applied sciences

MDPI

Article

Effect of Probabilistic Similarity Measure on Metric-Based Few-Shot Classification

Youngjae Lee and Hyeyoung Park *

School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; leeyj2711@gmail.com
* Correspondence: hypark@knu.ac.kr

Abstract: In developing a few-shot classification model using deep networks, the limited number of samples in each class causes difficulty in utilizing statistical characteristics of the class distributions. In this paper, we propose a method to treat this difficulty by combining a probabilistic similarity based on intra-class statistics with a metric-based few-shot classification model. Noting that the probabilistic similarity estimated from intra-class statistics and the classifier of conventional few-shot classification models have a common assumption on the class distributions, we propose to apply the probabilistic similarity to obtain loss value for episodic learning of embedding network as well as to classify unseen test data. By defining the probabilistic similarity as the probability density of difference vectors between two samples with the same class label, it is possible to obtain a more reliable estimate of the similarity especially for the case of large number of classes. Through experiments on various benchmark data, we confirm that the probabilistic similarity can improve the classification performance, especially when the number of classes is large.

Keywords: few-shot classification; metric-learning; probabilistic similarity; intra-class statistics

Citation: Lee, Y.; Park, H. Effect of Probabilistic Similarity Measure on Metric-Based Few-Shot Classification. *Appl. Sci.* **2021**, *11*, 10977. https://doi.org/10.3390/app112210977

Academic Editors: Andrea Prati, Carlos A. Iglesias, Vincent A. Cicirello and Luis Javier García Villalba

Received: 9 October 2021
Accepted: 15 November 2021
Published: 19 November 2021

1. Introduction

Pattern recognition methods using deep learning techniques have shown good results in many applications [1–5]. However, these results can only be obtained with a sufficiently large number of training data. Unlike the conventional deep learning models, humans can classify patterns with only a small number of samples. In order to realize this ability in a deep learning model, studies on few-shot learning have been attracting attention recently [6–8].

In few-shot learning for classification tasks, a classifier is required to recognize classes that are unseen in the learning phase, with a very limited number of samples. To achieve this goal, there have been proposed a number of few-shot classification models which are composed of two modules: an embedding module and a classification module [8–12]. The embedding module extracts appropriate features through mapping given input to an embedding space, and the classification module tries to classify newly given samples (query data) with only a few training samples (support data) by using the features from the embedding module. Since a new learning strategy, called episodic learning [8], was proposed to obtain a good embedding function under the few-shot scenario in [8], most of the subsequent works have mainly focused on designing a good embedding function model that can provide an efficient and general representation for recognizing unseen test classes [13–17].

On the other hand, there has been relatively little interest in the classification module where linear classifiers or simple distance-based classifiers have been mainly adopted [18–20]. This approach seems to be appropriate for the situation that the number of labeled samples for test classes is very limited because classifiers with high complexity can be easily overfitted to the few given samples. Although computational experiments have shown that a simple classifier combined with a well-generalized embedding module can achieve

good performance [21], there would be further room for improvement in the classification module [22–24]. In this paper, we try to improve the performance of the few-shot classifiers by elaborating on the distance-based classification module.

It is well-known that the accuracy of a distance-based classifier can be improved by using a statistical measure such as Mahalanobis distance [25] rather than simple geometric measures such as Euclidean distance. Under the situation of few-shot classification, however, it is difficult to utilize the statistical measure because the estimation of accurate distribution is hard due to the extremely limited number of samples. Inaccurate estimation on the class distributions lead to poor distance measure, resulting in low classification accuracy.

To overcome this limitation, we propose to combine the probabilistic similarity based on intra-class statistics [26–29] with the prototypical network [9] that is a representative few-shot classification model. In [26,27], the probabilistic similarity between two samples is defined as a probability that they belong to the same class, and its probability density function is estimated under the assumption that a data point is generated from two factors: a class-specific factor and a class-independent factor. The class-specific factor can be represented by a prototype vector that is defined as the mean vector of support samples in the prototypical network [9]. The class-independent factor can be considered as an environmental factor that is irrelevant to each class, and thus can be estimated through an episodic learning strategy developed for few-shot learning [8,9,30].

Based on these considerations, we develop a method for applying the probabilistic similarity measure in the learning of embedding function as well as in the recognition of unseen classes. Moreover, by exploiting the similarity in learning of the embedding function as well, it is also expectable to obtain better feature representation, which is more suitable to the assumption on the data distribution for the prototype-based classifier. Additionally, since the distribution of the class-independent factor can be estimated more accurately as the number of classes increases even when each class has few samples, the proposed method is expected to be more effective in the case of a large number of classes. This can be an advantage of the proposed method, which cannot be expected from the conventional works using Euclidean distance.

The aim and main contributions of our work are summarized below:

- In order to improve the performance of few-shot classification, we propose to combine the probabilistic similarity measure with deep embedding function networks.
- We define an explicit function of probabilistic similarity based on the intra-class statistics and propose a modified episodic learning algorithm that simultaneously performs estimation of the similarity and optimization of the embedding function.
- Whereas the conventional methods have been tested for a limited number of classes, we evaluate the change of performance as the number of classes increases, and confirm the apparent superiority of the proposed method, especially in the case of many classes.
- Although we adopted the prototypical network for the experiments, the proposed method is not constrained by the embedding network model, and thus it can be extended to various forms using more sophisticated deep network models.

In Section 2, we describe the few-shot classification problem and briefly review the previous works to solve it, focusing on the metric-based method. In Section 3, we explain the probabilistic similarity measure used in our proposed method, and the overall process of the proposed method is described in Section 4. Section 5 presents experimental results on benchmark datasets comparing its performance with the existing methods. Our conclusion is made in Section 6.

