electronics

MDPI

Article

Aircraft Type Recognition in Remote Sensing Images: Bilinear Discriminative Extreme Learning Machine Framework

Baojun Zhao [1], Wei Tang [1], Yu Pan [1], Yuqi Han [2] and Wenzheng Wang [3,*]

[1] Beijing Key Laboratory of Embedded Real-Time Information Processing Technology, School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China; zbj@bit.edu.cn (B.Z.); tgwi@bit.edu.cn (W.T.); panyu@bit.edu.cn (Y.P.)
[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China; yuqi_han@tsinghua.edu.cn
[3] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
* Correspondence: wang_wenzheng@pku.edu.cn

Abstract: Small inter-class and massive intra-class changes are important challenges in aircraft model recognition in the field of remote sensing. Although the aircraft model recognition algorithm based on the convolutional neural network (CNN) has excellent recognition performance, it is limited by sample sets and computing resources. To solve the above problems, we propose the bilinear discriminative extreme learning machine (ELM) network (BD-ELMNet), which integrates the advantages of the CNN, autoencoder (AE), and ELM. Specifically, the BD-ELMNet first executes the convolution and pooling operations to form a convolutional ELM (ELMConvNet) to extract shallow features. Furthermore, the manifold regularized ELM-AE (MRELM-AE), which can simultaneously consider the geometrical structure and discriminative information of aircraft data, is developed to extract discriminative features. The bilinear pooling model uses the feature association information for feature fusion to enhance the substantial distinction of features. Compared with the backpropagation (BP) optimization method, BD-ELMNet adopts a layer-by-layer training method without repeated adjustments to effectively learn discriminant features. Experiments involving the application of several methods, including the proposed method, to the MTARSI benchmark demonstrate that the proposed aircraft type recognition method outperforms the state-of-the-art methods.

Keywords: aircraft recognition; extreme learning machine; feature learning

Citation: Zhao, B.; Tang, W.; Pan, Y.; Han, Y.; Wang, W. Aircraft Type Recognition in Remote Sensing Images: Bilinear Discriminative Extreme Learning Machine Framework. *Electronics* **2021**, *10*, 2046. https://doi.org/10.3390/electronics10172046

Academic Editor: Donghyeon Cho

Received: 1 August 2021
Accepted: 23 August 2021
Published: 24 August 2021

1. Introduction

Aircraft type recognition is critical in both civil and military applications because it is a necessary component of target recognition in remote sensing images. However, the task is extremely difficult because of the existence of fine-grained features, which can cause small inter-class changes due to highly comparable subcategories and large intra-class changes due to variances in size, posture, and angle. For instance, Figure 1 illustrates three types of transport aircraft in series, namely, the C-5, C-17, and C-130. Although these aircraft have distinct purposes and roles, they are visually similar.

Numerous methods have been suggested for aircraft type identification, and they may be classified into three categories: deep-neural-network-based methods [1–5], template-matching-based methods [1,3,6], and handcrafted-feature-based methods [7–9].

The template-matching-based method involves constructing a template through image segmentation and key point extraction and then making similarity judgments by using the new image. For example, Wu et al. [6] presented a similarity measure based on reconstruction, which transforms the problem of type identification into one of reconstruction. Subsequently, the authors used a jigsaw reconstruction approach to solve the reconstruction problem to match the result with the standard template. To recognize airplanes, Zhao et al. [1] converted the aircraft identification problem into a landmark

detection problem and used keypoint template matching. Furthermore, to provide accurate and comprehensive representations of airplanes, Zuo et al. [3] developed an aircraft segmentation network and a keypoint detection network. Next, they performed template matching to recognize aircraft types. However, the template matching method is affected by the target attitude, weather, and other factors, and it cannot be used to accurately extract aircraft shapes in complex scenes [2].

Figure 1. Three kinds of transport aircraft: (**1**) C-5, (**2**) C-17 and (**3**) C-130.

Based on the method for the recognition of handcrafted features, aircraft images are extracted by using artificially designed features such as the scale-invariant feature transform (SIFT). Ref. [10] and histogram of oriented gradients (HOG) [11], and the extracted features are sent to a classifier such as SVM for classification and discrimination.For instance, by using several modular neural network classifiers, Rong et al. [7] recognized different types of aircraft. Three moment invariants, namely the wavelet moment, Zernike moment, and Hu moment, were derived from the airplane characteristics and utilized as the input variables for each modular neural network. Hsieh et al. [8] suggested a method based on the hierarchical classification of four distinct characteristics, namely the bitmap, wavelet transformation, distance transformation, and Zernike moment. However, with the use of artificially designed features, it is difficult to accurately describe prior knowledge of the target, and feature generalization is less robust and generalizable [4].

Recently, deep neural networks have been extensively used in a variety of areas, including classification [12–14], detection [15,16], object tracking [17–21], and segmentation [22,23], due to their capacity to learn robust features independently. Deep neural networks, in particular, have facilitated advances in aircraft recognition in remote sensing images. For instance, Diao et al. [4] presented a novel pixel-wise learning approach for object recognition based on deep belief networks. Zhao et al. [1] proposed an aircraft landmark detection method to address aircraft type recognition. This method detects the landmark points of an aircraft by using a vanilla network. Zuo et al. [3] adopted a convolutional neural network (CNN)-based image segmentation method to extract the keypoints of an aircraft object and later implemented template matching to perform object recognition. Zhang et al. [5] presented a conditional generative adversarial network (GAN)-based aircraft type recognition system. Without type labels, the proposed system can learn representative characteristics from images. To extract the discriminative portions of airplanes of various categories, Fu et al. [2] developed a multiple-class activation mapping method.

Compared to handcrafted feature-based machine learning, neural network-based models exhibit significant gains in terms of generalization and robustness. However, these models have the following limitations: (1) the backpropagation (BP) algorithm must be used to perform iterative optimization, meaning that the training process is time-intensive. (2) A considerable volume of training data is required to maintain an elevated level of performance. However, collecting aircraft model samples is a challenging task. Hence, the sample size is often excessively small to support deep neural networks in training, and the resulting models tend to suffer from overfitting. (3) Deep neural network training requires significant computing and storage resources and cannot be effectively implemented in certain resource-constrained environments.

By contrast, shallow feature learning algorithms require fewer computational resources, and their performance is comparable to that of neural-network-based models in certain tasks. For example, Chan et al. [24] suggested a straightforward deep learning network for image recognition, which was composed entirely of the most fundamental data-processing components: block-wise histograms, binary hashing, and principal component analysis.

