IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 March 2023, accepted 27 April 2023, date of publication 12 May 2023, date of current version 24 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3275743

RESEARCH ARTICLE

Quad-Channel Contrastive Prototype Networks for Open-Set Recognition in Domain-Specific Tasks

GUSTI AHMAD FANSHURI ALFARISY[1,2], OWAIS AHMED MALIK[1], AND ONG WEE HONG[1]
[1]School of Digital Science, Unversiti Brunei Darussaslam, Gadong BE 1410, Brunei Darussalam
[2]Department of Informatics, Institut Teknologi Kalimantan, Balikpapan 76127, Indonesia

Corresponding authors: Owais Ahmed Malik (owais.malik@ubd.edu.bn) and Gusti Ahmad Fanshuri Alfarisy (20h8562@ubd.edu.bn)

This work was supported by the Universiti Brunei Darussalam under Grant UBD/RSCH/1.18/FICBF(a)/2022/006.

ABSTRACT A traditional deep neural network-based classifier assumes that only training classes appear during testing in closed-world settings. In most real-world applications, an open-set environment is more realistic than a conventional approach where unseen classes are potentially present during the model's lifetime. Open-set recognition (OSR) provides the model with the capability to address this issue by reducing open-set risk, in which unknown classes could be recognized as known classes. Unfortunately, many proposed open-set techniques evaluate performance using ''toy'' datasets and do not consider transfer learning, which has become common practice in deriving a strong performance from deep learning models. We propose a quad-channel contrastive prototype network (QC-CPN) using quad-channel views of the input with contrastive prototype loss for real-world applications. These open-set techniques also require the tuning of new hyperparameters to justify their performance, so we first employ evolutionary simulated annealing (EvoSA) to find good hyperparameters and evaluate their performance with our proposed approach. The comparison results show that QC-CPN effectively outperforms other state-of-the-art techniques in rejecting unseen classes in a domain-specific dataset using the same backbone (MNetV3-Large) and could become a strong baseline for future study.

INDEX TERMS Machine learning, deep learning, prototype learning, open-set recognition, evolutionary simulated annealing, novelty detection, out-of-distribution detection.

I. INTRODUCTION

Current deep neural networks attain outstanding predictions in real-world cases. Many proposed architectures [1], [2] perform with roughly 90% accuracy on the popular large-scale dataset, ImageNet [3]. In some applications, the convolutional neural network achieves over 90% accuracy [4], [5]. These results demonstrate the effectiveness of deep learning models in recognizing images.

Unfortunately, the model may encounter unseen classes in some applications; therefore, enhancing the model to become open to unseen classes will generate a trustworthy classifier and benefit future learning. For instance, a waste detector could encounter a new type of waste during prediction that relies on the variety of the products. By detecting a novel type of waste, the model could learn different types in the future. Other critical applications are in the medical domain. Providing the capability to reject the unknown will help reduce false positive rates, which improves diagnosis.

The incapability of deep neural networks to identify unknown classes is due to the popular Softmax layer, as output probability that does not account for the unseen classes. It makes a model recognize the world that has been trained. Relying on current approaches may lead to incorrect predictions or overconfidence (a higher probability is assigned to a particular class) in unknown classes [6].

Many studies address this issue by introducing Open Set Recognition (OSR) [7], which is the relationship between novelty detection and multi-class recognition [8]. An additional risk to the standard empirical risk in machine

The associate editor coordinating the review of this manuscript and approving it for publication was Aasia Khanum.

learning is called open-space risk, where the model risks recognizing unknown classes as known classes. The model should reduce this risk in open-set problems and become the main objective. The unknown label dataset could be used to reduce the risk. However, it is problematic due to the infinite number of unknowns. Hence, utilizing available classes for identifying unknowns should be studied to minimize empirical risk (known class) and open-space risk (unknown class).

Unfortunately, most evaluations in OSR research have used experimental datasets, with MNIST, CIFAR10, CIFAR50, and/or TinyImageNet being the most popular [9], [10], [11], [12], [13], [14], [15], [16]. Only a few studies have used other datasets when constructing an open-set model [17], [18], [19], but these approaches may be ineffective when dealing with real-world applications in a specific domain.

In the CIFAR10 dataset, the class consists of vehicles and animals which could become contradictory in terms of the domain. In CIFAR100, the domain includes animals, food, flowers, household objects, nature scenes, people, and vehicles, which are not domain-specific. ImageNet also utilizes a broad domain. In some real-world applications, a specific domain is more useful, e.g., food, waste, medical images, or species. Thus, we conducted our study using a domain-specific dataset.

Alternatively, pre-trained deep learning models using transfer learning are common when enhancing performance in real-world cases [20], [21], [22]. This due to the utilization of the pre-trained parameters enhancing generalization of the targeted task. Yet, the effect of transfer learning when making models open to various proposed open-set techniques remains unclear as it has a generalization knowledge that could be employed to detect open set. This motivates us to employ transfer learning in OSR to domain-specific problems such as food, waste, pets, birds, and medical images. With this experimentation, the actual performance of OSR can be evaluated.

Additionally, the parameters in many open-set classifiers may be sensitive to the problem and may affect the performance of rejecting unknown classes. For instance, in the prototype-based deep neural networks proposed by [14], distance-based cross-entropy loss (DCE) and prototype loss (PL) require parameters to be adjusted. In DCE, distance is weighted by a predefined parameter γ that could affect the classification results. The importance of PL during training is affected by parameter λ. These two losses behave similarly, which pushes the feature vector toward its prototype. Adjusting both parameters to the target problem will affect open-set performance.

Based on these issues, we propose hyperparameter optimization to adjust the parameters of the open-set classifier to improve its performance in rejecting unseen classes. Evolutionary simulated annealing (EvoSA) approach has been chosen for the optimization as it performs better than the popular Bayesian Optimization. EvoSA could find the deep learning architecture with a smaller number of parameters and latency compared to simulated annealing and bayesian optimization [23]. EvoSA works by accepting a bad candidate solution followed by a recombination process after the simulated annealing which yields a better solution than bayesian optimization with the same number of objective evaluations. Moreover, the strength of the movement in searching for each hyperparameter can be controlled by a parameter in EvoSA. This level of movement is helpful in adjusting the movement of unique hyperparameters in open-set recognition for finding optimal open-set hyperparameters.

We employ prototype-based neural networks to design an open-set deep neural network as it offers interoperability of the recognition decision of the open set. We can assume that unseen classes should be at a distance from the known prototypes. To achieve this, the model should learn to generate prototypes far away from each other, which could be attained through contrastive learning. Even though contrastive learning is mostly studied through self-supervised learning by providing negative samples [24], it has not been studied with prototype-based neural networks for OSR. As the prototype is a representation of the samples, performing contrastive learning on only the prototypes (without using feature vectors of the samples) is plausible. Therefore, contrastive prototype loss is proposed in this paper so that the prototypes have a significant margin among them.

As humans recognize novel objects, multiple views could be helpful [25]. At the technical level, we use rotational geometric transformation to extract additional perspectives of the input images and use the combined features during prediction and training. Changing the orientation by rotation could enable the model to grasp the invariant property of the object in terms of area and angle. Therefore, we propose quad-channel contrastive prototype networks (QC-CPN) with evolutionary simulated annealing to open the deep neural networks model to the world.

The main contributions of this paper are as follows:

1) We present the evaluation of each Open Set Recognition (OSR) algorithm using hyperparameter optimization to gain its true performance for each domain-specific dataset.
2) We propose contrastive prototype loss that utilizes prototypes only in each mini-batch of training samples.
3) We propose using a quad-channel input image representation through geometric rotations to derive a better representation for rejecting unseen classes.

II. RELATED WORK

A. OPEN-SET RECOGNITION

OSR is a multi-class classification technique where the classifier can recognize unseen classes by minimizing open-set risk evaluated on unseen classes during testing phase [7]. Open-set risk is defined by minimizing the open-space risk (the risk of unseen class data recognized as known) and empirical risk that utilizes collected training data in known and unseen

samples (associated with data testing). Minimizing empirical risk is the main objective of conventional machine learning. The risk is alleviated by adopting an available training set to assess the true distribution of data. Meanwhile, open-space risk tries to avoid unseen classes being recognized as known classes. Merely reducing empirical risk does not meliorate the open-space risks.

To solve the OSR problem, both empirical and open-space risks are minimized. Unseen classes have to be combined with the test set, and these additional classes (unseen) can be utilized during training. However, state-of-the-art techniques use the set to find appropriate thresholds [16], [17] and do not employ it during training. Furthermore, generative approaches use the test set for assessment only [26].

An early attempt at OSR based on deep neural networks was conducted by Bendale and Boult [6] who introduced OpenMax as a replacement for Softmax. Here, The vectors from the penultimate layer are calibrated and a new element is added to represent an unknown class. Weibull distributions are employed to fit tail distances from the vector to its mean. However, two-stage tasks are required to construct OpenMax layers, i.e., training and calibration (including mean calculations). Moreover, considering class-incremental settings, the approach requires the re-computation on the second task.

Another study employed a modified conditional generative adversarial network (GAN) to synthesize new unknown samples. Using the same technique as OpenMax, i.e., fitting the Weibull distribution [9]. This technique also involved an additional mean activation vector for the unknown. MNIST [27] and HASYv2 [28] were used for the main experiment, which showed that the proposed method performed better than Softmax, generative Softmax, and OpenMax. Unfortunately, no apparent improvement was observed when applied to natural images using ImageNet12.