2. Few-Shot Classification Problem

The few-shot classification task is used to classify newly given samples by using an extremely limited number of training samples. The set of given training samples is called support set S, and the set of new samples to be classified is called query set Q. Usually, we consider the N-way K-shot problem, where the number of classes is N and the number of

support samples per class is K. Since the value of K is very small, it is difficult to obtain a good classification model with only support samples. Therefore, in order to develop a deep learning model for few-shot classification, it is normal to use a separate dataset that is in the same domain but has completely distinct class labels. In this approach, the main goal of the learning is to find a deep learning model that can recognize query samples from the new test classes with only a few support samples.

One of the representative methods for achieving this goal is the metric-based method which tries to find an appropriate metric for classification [8–10,12]. As shown in Figure 1, the overall structure of the metric-based method is largely composed of two modules: the embedding network and the few-shot classifier. During the learning phase, a deep network model learns to find an embedding function that maps raw inputs to feature vectors on the embedding metric space. The few-shot classifier then predicts the class of query data based on their similarity to the support data, which is measured on the metric space. The loss from the classifier is then transmitted for learning of the embedding network.

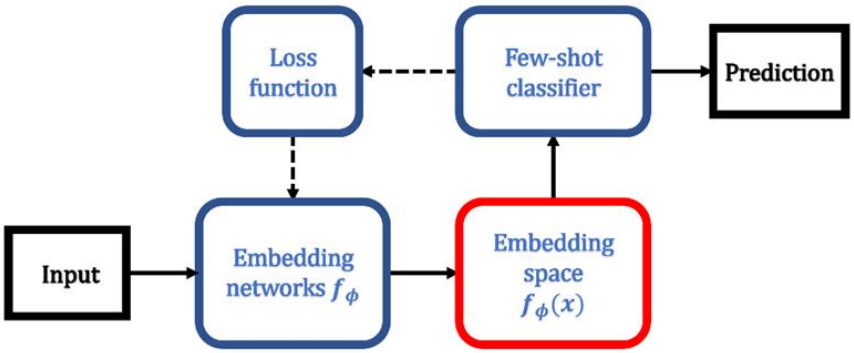

Figure 1. The overall structure of the metric-based few-shot classification model.

Unlike the usual metric-learning problem, the few-shot learning task assumes that the classes in the test phase are unseen during the learning phase, and thus it is important to acquire a good embedding function that can generally apply to unseen test classes. To address this problem, the matching networks [8] introduced the episodic learning strategy, which is one of the meta-learning techniques. In the episodic learning strategy, subsets in the form of an N-way K-shot classification are generated by random selections from the whole training data, which are called episodes. The neural network model updates the network parameters by learning one episode at a time. By proceeding through numerous episodes, the model learns a variety of cases composed of various classes and samples. In this way, the model is not limited to the given classes but can learn more generally about the domain of the classes. That is, information about unseen classes is obtained by the use of various combinations of classes that share some common factors of the domain.

Based on the episodic learning strategy, Snell et al. proposed the prototypical network [9], which combines a nonlinear embedding function network and a simple distance-based classifier. Under the assumption that there exists an embedding space on which samples from each class are clustered around a single prototype, it tries to find a good embedding network through episodic learning. Once a good embedding space is found, the classification is conducted by simply finding the nearest class prototype defined as the mean of the support samples. The Euclidean distances between queries and the prototype on the embedding space are used for defining the loss function for learning of the embedding network as well, so as to create an embedding space where samples from each class are gathered near its prototype.

Since the prototypical network [9] has shown better performance than more complicated few-shot learning models [6–8], there have been a number of extensions based on the same structure shown in Figure 1 [10,13–15]. Sung et al. [10] added the relation module after the embedding module for more fine-grained classification. Li et al. [13] proposed to use local features as additional information to image-level features in the embedding

module. Based on the prototypical networks, Wertheimer et al. [14] use a concatenation of foreground and background vector representations as feature vectors. Kim et al. [15] introduced the variational autoencoder (VAE) structure [31] into the embedding module for training the prototype images.

These works focused on obtaining a good embedding network and there has not been much interest in the classification module. This is primarily due to the limitations of the few-shot classification task. Even though it is known that other alternatives such as Mahalanobis distance [25] can be adopted instead of the simple deterministic distance, it is difficult to apply the statistical distance, because it needs distribution information of samples which are not sufficiently given in the few-shot classification tasks.

As an attempt to overcome these difficulties, Fort [22] tried confidence region estimation in the embedding space in the form of a Gaussian covariance matrix and used it to construct metrics. Liu et al. [24] proposed to use of a new metric learning formula based on Mahalanobis distance [25] to avoid the tendency to overfit the training class. However, these methods still have difficulty in estimating the covariance matrix of each class under the few-shot setting. Li et al. [23] proposed a method relatively free from this problem by defining a local covariance that is obtained from local features in the embedding modules. Though this method shows the efficiency of the second-order statistics, it needs a more complex classifier with specially designed metrics.

In this paper, we try to find some possibility of improving the classifier by using a probabilistic similarity based on the intra-class statistics, which can be estimated robustly especially when the number of classes increases. Unlike [20], our proposed method does not depend on the structure of the embedding module, and it can be combined with the original prototypical network as well as other sophisticated models.

3. Probabilistic Similarity Measure

In the probabilistic similarity measure [26–29], the similarity is defined as the probability that two data $\boldsymbol{x}$ and $\boldsymbol{x}'$ belong to the same class $\boldsymbol{c}_k$, which can be written as:

$$S(\boldsymbol{x}, \boldsymbol{x}') = prob\left[\boldsymbol{x} \in \boldsymbol{c}_k, \boldsymbol{x}' \in \boldsymbol{c}_k\right]. \tag{1}$$

An explicit function of the probability can be obtained by defining a generation model of data $\boldsymbol{x}$ with two components [28]: class component $\boldsymbol{c}$ and environmental component $\boldsymbol{\varepsilon}$ such as:

$$\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{\varepsilon} \tag{2}$$

The environmental components $\boldsymbol{\varepsilon}$ originates from some environmental variations such as illuminations and is assumed to be independent of the class source. On the contrary, the class component $\boldsymbol{c}$ is originated from a class-specific source determined differently for each class.