The extreme learning machine (ELM) [25] is a straightforward and extremely powerful feedforward network with a single hidden layer (SLFN). In comparison to standard SLFN training methods, the ELM can attain significantly higher training speeds while allowing for universal approximation [26]. These aspects can be attributed to the use of fixed hidden neurons and tunable output weights. The ELM can be used to accomplish a variety of tasks, including data representation learning [27–29] and classification [30]. Huang et al. [27] suggested an object recognition method based on the local-receptive-field-based extreme learning machine (ELM-LRF), which is often used to manage raw images directly. The framework generates random weights for the input and analytically calculates the output weights, which leads to a simple and deterministic solution. Zhu et al. [28] presented hierarchical neural networks based on an ELM autoencoder (ELM-AE) [29] to promptly learn local receptive filters and achieve trans-layer representation. Zong et al. [30] presented a weighted ELM to address data with an imbalanced class distribution.

To enhance the efficiency of the existing machine learning models, researchers have focused on facilitating learning by considering the local consistency of data. Peng et al. proposed a discriminative graph-regularized ELM (GELM) [31]. The GELM combines the discriminant information of multiple data samples to construct a Laplacian eigenmap (LE) [32] structure that is incorporated as a regular term in the ELM algorithm. In the generalized ELM autoencoder (GELM-AE) introduced by K. Sun et al. [33], manifold regularization is performed to restrict the ELM-AE to learn local-geometry-preserving representations. To determine both local geometry and global discriminatory information in the representation space, H. Ge et al. [34] developed a graph-embedded denoising ELM autoencoder (GDELM-AE) by integrating local Fisher discrimination analysis into the ELM-AE. Inspired by these studies, we incorporate the geometric information of given data into the recognition model to reduce the effect of small intra-class and large inter-class differences on aircraft recognition models.

To solve the problems mentioned earlier in this section, we propose a bilinear discriminative ELM network (BD-ELMNet) by drawing on the ideas of the local receptive field ELM, manifold regularization, and bilinear pooling. We optimize the CNN from the viewpoints of the training strategy and feature extraction. As the training strategy, layer-by-layer autoencoder training is performed to train the convolution parameters, which helps us to prevent the consumption of considerable computing resources due to BP and reduce the required sample size. We designed a four-step feature extraction procedure, and the steps are as follows: primary feature extraction, intermediate discriminative feature extraction, high-level feature extraction, and supervised classification. The primary feature extraction module uses the ELMConvNet network, which introduces multiple convolutions and pooling operations based on a single-layer ELM-LRF. To enhance the image classification and processing capabilities, the network structure can extract abstract image information and ensure the invariance of the displacement of data feature attributes. To realize intermediate discriminant feature extraction, a manifold regularized ELM autoencoder (MRELM-AE) is used to extract strong discriminative features, which can learn data representations from the local geometry and local discriminants extracted from the input data by minimizing the intra-class distance and maximizing the inter-class distance. In the MRELM-AE, the constraints imposed on the output weight force the outputs of similar and distinct samples to be close to and far from one another in the new space, respectively. The constraint is a manifold regularization term that is added to the goal of the original ELM-AE model. The output weights may then be solved analytically. In the high-order feature extraction module, the bilinear pool model is used as the high-order feature extractor

of the BD-ELMNet, which extracts second-order statistical information by calculating the outer product of the feature description vectors. This second-order statistical information can reflect the correlation between feature dimensions and generate an expressive global representation that can significantly enhance the classification performance of the model. Finally, we employ the weighted ELM as a supervised classifier to alleviate the problem of data imbalance.

The main contributions of this paper can be summarized as follows:

(1) We propose a novel aircraft recognition framework that not only inherits the characteristics of the ELM's training speed but also relies on convolution, MRELM-AE, and bilinear pooling to construct a three-level feature extractor, as a result of which the aircraft recognition model exhibits strong discrimination features.
(2) We propose a novel discriminant MRELM-AE, which adds the manifold regularization to the objective of the ELM-AE. The manifold regularization considers the geometric structure and distinguishing information of the data to enhance the feature expression ability of the ELM-AE.
(3) The experimental results on the MTARSI dataset [35] show that the BD-ELMNet outperforms the state-of-the-art deep learning method in terms of its training speed and accuracy.

The remainder of this article is organized as follows. In Section 2, we briefly introduce the works related to convolutional neural networks, pooling methods, data augmentation techniques and discriminative ELM. In Section 3, we introduce the proposed aircraft model recognition algorithm, BD-ELMNet. In Section 4, we discuss the performance of the proposed method and compare it with that of classic image recognition algorithms on the MTARSI dataset. Finally, we present a few concluding remarks in Section 5.

2. Related Work

This section provides a brief review of the CNN, pooling technologies, data augmentation, and discriminative ELMs , which are necessary to develop the proposed BD-ELMNet.

2.1. Convolutional Neural Networks

Deep neural networks have been used to achieve considerable progress in many areas such as image recognition and object detection algorithms in recent years. In image recognition, with the discovery and application of various CNN training techniques, such as Dropout [36] and Batch Normalization [37], as well as the increasing abundance of computing resources, CNN models have evolved continuously, and the results achieved in various image recognition competitions (such as ImageNet [38]) have only been increasing.

Among the more classic CNN models are AlexNet [12], VGG [13], GoogLeNet [39], ResNet [40], and DenseNet [41]. In terms of target detection, with the development of the aforementioned deep learning-based CNN models and the advancement of detection frameworks, considerable progress has been made in the field of natural image object detection; for example, two-stage detectors such as Faster Region-based CNN (RCNN) [42] and one-stage detectors such as SSD [43] and YOLO [44] have been developed. The application of these methods to two typical image-detection datasets, namely Pascalvoc [45] and COCO [46], has yielded excellent results.

However, due to the limitations of storage space and power consumption, the storage and calculation of neural network models on embedded devices remains hugely challenging. To solve the problem of deploying neural networks to resource-constrained embedded platforms, researchers have extensively investigated lightweight neural network designs. For example, SqueezeNet [47] uses 1×1 convolution and grouped convolution methods to achieve 50 higher performance than AlexNet. The level of compression is good, and the model has considerable precision. MobileNet [48] was proposed as a deep separable convolution method, and the method was applied very successfully. The model is 96.8% smaller and 27 times faster than the VGG-16 model. ShuffleNet [49] uses packet convolution

and channel shuffle methods to reduce the size of the network model and improve its operating efficiency.

Although CNN technology has progressed considerably, fundamentally, CNN is a nonconvex optimization model that uses BP to solve a problem iteratively in reverse to find the local optimal solution. Even in lightweight neural networks, a large number of parameters need to be optimized, which requires considerable computing power. In addition, the training set must be adequately large to properly train the network. These shortcomings severely restrict the CNN method when hardware platforms with limited computing resources and small-scale datasets are used.

