

# Reaching Nirvana: Maximizing the Margin in Both Euclidean and Angular Spaces for Deep Neural Network Classification

Hakan Cevikalp<sup>✉</sup>, Member, IEEE, Hasan Saribas<sup>✉</sup>, and Bedirhan Uzun

**Abstract**—The classification loss functions used in deep neural network classifiers can be split into two categories based on maximizing the margin in either Euclidean or angular spaces. Euclidean distances between sample vectors are used during classification for the methods maximizing the margin in Euclidean spaces whereas the Cosine similarity distance is used during the testing stage for the methods maximizing the margin in the angular spaces. This article introduces a novel classification loss that maximizes the margin in both the Euclidean and angular spaces at the same time. This way, the Euclidean and Cosine distances will produce similar and consistent results and complement each other, which will in turn improve the accuracies. The proposed loss function enforces the samples of classes to cluster around the centers that represent them. The centers approximating classes are chosen from the boundary of a hypersphere, and the pair-wise distances between class centers are always equivalent. This restriction corresponds to choosing centers from the vertices of a regular simplex inscribed in a hypersphere. The proposed loss function can be effortlessly applied to classical classification problems as there is a single hyperparameter that must be set by the user, and setting this parameter is straightforward. Additionally, the proposed method can effectively reject test samples from unfamiliar classes by measuring their distances from the known class centers, which are compactly clustered around their corresponding centers. Therefore, the proposed technique is especially suitable for open set recognition problems. Despite its simplicity, experimental studies have demonstrated that the proposed method outperforms other techniques in both open set recognition and classical classification problems. Interested individuals can access the source code for the proposed approach at <https://github.com/Cevikalp/dsc>.

**Index Terms**—Classification, computer vision, deep learning, neural collapse, open set recognition, simplex classifier.

## I. INTRODUCTION

**D**EEP neural network classifiers have been dominating many fields including computer vision by achieving the state-of-the-art accuracies in many tasks such as visual object, activity, face, and scene classification. Therefore, new deep neural network architectures and different classification losses have been constantly developing. The softmax loss

function is the most common function used for classification in deep neural network classifiers. Although the softmax loss yields satisfactory accuracies for general object classification problems, its performance for discrimination of the instances coming from the same class categories (e.g., face recognition) or open set recognition (a classification scenario that allows the test samples to come from the novel classes) is not satisfactory. The performance decrease is typically attributed to two factors: there is no mechanism for enforcing large-margin between classes and the softmax does not attempt to minimize the within-class scatter which is critical for obtaining good accuracies in open set recognition problems.

To improve the classification accuracies of the deep neural network classifiers, many researchers focused on maximizing the margin between classes. The recent methods can be roughly grouped into two categories based on maximizing the margin in either Euclidean or angular spaces. The methods targeting margin maximization in the Euclidean spaces attempt to minimize the Euclidean distances among the samples coming from the same classes and maximize the distances among the samples coming from different classes. Euclidean distances are used during the testing stage after the network is trained. In contrast, the methods maximizing the margin in the angular spaces use the cosine distances for classification.

In this article, we propose a novel method that maximizes the margin in both the Euclidean and angular spaces at the same time. The proposed methodology first selects class centers from the vertices of a regular simplex inscribed in a hypersphere and utilizes a loss function that minimizes the distances between the samples and their corresponding class centers.

### A. Related Work

Wen et al. [1], [2] introduced the center loss for face recognition to maximize the margin Euclidean space, and they reported significant improvements over the method using the softmax loss function in the context of face recognition. The range loss is combined with the softmax loss function in [3] to maximize the margin in Euclidean spaces. Wei et al. [4] proposed a classifier that combines the softmax loss and center loss functions with the minimum margin loss. A method combining the softmax loss function with the marginal loss is proposed by Deng et al. [5]. Cevikalp et al. [6] proposed a deep neural network based open set

Manuscript received 27 February 2023; revised 17 October 2023 and 4 April 2024; accepted 29 July 2024. Date of publication 12 August 2024; date of current version 5 May 2025. This work was supported by the Scientific and Technological Research Council of Türkiye (TUBİTAK) under Grant EEEAG-121E390. (Corresponding author: Hakan Cevikalp.)

Hakan Cevikalp and Bedirhan Uzun are with the Department of Electrical and Electronics Engineering, Eskişehir Osmangazi University, 26040 Eskişehir, Türkiye (e-mail: hakan.cevikalp@gmail.com).

Hasan Saribas is with the AIE Department, Huawei Türkiye Research and Development Center, 34768 Istanbul, Türkiye.

Digital Object Identifier 10.1109/TNNLS.2024.3437641

2162-237X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

recognition method that returns compact class acceptance regions for each known class. In this framework, hinge loss and polyhedral conic functions are used for the between-class separation. The methods using contrastive loss [7] also return compact class acceptance regions. To this end, they minimize the Euclidean distances of the positive sample pairs and penalize the negative pairs that have a distance smaller than a given margin threshold. In a similar manner, Schroff et al. [8], Hoffer and Ailon [9], Sohn [10], and Roy et al. [11] employ triplet loss function that uses triplets including a positive sample, a negative sample and an anchor. An anchor is also a positive sample, thus the within-class compactness is achieved by minimizing the Euclidean distances between the anchor and positive samples whereas the distances between the anchor and negative samples are maximized for the between-class separation. The employment of contrastive or triplet loss functions in methods has a significant drawback, which is the quadratic or cubic growth of the number of sample pairs or triplets compared to the total number of samples. This leads to slow convergence and instability in the training process, necessitating cautious data sampling/mining to mitigate these issues. Overall, the majority of the methods maximizing margin in the Euclidean spaces have shortcomings in a way that they are too complex since the user has to set many weighting and margin parameters. Furthermore, many of these methods are not suitable for open set recognition problems since they do not return compact acceptance regions for classes.

The methods that enlarge the margin in the angular spaces typically revise the classical softmax loss functions to maximize the angular margins between rival classes. These methods use either multiplicative or additive margins for the inter-class separations in the angular spaces. Among these, the SphereFace [12], [13] and the RegularFace [14] methods employ multiplicative margins whereas the CosFace [15] and the ArcFace [16] methods use additive margins. Majority of these methods normalize the feature vectors, classifier weights or both of them since the similarities are computed by using the angles. We would like to point out that almost all methods that maximize the margin in the angular space are proposed for face recognition. As indicated in [6], these methods use subspace approximations for the classes and the similarities are measured by using the angles between sample vectors. However, subspace approximations work well for the classification settings where the number of the features is much larger than the number of class specific samples. This is typically satisfied for the face recognition problems, but there are many classification tasks that do not satisfy this criterion. In addition to this problem, these methods are also complex since they have many parameters that must be set by the user as in the methods that maximize the margin in the Euclidean spaces.

The methods that are most closer to the proposed methodology are proposed in [17], [18], and [19]. These methods introduce loss functions for learning uniformly distributed representations on the hypersphere manifold through potential energy minimization. However, these studies consider the layer regularization problem rather than the direct classification problem and apply hyperspherical uniformity to the learned

weights. The main idea is to learn diverse deep neural network weights that are uniformly distributed on a hypersphere in order to reduce the redundancy. Therefore, these methods are more complex (in some sense it is also more sophisticated since it applies the hyperspherical uniformity to all neural network layers). Consequently, there are many hyperparameters that must be fixed in the resulting method. Also, when this idea is used in the classification layer, the distances between the resulting class representative weights are not equivalent as in our proposed method. A related study called UniformFace [20] used the same idea in the classification layer only and introduced uniform loss function to learn equidistributed representations for face recognition. Another similar method using class centroids is introduced in [21] for distance metric learning. Although this study focuses on distance metric learning, it uses class centers chosen as the basis vectors of  $C$ -dimensional space as anchors. Then, as in triplet loss, it attempts to minimize the distances between the data samples and the corresponding class centers and to maximize the distances between the samples and rival class centers. The selected class centers are fixed as in our proposed method and it has a restriction that the feature dimension size must be larger than or equal to the number of classes similar to our case. Compared to this method, our proposed method is much simpler and the run-time complexity of the proposed method is significantly less. Additionally, there are two significant oversights made by the authors in their proposed methodology. The initial oversight concerns their choice of centers, which are selected from the surface of a unit hypersphere (a hypersphere with a radius of 1). As expounded upon below, it becomes apparent that data samples tend to cluster near the surface of an expanding hypersphere as the dimensionality increases. Consequently, establishing the hypersphere radius as 1 is not well-suited for high-dimensional feature spaces, a viewpoint that is supported with findings reported in studies such as ArcFace [16] and CosFace [15]. The second concern revolves around the exclusive use of a fully connected layer to increase dimensionality, particularly when the feature dimension is smaller than the number of classes. A fully connected layer just uses the linear combination of existing features and the resulting space has the same dimensionality as in the original feature space in the best case scenario (this issue is explained in more details below). As a result, the dimensionality is not increased, and this method will not work for large-scale problems where the number of classes is very large.