Additionally, Ref. [28] further assumes that the class-specific component $\boldsymbol{c_k}$ for each class $\boldsymbol{c}_k$ can be regarded as a unique prototype, and the intra-class variations are caused by the environmental component $\boldsymbol{\varepsilon}$. The environmental component is also assumed to be independent of the class and be identical regardless of the class-label. Although these assumptions may be considered rather strict for application in real data, they are consistent with the assumption placed on the classifier of the prototypical network. More precisely, the simple classifier used in the prototypical network can be regarded as a particular case of the distance-based classifier using the probabilistic similarity used in [26–29].

In order to obtain an explicit form of the probabilistic similarity, let us consider a difference vector between two samples $\boldsymbol{x}$ and $\boldsymbol{x\prime}$ belonging to a single class $\boldsymbol{c}_k$, which can be written as:

$$\boldsymbol{\delta} = \boldsymbol{x} - \boldsymbol{x}' = (\boldsymbol{c_k} + \boldsymbol{\varepsilon}) - (\boldsymbol{c_k} + \boldsymbol{\varepsilon}') = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}'. \tag{3}$$

By subtracting two vectors belonging to the same class, we can infer that the class-specific component disappears and only the environmental component $\boldsymbol{\varepsilon}$ remains in the

intra-class difference vector. Then, the probability that the two data belong to the same class can be obtained by estimating the probability density function $p(\boldsymbol{\delta})$.

According to the assumption in [28], all classes have the same environmental component $\boldsymbol{\varepsilon}$. Therefore, all of the difference vectors will follow a single distribution regardless of class-labels. We can specify the distribution using this set of difference vectors and the characteristics of the environmental component $\boldsymbol{\varepsilon}$. Noting that the environmental component $\boldsymbol{\varepsilon}$ is caused by diverse sources, we can assume that $\boldsymbol{\varepsilon}$ is subject to Gaussian distribution, and so does the difference vector $\boldsymbol{\delta} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}'$.

In order to estimate the mean and variance of Gaussian pdf $p(\boldsymbol{\delta})$, we compose the set of intra-class difference vectors Ω using support samples, which can be defined as:

$$\Omega = \left\{ \boldsymbol{\delta}_{ij} \middle| \, \boldsymbol{\delta}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j, \, \boldsymbol{x}_i, \, \boldsymbol{x}_j \in \boldsymbol{c}_k, \, k = 1, \, \ldots, \, N \right\}. \tag{4}$$

Then, the mean vector $\boldsymbol{\mu}_\Omega$ and the covariance matrix $\sum_\Omega$ can be estimated from the set Ω, and the density function $p(\boldsymbol{\delta})$ that we want to know can be written as:

$$p(\boldsymbol{\delta}) \propto \exp\left(-\frac{1}{2} (\boldsymbol{\delta} - \boldsymbol{\mu}_\Omega)^T \sum\nolimits_\Omega{}^{-1} (\boldsymbol{\delta} - \boldsymbol{\mu}_\Omega) \right). \tag{5}$$

Noting that the higher value of $p(\boldsymbol{\delta})$ implies a higher likelihood that the two data making up $\boldsymbol{\delta}$ belong to the same class, the similarity measure $S_G(\boldsymbol{x}, \, \boldsymbol{x}\prime)$ for two samples $\boldsymbol{x}$ and $\boldsymbol{x}'$ is defined as the value proportional to $p(\boldsymbol{\delta})$, such as:

$$S_G(\boldsymbol{x}, \, \boldsymbol{x}\prime) = -(\boldsymbol{x} - \boldsymbol{x}' - \boldsymbol{\mu}_\Omega)^T \sum\nolimits_\Omega{}^{-1} (\boldsymbol{x} - \boldsymbol{x}' - \boldsymbol{\mu}_\Omega). \tag{6}$$

The efficiency of the obtained similarity value has been confirmed in various application problems [26,27,32]. In the prototypical network, its classifier uses Euclidean distance, which is the special case of the probabilistic similarity with $\boldsymbol{\mu}_\Omega = 0$ and unit covariance matrix. In this paper, we apply the general covariance matrix in the learning of embedding function as well as classification. It should also be remarked that this similarity is different from the conventional Mahalanobis distance [25] that uses covariance of original samples $\boldsymbol{x}$. By using intra-class difference vectors, the larger number of samples can be used to estimate the covariance matrix $\sum_\Omega$, and can obtain more accurate estimation.

4. Proposed Method

4.1. Few-Shot Classification Using Probabilistic Similarity

The assumption for deriving the explicit form of Equation (6) is rather impractical to deal with diverse variations of real data, but it could be effectively applied to the data representation obtained from the well-trained embedding function. Based on this consideration, we propose to combine the probabilistic similarity with the metric-based few-shot classification model. Though the probabilistic similarity does not depend on the structure of the embedding network and can be combined with various few-shot learning models, we adopt the prototypical network, the primary and representative model, in order to focus on the effect of the similarity.

When the probabilistic similarity $S_G(\boldsymbol{x}, \, \boldsymbol{x}')$ of Equation (6) is applied to the few-shot classification model, $\boldsymbol{x}$ is a query data, $\boldsymbol{x}'$ is a support data, and the average $\boldsymbol{\mu}_\Omega$ of the difference vectors can be set to zero. Additionally, with the few-shot classification model, we have an embedding function f_ϕ with parameter ϕ, and the embedding vector representation $f_\phi(\boldsymbol{x})$ can be used instead of the raw data $\boldsymbol{x}$. The similarity function is then written as:

$$S_G\left(f_\phi(\boldsymbol{x}), \, f_\phi(\boldsymbol{x}')\right) = -\left(f_\phi(\boldsymbol{x}) - f_\phi(\boldsymbol{x}')\right)^T \sum\nolimits_\Omega{}^{-1} \left(f_\phi(\boldsymbol{x}) - f_\phi(\boldsymbol{x}')\right). \tag{7}$$