To reduce the time required for CNN training based on the BP optimization algorithm, we designed the ELMConvNet network in BD-ELMNet to extract the key features and adopted a layer-by-layer pre-training strategy to optimize the model. Specifically, training can be stopped at any appropriate layer to create a model that captures the desired features. Compared with the BP optimization technology, the layered training method can improve the efficiency of neural network training. Moreover, by using the ELM optimization idea for reference, we can randomly generate the hidden layer node parameters of the network without adjustment and transform the output layer parameter solution into a simple linear convex optimization solution problem, which further improves the efficiency and reduces the sample size.

2.2. Pooling Methods

Pooling methods represent a key component of CNNs. The pooling layer reduces the dimensionality of the feature map after the convolutional layer through pooling calculation. Before CNNs began to be widely used, several analyses on pooling methods were reported. Bro et al. [50] analyzed the average and maximum pooling in the traditional method and proved that maximum pooling can help to retain more discriminative features than average pooling in terms of probability. Recent works on pooling have focused on how to better reduce the size of the feature map of a CNN by using a new pooling layer. In the process of mixing pools [51,52], various combinations of maximum pools and average pools are used to achieve this reduction. L_p pooling [53] aggregates activation in a norm L_p, which can be considered to be the continuum between maximum pooling and average pooling controlled by learning p. These methods can unify maximum pooling and average pooling and further improve the network performance.

Bilinear pooling is a widely used technique that emphasizes the most informative part of a feature map from an overall perspective by aggregating paired feature interactions. This method is widely used in fine-grained image recognition [54,55] to distinguish subordinate categories with similar appearances. The calculation of second-order statistics helps us to maintain feature selectivity and increase the expressive power of bilinear features.

Considering that bilinear pooling can be used to express high-order features and effectively distinguish object categories with similar appearances, we use it to extract high-order features in BD-ELMNet to further enhance the feature expression ability of the proposed method.

2.3. Data Augmentation Techniques

The data augmentation strategy is used to increase the quantity and diversity of limited data with the objective of extracting more useful information from limited data and generating value equivalent to more data. A large number of augmentation techniques and methods have been proposed to enrich and augment training datasets, improve the generalization ability of neural networks, and alleviate overfitting in deep learning models due to training using small samples.

Common image augmentation methods are mainly based on image transformations, such as luminosity change, flip, rotation, dithering, and blurring [56–58]. With the continuous expansion of the number of layers in deep learning neural networks and the continuous improvement of the expression ability of such networks, to prevent the model

from overfitting, a synthetic sample image augmentation method based on mix-up [59] was proposed [60–62], in addition to the use of GANs [63], as represented by the virtual image sample generation method for image augmentation [64–66]. In different application datasets and application scenarios, different image augmentation strategies and methods are used. Therefore, to identify the best image augmentation strategy for a specific image dataset and application scenario, studies on intelligent image augmentation based on an algorithm or model to search for an augmentation strategy have been conducted. For example, Fawzi [67] proposed adaptive image augmentation, and Cubuk [58] proposed an automatic augmentation framework based on recurrent neural networks. In addition, other studies [68–70] have explored intelligent or automated image augmentation technology.

The method based on data enhancement makes full use of source domain data to generate a large number of training samples for the target domain. For this reason, the small sample image classification problem can be solved using the classical machine learning algorithm in supervised learning. In this work, we focus on alleviating the problem of small samples from the perspective of model optimization and feature enhancement. Moreover, we use data augmentation based on image transformation for data augmentation.

2.4. Discriminative ELMs

Numerous extensions of ELM-AEs and ELMs have been introduced to effectively learn representations that maintain the local geometry of the input data. The GELM, introduced by Peng et al. [31], integrates the discriminant information from the data samples to create an LE structure [32], which is incorporated into the ELM algorithm as a standard concept. Yan et al. [71] suggested an information discriminative extreme learning machine that incorporates the geometric characteristics and discriminative information contained in the data sample into the ELM model to increase the generalization efficiency of the ELM classification results. Yan et al. [72] developed the SPELM model, which is a discriminative ELM with supervised sparsity. The SPELM is a subspace learning approach that considers the discriminative and sparse knowledge contained in the data as it progresses from the hidden to the output layer. Inspired by popular learning, K. Sun et al. [33] proposed a regularized ELM-AE algorithm, which adds popular regularization constraints to the ELM-AE loss function to learn local geometric preservation representations. Similarly, H. Ge et al. [34] incorporated local Fisher discriminant analysis (LFDA) into the ELM-AE loss function and proposed the GDELM-AE algorithm to identify local geometry and global discriminant knowledge in the representation space.

Unlike the methods based on discriminative ELMs, the ELM is not used as a classifier in this study. Specifically, we introduce several enhancements for the ELM to be used as a feature extraction model. The proposed MRELM-AE is essentially an unsupervised feature extraction model that simultaneously introduces the dataset structure and discriminant information into the ELM-AE. In terms of discriminating information techniques, GELM examines only the data's label consistency characteristics and compels samples belonging to the same class to provide a similar output. Unlike the GELM, the MRELM-AE leverages both the geometric structure and discriminant knowledge of the input data by maximizing inter-class compactness and intra-class separability. In addition, to mine the discriminative information of the input spatial data, the IELM adds an inter-class scatter degree and within-class dispersion into the ELM. The MRELM-AE also introduces the marginal fisheries analysis (MFA) [73] graphical penalty, which maximizes in-class compactness and input data separation by maximizing the related local geometrical structure and local discrimination information.

Furthermore, in terms of extracting the discriminant features, our work is significantly different from the GELM-AE and GDELM-AE. Although both the MRELM-AE and GELM-AE yield discriminative features based on popular regularization, the GELM-AE relies only on the local geometry of the input data for feature expression. In contrast, the local geometry and local discriminatory information of input data are concurrently used by the MRELM-AE to extract discriminating features. In addition, the GDELM-AE adds

LFDA [74] based on the assumption of a Gaussian distribution to the ELM-AE to mine the local and global structure of the input data. There is a risk in this method: when the input data do not conform to a Gaussian distribution, LDFA can not fully describe the separability of different categories, resulting in the GDELM-AE not being able to effectively extract the separation features between classes. The MRELM-AE algorithm we proposed is suitable for input data with arbitrary distribution and maximizes intra-class compactness and the separation of input data by maximizing the related local geometric structure and local discriminant information.