There are studies using or mentioning simplex centers as in our proposed method. Among these methods, Papyan et al. [22] shows that the samples of different classes cluster around the class centers forming the vertices of a regular simplex (as we proposed in this study) at the last stages of the learning process when the linear classifiers are used with the softmax loss function and feature dimension is higher than the number of classes. They show that the lengths of the vectors of the class means (after centering by their global mean) converge to the same length and the angles between pair-wise center vectors become equal during the last training stages (it is called terminal phase of the training in the study) of the deep neural

networks using linear classifiers. This method is different than our proposed method in the sense that they do not use fixed class centers chosen from the vertices of a simplex. Instead, they directly use the softmax loss function and learn class weights. In general, they simply provide theoretical arguments showing that using the softmax loss function with the linear classifiers yields embeddings where the class samples cluster around vertices of a regular simplex after some kind of normalizations. Pernici et al. [23], [24], Kasarla et al. [25], and Bytyqi et al. [26] use fixed centers chosen from the vertices of a regular simplex as in our proposed method. But, all of them utilize variants of the softmax loss function including hyperparameters that must be fixed by the user. None of them proposes a simple loss function as in our proposed method. Using the softmax loss function yields radial distributions as illustrated in these studies. Therefore, their success is not satisfactory especially in open set recognition problems since the resulting embeddings are not compact as in our proposed method, please see related discussion given in Section II-C below. Also, none of the studies considered the case when the dimension is smaller than the number of samples and conducted experiments on this setting. For such cases, we need to increase the dimension of the feature space and we propose solutions to handle this case. In contrast, none of these methods propose an effective solution for this case. Yang et al. [27] introduced an alternative loss function named dot regression loss, which, like our proposed method, utilizes centers selected from the vertices of a regular simplex. However, their approach requires the selection of two parameters, making our method comparatively simpler. Additionally, the loss function described in [27] mandates that feature samples conform to the surface of a hypersphere with a predefined radius, akin to the spherical embeddings used in the ArcFace method [16]. In contrast, our method does not impose such constraints, allowing the samples to occupy the full feature space for embedding.

## B. Contributions

The methods that maximize the margin in Euclidean or angular spaces mentioned above have the shortcomings in the ways that the objective loss functions include many terms that need to be weighted, the class acceptance regions are not compact, or they need additional hard-mining algorithms. In this study, we propose a simple yet effective method that does not have these limitations. Our proposed method maximizes the margin in both the Euclidean and angular spaces. To the best of our knowledge, our proposed method is the first method that maximizes the margin in both spaces. To accomplish this goal, we train a deep neural network that enforces the samples to gather in the vicinity of the class-specific centers that lie on the boundary of a hypersphere whose center is set to the origin. Each class is represented with a single center, and the distances between the class centers are equivalent. This corresponds to selection of class centers from the vertices of a regular simplex inscribed in a hypersphere. Both the Euclidean distances and angular distances between class centers are equivalent to each other.

Our proposed method has many advantages over other margin maximizing deep neural network classifiers. These advantages can be summarized as follows.

- 1) The proposed method is very straightforward in the sense that one needs to fix only one parameter, the hypersphere radius. Prior research on classification methods employing hyperspherical embeddings has already investigated the selection of this parameter, with [15] offering lower bounds for its determination. Therefore, setting this parameter is extremely easy for the users. For open set recognition, the user has to set two parameters if the background class samples are used for learning.
- 2) The proposed method returns compact and interpretable acceptance regions for each class, thus it is very suitable for open set recognition problems. Other methods utilizing simplex vertices for classification purposes use variants of the softmax loss function and return radial distributions which is not compact. Therefore, their accuracies are not satisfactory for open set recognition.
- 3) The distances between the samples and their corresponding centers are minimized independently of each other, thus the proposed method also works well for imbalanced datasets.
- 4) We investigate scenarios where the utilization of centers from a regular simplex is unfeasible due to the dimensionality of the feature space being less than the number of classes minus one ( $d < C - 1$ ). In such instances, neural collapse does not occur, and the case where  $d < C - 1$  remains largely unexplored with no proposed efficient solutions. Here, we address this issue by introducing a new module that augments the dimensionality of the feature space, as elaborated upon below.

Against all these advantages, there is only one limitation of the proposed method: The dimension of the CNN features must be larger than or equal to the total number of classes minus 1. To overcome this limitation, we introduced two solutions: The first solution uses a dimension augmentation module (DAM) whereas the second solution revises the existing deep neural network architectures.

## II. METHOD

### A. Motivation

In this study, we introduce a simple yet effective deep neural network classifier that maximizes the margin in both Euclidean and angular spaces. To this end, we propose a novel classification loss function that enforces the samples to compactly cluster around the class-specific centers that are selected from the outer boundaries of a hypersphere. The Euclidean distances and angles between the centers are equivalent. Please note that in terms of margin maximization the distances between the class centers are the maximum values for angular distances. In a similar manner, for Euclidean distances, if the class centers are enforced to lie on the boundary of a hypersphere, the distances among the classes again become the best optimal solution we can get. Theoretical proofs of this fact can be

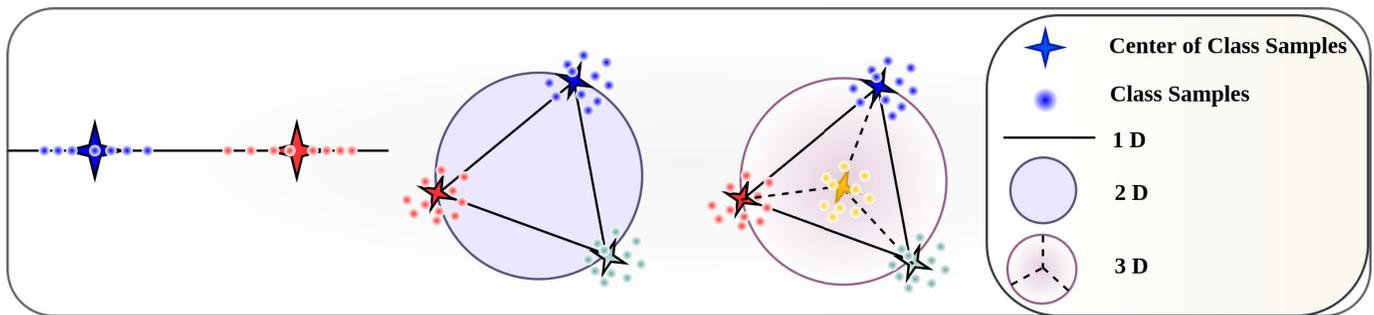


Fig. 1. In the proposed method, class samples are enforced to lie closer to the class-specific centers representing them, and the class centers are located on the boundary of a hypersphere. All the distances between the class centers are equivalent, thus there is no need to tune any margin term. The class centers form the vertices of a regular simplex inscribed in a hypersphere. Therefore, to separate  $C$  different classes, the dimensionality of the feature space must be at least  $C - 1$ . The figure on the left shows the separation of two classes in 1-D space, the middle figure depicts the separation of three classes in 2-D space, and the figure on the right illustrates the separation of four classes in 3-D space. For all cases, the centers are chosen from a regular  $C$ -simplex.

found in both [19] and [26]. Using simplex vertices as class centers is illustrated in Fig. 1. In this figure, the centers representing the classes are denoted by the star symbols whereas the class samples are represented with circles having different colors based on the class memberships. As seen in the figure, all pair-wise distances between the class centers are equivalent, and class centers are located on the boundary of a hypersphere. Moreover, if the hypersphere center is set to the origin, then the angles between the class centers are also same, and the lengths of the centers are equivalent, i.e.,  $\|\mathbf{s}_i\| = u$ , ( $u$  is the length of the center vectors). After learning stage, if the class samples are compactly clustered around the centers representing them, we can classify the data samples based on the Euclidean or angular distances from the class centers. Both distances yield the same results if the hypersphere center is set to the origin.

At this point, the question of whether enforcing data samples to lie around the simplex vertices is appropriate or not comes to mind. In fact, high-dimensional spaces are quite different than the low-dimensional spaces, and there are many studies showing that the data samples lie on the boundary of a hypersphere when the feature dimensionality,  $d$ , is high and the number of samples,  $n$ , is small. For example, Jimenez and Landgrebe [28] theoretically shows that the high-dimensional spaces are mostly empty and data concentrate on the outside of a shell (on the outer boundary of a hypersphere). The authors also show that as the number of dimensions increases, the shell increases its distance from the origin. More precisely, the data samples lie near the outer surface of a growing hypersphere in high-dimensional spaces (therefore setting the hypersphere radius to 1 as in [21] is not suitable for high-dimensional spaces). A more recent study [29] explicitly shows that the data samples lie at the vertices of a regular simplex in high-dimensional spaces. These two studies are not contradictory and they support each other since we can always inscribe a regular simplex in a hypersphere as seen in Fig. 1. In addition to these studies, Kumar et al. [30] and Weber [31] show that the eigenvectors of the Laplacian matrices (the matrices computed by operating on similarity matrices in spectral clustering analysis) form a simplex structure, and they use the vertices of resulting simplex for clustering of data samples. In other words, they prove that

when the data samples are mapped to Laplacian eigenspace, they concentrate on the vertices of a simplex structure. These studies are also complementary to the studies showing that the high-dimensional data samples lie on the boundary of a growing hypersphere. It is because, as proved in [32], normalized cuts (NCuts) [33] clustering algorithm, which is presented as a spectral relaxation of a graph cut problem, maps the data samples onto an infinite-dimensional feature space. Therefore, these data samples naturally concentrate on the vertices of a regular simplex due to the high-dimensionality of the feature space.