Synthesizing new unknown examples based on the assumption of counterfactual settings was proposed by [11] using an encoder-decoder GAN. At first, latent vectors close to its known classes were optimized so they did not belong to any known classes indicated by low probability value. Then, the generated samples were treated as open set data. The area under the ROC (AUROC) curve metrics showed that the proposed method could achieve a higher value compared to other studies. However, the improvement was small with a 0.4% AUROC score on CIFAR-10 compared to OpenMax.

OpenMax was extended by Yoshihashi et al. [15] by refining latent representation using encoder-decoder architecture, and additional latents for each layer were proposed for OSR. These latents were jointly concatenated with the label vector. The proposed technique illustrated higher F1 scores compared to supervised only and LadderNet on Omniglot and MNIST datasets. The authors did not compare their proposed approach with the latent vectors from the autoencoder to detect unknown classes, which could justify the effectiveness of the proposed technique.

Meanwhile, Chen et al. [17] proposed a different view of learning by classifying samples using reciprocal points. These points are the centroid of the classes but far from their sample's feature vectors, and they represent the dissimilarity of samples from other known classes. The prediction is performed by calculating the distance from the feature vector to the reciprocal point, and the maximum distance produces the highest probability. The AUROC score was higher than prototype learning. Furthermore, the study was extended by employing GAN to generate unknown samples [26]. However, the justification of the prediction based on reciprocal points is not appropriate for OSR, as the dissimilarity of other classes does not provide an actual representation of the recognized class. Furthermore, when a new class is added, the approach needs a revision of all reciprocal points to perform a prediction, as a new class provides additional dissimilarity to the respective reciprocal points.

In a study by Schlachter et al. [13], the unknown set could be derived using a misclassified set or low probability prediction. This set provides additional unknown classes. The results showed that dynamic intra-class splitting provides a better AUROC compared to counterfactual image generation and the Weibull support vector machine. However, when the model's prediction was nearly faultless, only a few sets could be trained as unknown for the K+1 class.

B. PROTOTYPE LEARNING FOR OSR

Prototype learning for the discriminative representation of the recognition model using a deep neural network was introduced by Yang et al. [14]. The neural network architecture was modified at the output layers, and could be interpreted as a feature vector. Additional trainable vectors that behave as the prototype of the respective class were added outside the neural networks. These vectors were updated for each computed loss via DCE. Generalized prototype loss was also proposed to learn each prototype close to its samples. Adding more weight to this loss contributed to a more distinctive representation. The output probability or distance to its prototype was then tuned to find the desirable threshold for rejecting unidentified classes during testing. A One-vs-all strategy was also proposed for prototype learning [29]. Using this technique, we can assume that unseen classes are far from the prototypes.

Contrastive learning on the prototypes was not considered in these studies. We argue that to identify unseen classes, the prototype should learn how dissimilar it is from other prototypes. By learning dissimilarity, the model knows the differences between all prototypes. When the model predicts any sample, this dissimilarity leads to better rejection and can be interpreted in prototype-based neural networks as each prototype being distant.

Prototype-based learning was developed by Shu et al. [18] with a multifold loss function consisting of classification loss, prototype loss, and prototype radius loss. The first loss utilized the cross-entropy loss of samples' features. Meanwhile,

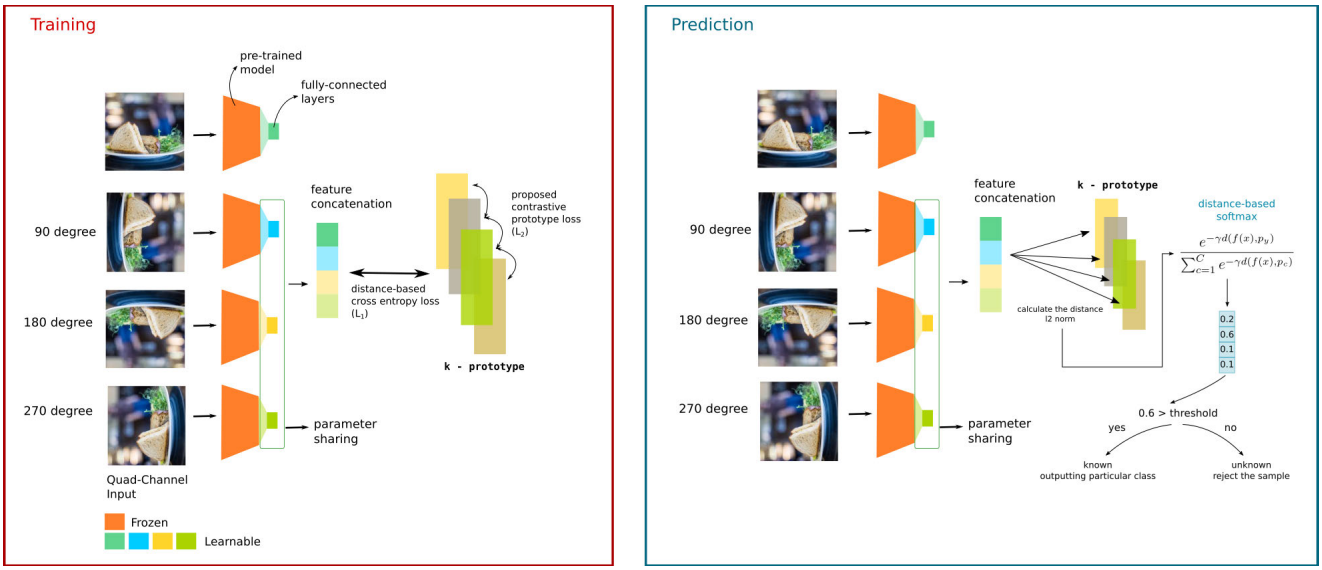


FIGURE 1. Quad-Channel Contrastive Prototype Networks (QC-CPN) architectures during training and prediction.

a radius was learned for eachal category that restricted the prototype's value. Three thresholds were needed to reject unknown classes: the acceptance threshold, rejection threshold, and distance-based threshold. The experiment showed that prototype radius loss was more accurate at rejecting unknown classes. This technique is only proven in action recognition tasks. The utilization of other open-set techniques for this domain needs more investigation.

Another study extended prototype learning using a margin [30]. The distance between samples and their prototypes was constrained by a margin called motorial prototype framework (MPF). Furthermore, the standard MPF was improved by a GAN with an additional classifier to generate samples that were close to unknown space. The generated samples were trained to be close to the centers of all prototypes which were assumed to stand in an open-set space. Their AUROC score was competitive compared to other open-set models. However, this method was not proven in a real world scenario that utilized transfer learning and did not consider the maximum distance between the prototypes.

C. CONTRASTIVE LEARNING

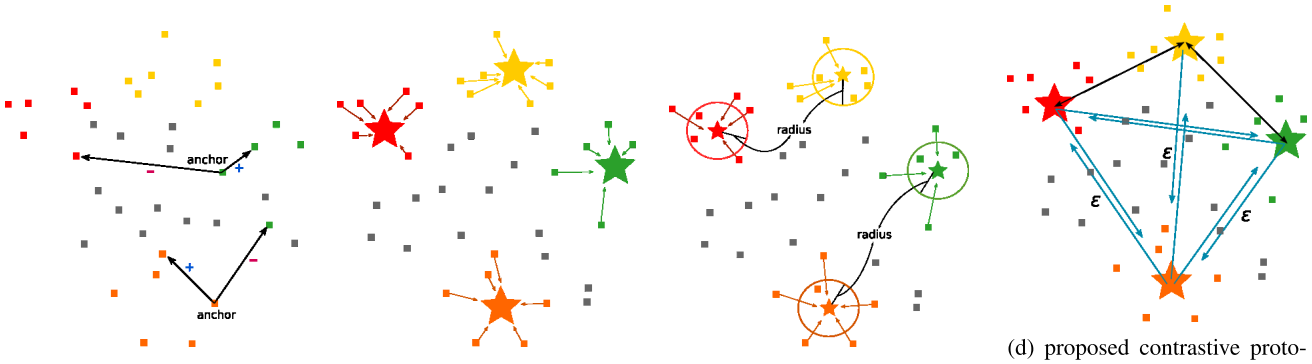
Contrastive learning is a representation learning type that utilizes positive (similar representation) and negative (dissimilar representation) data during training. The distance from the sample or anchor to the data is modified by minimizing the distance to positive data and pushing away from negative data [31]. The most popular implementation method is utilizing triplet loss [32] in deep neural networks, which is used in many applications [33], [34].

Deriving contrastive representation by training negative samples has proven to be effective in few-shot learning. In the study conducted by Gao et al. [35], a prototype was derived from the mean of the feature vectors from correctly classified support samples, and the exponent value from cosine similarity of positive samples was contrasted with other similarity values of negative samples that manifested in their contrastive loss. The results showed better classification accuracy compared to previous studies in terms of 5-way 1-shot and 5-way 5-shot. Furthermore, contrastive learning based on the prototype was also studied as representation learning for unlabelled data [36]. Using the expectation-maximization framework, K-means was employed in the expectation process to construct an estimated representation from the momentum encoder.

In a study by Kodama et al. [37], a self-supervised technique called SimCLR was proposed, based on contrastive learning. The encoder network was trained first using SimCLR, then its parameters were frozen to learn the fully-connected layers for classification. The anomaly was detected by using extreme value theory on the weighted average of two derived features. The AUROC score was better on the street view house numbers (SVHN) and Tiny Imagenet datasets; however, the comparison results may not represent their true performance as the authors took the results directly from the respective papers. The architecture used probably contributed to the performance.