3. Bilinear Discriminative ELM

3.1. Overall Framework

The ELM-LRF represents the first attempt to introduce the local receptive field theory into the ELM framework as a general ELM framework to solve image processing problems. In contrast to the CNN algorithm, the ELM-LRF can use the local receptive field to extract the local features, and the hidden layer parameter tuning does not require layer-by-layer debugging based on the BP algorithm, leading to faster training. Considering these characteristics, we choose the ELM-LRF as the baseline for our method.

However, the ELM-LRF is essentially a shallow network that cannot extract features with a robust discriminating ability to solve the difficult problem of distinguishing different types of aircraft with similar shapes. To address these issues, drawing on the popular regularization and bilinear pooling concepts, we propose a BD-ELMNet to enhance the feature expression ability of the ELM-LRF. The network structure of the BD-ELMNet, as shown in Figure 2, involves four modules. First, we design the deep convolutional ELM network (ELMConvNet) algorithm, which fuses the CNN and ELM algorithms to extract the primary local features in an image. This algorithm is described in Section 3.2. Second, inspired by popular regularization concepts, we construct a popular regularized extreme learning AE (i.e., the MRELM-AE) to solve the problem of similar aircraft being difficult to distinguish. The MRELM-AE extracts middle-level features with intense discrimination by maximizing the within-class compactness and between-class separability. The MRELM-AE algorithm is introduced in Section 3.3. Subsequently, we implement homologous bilinear pooling for feature fusion to extract higher-order features. The details regarding the bilinear pooling are presented in Section 3.4. Finally, considering the sample imbalance of various types of targets, we use the weighted ELM instead of the ELM classifier.

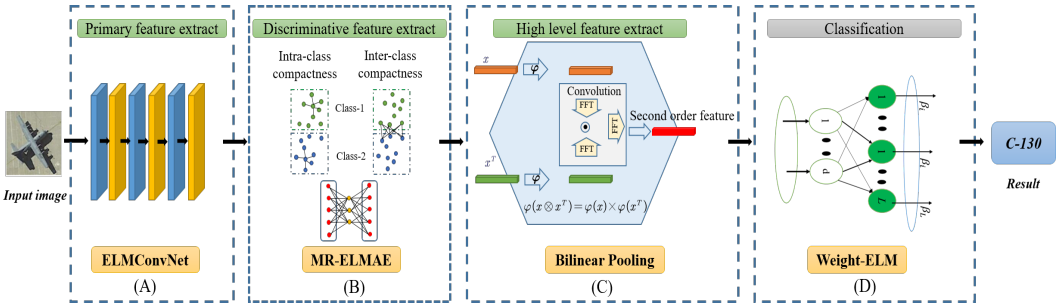


Figure 2. Pipeline of the BD-ELMNet method. (**A**) ELMConvNet structure. (**B**) Use of the MRELM-AE to extract mid-level robust discriminative features. (**C**) Enhancement of the feature expression ability of the ELM-LRF by extracting high-level features through bilinear pooling. (**D**) Use of the weighted ELM as a supervised classifier to solve the problem of unbalanced training samples.

3.2. ELMConvNet

The CNN can effectively mine the spatial features of objects through the convolution and pooling mechanisms. However, the CNN is a non-convex optimization model, and the parameter optimization of its hidden layer requires backpropagation (BP) to complete. The BP backpropagation algorithm makes it easy fo the CNN to fall into the problem of local optimization and a low convergence speed. Unlike classical deep learning, which uses

cumbersome iterative adjustment strategies for all network parameters, the hidden layer node parameters of the ELM network can be randomly generated without adjustment so that the output layer parameter solution is transformed into a simple linear optimization problem. Therefore, compared with the CNN, the ELM converges more quickly and is more efficient. In order to effectively take advantage of the powerful CNN feature extraction and ELM rapid learning, we propose the ELMConvNet network, which uses a random convolutional node that inherits the local receptive field mechanism of the CNN and the random projection mechanism of the ELM as the network unit.

Similar to the structure of the ELM-LRF and CNN, we adopt the convolutional layer, pooling, and activation function as the three components of ELMConvNet. However, in contrast to the ELM-LRF, the convolution kernel parameters in the ELMConvNet network are obtained by training based on the autoencoder framework. It is worth mentioning that the ELM-LRF convolution parameters are randomly selected and produced using a variety of different probability distributions. Simultaneously, the ELMConvNet's mechanism of learning convolution kernel parameters in a layer-by-layer manner through the autoencoder is different from the mechanism of the CNN using a BP neural network to learn the parameters.

The structure of ELMConvNet, as shown in Figure 2, includes operations pertaining to the convolutional layer, feature learning, activation function, and pooling layer. These four parts are described in the following text.

3.2.1. Convolutional Layer

For all potential locations of the convolutional layer in the network, we assume that the size of the input image is $d \times d$, and the receptive field of the convolution is $r \times r$; then, the size of the output feature map is $(d - r + 1) \times (d - r + 1)$. The specific convolution method can be expressed by Formula (1):

$$y_{i,j,k}(\mathrm{x}) = g\left(\sum_{m=1}^{r}\sum_{n=1}^{r} x_{i+m-1,j+n-1} \cdot w_{m,n} + b\right) i, j = 1, \cdots, (d - r + 1) \quad (1)$$

where $y_{i,j,k}$ represents the node of the kth feature map, and g represents the nonlinear activation function. It is worth noting that the convolution parameters w and bias b are not randomly generated but learned through ELM-AE.

3.2.2. Activation Function

The convolutional layer essentially extracts linear features. If the activation function is not added, the composition of several convolutional layers is regarded as a linear polynomial, and the network feature expression ability corresponds only to the linear feature expression ability. The activation function enables the network to learn non-linear feature mapping and thus improves the expressive capability of features. We select the ReLU function as the activation function in this case.

3.2.3. Pooling Layer

After convolution, pooling is implemented to minimize the function dimensionality and add translational invariance in the ELMConvNet network. Various pooling strategies, including averaging and maxpooling [35], are used over local areas.

3.2.4. Feature Learning

The learning of the filters is the most critical stage in the ELMConvNet algorithm. Inspired by the work of [75], we use ELM-based automatic encoding technology to calculate the parameters of the convolution filter, although we introduce several modifications to enhance the performance, such as reconstructing the normalized data instead of the original input. Specifically, the data matrix is first normalized with a mean and standard deviation (denoted as $\boldsymbol{X}_N$) of 0 and 1, respectively. Secondly, we use the intercept term to explain

the distortion of the convolution and learn to rebuild the normalized input term and to intercept the following target matrix: $\boldsymbol{T} = \left[\begin{array}{c|c} \boldsymbol{X}_N & \mathbf{1} \end{array} \right]$.

