explore relations between each basis vector pair and embed hierarchies when generating the proposed EBVs as the normal Euclidean space could not naturally embed hierarchies on datasets with known semantic hierarchies [16]. We also would like to explore the performance of EBVs in the case when the number of categories became very large. In addition, EBVs could also be regarded as advanced classifiers, it is interesting to adopt EBVs in other related tasks, *e.g.*, incremental learning, few-shot learning, etc.

References

[1] L Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 96–103, 1998. 2

[2] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. Clustering with Bregman divergences. *Journal of Mach. Learn. Research*, 6(10):1705–1749, 2005. 5

[3] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013. 3

[4] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10(5):1055–1064, 1999. 2

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 7

[6] MMSegmentation Contributors. MMSegmentation: Open MMLab semantic segmentation toolbox and benchmark. *Availabe online: https://github. com/open-mmlab/mmsegmentation (accessed on 18 May 2022)*, 2020. 7

[7] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Mach. Learn. Research*, 13:795–828, 2012. 8

[8] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005. 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009. 1, 5, 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 6

[11] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*, volume 2. Springer, 2010. 4

[12] Thomas Ericson and Victor Zinoviev. *Codes on Euclidean spheres*. Elsevier, 2001. 3

[13] Alexey Glazyrin and Wei-Hsuan Yu. Upper bounds for s-distance sets and equiangular lines. *Advances in Mathematics*, 330:810–833, 2018. 3

[14] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In *Adv. Neural Inform. Process. Syst.*, volume 17, pages 13–18, 2004. 2

[15] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022. 2

[16] Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X Yu. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11–20, 2022. 8

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9729–9738, 2020. 5

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. 7

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1, 2, 6, 7

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[21] Zilin Jiang, Jonathan Tidor, Yuan Yao, Shengtong Zhang, and Yufei Zhao. Equiangular lines with a fixed angle. *Annals of Mathematics*, 194(3):729–743, 2021. 3

[22] Haantjes Johannes. Equilateral point-sets in elliptic two- and three-dimensional spaces. *Nieuw Arch. Wiskunde*, 22(2):355–362, 1948. 3

[23] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[25] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6399–6408, 2019. 7

[26] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Int. Conf. Mach. Learn.*, pages 3519–3529, 2019. 8

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 2, 3, 6

[28] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8, 2008. 2

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[30] Daryl Lim, Gert Lanckriet, and Brian McFee. Robust structural metric learning. In *Int. Conf. Mach. Learn.*, pages 615–623, 2013. 2

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 7, 8

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 2, 6, 7, 8

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11976–11986, 2022. 2

[34] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011. 2

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 8

[36] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote sensing*, 28(5):823–870, 2007. 2

[37] Tomasz Malisiewicz, Abhinav Shrivastava, Abhinav Gupta, and Alexei A Efros. Exemplar-SVMs for visual object detection, label transfer and image retrieval. In *Int. Conf. Mach. Learn.*, pages 7–8, 2012. 2

[38] Aaron E Maxwell, Timothy A Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, 2018. 1, 2

[39] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Int. Conf. Mach. Learn.*, pages 591–598, 2000. 2

[40] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2624–2637, 2013. 3

[41] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Int. Conf. Comput. Vis.*, pages 774–782, 2021. 6

[42] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10428–10436, 2020. 2

[43] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Adv. Neural Inform. Process. Syst.*, pages 12116–12128, 2021. 8

[44] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. 2

[45] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Classification and clustering of arguments with contextualized word embeddings. *arXiv preprint arXiv:1906.09821*, 2019. 2

[46] Joseph M Renes, Robin Blume-Kohout, Andrew J Scott, and Carlton M Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, 2004. 3

[47] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951. 4, 6, 7, 8

[48] Fernando Rodriguez and Guillermo Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. Technical report, MINNESOTA UNIV MINNEAPOLIS, 2008. 2

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 2

[50] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419, 2007. 2

[51] Raied Salman and Vojislav Kecman. Regression as classification. In *Proceedings of IEEE Southeastcon*, pages 1–6, 2012. 2

[52] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, pages 4080–4090, 2017. 1, 3

[53] Thomas Strohmer and Robert W Heath Jr. Grassmannian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003. 3

[54] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2818–2826, 2016. 6

[55] Pieter Merkus Lambertus Tammes. On the origin of number and arrangement of the places of exit on the surface of pollen-grains. *Recueil Des Travaux Botaniques Néerlandais*, 27(1):1–84, 1930. 2, 3, 5

[56] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann. Review of classifier combination methods. *Mach. Learn. in Document Anal. and Recog.*, pages 361–386, 2008. 1

[57] Jacobus H van Lint and Johan J Seidel. Equilateral point sets in elliptic geometry. *Indag. Math*, 28(3):335–348, 1966. 2, 3

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 6000–6010, 2017. 1

[59] Vasilis Vryniotis. How to train State-of-The-Art models using TorchVision's latest primitives. https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/, 2021. 5, 6

[60] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1386–1393, 2014. 1, 2, 5

[61] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Mach. Learn. Research*, 10(2):207–244, 2009. 2

[62] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Eur. Conf. Comput. Vis.*, pages 499–515, 2016. 2, 3

[63] Ross Wightman, Hugo Touvron, and Hervé Jégou. ResNet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 5, 6

[64] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3733–3742, 2018. 5

[65] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Eur. Conf. Comput. Vis.*, pages 418–434, 2018. 7

[66] Han-Jia Ye, De-Chuan Zhan, Nan Li, and Yuan Jiang. Learning multiple local metrics: Global consideration helps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(7):1698–1712, 2019. 3

[67] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan. Visual classification with multitask joint sparse representation. *IEEE Trans. Image Process.*, 21(10):4349–4360, 2012. 2

[68] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Int. Conf. Comput. Vis.*, pages 6023–6032, 2019. 6

[69] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6

[70] Xingjian Zhen, Zihang Meng, Rudrasis Chakraborty, and Vikas Singh. On the versatile uses of partial distance correlation in deep learning. In *Eur. Conf. Comput. Vis.*, pages 327–346, 2022. 8

[71] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020. 6

[72] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9719–9728, 2020. 1, 2

[73] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20k dataset. *Int. J. Comput. Vis.*, 127(3):302–321, 2019. 7, 8