There are strong arguments that verify that high-dimensional data samples concentrate on the vertices of regular simplex as discussed above. Do the same arguments hold for the high-dimensional features produced by the deep neural network classifiers? A recent study [22] answers this question and reveals that the samples of different classes cluster around the class centers forming the vertices of a regular simplex (as we proposed in this study) at the last stages of the learning process when the feature dimension is higher than the number of classes. They show that the lengths of the vectors of the class means (after centering by their global mean) converge to the same length and the angles between pair-wise center vectors become equal during the last training stages (it is called terminal phase of the training in the study) of the deep neural networks using linear classifiers. They also demonstrate that the within-class scatter converges to zero indicating that the class-specific samples gather around their corresponding class center. A geometrical analysis of this study is given in [34]. However, both studies are not complete in the sense that they do not consider the cases when the dimension of the feature space is smaller than the number of classes so that it is impossible to fit the class centers to the vertices of a regular simplex. Also, the authors do not propose an efficient method as in our proposed method, instead they use the classical softmax loss function with the linear classifiers, and learn class weights for classification. In contrast, we propose an efficient method that enforces the samples to lie closer to the vertices of a regular simplex directly in this article. We do not learn class weights, instead we use fixed class centers chosen from the vertices of a regular simplex. In addition, we consider the dimension restriction (when the number of classes is larger

than the feature dimension) and introduce solutions to handle this problem as explained below.

### B. Maximizing Margin in Euclidean and Angular Spaces

Here, we propose a novel and simple method that enforces the samples of classes to cluster around the centers chosen from the vertices of a regular simplex. As shown in [22], all class samples cluster around the class centers forming the vertices of a regular simplex when the dimension of the feature space is larger than the number of classes. Therefore, there is no need to use complicated classifier layers, and the same effect can be accomplished by using much simpler classification layers as in our proposed method. In the proposed method, instead of using more complicated linear classifiers and learning class weights for each class, we directly enforce the class samples to compactly cluster around the fixed class centers chosen from the vertices of a regular simplex. All the pair-wise distances between the selected class centers are equivalent.

Let us assume that there are  $C$  classes in our dataset. In this case, we first need to create a  $C$ -simplex (some researchers call it  $C-1$  simplex considering the feature dimension, but we will prefer  $C$ -simplex definition). The vertices of a regular simplex inscribed in a hypersphere with radius 1 can be defined as follows:

$$\mathbf{v}_j = \begin{cases} (C-1)^{-1/2} \mathbf{1}, & j = 1 \\ \kappa \mathbf{1} + \eta \mathbf{e}_{j-1}, & 2 \leq j \leq C \end{cases} \quad (1)$$

where

$$\kappa = -\frac{1 + \sqrt{C}}{(C-1)^{3/2}}, \quad \eta = \sqrt{\frac{C}{C-1}}. \quad (2)$$

Here,  $\mathbf{1}$  is an appropriately sized vector whose elements are all 1,  $\mathbf{e}_j$  is the natural basis vector in which the  $j$ -th entry is 1 and all other entries are 0. Such a  $C$ -simplex is in fact a  $C$ -dimensional polyhedron where the distances between the vertices are equivalent. It must be noted that the distances between the vertices do not change even if the simplex is rotated or translated. But, the dimension of the feature space must be at least  $C-1$  in order to define such a regular  $C$ -simplex. Next, we must define the radius,  $u$ , of the hypersphere. This term is similar to the scaling parameter used in methods such as ArcFace [16] and CosFace [15], that maximize the margin in angular spaces. As the dimension increases, it must be also increased since the studies [28] show that the hypersphere whose outer shells include the data also grows as the dimension is increased. Wang et al. [15] provided a lower bound for the determination of this parameter. Then, we set the class centers that will represent the classes as

$$\mathbf{s}_j = u \mathbf{v}_j, \quad j = 1, \dots, C. \quad (3)$$

The order of selection of centers does not matter since the distances among all centers are equivalent. These distances are the best optimal values that we can get when the cosine distances are used as theoretically proved in [19] and [26]. In a similar manner, when the class centers are restricted to lie on the boundary of a hypersphere, the Euclidean distances

between the classes are again the maximum optimal value one can get. Therefore, there is no need of using a loss term for the interclass separation. Now, let us assume that the deep neural network features of training samples are given in the form  $(\mathbf{f}_i, y_i)$ ,  $i = 1, \dots, n$ ,  $\mathbf{f}_i \in \mathbb{R}^d$ ,  $y_i \in \{j\}$  where  $j = 1, \dots, C$ . Here,  $C$  is the total number of known classes, and we assume that the feature dimension  $d$  is larger than or equal to  $C-1$ , i.e.,  $d \geq C-1$ . Under these assumptions, the loss function of the proposed method can be written as

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{f}_i - \mathbf{s}_{y_i}\|^2. \quad (4)$$

The loss function includes a single term that targets to minimize the within-class variations by minimizing the distances between the samples and their corresponding class centers, which are set to the vertices of a regular simplex. There is no need for another loss term for the between-class separation since the selected centers have the maximum possible Euclidean and angular distances among them. As a result, there is no hyperparameter that must be fixed, and the proposed method is extremely easy for the users. Moreover, the data samples compactly cluster around their class centers, therefore the proposed method results in compact acceptance regions for classes, which is crucial for the success in the context of the open set recognition. It should be noted that our proposed method is quite different than the methods using vertices of a regular simplex as in our proposed method. It is because, all these methods use variants of the softmax loss function that typically require setting margin parameters for the interclass separation. Furthermore, these methods return noncompact radial distributions (see [24, Figs. 2, 4, 5, and 8] and [26, Fig. 2]). Therefore, their performance will not be satisfactory for open-set recognition problems. We call our proposed method as deep simplex classifier (DSC).

The running time of the proposed method will be more efficient compared to the methods using the softmax loss function and its variants, Arcface [16], Cosface [15], and regular polytope networks [24]. Because, these methods require to apply exponential function to each logit,  $(\mathbf{w}_c^T \mathbf{f}_i + b_c)$ , followed by a normalization by dividing with the sum of all these exponentials as seen in the softmax loss function given below

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{\mathbf{w}_{y_i}^T \mathbf{f}_i + b_{y_i}}}{\sum_{j=1}^C e^{\mathbf{w}_j^T \mathbf{f}_i + b_j}}. \quad (5)$$

On the other hand, we just need to extract the CNN features of the test samples during training and testing stages. Then, these features are compared to precomputed centers by using the Euclidean distances. Therefore, the proposed method is more efficient in terms of computational complexity. However, this does not affect testing times much since the most of the time is spent on convolutional layers of the deep neural network classifier during the testing stage.

### C. Including Background Class for Open Set Recognition

In open-set recognition scenarios, the training of classifiers commences by exclusively utilizing samples of known classes. Subsequently, both known and unknown class samples are

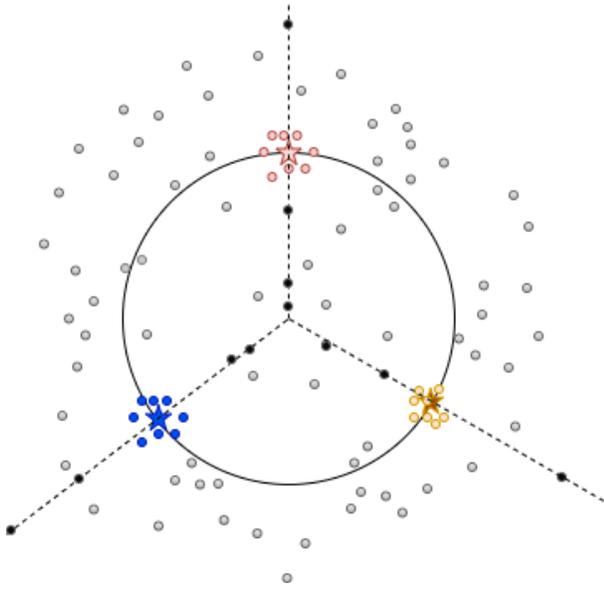


Fig. 2. If the cosine distances are used for measuring dissimilarities between the samples and centers as in SOTA methods, all the samples closer to the dashed lines will be very close to the known class centers, therefore they will not be rejected. As a result, the samples coming from the unknown classes denoted by the black circles (the ones closer to the origin and the ones closer to the end of the dashed lines outside the hypersphere), will be assigned to the known classes even though they lie very far from the known class regions. Therefore, using cosine distances and radially distributed CNN feature embeddings are not suitable for open-set recognition settings.

employed in testing the resulting classifiers. The primary objective in this task is to ensure accurate classification of known class samples, while also detecting and rejecting samples from unknown classes [35]. Prior methods for open-set recognition relied solely on the use of the known class samples during training. However, recent investigations [36], [37], [38], [39] have shown that augmenting the training dataset with the background dataset with samples from classes that differ from the known classes can greatly enhance accuracy. Let us represent the deep neural network features of the background samples by  $\mathbf{f}_k \in \mathbb{R}^d$ ,  $k = 1, \dots, K$ . In order to incorporate the background samples, we add an additional loss term that pushes the background samples away from the fixed known class centers as follows:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{f}_i - \mathbf{s}_{y_i}\|^2 + \lambda \sum_{i=1}^n \sum_{k=1}^K \max\left(0, m + \|\mathbf{f}_i - \mathbf{s}_{y_i}\|^2 - \|\mathbf{f}_k - \mathbf{s}_{y_i}\|^2\right) \quad (6)$$

where  $m$  is the selected threshold, and  $\lambda$  is the weighting term. The second loss term imposes a constraint on the distances between known class samples and their respective class centers, mandating that they be smaller than the distances between background class samples and the known class centers by a minimum margin of  $m$ . In contrast to our first proposed loss function, this loss function includes two terms that must be set by the users. But, this is necessary only if we use the background class samples.