In order to apply the ELM-AE algorithm to calculate the output weight, we need to determine the input $\boldsymbol{X}$ and the objective function T. Then, the convolution weight parameters and bias can be obtained according to the formula $\left[\begin{array}{c|c} \boldsymbol{F}_{\text{mat}}^T & \boldsymbol{B}^T \end{array} \right] = \boldsymbol{\beta}$, where $\boldsymbol{B}$ is the bias vector, defined as the transposition of the last column of β. The convolution weight parameter $\boldsymbol{F}$ can be obtained by reshaping the matrix $\boldsymbol{F}_{\boldsymbol{mat}}$.

The layered training algorithm can make the ELMConvNet algorithm hold the training under any specified feature layer, and the convex optimization mechanism makes the network converge fast. The convolutional neural network needs to train this model by backpropagating the classification error; the convergence speed is slow, and it is easy to fall into the local optimum. ELMConvNet uses the convolution mechanism of the local receptive field mechanism to propose the initial features of the aircraft target as the input of the subsequent discriminative MR-AE feature extraction.

3.3. Discriminative Feature Learning by the MRELM-AE

After the feature extraction through ELMConvNet, we obtain the low-level features of the aircraft. However, the geometry information is not effectively exploited, which hinders ELMConvNet from learning strong distinguishing features to overcome the issues associated with the presence of fine-grained characteristics. To overcome these challenges, we send the low-level features to the MRELM-AE to extract the high-level features with strong discriminative information.

Recently, it has been proved that retaining the geometric information of the original data points is the basic attribute of feature representation. In particular, preserving the local geometric structure can keep the spatial relationship between the data points in the original domain and their neighboring data points consistent with the spatial relationship after representing the space; for example, in the form of the Euclidean distance. This aspect helps to increase the compactness of the learning representation. Furthermore, the global geometry reflects the relationship within the entire dataset and can help in distinguishing the information from the original data space to the representation space. Therefore, preserving the local geometry can help to minimize the intra-class compactness, while preserving the global geometry can enable the maintenance of the inter-class separation.

To efficiently learn discriminative representations, we propose a novel ELM-based representation learning algorithm: the MRELM-AE. The MRELM-AE adds a graph-based penalty based on the ELM-AE. This penalty is inspired by the MFA framework [73], which extracts the geometric structure and geometry of the input data by maximizing the compactness between classes and separability within classes to discriminate information and enhance the ability to express features. The MRELM-AE has a similar network structure to the ELM-AE. First, an orthogonal random matrix with a nonlinear activation function is used to map the input data to the ELM feature space. Second, based on the reconstruction cost function with a discriminant penalty, the MRELM-AE uses the geometry structure and discriminant information to enhance the feature expression by minimizing the intra-class compactness and maximizing the inter-class compactness separability.

In the MRELM-AE, the characteristics of the intra-class compactness can be expressed as follows:

$$\begin{aligned} S_w &= \frac{1}{2} \sum_{i,j} W_{ij}^w \left\| \boldsymbol{h}_i \boldsymbol{\beta} - \boldsymbol{h}_j \boldsymbol{\beta} \right\|^2 \\ &= \mathrm{Tr}\Big((\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{L}_w (\boldsymbol{H}\boldsymbol{\beta}) \Big) \end{aligned} \tag{2}$$

where $\boldsymbol{h}_i$ is the output of the hidden layer for the sample $\boldsymbol{x}_i$ and β is the weight of the MRELM-AE output layer. $Tr(\cdot)$ is the trace of a matrix. The graph Laplacian $\boldsymbol{L}_w$ is defined as

$$\begin{aligned} \mathbf{L}_w &= \mathbf{D}^w - \mathbf{W}^w \\ W_{ij}^w &= \begin{cases} 1, \text{ if } \mathbf{x}_i \in \mathcal{N}_{k1}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_{k1}(\mathbf{x}_i) \\ \quad \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ arefromthesameclass} \\ 0, \text{ otherwise.} \end{cases} \end{aligned} \tag{3}$$

where $\boldsymbol{W}$ is the adjacent matrix and its elements, and $\boldsymbol{D}$ is the diagonal matrix with $D_{ii}^w = \sum_i W_{ij}^w$. $\mathcal{N}_{k1}(\mathbf{x}_j)$ consists of the NNs of the sample $\boldsymbol{x}_i$. The pairwise edge weights W_{ij}^w reflect the closeness between two samples. Traditionally, the edge weight is defined by the heat kernel $S_{ij} = e^{-\|v_i - v_j\|^2/\delta^2}$ with a predefined σ. By ignoring σ, the edge weight matrix reduces to a matrix with entries defined through function (8). Similar to that in the manifold regularization, D_{ii}^w represents a diagonal matrix with diagonal elements of $D_{ii}^w = \sum_i W_{ij}^w$.

Similarly, the characteristics of the inter-class compactness can be expressed as follows:

$$\begin{aligned} S_b &= \frac{1}{2}\sum_{i,j} W_{ij}^b \|\boldsymbol{h}_i\boldsymbol{\beta} - \boldsymbol{h}_j\boldsymbol{\beta}\|^2 \\ &= \mathrm{Tr}\Big((\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{L}_b (\boldsymbol{H}\boldsymbol{\beta})\Big) \end{aligned} \tag{4}$$

where

$$\begin{aligned} \mathbf{L}_b &= \mathbf{D}^b - \mathbf{W}^b \\ W_{ij}^b &= \begin{cases} 1, \text{ if } \mathbf{x}_i \in \mathcal{N}_{k2}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_{k2}(\mathbf{x}_i) \\ \quad \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from different classes} \\ 0, \text{ otherwise.} \end{cases} \end{aligned} \tag{5}$$

The shortest data pair in the data set k_b is represented by the weight $\boldsymbol{W}^b$. The weight value of a data pair is large when the distance between two data points is short.

Based on the definition of intra-class compactness S_W and inter-class compactness S_b, minimization $\mathrm{Tr}\big((\boldsymbol{H}\boldsymbol{\beta})^T \mathbf{L}_w (\boldsymbol{H}\boldsymbol{\beta})\big)$ can allow the features extracted by the ELM-AE to retain the original data geometry, and maximization $\mathrm{Tr}\big((\boldsymbol{H}\boldsymbol{\beta})^T \mathbf{L}_b (\boldsymbol{H}\boldsymbol{\beta})\big)$ can make the ELM-AE obtain strong discriminative features. When we perform the above process at the same time, we can obtain a new graph Laplacian operator $\mathbf{L}_{new}$, which is defined as $\left(\mathbf{L}_b^{-1/2}\right)^T \mathbf{L}_w \mathbf{L}_b^{-1/2}$ to preserve the geometric structure of the original data and obtain strong discriminant information.