With the nearest neighbor classifier, we assign query data $\boldsymbol{x}$ to the classes including the support data with maximum similarity values. For the case of prototype-based classifier, a

prototype vector c_k for each class c_k is calculated first by taking the mean of embedding support vectors $f_\phi(x)$ in c_k, which is written as:

$$c_k = \frac{1}{|S_k|} \sum_{x \in S_k} f_\phi(x). \tag{8}$$

Then, the class-label of query data x is determined by the similarity between embedding query vector $f_\phi(x)$ and the prototype c_k for each class c_k;

$$S_G\left(f_\phi(x),\ c_k\right) = -\left(f_\phi(x) - c_k\right)^T \sum_\Omega{}^{-1} \left(f_\phi(x) - c_k\right) \tag{9}$$

According to the format of the distance-based classifier, the similarity function defined above is rewritten as a distance function and we finally obtain:

$$dist\left(f_\phi(x),\ c_k,\ \sum_\Omega\right) = \left(f_\phi(x) - c_k\right)^T \sum_\Omega{}^{-1} \left(f_\phi(x) - c_k\right) \tag{10}$$

Here, the covariance matrix $\sum_\Omega$ is a parameter to be estimated during learning of embedding network as well as classification. Note that this probabilistic similarity has an advantage in that the number of samples for estimating $\sum_\Omega$ is relatively large even under the few-shot situation. Since the set of intra-class difference vectors is used for estimation, the number of samples in Ω is finally $N \times K^2$ in the case of the N-way K-shot problem.

In the few-shot classification, it is important to catch common distributional property shared by different classes at the learning phase and use them to classify the newly given classes in the test phase. Since the probabilistic similarity measure is derived from the distribution of environmental components shared by all classes, it can be estimated iteratively through episodic learning.

At t-th iteration of episodic learning, with the set of difference vectors Ω_t, the covariance $\sum_t$ is estimated as:

$$\sum_t = (1 - \alpha) \sum_{\Omega_t} + \alpha \sum_{t-1}, \tag{11}$$

where $\alpha (0 \leq \alpha \leq 1)$ is a user-defined parameter to control the proportion of the previously obtained estimation at $(t-1)$-th episode. In the test phase, we have the set of difference vectors Ω_{tst} composed of support samples in test classes, and the covariance $\sum_{tst}$ is estimated as:

$$\sum_{tst} = (1 - \alpha) \sum_{\Omega_{tst}} + \alpha \sum_{trn}, \tag{12}$$

where $\sum_{trn}$ is the covariance estimated by using the whole train set after learning is finished, and it is added to the covariance of the set of Ω_{tst} with a user-defined coefficient $\alpha (0 \leq \alpha \leq 1)$. We should note that the estimated covariance can be near singular under the few-shot settings, especially when the dimension of the embedding vector is larger than a number of samples in Ω. In that case, we need to add a regularization term (e.g., identity matrix) to prevent a singular condition of its inverse matrix.

4.2. Overall Process

Figure 2 shows the overall structure of the proposed few-shot classification model using probabilistic similarity. The overall process follows the conventional metric-based few-shot classifier illustrated in Figure 1, but there is an additional module for obtaining probabilistic similarity.

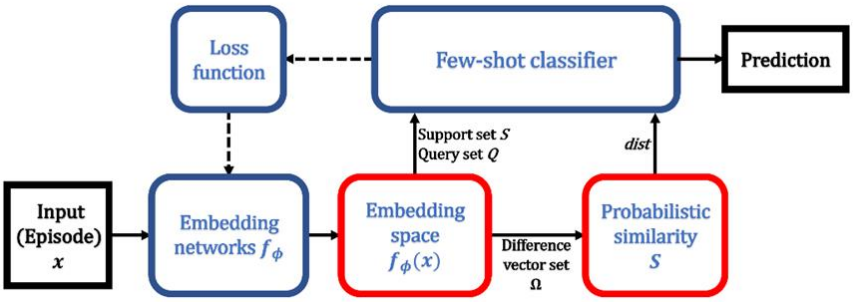

Figure 2. The overall structure of the proposed few-shot classification model.

Under the N-way K-shot classification scenario, a subset for an episode contains examples from N different classes, each of which is decomposed as the support set S_k with K samples and the query set Q_k with the remaining samples ($k = 1, \ldots, N$). Using the support set S_k, the prototype vector for each class is calculated and the set of intra-class difference vectors is also composed. The samples in the query set Q_k are given to a few-shot classifier for conducting classification and evaluating loss value. The loss L for the training episode is defined by using softmax over distances between queries and prototypes, which can be written as:

$$L = \sum_{k=1}^{N} \left[\sum_{\boldsymbol{x} \in Q_k} \frac{1}{N|Q_k|} \left\{ dist\left(f_\phi(\boldsymbol{x}), \boldsymbol{c}_k, \sum_t \right) + \log \sum_{\boldsymbol{x}\prime \in Q_k} exp\left(-dist\left(f_\phi(\boldsymbol{x}\prime), \boldsymbol{c}_k, \sum_t \right) \right) \right\} \right]. \tag{13}$$

The proposed episodic learning process is summarized in Algorithm 1.

Algorithm 1. A training algorithm for embedding network in N-way K-shot problem. N is the number of classes per episode, K is the number of support samples per class, N_{cls} is the total number of classes in the training set.

Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)\}$, where $y_i \in \{1, \ldots, N_{cls}\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ composed of all samples $(\boldsymbol{x}_i, y_i)$ with class label $y_i = k$.
Output: Trained embedding networks f_ϕ
for each episode t **do**
  Compose a subset V by randomly selecting N values from $\{1, \ldots, N_{cls}\}$
  Set $\Omega_t \leftarrow \varnothing$
  for each class label k in V **do**
    Compose a support set S_k by randomly selecting K examples from $\mathcal{D}_k$
    Compose a query set Q_k by subtracting S_k from $\mathcal{D}_k$ ($Q_k \leftarrow \mathcal{D}_k - S_k$)
    Calculate class prototype $\boldsymbol{c_k} \leftarrow \left(\sum_{x \in S_k} f_\phi(\boldsymbol{x})\right) / |S_k|$
    Compose set of difference vectors $\Omega_t \leftarrow \Omega_t \cup \{\boldsymbol{\delta} | \boldsymbol{\delta} = \boldsymbol{x} - \boldsymbol{x}', \boldsymbol{x} \in S_k, \boldsymbol{x}' \in S_k\}$
  end for k
  Calculate covariance matrix $\sum_{\Omega_t}$ using Ω_t
  Update covariance matrix $\sum_t \leftarrow (1 - \alpha) \sum_{\Omega_t} + \alpha \sum_{t-1}$
  Set loss $L \leftarrow 0$
  for each class k in V **do**
    for each $\boldsymbol{x}$ in Q_k **do**

$$L \leftarrow L + \frac{1}{N|Q_k|} \left[dist\left(f_\phi(\boldsymbol{x}), \boldsymbol{c_k}, \sum_t\right) + \log \sum_{\boldsymbol{x}' \in Q_k} exp\left(-dist\left(f_\phi(\boldsymbol{x}'), \boldsymbol{c_k}, \sum_t\right)\right) \right]$$

    end for $\boldsymbol{x}$
  end for k
  Update network parameters ϕ using a gradient descent optimizer with loss L
end for t

In the test phase, samples from new classes that are not seen during learning are given. Each test class is also decomposed as support data and query data. After calculating the prototype vector and similarity function by using the optimized embedding function through the learning phase, query samples are assigned to the class of the closest prototype.

In this case, the covariance matrix used for distance calculation is obtained by using Equation (12).

5. Experimental Results

In order to verify the performance of our proposed method, we conducted experiments using three datasets: Omniglot [33], Multi-PIE [32], and GTSRB [34]. Since the purpose of the experiments is to see the effect of the probabilistic similarity measure, we mainly compare its performance with the conventional model with Euclidean distance. Each dataset was divided into training data and test data. With the training set, the embedding network was trained using the Adam optimizer. The learning rate started at 10^{-3} and halved every 5000 episodes. The training continued until convergence of loss value, which took at least 50,000 episodes. For the performance evaluation, the classification accuracies for 600 test episodes were calculated and averaged various N-way K-shot settings. Since the proposed method needs at least two samples to compose the difference vector set Ω, we set $K = 5$, which is a common setting in the conventional works. Instead, we investigated the performance change according to the increase in the number of ways N, which is practically more important but is not addressed in the previous works.

5.1. Omniglot Character Recognition

Figure 3 shows some examples of Omniglot data [34] which are a handwritten dataset for various characters. It consists of 1623 characters collected from 50 alphabets, and each character has 20 samples drawn by different individuals. We follow the procedure of Vinyals et al. [8] for data preparation and augmentation. The original 105×105 data are resized to 28×28 and rotated by multiples of 90 degrees. By rotating the existing image, we obtained four times as many classes as the original one. The embedding network with four convolutional blocks transforms an image into a 64-dimensional feature.

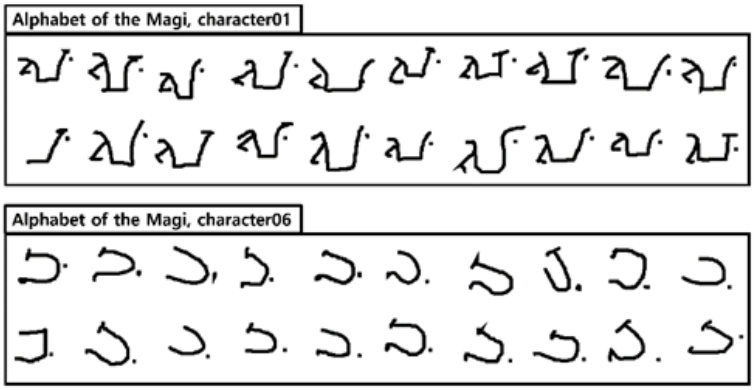

Figure 3. Character variation samples of Omniglot data.

Since the embedding vector is 64 dimensions, the covariance matrix is 64×64, and thus we have 4096 parameters to be estimated, which is much larger than the number of data given in the few-shot task. This is prone to cause a singularity in the inversion of the covariance matrix during distance calculation. To avoid this, we added an identity matrix as a regularization term for estimating covariance, as shown in Equation (11). In the test phase, we also added the covariance $\sum_{\text{trn}}$ obtained from training data as shown in Equation (12).

Table 1 compares the classification accuracy of the proposed method with the conventional methods, under the 5-shot settings. The Omniglot data are one of the representative benchmark sets for few-shot classification, but it is relatively simple. Thus, as shown in Table 1, all the methods show good results while the proposed one achieves the best results. In Figure 4, we compared the performance changes according to the number of test classes. In order to see the effectiveness of the proposed method, we compare the performance with

the original prototypical network [9] as well as the Gaussian prototypical network [22] that uses class-wise covariance information. As shown in the graph, the performance degradation of the proposed method is gentler than the original prototypical network. Though the Gaussian prototypical network shows better performance than the original one, it can be seen that the proposed intra-class covariance gives a more effective distance measure than the class-wise covariance used in [19]. Since the proposed probabilistic similarity is estimated by using intra-class difference vectors which increase in proportion to the number of classes, it is possible to estimate the covariance matrix $\sum_{tst}$ more accurately as N increases. This may act as a strength of the proposed method, showing better performance in the case of large N. In particular, when compared with the gaussian prototypical network using statistical characteristics, it can be seen that the performance gap increases as N increases.

Table 1. The 5-shot classification accuracies on Omniglot data. The values marked with * are quoted from the original works [8,10,12,17,19], and the ones with ** are measured by experiments using codes provided by the authors [9,22].