III. THE PROPOSED METHOD

In this section, we present the architecture of the proposed Quad-Channel Contrastive Prototype Network (QC-CPN) along with its prediction and training mechanisms. The hyperparameters of QC-CPN (λ, feature dimension, learning rate, and ϵ) are tuned using the Evolutionary Simulated Annealing (EvoSA) [23]. Each hyperparameters will be discussed in Section IV-A. We only optimize the number of feature dimensions in one layer, not the combination of layers as it is used in the EvoSA paper. Thus, we did not use the flipped mutation.



(a) triplet loss (b) prototype loss (c) motorial margin loss (d) proposed contrastive prototype loss

FIGURE 2. The illustration of the proposed loss, triplet loss, and other prototype-based loss. The grey boxes show the open-set samples, and the stars symbol shows the prototypes. (a) triplet loss uses the three samples to learn the features; (b) prototype loss pushes the feature representation to its representative prototype; (c) motorial margin loss provides mitigation to the feature's sample by using a radius; (d) our proposed contrastive prototype loss utilizes the prototype in each mini-batch of training data without using any sample and adjusts the prototypes to have a margin of ϵ.

A. QUAD-CHANNEL CONTRASTIVE PROTOTYPE NETWORKS (QC-CPN)

1) ARCHITECTURE OF THE QC-CPN

The overall architecture of QC-CPN is depicted in Figure 1 which consists of two main components: a feature extractor (FE) and a prototype representation. FE extracts the feature from quad-channel inputs using a pre-trained model with frozen parameters (MNetV3-Large) followed by fully connected layers, while a prototype represents a class. Each class has one prototype in our case which is parameterized as a vector and initialized randomly.

Transfer learning is utilized by keeping the parameters in the backbone of MNetV3-Large followed by the training of the fully-connected layers. The parameters in the MNetV3-Large contain the knowledge from the ImageNet classification task, by using the output vector from the backbone, the low-level features derived from diverse classes can be utilized which could improve the generalization capability in the open-set model. This utilization will lead to better rejection capability as the output vector from unknown classes is obtained from the parameters of the ImageNet, which could affect the output probability distribution.

In quad-channel views, the input image is rotated in three ways which are 90, 180, and 270 degrees. Thus, four images are passed as an input to the QC-CPN model. Each image produces its feature vector concatenated as one vector to represent the image with perspective at different angles. The parameters in the pre-trained model are frozen to preserve its knowledge. The fully-connected parts of the pre-trained deep learning models, intended for previous training (ImageNet), are replaced with one new fully-connected layer, and the parameters on the rotated images are shared to accelerate the training.

Meanwhile, each class has its prototype representing a centroid of the class that is parameterized. The distance between concatenated feature vectors derived from quad-channel input and all prototypes are mapped to the probability distribution through softmax layers. The maximum probability shows the prediction of this model. For the rejection mechanism, the maximum probability is compared with the threshold. A lower value will make the model identify the sample as unseen classes. The best threshold is determined from ROC [38].

2) QC-CPN FOR PREDICTION

Prediction is performed by calculating the Euclidean distance between a feature from FE and the prototypes. Let $x \in X \subset \mathbb{R}^{w \times h \times q}$ be an image with w width, h height, and q number of channels. Let $\boldsymbol{z} \in Z \subset \mathbb{R}^n$ be a final feature vector with n number of dimensions. Then, FE can be formalized as a function which is shown in Equation (1).

$$f_{\phi,\theta} : X \to Z \tag{1}$$

FE is parameterized by two parameters: ϕ and θ. This is intended to differentiate functionality in a model for extracting the features. Parameter ϕ is used to extract the vector before the fully-connected layers. It is frozen from the pre-trained model and shared across all quad-channel inputs. We can see this as the parameter for the pre-trained backbone. Parameter θ is used to get the partial feature vector from an original image and a rotated image. For rotated images, θ is shared in producing the partial feature. The four partial feature vectors are concatenated to produce the final feature vector.

In the quad-channel itself, four images are used as input. Three images are derived from the original image through the rotation function that is defined as $r_i : X \to X$ where $i \in \{90°, 180°, 270°\}$, e.g, $r_{90°}(x)$ means that an image x is rotated by 90°.

Parameter ϕ is used to get the feature before the fully-connected parts. Let $\boldsymbol{a} \in A \subset \mathbb{R}^u$ where $u = \frac{1}{4}n$, then a function to extract feature from the backbone can be defined as shown in Equation (2). Beside that, a function to extract the partial final feature with parameter θ can be defined in Equation (3).

$$g_\phi : X \to A \tag{2}$$

$$h_\theta : A \rightarrow A \tag{3}$$

Then, the final feature extractor $f_{\phi,\theta}$ can be defined in Equation (4). Three rotated images are produced by $r_i(x)$. All of them, including the original one, will be passed to the g_ϕ to extract the feature from the pre-trained model. Afterward, the interim feature of the four images is passed to h_θ. All of the derived vectors are concatenated through concat function.

$$f_{\phi,\theta}(x) = \text{concat}\Big(h_\theta(g_\phi(x)), h_\theta(g_\phi(r_{90^\circ}(x))), h_\theta(g_\phi(r_{180^\circ}(x))), h_\theta(g_\phi(r_{270^\circ}(x)))\Big) \tag{4}$$

Let $\boldsymbol{m} \in M \subset \mathbb{R}^n$ be the prototype and $|M| = C$ where C is the number of prototypes that is the same with the number of classes, meaning that one class is represented by one prototype. Let c be the class index. For the sake of simplicity, ϕ and θ are omitted. The distance function can be defined as shown in Equation (5) which uses L2 norm or Euclidean distance.

$$d(f(x), m_c) = \|f(x) - \boldsymbol{m_c}\|_2 \tag{5}$$

The prediction through probability value can be derived from the softmax layer based on the negative distance controlled with parameter $\gamma \in [0, 1]$ as shown in Equation (6). Negative exponential behaves as an inverse of the distance such that a higher probability indicates that the model predicts strongly into the respective particular class.

$$p(y|x; f, M) = \frac{e^{-\gamma d(f(x), \boldsymbol{m_y})}}{\sum_{c=1}^{C} e^{-\gamma d(f(x), \boldsymbol{m_c})}} \tag{6}$$

The time complexity of the prediction of an instance of an image is $\mathcal{O}(n^2)$ where n is the number of partial features. The asymptotic notation is derived from $\mathcal{O}(4 \cdot a \cdot b \cdot l + 4 \cdot n_1 \cdot n_2 + 4 \cdot n_2 \cdot C)$ where a is the time complexity in forwarding the image into convolution layer, b is the number of the output feature map, l is the number of layers in convolution part, the number 4 is the number of input, n_1 is the number of the dimension or neuron of last layer of the backbone, n_2 is the number of dimension of partial feature, and C is the number of classes. The first terms represent the backbone, the second term represents the partial features, and the last term represents the distance computation for the output score. If $n_1 = n_2$, then we will obtain $\mathcal{O}(4 \cdot a \cdot b \cdot l + 4 \cdot n^2 + 4 \cdot n \cdot C)$. Considering the backbone as a constant operation, we have an upper bound of $\mathcal{O}(n^2)$ in which the complexity of the networks depends on the number of partial features.

3) TRAINING QC-CPN FOR OPEN-SET RECOGNITION

Training QC-CPN requires computing two losses: the discriminative loss and the contrastive prototype loss. In the discriminative loss, categorical cross-entropy based on the distance is employed, similar to [14]. Optimizing this loss will make a sample having a shorter distance to its actual prototype and a longer distance to others. This loss is defined in Equation (7). The sample of mini-batch k is denoted as $x_k = \{x_k^i\}_{i=1}^S$ with label $y_k = \{y_k^i\}_{i=1}^S$ where S is the total samples in the mini-batch.

$$\mathcal{L}_1(x_k, y_k) = -\frac{1}{S}\sum_{i=1}^{S} \log p(y_k^i | x_k^i; f, M) \tag{7}$$

The proposed contrastive prototype loss is defined in Equation (8). This loss is intended to make each prototype representing a particular class far from each other by a margin which is denoted as a contrastive prototype parameter (ϵ). Furthermore, this loss only calculates the distance based on mini-batch, which may provide faster learning. Compared to triplet loss that is used in many studies [39], [40], our proposed loss function does not need to use an algorithm or random sampling to choose the negative or positive samples or a synthetic image. Original triplet loss requires sampling of negative samples, which shows that randomly choosing the samples leads to a sub-optimal solution [41]. We utilize the prototypes only as they represent the classes. Each prototype that belongs to the data label in the mini-batch will be forced away. The illustration of the proposed loss and its difference from others is shown in Figure 2.

$$\mathcal{L}_2(y_k) = \frac{1}{S}\sum_{i=1}^{S}\sum_{j=1}^{C} \max(0, \epsilon - d(m_{y_k^i}, m_{y_k^j})), y_k^i \neq y_k^j \tag{8}$$

The total loss is defined in Equation (9) where x_k is the images in the training set for mini-batch k with its associate labels y_k. The training set X_{train} and Y_{train} consist of multiple mini-batch in which $X_{train} = \{x_k\}_{k=1}^K$ and $Y_{train} = \{y_k\}_{k=1}^K$ where K is the total mini-batch for each epoch. The weight for the $\mathcal{L}_2$ is denoted as w_1. The parameter will be updated based on the loss of each mini-batch of samples using Equation (9).

$$\mathcal{L}(x_k, y_k) = \mathcal{L}_1(x_k, y_k) + w_1 \cdot \mathcal{L}_2(y_k) \tag{9}$$

The objective of the training phase is to minimize Equation (9). Using both proposed losses, prototypes are ensured to be far from each other with the discriminative functionality. With optimized ϵ, open samples will have more space between prototypes that are not identified as known classes leading to higher rejection performance of unseen classes. Based on the formalization, the algorithm for training is summarized in Algorithm 1.