Therefore, we formulate the objective of the MRELM-AE as

$$\begin{aligned} &\arg\min_{\beta} V \\ &V = \tfrac{1}{2}\|\beta\|_F^2 + \tfrac{C}{2}\|\beta\boldsymbol{H} - \boldsymbol{X}\|_F^2 + \tfrac{\lambda}{2}\,\mathrm{Tr}\big((\boldsymbol{H}\beta)^T \boldsymbol{L}_{\mathrm{new}} (\boldsymbol{H}\beta)\big) \end{aligned} \tag{6}$$

where C, λ and γ represent the balance hyper-parameters. Since the objective function (6) is convex, the output weights can be analytically solved as

$$\boldsymbol{\beta}^* = \begin{cases} \left(\frac{1}{C}\left(\boldsymbol{I}_l + \lambda \boldsymbol{H}^T \boldsymbol{L}_{\mathrm{new}} \boldsymbol{H}\right) + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{X} \text{ if } N \geq l \\ \boldsymbol{H}^T \left(\frac{1}{C}\left(\boldsymbol{I}_N + \lambda \boldsymbol{L}_{\mathrm{new}} \boldsymbol{H}\boldsymbol{H}^T\right) + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1} \boldsymbol{X} \text{ if } N < l \end{cases} \tag{7}$$

where $\boldsymbol{I}_l$ and $\boldsymbol{I}_N$ are identity matrices of dimensions l and N, respectively. For the given training data $\boldsymbol{X}$, the representations $\boldsymbol{F} \in R^{N\times 1}$ can be determined as $\boldsymbol{F} = \boldsymbol{X}\boldsymbol{\beta}^T$.

3.4. High-Order Feature Extraction through Compact Bilinear Pooling

Bilinear pooling represents a new feature fusion method that uses high-order information to fuse features to capture the pairwise correlations among features [76]. Various studies have demonstrated the superior fusion performance related with bilinear representation in other aspects, such as concatenation, sum by element, the Hadamard product, and the vector of local set descriptors (VLADs) [75]. Considering the associated inheritance of advantages of both concatenation and element-wise multiplication [54], we implement bilinear pooling after the discriminant MRELM-AE to extract higher-order features, thereby enhancing the discrimination of the features.

Bilinear pooling calculates the outer product between two vectors, which allows a multiplication of the interaction between all elements of the two vectors compared to the element-wise product. However, the feature dimension after bilinear pooling is very high (n^2), which makes bilinear pooling calculation inefficient and difficult to apply. Therefore, in order to solve the problem of inefficient bilinear pooling calculation, we adopt the idea of Multimodal Compact Bilinear pooling [55], as shown in Figure 1, randomly project the features obtained by the MRELM-AE to a higher-dimensional space (using Count Sketch [77]), and then efficiently convolve the two vectors by using the element-wise product in the Fast Fourier Transform (FFT) space.

3.5. Supervised Learning by Using the Weighted ELM

After extracting the features through compact bilinear pooling, the high-order feature expressions for aircraft targets are obtained. Subsequently, the high-order features are sent to the supervised classifier to determine the category of the aircraft objects. Because the data samples of each type of aircraft sample are not balanced, the ELM training and performance analysis are difficult to realize. To mitigate the impact of the abovementioned category imbalance problems, we use a weighted ELM [30] to perform supervised learning. The weighted ELM classifier does not aim at minimizing the classification error rate but at minimizing the weighted classification cost. For the categories with a small number of samples, we artificially set a larger classification error cost to affect the training of the classifier process, increase the impact of small samples on the classification performance, and "re-balance" the number of category labels.

4. Experiments

As stated in this section, we begin by examining the impact of the hyperparameters on the model's performance. Additionally, the proposed BD-ELMNet is compared against various state-of-the-art image recognition systems utilizing the difficult MTARSI dataset [35] for aircraft type recognition.

4.1. MTARSI Dataset

The multitype aircraft remote sensing images (MTARSI) dataset represents the first public, fine-grained aircraft type classification dataset for remote sensing images. Seven specialists in the field of remote sensing image interpretation painstakingly labeled all of the example images. Thus, this dataset possesses high authority. Overall, MTARSI has collected 9385 remote sensing images from Google Earth satellites. Boeing C-5, P-63, T-43, B-1, KC-10, C-130, B-2, B-52, B-29, C-135, C-17,E-3, F-16, C-21, U-2, A-10, A-26, T-6, and F-22 aircraft are included in the 20 aircraft images in the dataset. The number of sample images for different aircraft types is different (see Table 1) and ranges from 230 to 846. In other words, the number of different types of aircraft is imbalanced, which increases the difficulty in aircraft type recognition. Furthermore, the MTARSI includes pictures with varying spatial resolutions as well as complex changes in posture, geographic position, lighting, and time period. This aspect enriches the intra-class variation, rendering aircraft type recognition more challenging. Furthermore, all the aircraft types are similar in appearance and difficult to distinguish, and thus certain inter-class similarities exist in the dataset for

each aircraft type. Several examples of aircraft images in the MTARSI dataset are shown in Figure 3.

Table 1. MTARSI has different aircraft classes and a different number of images for each class.

Types	Images	Types	Images	Types	Images
B-1	513	C-130	763	F-16	372
B-2	619	E-3	452	F-22	846
B-52	548	C-135	526	KC-10	554
B-29	321	C-5	499	C-21	491
Boeing	605	C-17	480	U-2	362
A-10	345	T-6	248	A-26	230
P63	305	T-43	306	-	-

Figure 3. Samples of 20 aircraft types in the MTARSI dataset.

4.2. Evaluation Metrics

In the experiments, we adopted the accuracy and confusion matrix to quantitatively evaluate the aircraft recognition performance. The confusion matrix is a visualization tool that reflects the classification performance of the model, especially for supervised learning. The matrix was determined by comparing the position and classification of each measured pixel with the corresponding status and category in the classified image.

4.3. Implementation Details

Parameter-settings: To effectively extract the features, we followed the AlexNet parameter settings to set the parameters of the BD-ELMNet, as shown in Table 2, which

summarizes the parameters such as the convolution kernel size, number of feature maps, and pooling size. We selected hyper-parameters through cross-validation, such as the number of hidden nodes k in the autoencoder (MRELM-AE), the normalization parameter C and λ. As shown in Table 3, the number of hidden neurons ranged from 100 to 5000, and the interval of all AEs was 100. We selected the hyperparameter C in the range of the exponential gap [1.0×10^{-5}, 1.0×10^{10}] based on the validation set test results.In addition, we set the activation function of the ELM as a non-linear sigmoid function.