Model	5-Shot Acc. (%)	
	5-Way	20-Way
MatchingNet [8]	98.9 *	97.0 *
RelationNet [10]	99.6 *	98.6 *
MAML [17]	99.7 *	98.7 *
ConvNet [19]	99.6 *	98.6 *
IMP [12]	99.5 *	98.6 *
ProtoNet [9]	99.50 **	98.40 **
GaussianProtoNet [22]	99.50 **	98.40 **
Proposed	**99.71**	**98.71**

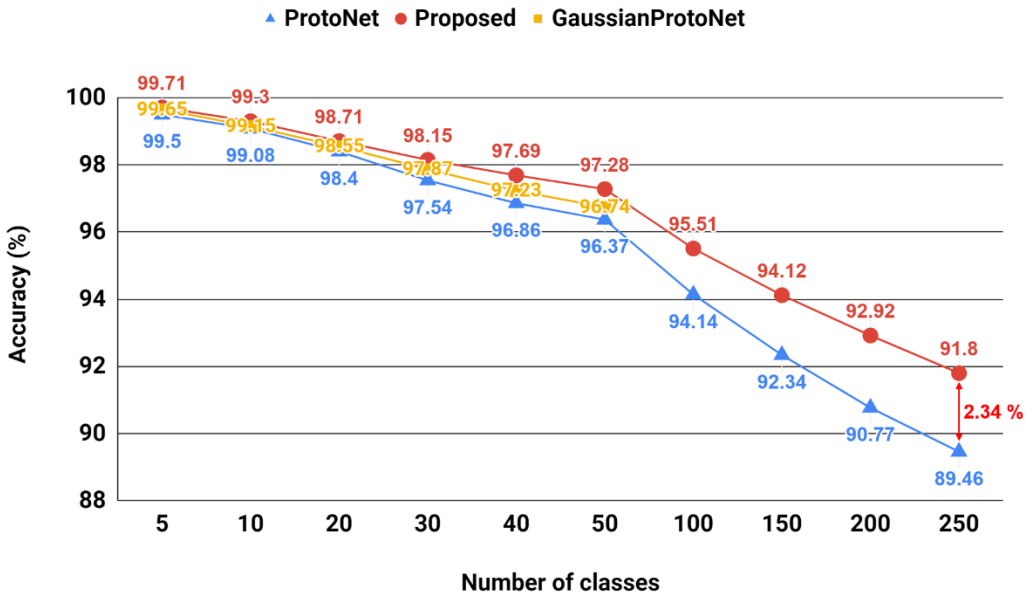

Figure 4. Change of Classification accuracy on Omniglot data depending on the number of ways (in the case of the Gaussian prototypical network, experiments were possible up to 50 ways with the provided code.).

5.2. Multi-PIE Face Recognition

Figure 5 shows some examples of the Multi-PIE dataset that has been created as a benchmark for facial recognition. A total of 337 subjects participated in data collection, and shooting was conducted in four-time sessions. In the shooting, 20 patterns of lighting effects, 15 poses, and 6 types of emotional expression were mobilized, and over 2000 images were collected per subject in a one-time session. We transform the original data to suit

the intention of the experiment according to the previous work [32]. The whole image is cropped so that only the face appears as shown in Figure 6. It is then converted to a black and white image and then resized to 28 × 28 pixels. The Multi-PIE data have rather simple variations compared to the recent benchmark for face verification. However, considering that this experiment is conducted with the simple convolutional network with the purpose of verifying the effect of the probabilistic similarity, Multi-PIE data are appropriate in the sense that it assorted environmental variations such as illumination, poses, expression, and time sessions.

Figure 5. Original Multi-PIE dataset.

Figure 6. Modified Multi-PIE image samples used in the experiment.

The modified dataset consists of a total of 184 classes, and we divide them into 122 training classes and 62 test classes for a few-shot classification problem. The training set contains 600 samples per class, and the test set contains 370 samples per class. For each training episode, 45 queries per class are used. We also should note that each class of Multi-PIE data has much more samples with diverse variations than Omniglot data while only five samples per class are used for support. This may cause some difficulties in estimating the covariance of difference vectors.

Figure 7 compares the performance of the proposed method using prototypical networks. From the graph, we can see that the proposed method can improve the accuracy by using probabilistic similarity, and the effect of performance improvement appears more clearly as the number of ways increases. This result is consistent with our argument that, as the number of ways increases, the accuracy of the estimation increases and thus more sophisticated classification becomes possible. Recognition performance could be further improved by using a more complex backbone network, but this is somewhat out of the scope of this study. In this experiment, we focused on confirming the effect of probabilistic similarity.

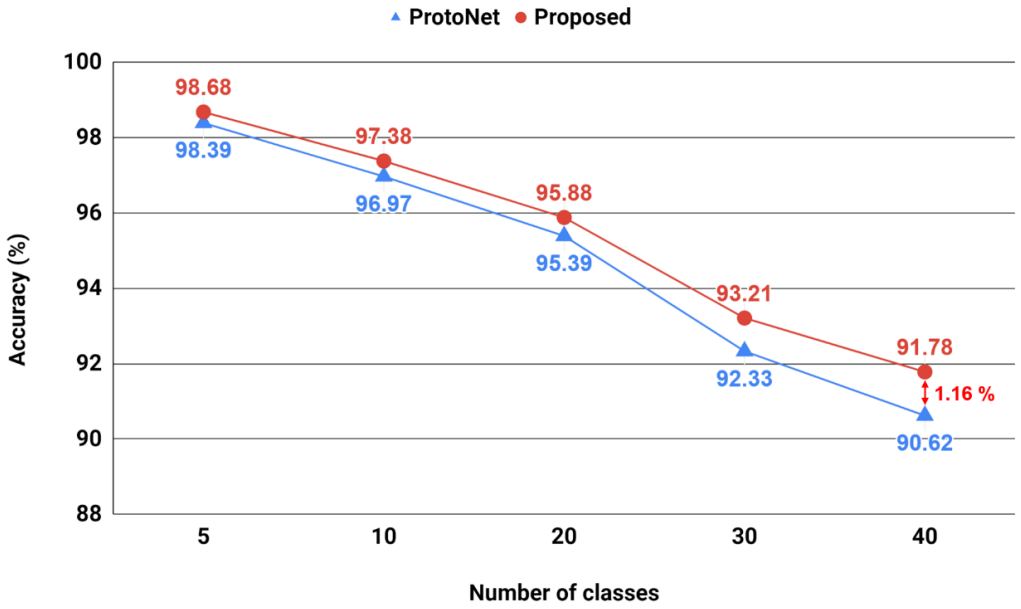

Figure 7. Change of Classification accuracy on Multi-PIE depending on the number of ways.