The time complexity for the training is $\mathcal{O}(s \cdot n^2)$. Since we froze the parameters in the backbone part, updating these parameters is unnecessary in which can be omitted. Hence, the updating parameters are applied for partial features and the prototypes only with complexity $\mathcal{O}(s \cdot n^2 + s \cdot n \cdot c)$. The first term represents the operations of updating parameters of partial features and the second term is the parameters update of the prototypes. Hence, the complexity depends on the number of mini-batch of samples (s) and the number of dimensions of partial features.

TABLE 1. Hyperparameter search settings.

Method	Hyperparameter	Lower Bound	Upper Bound	δ
All	Feature	100	4000 (1000 for QC-CPN)	500
	Learning Rate	1×10^{-6}	0.1	0.1
QC-CPN	γ	0	1	0.2
	ϵ	1	500	10
CPL	γ	0	1	0.2
GCPL	γ	0	1	0.2
	λ	0	1	0.2
RPL	T (temperature)	0	1	0.2
	λ	0	1	0.2
ARPL	T (temperature)	0	1	0.2
	λ	0	1	0.2
MPF	λ	0	1	0.2

Algorithm 1 QC-CPN Training Algorithm

Input : set of the mini-batch of data points or images from the training set X_{train} with their associate labels Y_{train}, weight for prototype contrastive loss w_1, number of feature dimension n, learning rate α, contrastive prototype parameter ϵ, the strength of the distance in softmax normalization γ, and *total_epoch*.

Output : the parameter θ and M of deep learning model h

$i \leftarrow 0$
while $i <$ *total_epoch* **do**
  for each (x_k, y_k) in (X_{train}, Y_{train}) **do**
    compute the loss $\mathcal{L} = \mathcal{L}_1(x_k, y_k) + w_1 \cdot \mathcal{L}_2(y_k)$
    update parameter θ by $\nabla_\theta \mathcal{L}(x_k, y_k)$, in case of vanilla gradient descent: $\theta = \theta - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \theta}$
    update parameter M by $\nabla_M \mathcal{L}(x_k, y_k)$ in case of vanilla gradient descent: $M = M - \alpha \cdot \frac{\partial \mathcal{L}}{\partial M}$
  end for
  $i \leftarrow i + 1$
end while

B. UNKNOWN (UNSEEN) CLASS REJECTION

We simply use the maximum from the probability distribution, which is also used in out-of-distribution detection [42] as a baseline algorithm. This value is compared with the threshold, and a high value represents the short distance to the corresponding prototype, which is more likely known classes rather than unknown ones. The illustration can be seen in the right box in Figure 1.

The threshold itself is determined through ROC, which maps the false positive rate with the true positive rate with various thresholds. The threshold with a low false positive rate and a high true positive rate is chosen.

IV. EXPERIMENTAL SETTINGS

This section outlines the three aspects of our experiment settings: hyperparameters, the dataset used, and the evaluation of the model's performance. We compare the optimized hyperparameters for each algorithm with our approach to give a fair comparison between all methods. The dataset used will be explained along with its settings for unknown class selection. The section concludes with a description of the metric used, the ablation study, and the effect of the contrastive prototype parameter.

For the computation environment, we employed the GPU to train the model through a popular deep-learning library. we run the experimentation using a single GPU with RTX 2060. We used the PyTorch library for implementing our method.

A. EVOSA SETTINGS IN SEARCHING FOR HYPERPARAMETERS

We performed hyperparameter optimization using EvoSA to obtain the best settings in the test set before making a comparison using the same backbone, MNetV3-Large [43], as a feature extractor. Details of the search settings in EvoSA are shown in Table 1. The method is the algorithm or learning technique for OSR, hyperparameters control the learning scheme in OSR, the lower and upper bounds are the minimum and maximum values that the solution can provide, and δ is the parameter in EvoSA that controls the strength of the neighboring movement for each specific hyperparameter.

The objective or cost function of EvoSA is defined in Equation (10). The main objective is to find the hyperparameter p for each method trained using training dataset D_{train} that maximizes the AUROC score between the test dataset of known classes (D_{test}) and the dataset containing unknown classes ($D_{unknown}$). We only use one combination (ordered labels) to perform hyperparameter searching and use the best p to evaluate the different combinations of labels.

$$\min_p 1 - \text{AUROC}(p, D_{train}, D_{test}, D_{unknown}) \tag{10}$$

The number of feature dimensions represented by the number of neurons in the penultimate layer and the learning rate was optimized for all algorithms. In QC-CPN, CPL [14], GCPL [14], RPL [17], and ARPL [26] γ or temperature provides the weight for the distance before outputting probability distribution through Softmax. We argue that this parameter is sensitive to the AUROC score and needs to be optimized. Meanwhile, ϵ in QC-CPN shows the minimal distance between the prototypes, and parameter λ in GCPL shows the strength of the prototype loss [14]. In RPL, λ is a regularization term to reduce open space risk by constraining the distance between samples and reciprocal points by a

TABLE 2. Cost value and best hyperparameters after EvoSA.

Dataset	Method	Cost	Hyperparameters			
			γ/ Temperature	Feature	Learning Rate	ϵ/λ
Garbage6 *(waste domain)*	Softmax	0.1756	-	2609	0.00002	-
	CPL [14]	0.1905	0.7994	700	0.00019	-
	GCPL [14]	0.2606	0.7685	2069	0.05874	0.0027
	RPL [17]	0.1563	1	2532	0.01972	0.4031
	ARPL [26]	0.1325	0.5602	2996	0.00587	0.0754
	MPF [30]	0.1129	-	2849	0.02858	0
	QC-CPN	**0.1110**	0.1845	141 x4	0.00531	386.2027
OxfordPet *(pets domain)*	Softmax	0.2459	-	375	0.00120	-
	CPL [14]	0.2533	0.6090	533	0.00657	-
	GCPL [14]	0.2488	0.9986	100	0.06525	0
	RPL [17]	0.2364	0.9270	2655	0.00514	0.4737
	ARPL [26]	0.2544	0.6545	665	0.00360	0.2314
	MPF [30]	0.2895	-	2114	0.00167	0.3662
	QC-CPN	**0.2236**	0.5213	921 x4	0.04910	79
IndianFood *(food domain)*	Softmax	0.2186	-	100	0.00422	-
	CPL [14]	0.2316	1	189	0.02546	-
	GCPL [14]	0.1811	0	349	0.00086	1
	RPL [17]	0.1794	1	1376	0.00011	0.1863
	ARPL [26]	0.1545	0.4100	399	0.00159	0.0151
	MPF [30]	0.1968	-	317	0.00161	0.0993
	QC-CPN	**0.1255**	0.3942	884 x4	0.00355	65.7975
CUB 200 *(bird species domain)*	Softmax	0.2916	-	1982	0.00040	-
	CPL [14]	0.2519	0.9797	250	0.00254	-
	GCPL [14]	0.2753	0.5963	881	0.00450	0
	RPL [17]	0.3218	0.6087	2007	0.86890	0.0046
	ARPL [26]	0.3420	1	1686	0.00746	0.2059
	MPF [30]	0.3172	-	1352	0.00160	0.3450
	QC-CPN	**0.1937**	0.3529	827 x4	0.00482	45.9587
HAM10000 *(skin lesion domain)*	Softmax	0.3528	-	3586	0.000001	-
	CPL [14]	0.3254	0.2184	2922	0.00455	-
	GCPL [14]	0.2897	0.2462	1598	0.00774	0.0667
	RPL [17]	**0.0271**	1	3240	0.00327	0.7291
	ARPL [26]	0.0766	0.5945	4000	0.07186	0.7893
	MPF [30]	0.1033	-	1815	0.05045	0.9223
	QC-CPN	0.0698	0.1053	579 x4	0.06212	305.2671

learnable margin [17]. In the MPF [30], λ behaves similarly to RPL which is the minimum margin between features and their prototype. Meanwhile, hyperparameter optimization is not considered for OpenMax because it involves a two-stage approaches that utilizes a trained Softmax-based model (the hyperparameters of the Softmax model itself have been optimized).

B. DATASETS

The summary of the dataset used in the OSR experimentation is shown in Table 3. The number of classes is split into two sets for obtaining known classes and unknown classes. Meanwhile, the detailed description of each dataset is described below.

1) Garbage6

This dataset consists of 6 different types of garbage collections (i.e. six classes): cardboard, glass, metal, paper, plastic, and trash. For evaluating open-set tasks, four classes were selected as *known* and two classes were chosen as *unknown*. In optimizing the hyperparameters, four classes of this dataset were used as the known dataset (from one to four). For five trials of experimentation, we selected the classes randomly. The dataset can be found at https://www.kaggle.com/harshul23/garbage.

2) OxfordPet

OxfordPet dataset contains images different species of cats and dogs [44]. It consists of 37 species intended for fine-grained object categorization. We selected the labels from 1 to 22 for searching best hyperparameters. For open-set experimentation, we selected 15 classes as *unknown*. Five trials were performed with a random combination of classes. This dataset can be found at https://www.robots.ox.ac.uk/ vgg/data/pets/.