Programming-environment-settings: As the experimental platform for all experiments, we used a PC with an Intel i7-6700 CPU, 2.60 GHz, 8 GB of RAM, and GeForce GTX 1080ti GPUs. The algorithms were implemented and executed in MATLAB 2020b.

Table 2. Parameter settings of the DCAE-ELMNet and MRELM-AE joint network.

Layer	Layer Name	Size/Stride	Output
L0	Input layer	224×224 , 3 channels	-
L1	Convolutional layer	3×3 /32, s = 1	224×224
L2	Combined pooling layer	2×2/s = 2	112×112
L3	Convolutional layer	3×3 /64, s = 2	56×56
L4	Combined pooling layer	2×2/s = 2	28×28
L5	Convolutional layer	3×3 /128, s = 2	14×14
L6	Combined pooling layer	2×2/s = 2	7×7

Table 3. Hyperparameter selection range for cross-validation.

Hyperparameter	Range
Number of hidden neurons	100 to 5000
C	1.0×10^{-5} to 1.0×10^{10}
λ	1.0×10^{-5} to 1.0×10^{10}

4.4. Hyper-Parameter Study

The key hyperparameters that affect the performance of BD-ELMNet are mainly the balance parameters (C), λ and the number of hidden neurons (k). We performed multiple crossover experiments to determine the hyperparameter values.

Selection of the number of hidden neurons. We studied the effect of the number of hidden neurons k on the accuracy of aircraft recognition in this experiment. As shown in Figure 4, as the number of buried neurons increased, the algorithm obviously achieved better accuracy and a lower standard deviation. Even with a larger number of buried neurons than 300, the average accuracy remained constant, ranging from 72.15% to 77.1%. When there were more than 450 hidden neurons, the standard deviation was less than 0.05. As a result, we set the number of hidden neurons to 450 in the next trials.

Selection of the balance parameters. The purpose of this experiment is to determine the effect of limited parameters on the accuracy of aircraft recognition. Figure 5 shows the recognition accuracy for different combinations of parameters and C. Clearly, the accuracy of aircraft recognition tends to be stable when $\log_{10}(C) \geq 0$ and $\log_{10}(\lambda) \leq 5$.

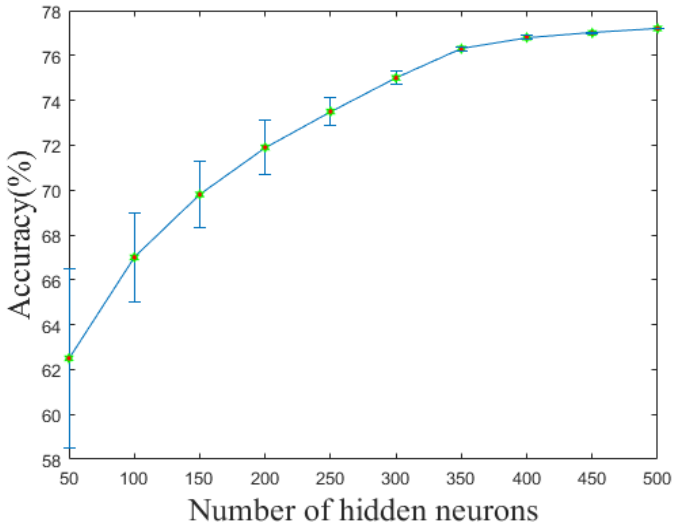


Figure 4. Aircraft recognition performance under different numbers of hidden neurons.

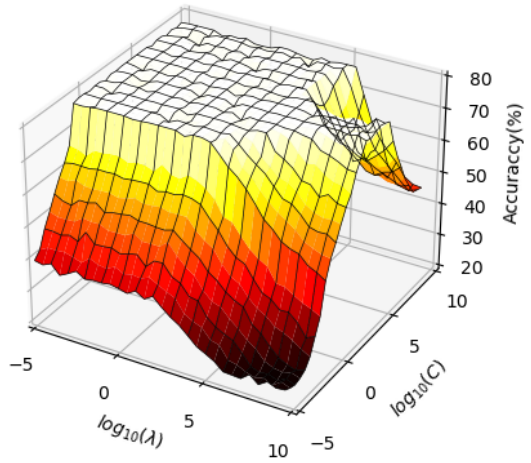


Figure 5. Aircraft recognition performance under different combinations of C and λ.

4.5. Ablation Studies

We provide various comparisons in Table 4 to assess the contribution of each module, where the MRELM-AE, bilinear pooling, and W-ELM correspond to the BD-ELMNet. First, we evaluated the contribution of several elements to our baseline recognizer as a reference. As shown in Table 4, all the techniques contributed to an increase in accuracy, and the final baseline attained an accuracy of 0.781.

(1) Convolution-pool layer. We expanded the single-layer ELM-LRF to a multi-layer neural network structure by introducing multiple convolution and pooling operations. A deep neural network structure can not only extract the abstract information of the image but also can ensure the displacement invariance of the data feature attributes. An increment of 4.9% was achieved. This finding proves that the multi-layer convolution-pooling operation can enhance the feature extraction ability of the ELM-LRF.

(2) MRELM-AE. The MRELM-AE achieved a performance enhancement of 3.8% compared to that of the baseline. This finding shows that the discriminative features learned by the MRELM-AE can effectively increase the accuracy of target recognition.

(3) Compact bilinear pooling. In comparison with the baseline, compact bilinear pooling achieved certain enhancements, which may be attributed to its ability to extract the pairwise correlations between feature. The relative increase was approximately 8.9% in aircraft type recognition tasks.

(4) W-ELM. The performance gain (approximately 6.4%) associated with the W-ELM was relatively large due to the use of the weighted classification cost function, which can mitigate the impact of the category imbalance problems in the considered tasks.

Table 4. Ablation study on the components of the bilinear discriminative ELM network. The bold numbers represent the optimal detection result.