Our proposed method returns compact acceptance regions as illustrated in Fig. 2. The background samples are pushed away from the known class centers. Therefore, the unknown class samples can be easily rejected based on the Euclidean distances from the test samples to the known class centers. Here, using Euclidean distances is important since it is impossible to reject background samples far from the centers if these samples have the similar orientations as in selected centers. This is the main reason why the state-of-the-art classifiers such as ArcFace [16] and CosFace [15], fail for open-set recognition problems. Compared to other related methods using simplex vertices and variants of the softmax loss function, our proposed method is more suitable for open-set recognition tasks since the proposed methodology returns compact class acceptance regions. Please note that almost all methods using the fixed simplex vertices and the variants of softmax loss function return noncompact radial distributions. Since the class acceptance regions are not compact as in our proposed method, their performances are not satisfactory for open-set recognition problems (e.g., our proposed method achieve much better open-set accuracies compared to [25] on open-set recognition).

#### D. Dimensionality Restriction and Solution Techniques

The major limitation of the proposed method is the restriction that the dimension of the feature space must be larger than or equal to  $C - 1$ , i.e.,  $d \geq C - 1$ . A similar restriction exists in [21], and their proposed method requires  $d \geq C$  since they choose the class centers as the standard basis vectors of  $C$ -dimensional space as opposed to our proposed method that selects the centers from the vertices of a regular simplex. The typical feature dimension size returned by the classical deep neural network classifiers is 2048 or 4096. In this case, the number of classes in our training set cannot exceed 2049 or 4097. However, the number of classes can be larger than these values for some classification tasks, and we cannot use the proposed method in such cases.

There are several procedures to solve this problem: As a first solution, we can use a method similar to [40] that returns more centers where the distances between centers are approximately equivalent. In this case, the number of centers is increased to  $2d + 4$  for  $d$ -dimensional feature spaces. However, this procedure may not solve the problem if the number of classes is still larger than  $2d + 4$ . For more complete solutions, we can revise the existing CNN architectures so that they yield the desired feature size or integrate a module that increases the feature dimension. This is illustrated in Fig. 3. These two procedures are explained below.

1) *Dimension Augmentation Module*: To solve the dimension restriction, we first introduce a plug and play module called the DAM that increases the feature dimension size to any desired value. The module is visualized in Fig. 4, and it includes several fully connected layers supported with nonlinear activation functions. The first fully connected layer maps the  $d$ -dimensional feature space onto a higher  $(C - 1)/2$  (half of the desired feature dimension size) dimensional space. Then, we apply parametric rectified linear unit (PReLU) activation functions [41] followed by the second fully connected

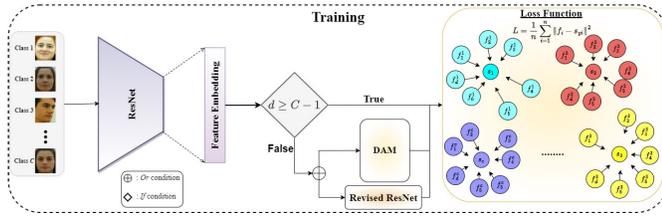


Fig. 3. Illustration of the proposed method: we use well-known architectures (such as ResNet-19 and ResNet-101) as backbones and we only change the classification loss layer. If the dimension of the CNN feature space is smaller than  $C - 1$ , we increase the dimension to desired size by using DAM module or revising the network architecture, and then apply the proposed loss function.

layer. The second fully connected layer increases the dimension to desired feature space size,  $(C - 1)$ . Then, we apply another PReLU function followed by the last fully connected layer. It should be noted that, following the ReLU (or PReLU) operation, the majority of values may become positive, despite their corresponding centers having negative values. Therefore, the last layer in the module includes a fully connected layer that maps  $(C - 1)$  dimensional feature space back to  $(C - 1)$  dimensional feature space so that the sample features may have negative values. The proposed module increases the dimension in two steps as explained above. The dimension can be directly increased from  $d$  to  $C - 1$  in the first fully connected layer. In a similar manner, we can increase the dimension in more than two steps if desired.<sup>1</sup> The main idea of the proposed DAM is similar to kernel mapping idea used in kernel methods [42], [43] in the spirit with the exception that we explicitly map the data to higher dimensional feature space as in [44] and [45]. It should be noted that Do et al. [21] proposed to use a fully connected layer alone for increasing the dimensionality of the feature space. However, a fully connected layer just uses the linear combination of existing features and the resulting space has the dimensionality which is lower than or equal to the original feature space dimension. Therefore, one has to use activation functions to introduce nonlinearity and increase the dimension as in our proposed module.

2) *Revising Network Architecture:* We can also solve the dimension problem by slightly changing the existing CNN architectures instead of using our proposed plug and play DAM. To this end, we can avoid the fully connected layers that are used for dimension reduction in the last layers of deep CNNs. For example, in the ResNet architectures we used for face recognition in our experiments, the dimension of the feature space is 25088 just before the fully connected layers, and it is reduced to 512 after fully connected layers. Instead of reducing the dimension to 512, we can reduce it to values that solve the current problem. If the number of classes is much larger than 25088, we can use more filters at the last layers to increase this number. In this study, we used 25088 dimensional feature space and reduced the feature size to 12500 by using a fully connected linear layer (without PReLU) for training the large-scale dataset sampled from MS1MV3 dataset [46] without any need for dimension

<sup>1</sup>Our shared software allows to select any desired number of steps for increasing dimensionality.

increase for face verification experiments conducted in this study.

### III. EXPERIMENTS

#### A. Illustrations and Ablation Studies

Here, we first conducted some experiments to visualize the embedding spaces returned by the various loss functions using the vertices of the regular simplex. To this end, we utilized a small deep neural network that yields 2D CNN features. As training data, we selected three classes from the Cifar-10 dataset since the maximum number of classes is bounded by 3 in 2-D spaces in the proposed method. We would like to point out that we can use different loss functions in addition to our default loss function given in (4) once we determine the vertices of the simplex that will represent the classes. For this experiment, we used two other loss functions: The first one is the hinge loss that minimizes the distances between the samples and their corresponding class center if the distance is larger than a selected threshold

$$\mathcal{L}_{\text{hinge}} = \frac{1}{n} \sum_{i=1}^n \max(0, \|\mathbf{f}_i - \mathbf{s}_{y_i}\|^2 - m). \quad (7)$$

This loss function does not minimize the distances between the samples and their corresponding centers if the distances are already smaller than the selected threshold,  $m$ . This way class-specific samples are compactly clustered in a hypersphere with radius,  $m$ . For the second loss function, we used the variant of the softmax loss function where the weights are fixed to the simplex vertices as in

$$\mathcal{L}_{\text{softmax}} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{\mathbf{s}_{y_i}^\top \mathbf{f}_i + b_{y_i}}}{\sum_{j=1}^C e^{\mathbf{s}_j^\top \mathbf{f}_i + b_j}}. \quad (8)$$

For the softmax loss, we fix the classifier weights to the predefined class centers and we only update features of the samples by using backpropagation. We set the hypersphere radius to,  $u = 5$ , since this is a simple dataset.

The embeddings returned by the deep neural networks using different loss functions are plotted in Fig. 5. The first figure on the left is obtained by our default loss function that does not need any parameter selection. All data samples are compactly clustered around their class means as expected. The second loss function using the hinge loss returns spherical distributions based on the selected margin,  $m$ , and the classes are still separable by a margin. In contrast, when the softmax is used with the simplex vertices, the data samples are very close and they overlap since there is no margin among the classes. Therefore, our default loss function seems to be the best choice among all tested variants since it does not need fixing any parameter and returns compact class regions.

We also conducted tests on imbalanced datasets. In our proposed method, the distances between the samples and their corresponding class centers are minimized independently of each other. Therefore, we expect that the proposed method will be more robust against to imbalanced datasets. To verify this, we conducted experiments on the same three classes used before. We used the same deep neural network classifier yielding 2-D feature spaces for this experiment. The number

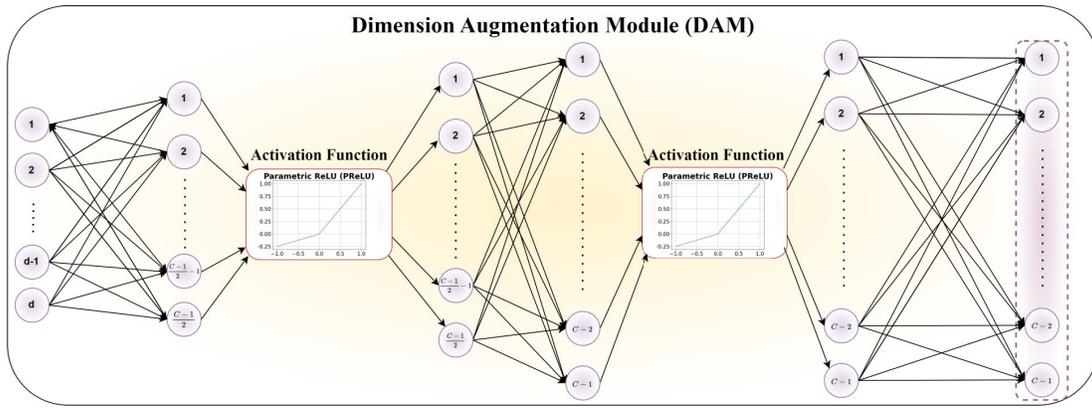


Fig. 4. Plug and play module that will be used for increasing feature dimension. It maps  $d$ -dimensional feature vectors onto a much higher  $(C - 1)$ -dimensional space. The DAM module was specifically designed to allow users to choose any desired number of steps for increasing dimensionality. It is possible to increase the dimension in a single step or gradually increase it using multiple steps. This figure depicts the case when two steps are used for increasing the dimension.