3) IndianFood

IndianFood dataset has images for 20 classes of Indian cuisine. The 12 classes were selected as *known* samples while other eight classes were chosen as *unknown* for hyperparameter optimization. For five random trials, 12 different combinations of classes were used. This dataset is available at https://www.kaggle.com/theeyeschico/indian-food-classification.

4) CUB-200

CUB-200 is a dataset of bird species with 200 classes [45]. This dataset can be used to assess the performance of OSR in terms of fine-grained classification. For hyperparameter optimization, we have chosen 120 classes and

TABLE 3. Summary of the dataset used in the experimentation.

Dataset	Domain	the Number of Classes	the Number of Known Classes	the Number of Unknown Classes	Total Images
Garbage6	waste or garbage	6	4	2	2533
OxfordPet	cats and dogs species	37	22	15	7393
IndianFood	food	20	12	8	6271
CUB-200	bird species	200	120	80	11988
HAM10000	dermatoscopic images	7	4	3	10015

TABLE 4. Closed-set accuracy, AUROC, and BACCU score over 5 random trials.

Method	Garbage6 *(waste)*	OxfordPet *(pets)*	IndianFood *(food)*	CUB-200 *(bird species)*	HAM10000 *(skin lesion)*	Mean
Closed-Set Accuracy						
Softmax	0.934 ± 0.014	$\mathbf{0.899} \pm 0.011$	$\mathbf{0.883} \pm 0.026$	$\mathbf{0.630} \pm 0.024$	0.932 ± 0.022	0.856
OpenMax [6]	0.934 ± 0.014	$\mathbf{0.899} \pm 0.011$	$\mathbf{0.883} \pm 0.026$	$\mathbf{0.630} \pm 0.024$	0.932 ± 0.022	0.856
CPL [14]	0.911 ± 0.010	0.817 ± 0.018	0.845 ± 0.019	0.530 ± 0.013	0.954 ± 0.015	0.812
GCPL [14]	0.901 ± 0.013	0.817 ± 0.015	0.083 ± 0.009	0.529 ± 0.010	0.958 ± 0.018	0.657
RPL [17]	0.908 ± 0.016	0.799 ± 0.036	0.795 ± 0.024	0.009 ± 0.002	0.997 ± 0.003	0.702
ARPL [26]	0.918 ± 0.023	0.804 ± 0.028	0.820 ± 0.021	0.406 ± 0.012	0.996 ± 0.004	0.789
MPF [30]	0.912 ± 0.016	0.786 ± 0.024	0.808 ± 0.025	0.454 ± 0.011	0.997 ± 0.002	0.791
QC-CPN	$\mathbf{0.950} \pm 0.010$	0.862 ± 0.015	0.866 ± 0.020	0.609 ± 0.015	$\mathbf{0.998} \pm 0.002$	**0.857**
Area Under ROC (AUROC)						
Softmax	0.722 ± 0.058	0.794 ± 0.020	0.801 ± 0.023	0.650 ± 0.017	0.732 ± 0.043	0.740
OpenMax [6]	0.707 ± 0.068	0.668 ± 0.018	0.738 ± 0.031	0.553 ± 0.011	0.571 ± 0.037	0.647
CPL [14]	0.770 ± 0.054	0.741 ± 0.011	0.708 ± 0.016	0.631 ± 0.012	0.729 ± 0.033	0.716
GCPL [14]	0.713 ± 0.017	0.730 ± 0.009	0.500 ± 0.000	0.633 ± 0.009	0.795 ± 0.030	0.674
RPL [17]	0.780 ± 0.049	0.760 ± 0.036	0.812 ± 0.024	0.499 ± 0.010	$\mathbf{0.962} \pm 0.022$	0.763
ARPL [26]	0.790 ± 0.046	0.744 ± 0.046	0.799 ± 0.011	0.592 ± 0.027	0.900 ± 0.035	0.765
MPF [30]	0.770 ± 0.048	0.711 ± 0.043	0.814 ± 0.019	0.607 ± 0.021	0.872 ± 0.036	0.755
QC-CPN	$\mathbf{0.818} \pm 0.039$	$\mathbf{0.801} \pm 0.019$	$\mathbf{0.848} \pm 0.025$	$\mathbf{0.668} \pm 0.024$	0.849 ± 0.064	**0.797**
Balanced Accuracy (BACCU)						
Softmax	0.667 ± 0.045	0.728 ± 0.015	0.735 ± 0.026	0.601 ± 0.013	0.676 ± 0.032	0.681
OpenMax [6]	0.647 ± 0.045	0.599 ± 0.011	0.642 ± 0.023	0.431 ± 0.012	0.480 ± 0.030	0.560
CPL [14]	0.709 ± 0.052	0.677 ± 0.009	0.652 ± 0.017	0.590 ± 0.015	0.669 ± 0.024	0.659
GCPL [14]	0.653 ± 0.017	0.669 ± 0.008	0.500 ± 0.000	0.588 ± 0.010	0.725 ± 0.022	0.627
RPL [17]	0.709 ± 0.044	0.703 ± 0.046	0.725 ± 0.027	0.501 ± 0.012	$\mathbf{0.908} \pm 0.035$	0.709
ARPL [26]	0.725 ± 0.046	0.687 ± 0.037	0.733 ± 0.011	0.558 ± 0.022	0.800 ± 0.039	0.701
MPF [30]	0.700 ± 0.040	0.657 ± 0.029	0.745 ± 0.024	0.578 ± 0.013	0.791 ± 0.039	0.694
QC-CPN	$\mathbf{0.744} \pm 0.034$	$\mathbf{0.731} \pm 0.018$	$\mathbf{0.773} \pm 0.033$	$\mathbf{0.620} \pm 0.020$	0.785 ± 0.062	**0.731**

considered other labels as *unknown*. For five trials, 120 random classes were chosen. This dataset can be found at https://www.vision.caltech.edu/datasets/cub_200_2011/.

5) HAM10000

HAM10000 dataset has a collection of roughly 10,000 dermatoscopic images [46]. It has seven classes including Melanocytic nevi (nv), Melanoma (mel), Benign keratosis-like lesions (bkl), Basal cell carcinoma (bcc), Actinic keratoses (akiec), Vascular lesions (vas), and Dermatofibroma (df). We selected 10% of the training dataset for validation. Four classes were selected as *known* and the other three classes were used as *unknown*.

The OSR technique can be beneficial for medical domains which need a rejection mechanism to aid good diagnosis. It will be harmful to the patient if the unknown lesion is classified as a known skin lesion. This dataset serves as a test bed for application of OSR in medical domains. This dataset can be found at https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T.

C. EXPERIMENTATION EVALUATION

We used the best hyperparameters for each method to perform five random trials. We evaluated performance based on the mean and standard deviation of closed-set accuracy, AUROC (known and unknown classes) [38], and balanced accuracy (BACCU) [47]. Closed-set accuracy is important to assess the model's capability in categorizing the samples for known classes. AUROC is used to assess the model's performance in rejecting unknown classes without any dependency on the threshold. Meanwhile, as the proportion of known and unknown is unbalanced, BACCU assesses true performance by fixing the threshold derived from the ROC curve. The BACCU equation is shown in Equation (11), where TP is the number of true positive predictions, P is the total number of samples in the positive class (known class), TN is the number of true negative predictions, and T is a total number of samples in the negative class (unseen class). The results for the best hyperparameters are discussed in Section V-A, and performance in the five random trials is discussed in Section V-B.

$$\frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) \tag{11}$$

We also performed an ablation study to investigate the effect of the proposed method, and this is discussed in Section V-C. Some components of QC-CPN were removed, resulting in the following models:

TABLE 5. Comparison of cost value and best hyperparameters derived from EvoSA by removing some components of QC-CPN.

Dataset	Method	Cost	Hyperparameters			
			γ/Temperature	Feature	Learning Rate	ϵ/λ
Garbage6 *(waste domain)*	QC-S	0.1460	-	987 x4	0.00058	-
	QC-PN	**0.1046**	0.7803	921 x4	0.00281	-
	OC-CPN	0.1417	0.9139	448	0.00438	114.5572
	QC-CPN*	0.1059	0.4406	236 x4	0.00505	281.5650
	Softmax*	0.1621	-	452	0.00036	-
	QC-CPN	0.1110	0.1845	141 x4	0.00531	386.2027
OxfordPet *(pets domain)*	QC-S	0.3312	-	385 x4	0.00382	-
	QC-PN	**0.2218**	0.5971	401 x4	0.06857	-
	OC-CPN	0.2193	0.6719	661	0.00209	166.3667
	QC-CPN*	0.2302	0.4721	533 x4	0.00228	59.2252
	Softmax*	0.3189	-	523	0.00069	-
	QC-CPN	0.2236	0.5213	921 x4	0.04910	79
IndianFood *(food domain)*	QC-S	0.1597	-	996 x4	0.00021	-
	QC-PN	0.1383	0.8934	534 x4	0.00123	-
	OC-CPN	0.1776	0.7300	429	0.03622	21.1563
	QC-CPN*	0.1559	1	541 x4	0.00410	65
	Softmax*	0.2125	-	633	0.00094	-
	QC-CPN	**0.1255**	0.3942	884 x4	0.00355	65.7975
CUB 200 *(bird species domain)*	QC-S	0.2600	-	517 x4	0.00014	-
	QC-PN	0.2096	0.2886	927 x4	0.03208	-
	OC-CPN	0.2394	0.3870	804	0.02321	36.1099
	QC-CPN*	0.2008	0.5641	634 x4	0.00477	73.1155
	Softmax*	0.3855	-	1000	0.000001	-
	QC-CPN	**0.1937**	0.3529	827 x4	0.00482	45.9587
HAM10000 *(skin lesion domain)*	QC-S	0.1627	-	288 x4	0.00119	-
	QC-PN	0.0761	0.0998	611 x4	0.05230	-
	OC-CPN	0.0746	0.3722	859	0.01646	100
	QC-CPN*	**0.0505**	0.2284	544 x4	0.04585	257.1219
	Softmax*	0.1560	-	951	0.00056	-
	QC-CPN	0.0698	0.1053	579 x4	0.06212	305.2672