5.3. GTSRB Traffic Sign Recognition

As the third dataset, we chose a more practical one that is likely to be observed in real applications. GTSRB dataset [34] consists of various types of sign images as shown in Figure 8. There are 43 types of signs and total 51,839 images, which are color images taken from various angles and lighting conditions with various resolutions. We resize all the images to 84×84 pixel size. A total of 43 classes are divided into 22 and 21 classes used for training and testing, respectively. In order to maximize the generality of the embedding network through learning, data augmentation for the training set was performed according to the previous work [13]. Similar to the case of Omniglot, the number of training classes was increased by rotating the training images. Since it is a color image, the raw data format becomes $84 \times 84 \times 3$ and is converted into a 1600-dimensional feature vector through the embedding network. In the learning phase, the embedding network is trained through episodes in the form of 20-way 5-shot. In the test phase, we start at 5-way 5-shot classification, and increase the ways by 5, finally reaching up to 21-way.

Figure 8. GTSRB image sample used in the experiment.

Figure 9 shows the change of accuracy according to the number of classes for the proposed method and prototypical network. In these practical data, the effect of probabilistic similarity is observed more clearly. Especially, the superiority of the proposed method becomes clearer as the number of ways increases. The results are consistent with the assumptions about the correlation between the number of ways and the accuracy. In order to verify the efficiency of the proposed method compared with state-of-the-art methods, we also conducted experiments for 1-shot classification according to [34]. Since the proposed

method cannot obtain difference vectors set Ω_{tst} with a single support sample, episodic training was carried out with five support samples during the learning phase, and the covariance obtained from the train dataset was used for testing. From Table 2, we can see that the performance of the proposed method was higher than most conventional models except the VPE model with data augmentation, which is well-designed for the specific GTSRB data.

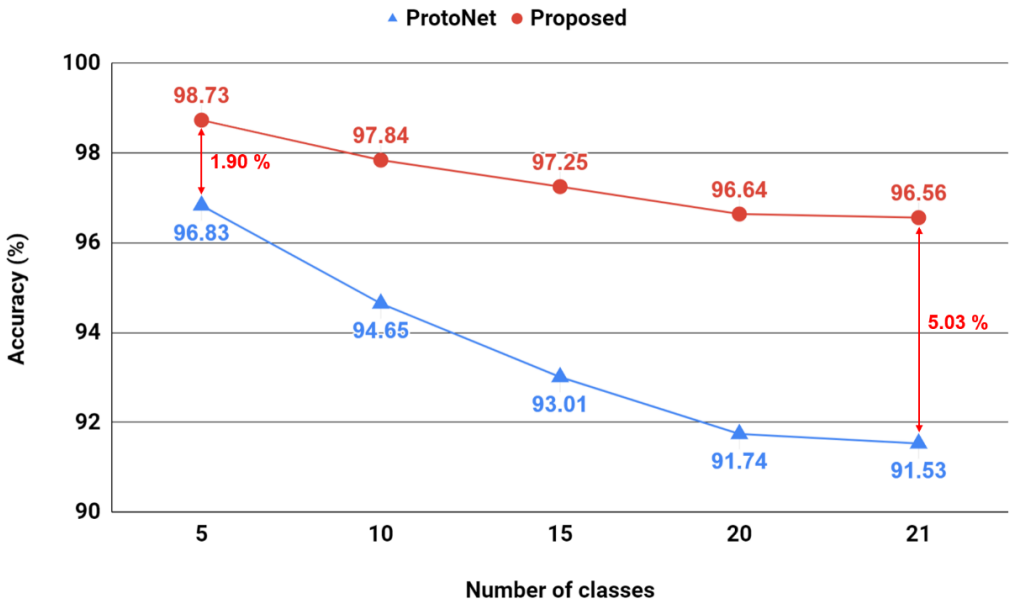

Figure 9. Change of Classification accuracy on GTSRB depending on the number of ways.

Table 2. The 1-shot classification accuracies on GTSRB. The values marked with * are quoted from the VPE paper [15], and the ones with ** are measured by experiments using codes provided by the authors [9].

Method	21-Way 1-Shot Acc. (%)	
	Original Training Data	Train with Data Augmentation
ProtoNet [9]	67.10 **	74.58 **
QuadNet [20]	45.20 *	-
SiamNet [6]	22.45 *	33.62 *
MatchingNet [8]	26.03 *	53.30 *
VAE [31]	20.67 *	22.24 *
VPE [15]	56.98 *	81.27 *
Proposed	**69.62**	**75.56**

To summarize the results of the three experiments, when the size of the way is low, the difference in performance from the existing models does not appear much. However, from Figures 4, 7 and 9, we can see a noticeable difference from the original model for the task with a larger number of ways, which has not been investigated in previous works. The larger the way, the more samples can be used to estimate the environmental distribution represented as the covariance matrix. Thanks to this advantage of the proposed method based on intra-class statistics, the performance degradation according to the increase in ways is gentler than that of the conventional model. Although the classification task for a large number of classes has great practical importance, it has rarely been dealt with in conventional works on few-shot classification. This paper has significance in that it presents a method to solve the many-class few-shot classification problem.

6. Conclusions

For the conventional metric-based few-shot classification methods, the main focus is to find a good metric space on which the intra-class variations of unseen classes are minimized. Under the premise that this can be successfully achieved by using a deep embedding network and episodic learning strategy, the classification is performed by a simple distance-based classifier using standard distance such as cosine and Euclidean. In this paper, we suggest a way of improving the distance-based classifier by using a probabilistic similarity, which is derived from a class-independent environmental factor estimated by using intra-class difference vectors. Taking the intra-class difference vector, we can exclude the class-specific components that are hard to estimate with a limited number of samples per class.