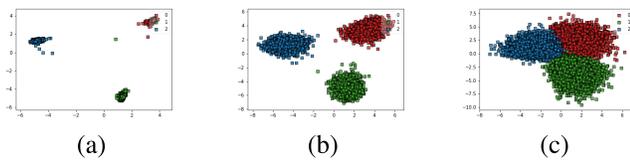


Fig. 5. Outputs of the deep neural network classifiers trained by using different simplex loss functions: (a) 2D CNN features returned by the proposed method trained with the default loss function given in (4), (b) 2D CNN features returned by the proposed method trained with the hinge loss, and (c) 2D CNN features returned by the proposed method trained with the softmax loss function.

of training samples per class is 5000 for the selected classes and we first trained the proposed method by using the same amount of samples for each class. Then, we extracted the CNN features of test samples. After that, we decreased the number of samples of the blue colored classes to 500 (which is 10% of the original size) to create an imbalanced training set. We trained another network by using this imbalanced dataset and extracted the CNN features of the testing samples. The visualization of the extracted features is shown in Fig. 6, where the first row shows the CNN features of the training and test samples extracted by using the network trained with the balanced dataset and the second row shows the extracted features by using the network trained with imbalanced dataset. As seen in the figure, the extracted features of the test samples obtained by using the imbalanced dataset are similar to the ones obtained by using the balanced dataset. This verifies that the proposed method is more robust against to imbalanced datasets as expected.

### B. Open-Set Recognition Experiments

The datasets are split into *known* and *unknown* classes in open-set recognition settings. By following the standard settings, we split the datasets into known and unknown classes five times, trained our classifiers and computed the accuracies. The final accuracies are obtained by averaging the accuracies obtained in each trial. The details of the each dataset are given below:

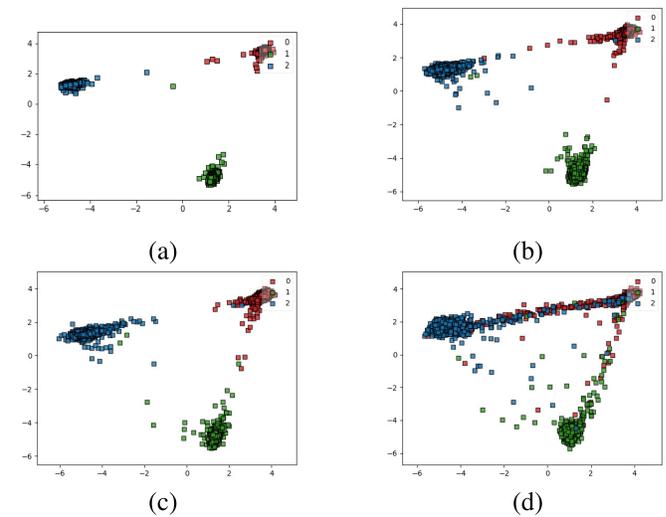


Fig. 6. Learned feature representations of image samples. (a) Embeddings of the training samples returned by the proposed method trained with the balanced dataset. (b) Embeddings of the test samples returned by the proposed method trained with the balanced dataset. (c) Embeddings of the training samples returned by the proposed method trained with the imbalanced dataset. (d) Embeddings of the test samples returned by the proposed method trained with the imbalanced dataset.

1) **Datasets: Mnist, Cifar-10, SVHN:** These datasets are split randomly into six known and four unknown classes by using the common testing setting. The 80 Million Tiny Images dataset [47] is used as the background class.

**Cifar + 10, Cifar + 50:** For Cifar +  $N$  experiments, four randomly chosen classes from Cifar-10 dataset are used for training, and  $N$  nonoverlapping classes chosen from the Cifar-100 dataset are used as unknown classes as in [37], [48], [49], and [50]. The 80 Million Tiny Images dataset [47] is used as the background class.

**TinyImageNet:** For TinyImageNet [51] experiments, 20 classes are randomly chosen as known classes and 180 classes as unknown classes by following the standard setting. The 80 Million Tiny Images dataset [47] is used as the background class.

TABLE I

AUC SCORES (%) OF OPEN SET RECOGNITION METHODS ON TESTED DATASETS (*n.r.* STANDS FOR NOT REPORTED). THE BEST ACCURACIES ARE SHOWN WITH RED FONTS WHEREAS STATISTICALLY SIMILAR PERFORMANCES ARE SHOWN WITH BLUE FONTS. THE METHODS THAT STATISTICALLY PERFORM POORLY ARE SHOWN WITH STANDARD BLACK FONT. THE STANDARD DEVIATION OF OBJECTTOSPHERE METHOD IS ASSUMED AS 1 FOR THE CIFAR-10 DATASET

| Methods                    | Mnist              | Cifar-10                  | SVHN               | Cifar+10           | Cifar+50           | TinyImageNet       |
|----------------------------|--------------------|---------------------------|--------------------|--------------------|--------------------|--------------------|
| DSC (Ours)                 | <b>99.6 ± 0.1</b>  | <b>93.8 ± 0.3</b>         | 95.3 ± 0.8         | <b>99.1 ± 0.2</b>  | <b>98.4 ± 0.3</b>  | <b>82.5 ± 1.8</b>  |
| Softmax                    | 97.8 ± 0.2         | 67.7 ± 3.2                | 88.6 ± 0.6         | 81.6 ± <i>n.r.</i> | 80.5 ± <i>n.r.</i> | 57.7 ± <i>n.r.</i> |
| OpenMax                    | 98.1 ± 0.2         | 69.5 ± 3.2                | 89.4 ± 0.8         | 81.7 ± <i>n.r.</i> | 79.6 ± <i>n.r.</i> | 57.6 ± <i>n.r.</i> |
| G-OpenMax                  | 98.4 ± 0.1         | 67.5 ± 3.5                | 89.6 ± 0.6         | 82.7 ± <i>n.r.</i> | 81.9 ± <i>n.r.</i> | 58.0 ± <i>n.r.</i> |
| C2AE                       | 98.9 ± 0.2         | 89.5 ± 0.9                | 92.2 ± 0.9         | 95.5 ± 0.6         | 93.7 ± 0.4         | 74.8 ± 0.5         |
| CAC                        | <b>99.1 ± 0.5</b>  | 80.1 ± 3.0                | 94.1 ± 0.7         | 87.7 ± 1.2         | 87.0 ± 0.0         | 76.0 ± 1.5         |
| CPN                        | 99.0 ± 0.2         | 82.8 ± 2.1                | 92.6 ± 0.6         | 88.1 ± <i>n.r.</i> | 87.9 ± <i>n.r.</i> | 63.9 ± <i>n.r.</i> |
| OSRCI                      | 98.8 ± 0.1         | 69.9 ± 2.9                | 91.0 ± 0.6         | 83.8 ± <i>n.r.</i> | 82.7 ± —           | 58.6 ± <i>n.r.</i> |
| CROSR                      | 99.1 ± <i>n.r.</i> | 88.3 ± <i>n.r.</i>        | 89.9 ± <i>n.r.</i> | 91.2 ± <i>n.r.</i> | 90.5 ± <i>n.r.</i> | 58.9 ± <i>n.r.</i> |
| RPL                        | 98.9 ± 0.1         | 82.7 ± 1.4                | 93.4 ± 0.5         | 84.2 ± 1.0         | 83.2 ± 0.7         | 68.8 ± 1.4         |
| GDFRs                      | <i>n.r.</i>        | 83.1 ± 3.9                | 95.5 ± 1.8         | 92.8 ± 0.2         | 92.6 ± 0.0         | 64.7 ± 1.2         |
| Objecttosphere             | <i>n.r.</i>        | <b>94.2 ± <i>n.r.</i></b> | 91.4 ± <i>n.r.</i> | 94.5 ± <i>n.r.</i> | 94.4 ± <i>n.r.</i> | 75.5 ± <i>n.r.</i> |
| Maximally Separated Matrix | <i>n.r.</i>        | <b>95.3 ± 1.5</b>         | <b>97.6 ± 0.5</b>  | 98.3 ± 0.4         | 96.7 ± 0.3         | <i>n.r.</i>        |

TABLE II

CLOSED-SET ACCURACIES (%) OF OPEN-SET RECOGNITION METHODS ON TESTED DATASETS

| Methods    | Mnist             | Cifar-10          | SVHN              | Cifar+10          | Cifar+50          | TinyImageNet      |
|------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| DSC (Ours) | <b>99.8 ± 0.1</b> | <b>96.1 ± 1.4</b> | <b>96.5 ± 0.3</b> | <b>97.6 ± 0.5</b> | <b>97.9 ± 0.5</b> | <b>83.3 ± 2.2</b> |
| Softmax    | 99.5 ± 0.2        | 80.1 ± 3.2        | 94.7 ± 0.6        | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |
| OpenMax    | 99.5 ± 0.2        | 80.1 ± 3.2        | 94.7 ± 0.6        | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |
| G-OpenMax  | 99.6 ± 0.1        | 81.6 ± 3.5        | 94.8 ± 0.8        | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |
| CPN        | <b>99.7 ± 0.1</b> | 92.9 ± 1.2        | <b>96.7 ± 0.4</b> | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |
| OSRCI      | 99.6 ± 0.1        | 82.1 ± 2.9        | 95.1 ± 0.6        | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |
| CROSR      | 99.2 ± 0.1        | 93.0 ± 2.5        | 94.5 ± 0.5        | <i>n.r.</i>       | <i>n.r.</i>       | <i>n.r.</i>       |