- Quad-channel with a Softmax layer **(QC-S)**: The prototypes and contrastive prototype loss were removed and the quad-channel was used. This model was designed to examine performance without prototypes and revealed the effect of the quad-channel based on the geometric transformation.
- Quad-channel prototype without contrastive learning **(QC-PN)**: This model did not use contrastive prototype loss and was meant to examine the effect of the $\mathcal{L}_2$ in QC-CPN.
- One-channel contrastive prototype networks **(OC-CPN)**: This model was designed to examine the effect of the quad-channel using only one channel perspective without a rotation feature.
- Quad-channel with shared parameters for all partial features **(QC-CPN*)**: This QC-CPN model shared all parameters for all orientations of the inputs. The model observed the effect of parameter separation between the original image and rotated images.
- Softmax with rotated data augmentation **(Softmax*)**: The standard deep learning model with data augmentation was compared, and the effect of concatenated features was observed and compared to the rotation with data augmentation (using the same rotation). The augmentation increased the total samples of data training four times.

The effect of our proposed contrastive prototype loss and different rotation degree were also studied and were discussed in Section V-D and V-E respectively.

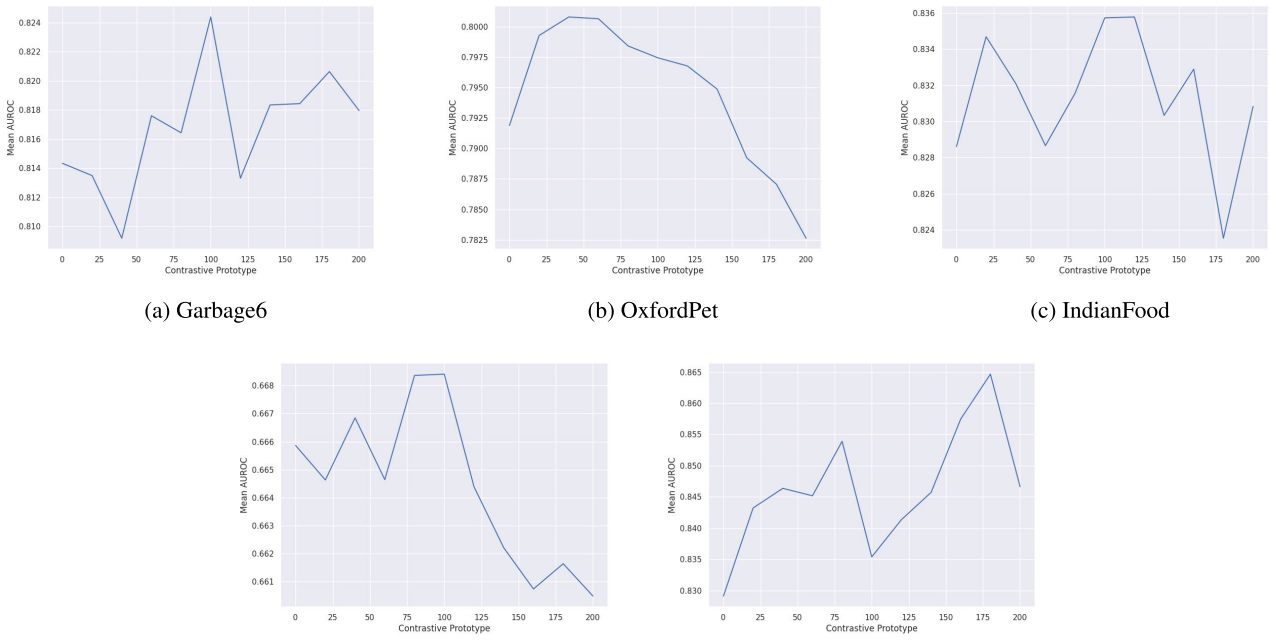
In contrastive prototype loss, we tuned the different values of ϵ from 0 to 200 in steps of 20 and analyzed the AUROC performance over five domain-specific datasets with five different combinations of classes similar to the five random trials. Using the same combinations, we also observe the effect of different rotation degrees, which are 30, 60, and 90 degrees that are used to generate additional images for quad-channel features.

V. RESULTS AND DISCUSSION

A. OPTIMIZED HYPERPARAMETERS

The important hyperparameters for each algorithm were optimized using EvoSA to maximize the performance of discrimination between known and unknown classes via the AUROC metric. The best hyperparameters with the best costs are presented in Table 2. Using this hyperparameter will give us insight into the actual performance of each OSR algorithm in domain-specific tasks.

Overall, QC-CPN has the least cost across domain-specific datasets (except HAM10000). In the Garbage6 dataset, the MPF provides a competitive cost compared to QC-CPN. In OxfordPet, most methods yield a competitive performance in which RPL has roughly 0.01 lower performance



(a) Garbage6 (b) OxfordPet (c) IndianFood

(d) CUB-200 (e) HAM10000

FIGURE 3. The effect of contrastive prototype parameters over different domains and values.

TABLE 6. Performance over ablation study scenarios of QC-CPN with five random trials.

Method	Garbage6 *(waste)*	OxfordPet *(pets)*	IndianFood *(food)*	CUB-200 *(bird species)*	HAM10000 *(skin lesion)*	Mean
Closed-Set Accuracy						
QC-S	0.928 ± 0.012	0.833 ± 0.010	$\mathbf{0.900} \pm 0.010$	0.585 ± 0.012	0.997 ± 0.003	0.848
QC-PN	0.947 ± 0.009	$\mathbf{0.863} \pm 0.013$	0.866 ± 0.024	0.583 ± 0.018	$\mathbf{0.998} \pm 0.002$	0.851
OC-CPN	0.920 ± 0.014	0.839 ± 0.018	0.817 ± 0.023	0.525 ± 0.016	$\mathbf{0.998} \pm 0.003$	0.820
QC-CPN*	0.943 ± 0.019	0.849 ± 0.015	0.854 ± 0.022	0.591 ± 0.019	0.997 ± 0.002	0.847
Softmax*	0.922 ± 0.012	0.739 ± 0.021	0.856 ± 0.013	0.273 ± 0.012	0.996 ± 0.002	0.757
QC-CPN	$\mathbf{0.950} \pm 0.010$	0.862 ± 0.015	0.866 ± 0.020	$\mathbf{0.609} \pm 0.015$	$\mathbf{0.998} \pm 0.002$	**0.857**
Area Under ROC (AUROC)						
QC-S	0.741 ± 0.067	0.740 ± 0.023	0.818 ± 0.014	0.637 ± 0.018	0.798 ± 0.079	0.747
QC-PN	0.809 ± 0.054	0.794 ± 0.018	0.842 ± 0.029	0.658 ± 0.023	0.852 ± 0.062	0.791
OC-CPN	0.774 ± 0.039	0.777 ± 0.018	0.810 ± 0.019	0.651 ± 0.021	$\mathbf{0.874} \pm 0.062$	0.777
QC-CPN*	0.805 ± 0.051	0.794 ± 0.019	0.836 ± 0.011	0.656 ± 0.022	0.837 ± 0.062	0.786
Softmax*	0.701 ± 0.057	0.674 ± 0.011	0.779 ± 0.010	0.536 ± 0.021	0.751 ± 0.055	0.688
QC-CPN	$\mathbf{0.818 \pm 0.039}$	$\mathbf{0.801} \pm 0.019$	$\mathbf{0.848} \pm 0.025$	$\mathbf{0.668} \pm 0.024$	0.849 ± 0.064	**0.797**
Balanced Accuracy (BACCU)						
QC-S	0.691 ± 0.054	0.683 ± 0.017	0.751 ± 0.021	0.591 ± 0.011	0.732 ± 0.077	0.690
QC-PN	$\mathbf{0.753} \pm 0.049$	0.728 ± 0.015	0.762 ± 0.028	0.616 ± 0.022	0.784 ± 0.060	0.729
OC-CPN	0.714 ± 0.021	0.711 ± 0.016	0.738 ± 0.020	0.606 ± 0.013	$\mathbf{0.811} \pm 0.065$	0.716
QC-CPN*	0.742 ± 0.043	0.726 ± 0.013	0.752 ± 0.012	0.610 ± 0.016	0.768 ± 0.056	0.720
Softmax*	0.658 ± 0.043	0.625 ± 0.007	0.711 ± 0.016	0.525 ± 0.014	0.691 ± 0.076	0.642
QC-CPN	0.744 ± 0.034	$\mathbf{0.731} \pm 0.018$	$\mathbf{0.773} \pm 0.033$	$\mathbf{0.620} \pm 0.020$	0.785 ± 0.062	**0.731**

than QC-CPN. We observed that the MPF had the worst performance, which was different when applied to Garbage6. In IndianFood and CUB 200, QC-CPN produced significant results in terms of cost value compared to other methods. In HAM10000, the least cost was achieved by RPL, while Softmax had the highest cost. QC-CPL has the second-best performance results in all comparison methods. Further investigation is needed in this domain, as the problem is potentially due to the prototype derived from the rotation of the quad-channel input, which may not help in the related domain.