Although the probabilistic similarity based on intra-class statistics has already been used in classical pattern recognition studies, the conventional works suggest a premise that a good feature representation for the input data is provided in advance. In the proposed method, however, the feature extraction module (embedding network) is also trained by using loss signals from the classifier with probability similarity. Essentially, the probabilistic similarity assumes that all the classes in a domain have the same intra-class variations, and this is consistent with the prototypical network model, which assumes that it is possible to find a good embedding space where each class has a single prototype and its variations are very limited. Owing to this consistency, the proposed method achieves improved performance in the experiments. However, it is noteworthy that the proposed algorithm has no constraint on the embedding network model and better performance can be expected by using a more complex embedding network. Finally, since the good performance for problems with many classes and the simplicity of implementation can be practical strengths of the proposed method, its practical applications in various fields can also be an interesting follow-up.

Author Contributions: Conceptualization, H.P.; data curation, Y.L. and H.P.; formal analysis, Y.L. and H.P.; methodology, Y.L. and H.P.; software, Y.L.; validation, Y.L. and H.P.; visualization, Y.L.; writing—original draft, Y.L. and H.P.; writing—review and editing, Y.L. and H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: This work was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20204010600060). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
3. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.

4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Franciscom, CA, USA, 4–9 February 2017.
6. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese Neural Networks for One-Shot Image Recognition. In Proceedings of the ICML Deep Learning Workshop, Paris, France, 10–11 July 2015.
7. Ravi, S.; Larochelle, H. Optimization as a Model for Few-Shot Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
8. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3630–3638.
9. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. NIPS. 2017. Available online: https://github.com/jakesnell/prototypical-networks (accessed on 1 October 2021).
10. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to Compare: Relation Network for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 119–1208.
11. Garcia, V.; Bruna, J. Few-Shot Learning with Graph Neural Networks. In Proceedings of the 36th International Conference on Learning Representaions, Vancouver, BC, Canada, 30 April–3 May 2018.
12. Allen, K.; Shelhamer, E.; Shin, H.; Tenenbaum, J. Infinite Mixture Prototypes for Few-Shot Learning. In Proceedings of the 36th Conference on Machine Learning, Long Beach, CA, USA, 16–20 June 2019; Volume 97, pp. 232–241.
13. Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; Luo, J. Revisiting Local Descriptor based Image-to-Class Measure for Few-Shot Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7260–7268.
14. Wertheimer, D.; Hariharan, B. Few-Shot Learning with Localization in Realistic Settings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 6558–6567.
15. Kim, J.; Oh, T.H.; Lee, S.; Pan, F.; Kweon, I.S. Variational Prototyping-Encoder: One-Shot Learning with Prototypical Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9462–9470.
16. Sun, Q.; Liu, Y.; Chua, T.S.; Schiele, B. Meta-Transfer Learning for Few-Shot Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 403–412.
17. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 36th Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 1126–1135.
18. Hao, F.; He, F.; Cheng, J.; Wang, L.; Cao, J.; Tao, D. Collect and Select: Semantic Alignment Metric Learning for Few-Shot Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2020; pp. 8460–8469.
19. Kaiser, Ł.; Nachum, O.; Roy, A.; Bengio, S. Learning to Remember Rare Events. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
20. Kim, J.; Lee, S.; Oh, T.H.; Kweon, I.S. Co-Domain Embedding using Deep Quadruplet Networks for Unseen Traffic Sign Recognition. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
21. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.F.; Huang, J.B. A Closer Look at Few-Shot Classification. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
22. Fort, S. Gaussian Prototypical Networks for Few-Shot Learning on Omniglot. In Proceedings of the NIPS 2017 Bayesian Deep Learning Workshop, Long Beach, CA, USA, 9 December 2017. Available online: https://github.com/stanislavfort/gaussian-prototypical-networks (accessed on 1 October 2021).
23. Li, W.; Xu, J.; Huo, J.; Wang, L.; Gao, Y.; Luo, J. Distribution Consistency based Covariance Metric Networks for Few-Shot Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8642–8649.
24. Liu, B.; Kang, H.; Li, H.; Hua, G.; Vasconcelos, N. Few-Shot Open-Set Recognition using Meta-Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 8798–8807.
25. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D.L. The mahalanobis distance. *Chemom. Intell. Lab. Syst.* **2020**, *50*, 1–18. [CrossRef]
26. Lee, K.; Park, H. A New Similarity Measure Based on Intraclass Statistics for Biometrie Systems. *ETRI J.* **2003**, *25*, 401–406. [CrossRef]
27. Lee, K.; Park, H. Probabilistic learning of similarity measures for tensor PCA. *Pattern Recognit. Lett.* **2012**, *33*, 1364–1372. [CrossRef]
28. Cho, M.; Park, H. A feature analysis for dimension reduction based on a data generation model with class factors and environment factors. *Comput. Vis. Image Underst.* **2009**, *113*, 1005–1016. [CrossRef]
29. Moghaddam, B.; Jebara, T.; Pentland, A. Bayesian face recognition. *Pattern Recognit.* **2000**, *33*, 1771–1782. [CrossRef]
30. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-Learning with Memory-Augmented Neural Networks. In Proceedings of the 33th Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1842–1850.

31. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
32. Choi, H.; Park, H. Measuring Similarity Between Matrix Objects for Pattern Recognition. In Proceedings of the 3rd International Conference on Human-Agent Interaction, Daegu Kyungpook, Korea, 21–24 October 2015; pp. 175–177.
33. Lake, B.; Salakhutdinov, R.; Gross, J.; Tenenbaum, J. One Shot Learning of Simple Visual Concepts. In Proceedings of the Annual Meeting of the Cognitive Science Society, Boston, MA, USA, 20–23 July 2011; Volume 33.
34. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [CrossRef]