2) *Results*: The main goal of open set recognition is to detect and reject the samples that come from the novel classes. The performance of open set recognition is often measured using area under the ROC curve (AUC) scores. Additionally, the closed set accuracy is also reported to evaluate classification performance on known data by disregarding unknown samples, as demonstrated in previous works such as [48] and [52]. We trained our proposed method by using the loss function given at (6), which is especially designed for the open-set recognition settings. Our proposed method, DSC, is compared against to other state-of-the-art open set recognition methods including maximally separating matrix method of [25] using simplex vertices, C2AE [53], Softmax, OpenMax [35], OSRCI [52], CAC [37], RPL [50], CROSR [49], ROSR [49], generative-discriminative feature representations (GDFRs) [54], and Objecttosphere [55] methods. Except for the TinyImageNet dataset, we employed the identical network backbone as in [52] for all datasets. To achieve higher accuracies for the TinyImageNet dataset, we utilized a deeper Resnet-50 architecture. The hypersphere radius is set to  $u = 64$  as in ArcFace method. The proposed methods demonstrated accuracies that are directly comparable to those reported in [52] for most of the tested datasets, as the network weights were randomly initialized during the training stage. AUC scores were summarized in Table I, which showed that the proposed method achieved the best accuracies across all datasets except for the Cifar-10 and SVHN. We also conducted statistical significance tests to assess the variances in accuracy between the proposed method and its competitors listed in Table I. This examination employs a null hypothesis statistical

test utilizing the t-distribution. If the obtained significance falls below the predefined significance threshold (set at 0.05), we reject the null hypothesis, indicating that there is a statistically significant difference in performance between the two methods. The highest accuracy scores are highlighted in bold red text, while methods exhibiting statistically similar performance are indicated in bold blue. Results for methods that perform poorly from a statistical perspective are presented in standard black font. Notably, there were significant performance differences observed for the Mnist, Cifar + 10, Cifar + 50, and TinyImageNet datasets. Our proposed method achieves significantly better accuracies compared to other tested methods. For the Cifar-10 dataset, our proposed method performs statistically similar to the best performing method whereas all tested methods perform worse compared to the best performing method for the SVHN dataset. Closed-set accuracies for open-set recognition methods were reported in Table II, where the proposed method achieved the best accuracies among the tested methods, with the exception of the SVHN dataset. Obtaining the best accuracies in terms of AUC scores and closed-set accuracies indicates that our proposed method can easily identify and reject the novel class samples and correctly classify the known class samples as expected.

### C. Closed-Set Recognition Experiments

1) *Experiments on Moderate Sized Datasets*: Here, we conducted closed-set recognition experiments on moderate sized datasets. Our proposed method did not need DAM since the feature dimension is much larger than the number of classes

TABLE III

CLASSIFICATION ACCURACIES (%) ON MODERATE SIZED DATASETS

| Methods     | Mnist       | Cifar-10    | Cifar-100   |
|-------------|-------------|-------------|-------------|
| DSC (Ours)  | <b>99.7</b> | <b>95.9</b> | <b>79.5</b> |
| Softmax     | 99.4        | 94.4        | 75.3        |
| Center Loss | <b>99.7</b> | 94.2        | 76.1        |
| ArcFace     | <b>99.7</b> | 94.8        | 75.7        |
| CosFace     | <b>99.7</b> | 95.0        | 75.8        |
| SphereFace  | <b>99.7</b> | 94.7        | 75.1        |

TABLE IV

CLASSIFICATION ACCURACIES (%) ON IMAGENET DATASET

| Methods                     | top 1       | top 5       |
|-----------------------------|-------------|-------------|
| DSC (Ours)                  | <b>80.7</b> | <b>95.8</b> |
| Softmax                     | 79.3        | 94.5        |
| Large-Margin Softmax        | 80.2        | 95.3        |
| Maximally Separating Matrix | 78.5        | 95.1        |

in the training set for these experiments. We compared our results to the methods that maximize the margin in Euclidean or angular spaces. We implemented the compared methods by using provided source codes by their authors, and we used the ResNet-18 architecture [56] as backbone for all tested methods. Therefore, our results are directly comparable. We set the hypersphere radius to  $u = 64$  as before.

Classification accuracies are given in Table III. For the Mnist dataset, majority of the tested methods yield the same accuracy, but our proposed DSC method outperforms all tested methods on the Cifar-10 and Cifar-100 datasets. The performance difference is significant, especially on the Cifar-100 dataset. These results verify the superiority of the margin maximization in both Euclidean and angular spaces. Achieving the best accuracies is encouraging, because our proposed method is very simple and does not need any parameter tuning, yet it outperforms more complex methods.

We also conducted tests on ImageNet dataset [57]. We used a deeper architecture ResNet-101 since this dataset is a large-scale dataset including 1000 classes. The results are given in Table IV. We compared our results to the method using the softmax loss function, large-margin softmax loss function [13], and a very related method, maximally separating matrix method of [25], using simplex vertices as fixed class centers as in our proposed method. As seen in the table, our proposed method outperforms all methods and achieve the best accuracies for both top-1 and top-5 accuracies.

2) *Experiments on Large-Scale Datasets:* We also tested the proposed method in the classification setting where the number of classes is much larger than the feature dimensionality. As stated earlier, the dimension restriction occurs in such settings. To overcome this, we utilized DAM and revised network architecture as explained in Section II-D.  $DSC_{DAM}$  represents the classifier using DAM, and  $DSC_{RNA}$  represents the classifier using the revised network architecture. We tested the proposed methods on face verification and recognition problems.

To conduct every face verification test, the standard procedure is followed by employing the same network that has been trained on a large-scale face dataset. The network that

TABLE V

VERIFICATION RATES (%) ON DIFFERENT DATASETS

| Method               | LFW         | CALFW       | CPLFW       | CFP         | AgeDB       |
|----------------------|-------------|-------------|-------------|-------------|-------------|
| $DSC_{DAM}$          | <b>99.8</b> | <b>95.6</b> | 91.5        | <b>98.2</b> | <b>97.9</b> |
| $DSC_{RNA}$          | 99.6        | 94.2        | 87.7        | 94.9        | 96.2        |
| VGGFace2             | 99.4        | 90.6        | 84.0        | ---         | ---         |
| Center Loss          | 99.3        | 85.5        | 77.5        | ---         | ---         |
| ArcFace (ResNet-101) | <b>99.8</b> | 95.5        | <b>92.1</b> | <b>98.2</b> | ---         |
| CosFace              | 99.7        | 93.3        | <b>92.1</b> | ---         | 97.7        |
| SphereFace           | 99.4        | 93.3        | <b>92.1</b> | 94.4        | 97.7        |

is utilized for this purpose has been trained on the MS1MV3 dataset [46], which is a refined variant of the MS-Celeb-1M dataset [58], and incorporates the proposed loss function. The MS1MV3 dataset includes approximately 91K individuals. We have used the first 12K individuals having the most samples per class in our experiments (using more classes yielded memory problems with the GPUs we have used for the experiments). The ResNet-101 architecture is used as backbone, and this backbone yields CNN features whose dimension is  $d = 512$ . Therefore, the number of classes is much larger than the feature dimension,  $d = 512$ . For both proposed classifiers, we mapped the feature dimension to 12 500 rather than  $C - 1 = 11999$ . For  $DSC_{DAM}$ , we used only one layer with PReLU activation functions, which required to estimate additional  $512 \times 12500 + (12500)^2 - 512 \times 12000$  weight parameters for the utilized network. We also applied batch normalization after PReLU layer. For  $DSC_{RNA}$ , we first removed the original fully connected layer that maps the 25 088 dimensional CNN features to 512 dimensional space. Then, we added a fully connected layer (without PReLU) that maps 25 088 dimensional CNN features to 12 500 dimensional feature space. Therefore, this revision is required to estimation of additional  $25088 \times 12500 - [25088 \times 512 + 512 \times 12000]$  weights. The hypersphere radius is set to 2000. Training the network,  $DSC_{RNA}$ , using the revised network architecture took 11 444 s (3.178 h) to finish one epoch whereas the network using DAM,  $DSC_{DAM}$ , completed an epoch in 11 137 s (3.093 h). In contrast, a network that uses the 512-D CNN feature space with the classical softmax loss function finishes an epoch in 8962 s (2.489 h). Therefore,  $DSC_{RNA}$  is approximately 1.28 times slower and  $DSC_{DAM}$  is 1.24 times slower compared to a classical network that uses the softmax loss function. Once the networks are trained, we used the resulting architectures to extract deep CNN features of the face images coming from the test datasets.