From the observed best hyperparameter found via EvoSA, the proposed loss in GCPL and the MPF are not essential for maximizing the AUROC score. Parameter λ in the MPF in the Garbage6 dataset was 0, representing the coefficient to control the motorial margin constraint, which was considered ineffectual in gaining optimal AUROC. The same hyperparameter value for λ was found in GCPL on OxfordPet and

CUB 200 datasets. This represents the weight of generative loss that pushes the representation of the embedding sample close to its prototype. We argue that this parameter is ineffective in constructing a better representation for bounding the open-set samples if used with DCE because it behaves the same way. Minimizing DCE will attract the embedding sample close to its prototype, similar to the λ or prototype loss in GCPL.

In the IndianFood dataset, GCPL needed zero value for λ in DCE (discriminative loss), which is considered unnecessary for rejecting unseen classes. This suggest that the generative approach may help identify unseen classes rather than discriminative ones, as multi-class classification and identifying unseen classes are probably not directly related. One possible approach for future work is to consider multi-task learning for accuracy and rejection, which can be seen as two different tasks with shared parameters in the neural networks that utilize an advanced generative model to help identify unseen classes.

Using the same backbone (MNetV3-Large) as a feature extractor and the best hyperparameters, Softmax provides competitive results with other algorithms that were underestimated in the literature [12], [26], [30], [37]. In Garbage6, CPL and GCPL are inferior to Softmax, and in the OxfordPet dataset, CPL, ARPL, and the MPF have higher costs than Softmax. In IndianFood, only CPL is worse than Softmax. Furthermore, in CUB 200, most algorithms are competitive with Softmax except QC-CPN. This shows that in real-world applications in domain-specific tasks using transfer learning, Softmax is still considered a good baseline.

B. OSR PERFORMANCE

The statistical score of OSR performance for each algorithm in the five random trials is shown in Table 4. In terms of closed-set accuracy, Softmax had the highest score in three datasets (OxfordPet, IndianFood, and CUB-200). OpenMax had the same score due to meta recognition that utilized the trained Softmax-based model. GCPL had the lowest score in the IndianFood dataset because the γ parameters had zero weight, which avoided the optimization of cross-entropy loss. Overall, QC-CPN had the highest mean of closed-set accuracy.

In terms of the AUROC score, the proposed QC-CPN produced the highest value in the most domain-specific dataset as shown in the mean value. In HAM10000, RPL gave its peak performance in distinguishing known and unknown classes. For OpenMax, the score for all datasets was lower compared to the original model in Softmax; therefore, meta recognition via Weibull distribution was not as effective and actually reduced the original performance.

Regarding the BACCU score, QC-CPN was superior compared to other methods in the most datasets, and Softmax produced a competitive score. As shown in the mean column in Table 4, QC-CPN has a significantly higher score compared to other open-set methods. Furthermore, Softmax had a competitive baseline and a higher score than the popular OpenMax, which significantly reduced the performance of the canonical Softmax in the OxfordPet, IndianFood, CUB-200, and HAM10000 datasets by roughly 0.13, 0.09, 0.17, and 0.20, respectively. This shows that OpenMax is not a suitable replacement for Softmax when rejecting unseen classes in domain-specific tasks. We suggest reconsidering OpenMax as a good baseline when dealing with domain-specific tasks using a pre-trained model.

C. ABLATION STUDY

The best hyperparameters of ablated QC-CPN and Softmax with data augmentation (Softmax*) are shown in Table 5. Softmax* had the highest cost in most datasets (Garbage6, IndianFood, and CUB 200) compared to other components. Increasing data using different orientation does not significantly increase the capability of rejection. Furthermore, in the OxfordPet and CUB-200, Softmax* underperformed by approximately 0.07 and 0.09, respectively, compared to standard Softmax.

Extending the input model by quad-channel in Softmax (QC-S) increased the AUROC through lower cost results in most datasets with the best hyperparameters. Moreover, the cost could be reduced by changing the output of QC-S with prototype learning (QC-PN). We also observed that using one channel only (OC-CPN) had a higher cost than QC-PN in most datasets. In OxfordPet and HAM10000, the differences were insignificant.

QC-CPN* had a competitive cost compared to QC-PN in the Garbage6 and Oxfordpet datasets. In this case, contrastive prototype loss with shared parameters for all partial features did not provide significant results. In IndianFood, it had the opposite effect and had higher costs compared to QC-PN. However, in the CUB-200 and HAM1000 datasets, QC-CPN produced a lower cost compared to QC-PN. Meanwhile, QC-CPN had competitive costs in all datasets, with the lowest costs in the IndianFood and CUB-200 datasets.

Table 6 presents the performance of five random combinations of known and unknown classes. These results reveal the excellent performance achieved by QC-CPN for all datasets. For all metrics, QC-CPN performed the best in most datasets, as shown in the mean column. The performance varies for closed-set accuracy, and no single method performed the best on most dataset. From these results, Softmax* produced the worst performance, showing that data augmentation alone cannot substantially improve performance.

D. THE EFFECT OF CONTRASTIVE PROTOTYPE PARAMETER (ϵ)

As shown in Figure 3, the value of ϵ effected the performance of rejecting unseen classes. On all datasets, using $\epsilon = 0$ for the model revealed that the mean AUROC score was not the best performance. There exist a good performance when $\epsilon > 0$. We also observed that incorrectly settings the parameter values reduced performance compared to without using contrastive prototype loss.

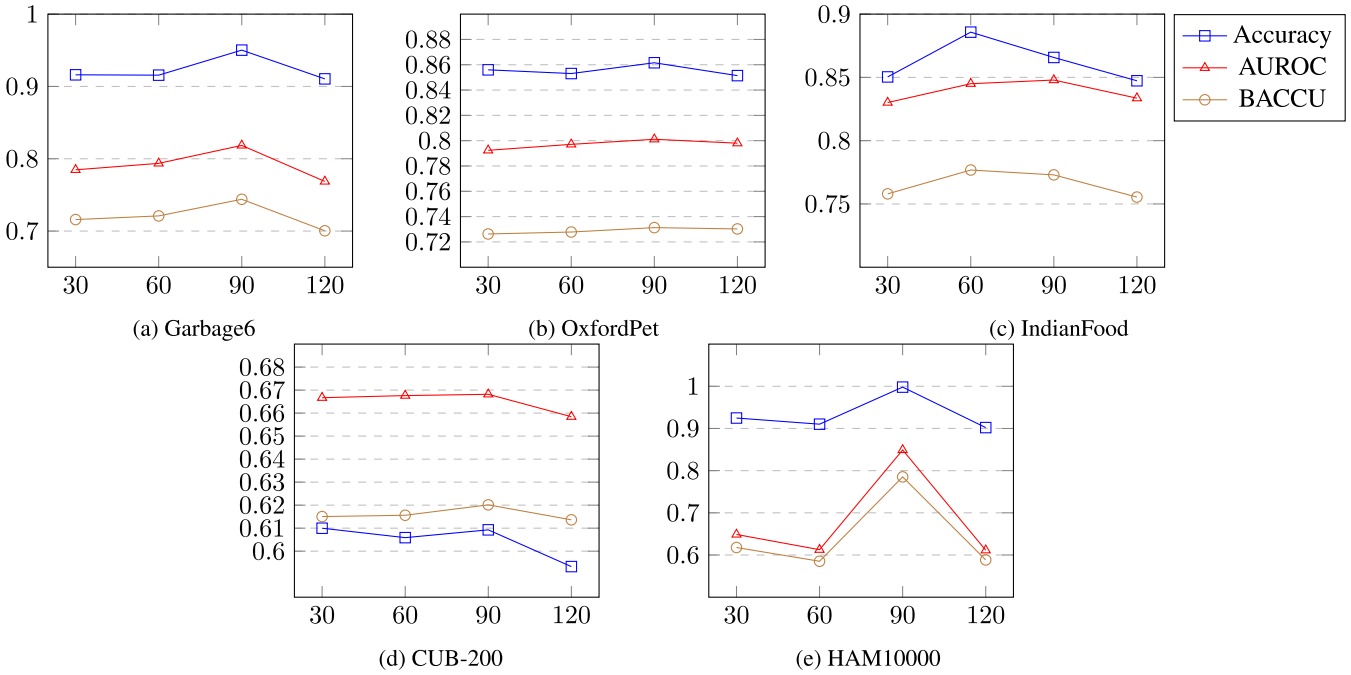


FIGURE 4. The effect of different degrees of rotation on the different datasets in QC-CPN. The horizontal axis shows the different degrees while the vertical axis shows the performance of different metrics at the same scale (higher is better).

In Garbage6, $\epsilon \approx 100$ was a good parameter. After this value, the mean AUROC score fell at $\epsilon = 120$ and increased afterward. In OxfordPet, ϵ between 40 and 60 provided a good mean AUROC, and as the number increased, the performance gradually decreased. In this dataset, the high value of ϵ significantly reduced the capability of rejecting unseen classes, as shown in the Figure at $\epsilon = 200$. In the IndianFood dataset, ϵ between 100 and 120 provided the best score. Increasing the value over 120 would potentially reduce the model's performance. Meanwhile, in CUB-200, the best performance was clearly shown when ϵ was between 80 and 100. Increasing this value over 120 drastically reduced the performance. In HAM10000, $\epsilon = 180$ performed well compared to the others. With this trend, there is a possibility of better performance with $\epsilon > 200$.