As test datasets, we used labeled faces in the wild (LFW) [59], celebrities in frontal-profile dataset (CFP-FP) [60], cross-age LFW (CALFW) [61], AgeDB [60], and cross-pose LFW (CPLFW) [62]. For evaluation, the standard protocol of unrestricted with labeled outside data [59] is used and the accuracies are obtained by using 6000 pair testing images on LFW, CALFW, AgeDB, and CPLFW. For CFP-FP dataset, the accuracies are obtained by using 7000 pairs of testing images by following the standard testing setting. Table V reports the accuracies. As seen in the results, the proposed method using DAM achieves the best accuracies for four datasets among all tested five datasets.  $DSC_{RNA}$  method also obtains competitive

TABLE VI  
IDENTIFICATION ACCURACIES (%) ON THE IJB-B AND IJB-C BENCHMARKS

| Method                 | IJB-B Dataset                                |           |                             |        |         |
|------------------------|--|-----------|-----------------------------|--------|---------|
|                        | True Positive Identification Rate (TPIR) (%) |           | Rank- <i>N</i> Accuracy (%) |        |         |
|                        | @FPIR=0.01                                   | @FPIR=0.1 | Rank-1                      | Rank-5 | Rank-10 |
| DSC <sub>DAM</sub>     | 88.86  | 93.80     | 95.80                       | 97.10  | 97.50   |
| DSC <sub>RNA</sub>     | 85.46  | 91.63     | 93.92                       | 96.10  | 96.90   |
| ArcFace                | 91.61  | 95.18     | 96.06                       | 97.23  | 97.71   |
| VGGFace2 (Softmax)     | 74.30  | 86.30     | 90.20                       | 94.60  | 95.90   |
| FPN [64]               | --   | --        | 91.10                       | 95.30  | 96.50   |
| PRN [65]               | 81.40  | 90.70     | 93.50                       | 96.50  | 97.50   |
| AFRN (model B) [66]    | 80.30  | 88.50     | 92.30                       | 96.20  | 97.40   |
| AFRN (model C) [66]    | 86.40  | 93.70     | 97.30                       | 97.60  | 97.70   |
|                        | IJB-C Dataset                                |           |                             |        |         |
|                        | True Positive Identification Rate (TPIR) (%) |           | Rank- <i>N</i> Accuracy (%) |        |         |
|                        | @FPIR=0.01                                   | @FPIR=0.1 | Rank-1                      | Rank-5 | Rank-10 |
| DSC <sub>DAM</sub>     | 91.80  | 94.71     | 95.80                       | 97.80  | 97.90   |
| DSC <sub>RNA</sub>     | 88.95  | 92.60     | 94.62                       | 96.17  | 96.95   |
| ArcFace                | 89.91  | 93.50     | 95.20                       | 96.96  | 97.63   |
| VGGFace2 (Softmax)     | 76.30  | 86.50     | 91.40                       | 95.10  | 96.10   |
| Center Loss            | 77.20  | 85.30     | 90.70                       | 94.10  | 95.20   |
| AFRN (model B) [66]    | 85.30  | 90.50     | 93.10                       | 95.60  | 96.40   |
| AFRN (model C) [66]    | 88.40  | 93.10     | 95.70                       | 97.60  | 97.70   |
| Discriminative CH [67] | --   | --        | 93.50                       | 96.60  | --      |

accuracies, but its accuracies are lower than DSC<sub>DAM</sub>. These results verify that the proposed techniques for solving dimension problem successfully resolve this problem. However, the weight parameters of the networks are greatly increased.

We also conducted identification (recognition) tests on the challenging IJB-B and IJB-C datasets [63]. These datasets present considerable difficulties due to their inclusion of full pose variations and wide-ranging imaging conditions. The IJB-B dataset is characterized by its template-based approach, encompassing 1845 subjects with 11 754 images and 55 025 frames from 7011 videos. Images and videos were sourced from the web, showcasing significant variations in pose, illumination, and image quality, among other factors. The IJB-C dataset serves as an extension of the IJB-B dataset, featuring 3531 unique subjects in unconstrained environments. This mixed media set-based dataset comprises 31 334 still images, averaging approximately six images per subject, and 117 542 video frames, averaging about 33 frames per video. Each subject is represented by a template consisting of multiple images, rendering the set-based face recognition approach ideal for subject identification. These datasets are widely recognized as benchmark datasets for evaluating state-of-the-art face recognition methodologies.

For reporting accuracies, we follow the standard benchmark procedure for IJB-B and IJB-C to evaluate the proposed methods on “search” protocol for 1:N face identification. Here, the Rank-*N* classification accuracies are reported for identification, and the classification rate is the percentage of probe searches, which correctly finds the probe’s gallery mate in the gallery set within top *N* rank-ordered results. In addition, we also report the true positive identification rate (TPIR) accuracies obtained for different false positive identification rate (FPIR) values. The results are given in Table VI, where the red and blue fonts successively denote the best and the second best accuracies. As seen in the table, the proposed method using DAM module achieves the best accuracies in all metrics on the IJB-C dataset, whereas it obtains the second

best accuracies in terms of TPIR accuracies on the IJB-B dataset.

#### IV. SUMMARY AND CONCLUSION

This article proposed a neural network classifier that aims to maximize the margin in both the Euclidean and angular spaces. Specifically, the method generates embeddings such that class-specific samples cluster around the class centers chosen from the vertices of a regular simplex. The technique is particularly straightforward as it requires to fix a simple parameter for classical closed set recognition settings. Despite its simplicity, the proposed method achieves state-of-the-art accuracies on open-set recognition problems by rejecting samples of unknown classes based on their distances from the class-specific centers. Additionally, the proposed method outperforms other current classification methods on closed set recognition settings, particularly with moderate-sized datasets. Nonetheless, the method exhibits a limitation in learning large-scale datasets, which can be addressed by introducing a DAM and revising existing deep neural network architectures. The proposed classifier using the DAM achieves state-of-the-art accuracies on face verification problems, but the weight parameters of the deep neural network classifier greatly increase. In summary, the proposed method is an ideal choice for open set recognition and classical classification problems, particularly when the feature dimension is larger than the number of classes, and the proposed classifier is straightforward to use with a simple hyperparameter that requires setting. For large-scale datasets with many classes, the proposed method using DAM still yields good accuracies, but it increases the complexity of the deep neural network architectures.

#### APPENDIX

##### IMPLEMENTATION DETAILS

The learning rate is set to 0.1 for the proposed DSC in open-set recognition experiments. We set

TABLE VII  
CLASSIFICATION ACCURACIES (%) FOR DIFFERENT  $u$   
VALUES ON CIFAR-100 DATASET

| $u$ values | Accuracies (%) |
|------------|----------------|
| $u = 32$   | 76.2           |
| $u = 64$   | 79.5           |
| $u = 100$  | 79.4           |
| $u = 150$  | 79.9           |
| $u = 200$  | 79.0           |

$\lambda = (1/2 \times \text{batch\_size}^2)$ , and  $m = u/2$ , where  $u$  is the hypersphere radius.

We do not need  $\lambda$  and  $m$  parameters for closed-set recognition. For closed-set recognition experiments, we used the ResNet-18 architecture as backbone for moderate sized datasets, and the ResNet-101 architecture is used for large-scale face recognition dataset. For updating network weights, we used Adam optimization strategy for large-scale face recognition whereas stochastic gradient descent (SGD) is used for moderate size datasets. The learning rate is set to  $10^{-3}$  for face recognition and to 0.5 for moderate-sized datasets.

Regarding the scale parameter  $u$ , we have conducted experiments by selecting different values. Experiments verify that the selection of  $u$  is not very important as long as it is not fixed to small values such as 1. Theoretically, the data samples lie on the surface of a growing hypersphere as the dimension increases. For smaller dimensions, we can choose smaller values of  $u$  as we did for illustrations experiments (we fixed  $u$  to 5 for 2-D inputs). But, for larger dimensions, we need higher values. Also, after some value, increasing  $u$  value does not change the results much. The accuracies that are obtained for Cifar-100 dataset for various  $u$  values are given in Table VII.

The threshold,  $m$ , parameter is only used for open set recognition problems. Moreover, we did not have any trouble for fixing it since our centers are fixed to certain positions. We already know the distances between the class centers chosen as simplex vertices. All distances are equal, we simply checked the largest intraclass distances within classes and determined a margin based on this. Setting margin term to half of the hypersphere radius worked well for all cases. For all experiments except for face verifications ones, we did not fine-tune our classification network from a pretrained network and started the network weights from scratch by initializing with random weights which is the common practice used for initializing network weights. To train on the large-scale face recognition dataset, we fine-tuned our backbone network from a pretrained ArcFace network. To this end, we first froze the backbone network weights and updated only DAM parameters for DSC<sub>DAM</sub> method and fully connected layer weights for DSC<sub>RNA</sub> method. Once these weights are learned, we stopped the freezing the backbone network weights and trained the full architectures end-to-end manner.

#### SEMANTICALLY RELATED FEATURE EMBEDDINGS

Experiments were also conducted to evaluate if the proposed method yields feature embeddings that effectively cluster semantically and visually similar classes in open-set recognition settings. It should be noted that the semantic relationships

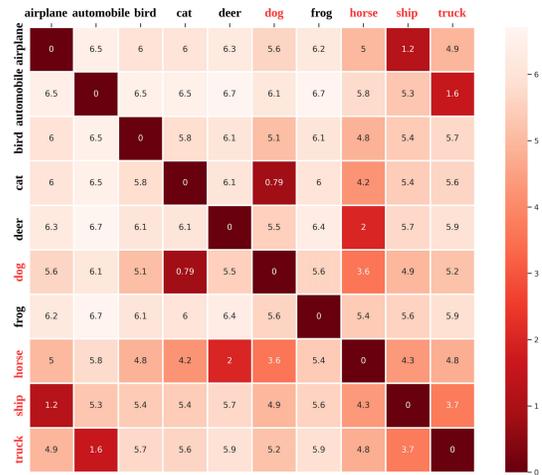


Fig. 7. Distance matrix is computed by using the centers of the testing classes. The four classes that are not used in training are closer to their semantically related classes in the learned embedding space.

are not preserved for the training classes since the Euclidean and angular distances between the class centers are equivalent. However, if the proposed method returns good CNN features, we expect the samples belonging to classes not used in training to lie closer to their semantically related training classes. To verify this, we trained our proposed method by using six classes from the Cifar-10 dataset: airplane, automobile, bird, cat, deer, and frog. Then, we extracted the CNN features of all testing data coming from ten classes by using the trained network. Then, we computed the average CNN feature vector of each class, and computed the distances between them. Fig. 7 illustrates the computed distances between the centers. The distances between the classes used for training are similar and they change between 5.8 and 6.7. The four classes, the dog, horse, ship, and truck classes, that are not used for training are represented with red color in the figure. As seen in the figure, the dog class is closest to its semantically similar cat class, the truck class is closer to its semantically similar automobile class, the horse class is closest to the deer class, and the ship class is closer to the visually similar airplane class (since the backgrounds—blue sky and sea—are mostly similar for these two classes). This clearly shows that the proposed method returns semantically meaningful embeddings.