E. ROTATION EFFECT

The plot between rotation degree and different metrics (closed-set accuracy, AUROC, and BACCU) is shown in Figure 4. This degree is used to synthesize the additional three images, e.g., for 90 degrees, the images will be rotated with 90, 180, and 270 degrees.

For the most dataset, it was observed that 90 degrees provide the highest score in most dataset. In IndianFood dataset, 60 degrees provide the highest performance for all metrics. These empirical results indicate the importance of the choice of rotation degree for different cases.

The figure also shows that generating additional features from rotation could enhance the performance of the models. With cautions that the choice of rotation degree is important. In the Garbage6 dataset, setting 120 degrees as the basis could not provide higher performance than using without the quad-channel feature, no additional features, (OC-CPN) as well as the HAM1000 dataset. Further study is necessary to assess whether the choice of rotation degree is similar in the same domain or not. Wrongly set the rotation degree potentially could reduce the performance.

VI. CONCLUSION AND FUTURE STUDY

This paper presents a simple yet effective technique to address the open-set problem in real-world applications in domain-specific tasks. We propose using multiple views at the input level to derive a strong representation through the quad-channel by geometric rotation with contrastive prototype loss. The extensive experimentation with hyperparameter optimization using evolutionary simulated annealing and practical approaches using transfer learning with the same backbone reveals that our approach outperforms state-of-the-art techniques in terms of rejecting unseen classes. Quad-channel contrastive prototype networks could become a strong baseline for domain-specific tasks in the future. Our approach needs contrastive prototype parameters to be tuned first to obtain peak performance.

For future study, we will explore multiple views with contrastive learning on the prototype levels and data levels of the feature vector, which may improve open-set performance. Additionally, recognizing unseen classes in domain-specific problems will be a good direction for future work. Recognizing two different unseen classes in the domain and out-of-domain interests in deep neural networks could

improve the model to learn continually with a relevant training set.

REFERENCES

[1] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers, and N. Houlsby, "Scaling vision with parse mixture of experts," in *Advances in Neural Information Processing Systems*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. New York, NY, USA: Curran, 2021, pp. 8583–8595. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/48237d9f2dea8c74c2a72126cf63d933-Paper.pdf

[2] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1204–1213.

[3] 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.

[4] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecol. Informat.*, vol. 48, pp. 257–268, Nov. 2018.

[5] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "LeafNet: A computer vision system for automatic plant species identification," *Ecological Informat.*, vol. 40, pp. 50–56, Jul. 2017.

[6] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1563–1572.

[7] W. J. Scheirer, A. D. R. Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[8] T. E. Boult, S. Cruz, A. Dhamija, M. Gunther, J. Henrydoss, and W. Scheirer, "Learning and the unknown: Surveying steps toward open world recognition," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 9801–9807. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5054

[9] Z. Ge, S. Demyanov, and R. Garnavi, "Generative OpenMax for multi-class open set classification," in *Proc. Brit. Mach. Vis. Conf.*, G. T.-K. Kim, S. Zafeiriou and K. Mikolajczyk, Eds. 2017, p. 42, doi: 10.5244/C.31.42.

[10] C. Geng, S. 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.

[11] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 613–628.

[12] 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.

[13] P. Schlachter, Y. Liao, and B. Yang, "Deep open set recognition using dynamic intra-class splitting," *Social Netw. Comput. Sci.*, vol. 1, no. 2, p. 77, Mar. 2020.

[14] H. Yang, X. Zhang, F. Yin, and C. Liu, "Robust classification with convolutional prototype learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3474–3482.

[15] 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.

[16] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4399–4408.

[17] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *Proc. ECCV*, 2020, pp. 507–522.

[18] Y. Shu, Y. Shi, Y. Wang, T. Huang, and Y. Tian, "P-ODN: Prototype-based open deep network for open set recognition," *Sci. Rep.*, vol. 10, no. 1, p. 7146, Apr. 2020.

[19] L. Shu, H. Xu, and B. Liu, "DOC: Deep open classification of text documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2911–2916. [Online]. Available: https://aclanthology.org/D17-1314

[20] S. Sachar and A. C. Kumar, "Automatic plant identification using transfer learning," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1022, Jan. 2021, Art. no. 012086.

[21] G. VijayaKumari, P. Vutkur, and P. Vishwanath, "Food classification using transfer learning technique," *Global Transitions Proc.*, vol. 3, no. 1, pp. 225–229, Jun. 2022.

[22] Q. Zhang, Q. Yang, X. Zhang, Q. Bao, J. Su, and X. Liu, "Waste image classification based on transfer learning and convolutional neural network," *Waste Manag.*, vol. 135, pp. 150–157, Nov. 2021.

[23] G. A. F. Alfarisy, O. A. Malik, and O. W. Hong, "Evolutionary simulated annealing for transfer learning optimization in plant-species identification domain," in *Proc. IEEE 9th Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Jun. 2022, pp. 1–6.

[24] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020, doi: 10.3390/technologies9010002.

[25] N. K. Logothetis and D. L. Sheinberg, "Visual object recognition," *Annu. Rev. Neurosci.*, vol. 19, no. 1, pp. 577–621, Mar. 1996. [Online]. Available: https://www.annualreviews.org/doi/10.1146/annurev.ne.19.030196.003045

[26] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8065–8081, Nov. 2022.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[28] M. Thoma, "The HASYv2 dataset," 2017, *arXiv:1701.08380*.

[29] 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.

[30] Z. Xia, P. Wang, G. Dong, and H. Liu, "Adversarial motorial prototype framework for open set recognition," 2021, *arXiv:2108.04225*.

[31] N. Saunshi, J. Ash, S. Goel, D. Misra, C. Zhang, S. Arora, S. Kakade, and A. Krishnamurthy, "Understanding contrastive learning requires incorporating inductive biases," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds. Jul. 2022, pp. 19250–19286. [Online]. Available: https://proceedings.mlr.press/v162/saunshi22a.html

[32] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. SIMBAD*, 2015, pp. 84–92.

[33] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1335–1344.

[34] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 459–474.

[35] Y. Gao, N. Fei, G. Liu, Z. Lu, and T. Xiang, "Contrastive prototype learning with augmented embeddings for few-shot learning," in *Proc. 37th Conf. Uncertainty Artif. Intell.*, vol. 161, C. De Campos and M. H. Maathuis, Eds. Jul. 2021, pp. 140–150. [Online]. Available: https://proceedings.mlr.press/v161/gao21a.html

[36] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," 2020, *arXiv:2005.04966*.

[37] Y. Kodama, Y. Wang, R. Kawakami, and T. Naemura, "Open-set recognition with supervised contrastive learning," in *Proc. 17th Int. Conf. Mach. Vis. Appl. (MVA)*, Jul. 2021, pp. 1–5.

[38] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[39] R. Cao, Q. Zhang, J. Zhu, Q. Li, Q. Li, B. Liu, and G. Qiu, "Enhancing remote sensing image retrieval with triplet deep metric learning network," 2019, *arXiv:1902.05818*.

[40] A. Thakur, D. Thapar, P. Rajan, and A. Nigam, "Multiscale CNN based deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss," 2019, *arXiv:1903.10713*.

[41] G. Kertesz, "Different triplet sampling techniques for lossless triplet loss on metric similarity learning," in *Proc. IEEE 19th World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2021, pp. 000449–000454. [Online]. Available: https://ieeexplore.ieee.org/document/9378628/

[42] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Oulon, France, Apr. 2017, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=Hkg4TI9xl

[43] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[44] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3498–3505.
[45] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD birds 200," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., CNS-TR-2010-001, 2010.
[46] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, no. 1, pp. 1–9, Aug. 2018.
[47] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.

GUSTI AHMAD FANSHURI ALFARISY received the B.C.S. and M.C.S. degrees from Brawijaya University, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Digital Science, Universiti Brunei Darussalam. He is a Lecturer with the Department of Informatics, Institut Teknologi Kalimantan, Indonesia. His research interests include applied machine learning, deep learning, neural architecture search, open-world lifelong machine learning, computer vision, and natural language processing.

OWAIS AHMED MALIK received the B.E. degree in computer systems engineering from NED, Pakistan, in 1998, the M.S. degree in computer science from KFUPM, Saudi Arabia, in 2002, and the Ph.D. degree in computer science from Universiti Brunei Darussalam (UBD), in 2015. He is currently a Senior Assistant Professor with the School of Digital Science, UBD. He has more than ten years of progressive experience in academia and research in the field of computer science and engineering. He has been teaching various undergraduate courses, including machine learning, data mining, machine perception, programming fundamentals, design and analysis of algorithms, software engineering, and operating systems in different national/international universities. He has published a number of papers in internationally reputable journals and conferences. His research interests include designing and exploring different intelligent/pattern recognition algorithms for the analysis and classification of biodiversity and cybersecurity data, applied biomechanics, bio-signal processing, and big data analytics.

ONG WEE HONG joined Universiti Brunei Darussalam (UBD), in 2007, where he is currently an Assistant Professor of computer science. Before joining UBD, he was with the Jefri Bolkia College of Engineering, from 1998 to 2007. He is also leading the Robotics and Intelligent Systems Laboratory (Robolab). In particular, he is exploring the application of artificial intelligent techniques and info-communication technologies in developing intelligent systems. His research interests include the development of intelligent systems for personal robots, ambient intelligence, unsupervised human activities recognition, storage cloud-based IoT systems, mobile robot navigation, human emotion perception for human–robot interaction, and self-supervised object recognition.