#### REFERENCES

- [1] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A comprehensive study on center loss for deep face recognition," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 668–683, Jun. 2019.
- [2] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.
- [3] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5419–5428.
- [4] X. Wei, H. Wang, B. Scotney, and H. Wan, "Minimum margin loss for deep face recognition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107012.
- [5] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2006–2014.

- [6] H. Cevikalp, B. Uzun, O. Köpüklü, and G. Ozturk, "Deep compact polyhedral conic classifier for open and closed set recognition," *Pattern Recognit.*, vol. 119, Nov. 2021, Art. no. 108080.
- [7] C. Qi and F. Su, "Contrastive-center loss for deep neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2851–2855.
- [8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
- [9] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Conf. Learn. Recognit. (ICLR) Workshops*, 2015, pp. 84–92.
- [10] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1–9.
- [11] S. Roy, M. Harandi, R. Nock, and R. Hartley, "Siamese networks: The tale of two manifolds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3046–3055.
- [12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6738–6746.
- [13] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1–10.
- [14] K. Zhao, J. Xu, and M.-M. Cheng, "RegularFace: Deep face recognition via exclusive regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1136–1144.
- [15] H. Wang et al., "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.
- [16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4685–4694.
- [17] W. Liu et al., "Learning towards minimum hyperspherical energy," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.
- [18] R. Lin et al., "Regularizing neural networks via minimizing hyperspherical energy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6916–6925.
- [19] W. Liu, R. Lin, Z. Liu, L. Xiong, B. Scholkopf, and A. Weller, "Learning with hyperspherical uniformity," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2021, pp. 1–13.
- [20] Y. Duan, J. Lu, and J. Zhou, "UniformFace: Learning deep equidistributed representation for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3415–3424.
- [21] T.-T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10396–10405.
- [22] V. Pappas, X. Y. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 40, pp. 24652–24663, Oct. 2020.
- [23] F. Pernici, M. Bruni, C. Baecchi, and A. D. Bimbo, "Maximally compact and separated features with regular polytope networks," in *Proc. CVPR Workshops*, 2019, pp. 46–53.
- [24] F. Pernici, M. Bruni, C. Baecchi, and A. D. Bimbo, "Regular polytope networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4373–4387, Sep. 2022.
- [25] T. Kasarla, G. J. Burghouts, M. van Spengler, E. van der Pol, R. Cucchiara, and P. Mettes, "Maximum class separation as inductive bias in one matrix," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–14.
- [26] Q. Bytyqi, N. Wolpert, E. Schomer, and U. Schwanecke, "Prototype softmax cross entropy: A new perspective on softmax cross entropy," in *Proc. Scand. Conf. Image Anal.*, 2023, pp. 16–31.
- [27] Y. Yang, S. Chen, X. Li, L. Xie, Z. Lin, and D. Tao, "Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 37991–38002.
- [28] L. O. Jimenez and D. A. Landgrebe, "Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data," *IEEE Trans. Syst., Man Cybern., C*, vol. 28, no. 1, pp. 39–54, Feb. 1998.
- [29] P. Hall, J. S. Marron, and A. Neeman, "Geometric representation of high dimension, low sample size data," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 67, no. 3, pp. 427–444, Jun. 2005.
- [30] P. Kumar, L. Niveditha, and B. Ravindran, "Spectral clustering as mapping to a simplex," in *Proc. ICML Workshops*, 2013, pp. 1–9.
- [31] M. Weber, "Clustering by using a simplex structure," ZIB, Berlin, Germany, ZIB-Rep. 04-03, 2003.
- [32] A. Rahimi and B. Recht, "Clustering with normalized cuts is clustering with a hyperplane," in *Proc. Stat. Learn. Comput. Vis.*, 2004, pp. 56–69.
- [33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [34] Z. Zhu et al., "A geometric analysis of neural collapse with unconstrained features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–15.
- [35] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1757–1772, 2013.
- [36] A. R. Dhamija, M. Gunther, and T. E. Boult, "Reducing network agnostophobia," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.
- [37] D. Miller, N. Sündnerhauf, M. Milford, and F. Dayoub, "Class anchor clustering: A loss for distance-based open set recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3569–3577.
- [38] H. Cevikalp, B. Uzun, Y. Salk, H. Saribas, and O. Köpüklü, "From anomaly detection to open set recognition: Bridging the gap," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109385.
- [39] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021.
- [40] M. Balko, A. Pór, M. Scheucher, K. Swanepoel, and P. Valtr, "Almost-equidistant sets," *Graphs Combinatorics*, vol. 36, no. 3, pp. 729–754, May 2020.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [42] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Sep. 1995.
- [43] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proc. Neural Netw. Signal Process., IEEE Signal Process. Soc. Workshop*, Aug. 1999, pp. 41–48.
- [44] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.
- [45] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. NIPS*, 2007, pp. 1–8.
- [46] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 741–757.
- [47] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
- [48] H.-M. Yang, X.-Y. Zhang, F. Yin, Q. Yang, and C.-L. Liu, "Convolutional prototype network for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2358–2370, May 2022.
- [49] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4011–4020.
- [50] G. Chen et al., "Learning open set network with discriminative reciprocal points," in *Proc. ECCV*, 2020, pp. 507–522.
- [51] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [52] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proc. ECCV*, 2018, pp. 1–16.
- [53] P. Oza and V. M. Patel, "C2AE: Class conditioned auto-encoder for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2302–2311.
- [54] P. Perera et al., "Generative-discriminative feature representations for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11811–11820.
- [55] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. CVPR*, 2016, pp. 1563–1572.
- [56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

- [58] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Switzerland: Springer, 2016, pp. 87–102.
- [59] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Proc. Workshop Faces 'Real-Life' Images, Detection, Alignment, Recognit.*, 2008, pp. 1–15.
- [60] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "AgeDB: The first manually collected, in-the-wild age database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1997–2005.
- [61] T. Zheng, W. Deng, and J. Hu, "Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments," 2017, *arXiv:1708.08197*.
- [62] T. Zheng and W. Deng, "Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments," Dept. Posts Telecommun., Beijing Univ., Beijing, China, Tech. Rep. 18.01, 2018.
- [63] B. Maze et al., "IARPA Janus benchmark-C: Face dataset and protocol," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 158–165.
- [64] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "FacePoseNet: Making a case for landmark-free face alignment," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1599–1608.
- [65] B.-N. Kang, Y. Kim, and D. Kim, "Pairwise relational networks for face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 628–645.
- [66] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, "Attentional feature-pair relation networks for accurate face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5471–5480.
- [67] H. Cevikalp and G. G. Dordinejad, "Video based face recognition by using discriminatively learned convex models," *Int. J. Comput. Vis.*, vol. 128, no. 12, pp. 3000–3014, Dec. 2020.



processing, machine learning, deep learning, and the control of unmanned aerial vehicles.

**Hasan Saribas** received the bachelor's degree in electrical and electronics engineering from Ataturk University, Erzurum, Türkiye, in 2011, the master's degree from the Department of Avionics, Anadolu University, Eskişehir, Türkiye, in 2015, and the Ph.D. degree from the Department of Avionics, Eskişehir Technical University, Eskişehir, in 2020.

He is currently employed as a Senior AI Research Engineer with Huawei Türkiye Research and Development Center, Istanbul, Türkiye. His research interests include recommendation systems, image



**Hakan Cevikalp** (Member, IEEE) received the M.S. degree from the Department of Electrical and Electronics Engineering, Eskişehir Osmangazi University, Eskişehir, Türkiye, in 2001, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA, in 2005.

He worked as a Post-Doctoral Researcher with the Lear Team, INRIA Rhone-Alpes, Montbonnot-Saint-Martin, France, in 2007, and at Rowan University, Glassboro, NJ, USA, in 2008. He is currently working as a Full Professor with the Department of Electrical and Electronics Engineering, Eskişehir Osmangazi University. His research interests include pattern recognition, machine learning, neural networks, image and signal processing, optimization, and computer vision.



**Bedirhan Uzun** received the Master of Science degree from the Department of Electrical and Electronics Engineering, Eskişehir Osmangazi University, Eskişehir, Türkiye, in 2020, where he is currently pursuing the Ph.D. degree.

He served as a Research Assistant with the Department of Electrical and Electronics Engineering, Eskişehir Osmangazi University. Currently, he is working as a Blockchain Developer with API3, George Town, Grand Cayman. His research interests lie in the fields of computer vision and blockchain technologies.