Baseline (ELM-LRF)	Conv-Pool Layer	MRELM-AE	Compact Bilinear Pooling	W-ELM	Accuracy	Training Time (s)
+	–	–	–	–	0.517	187
+	+	–	–	–	0.566	205
+	+	+	–	–	0.628	438
+	+	+	+	–	0.717	789
+	+	+	+	+	**0.781**	**889**

4.6. Comparison with State-of-the-Art Methods

On the MTARSI dataset, we compared the proposed BD-ELMNet with various similar techniques to evaluate the efficacy of the proposed methods; see Table 5. The following related methods were considered: (1) Handcrafted-feature-based approaches: BOVW [78], LBP–SVM [79]. (2) Deep-learning-based approaches: PCANet [24], SqueezeNet [47], AlexNet [12], MobileNet [61]. (3) ELM-based approaches: ELM-LRF [27], ELM-CNN [75].

The handcrafted-feature-based approaches [78,79] are state-of-the-art methods in the field of image recognition; thus, we compared these methods with the proposed method. We utilized a fixed grid size (16 × 16 pixels) with an interval step of 8 pixels to extract all the descriptors in the picture for local patch descriptors such as the SIFT, and we used the average pooling pixels in each dimension of the descriptors to obtain the final image characteristics. For the BOVW, we set the dictionary size at 4096.

In addition, to perform comparative analysis, we applied deep-learning-based approaches and ELM-based approaches. Among these methods, SqueezeNet and AlexNet require the BP algorithm for iterative optimization, while the PCANet, ELM-LRF, and ELM-CNN do not require trivial BP fine-tuning. Note that both AlexNet and LeNet are trained from scratch, and ImageNet-based pre-training models are not used. Furthermore, the PCANet and ELM-LRF methods have two hidden layers. The MTARSI dataset is divided into training and test sets in a ratio of 7:3, and the size of the images is fixed as 224 × 224 pixels. To ensure a fair comparison, the abovementioned methods were compared according to the training set and test set.

Table 5. Experimental results of different classification algorithms on the MTARSI dataset.

Method	Accuracy
LBP–SVM [79]	0.457
ELM-LRF [27]	0.517
PCANet [24]	0.595
SIFT + BOVW [78]	0.609
ELM-CNN [75]	0.715
AlexNet [12]	0.753
SqueezeNet [47]	0.765
MobileNet [48]	0.776
BD-ELMNet (Our method)	0.781

Table 5 indicates that the BD-ELMNet, MobileNet, SqueezeNet, AlexNet, and ELM-CNN exhibit the highest performance, which demonstrates that deep learning based on multi-layer network feature learning can help to learn strong distinguishing features from shallow to deep learning. Since shallow learning methods such as the PCANet and ELM-LRF have only two hidden layers, their feature expression ability is weak and cannot overcome the problem of indistinguishable aircraft with similar shapes.

The performance of the methods based on manual features is lower than that of AlexNet and the proposed method. In particular, the manual feature methods cannot effectively overcome the interference of factors such as illumination and rotation, although a method based on deep learning can independently learn robust features. The BD-ELMNet performs somewhat better than deep learning methods such as AlexNet, demonstrating the method's efficacy. This finding shows that the bilinear pooling and manifold regularization can be used to effectively enhance feature discrimination.

Table 6 presents the experimental results of the different classification algorithms in terms of the computational complexity.

Table 6. Experimental results of different classification algorithms in terms of the computational complexity.

Method	Training Time (s)
PCANet [24]	392
MobileNet [48]	6480
SqueezeNet [47]	4979
AlexNet [12]	3654
ELM-LRF [27]	187
ELM-CNN [75]	498
BD-ELMNet (Our method)	889

The suggested method's training duration was compared to that of the SqueezeNet, MobileNet, and AlexNet techniques in order to assess its computational efficiency and show its recognition accuracy. As indicated in Table 6, the training speed of the shallow learning network (not requiring BP adjustment) was considerably higher than that of the deep learning network. The training time of the BD-ELMNet was higher than that of the ELM-LRF and ELM-CNN because its network involved the additional MRELM-AE and bilinear pooling module. Compared with SqueezeNet, MobileNet, and AlexNet, the training time of BD-ELMNet was reduced by three times, which proves that the layer-wise training procedure can shorten the training time compared with that of the BP optimization method. Since they have a non-convex function, deep learning methods such as SqueezeNet require BP optimization to perform multiple iterations of training to find the local optimal solutions.

To prove the effectiveness of the proposed BD-ELMNet algorithm on the small-scale MSTARSI dataset, we conducted performance tests by using three classical deep learning algorithms, namely AlexNet, SqueezeNet, and MobileNet, with different training methods. We trained these deep learning networks from scratch and trained them with the ImageNet pre-training models. In these two training methods, we added a data augmentation method based on image transformation. The training results are summarized in Table 7.

Table 7. Comparative experiment of different training methods. "_scratch" means training from scratch. "_pre-train" means using the ImageNet pre-training model for training.

Methods	Accuracy
AlexNet_scratch	0.594
AlexNet_pre-train	0.753
SqueezeNet_scratch	0.609
SqueezeNet_pre-train	0.765
MobileNet_scratch	0.658
MobileNet_pre-train	0.776
Our method_scratch	**0.781**

The following observations can be made from Table 7: (1) the effect of the training method fine-tuned using the pre-training model is superior to that of the method trained from scratch; (2) although the proposed method adopts the method of training from scratch, its effect is superior to that of the deep learning algorithm trained using the pre-trained model. The first observation demonstrates that the use of pre-trained models to fine-tune training can alleviate the problem of model overfitting caused by small samples. Such a pre-trained model that is easy to generalize is obtained after training with a large number of sample sets similar to ImageNet. Compared to training using the ImageNet pre-training model, even if the data-enhanced de novo training method is adopted, the effect is far poorer than that of the training method in which fine-tuning is performed using the pre-training model. The above observation can be attributed to three main reasons: (1) the feature learning spaces of deep learning models such as AlexNet are high-dimensional spaces. As the dimensionality increases, the number of samples required increases exponentially. Small samples can easily lead to overfitting when overly complex training models are used. (2) Deep learning models such as AlexNet are nonconvex optimization models with high nonlinearity. The nonconvex optimization method often uses gradient descent optimization, which leads to limited changes in the parameters of each node when the sample size is limited, causing deep learning to easily fall into a local optimum. (3) Deep learning uses only the data calibration drive mechanism, only relevant learning abilities, and no causal reasoning ability of knowledge rules. (4) Even if the data augmentation method is used to enhance the sample size, the enhanced sample size cannot reach the level of ImageNet in terms of magnitude or diversity, and for this reason, a deep learning algorithm with the above shortcomings cannot perform well in terms of feature generalization. The second observation indicates that the proposed method is superior to deep learning models when applied to small sample datasets. The reasons for this are as follows: (1) The proposed method draws on the ELM optimization strategy, and the ELM model seeks the global optimal solution. The hidden layer ELM parameter must learn only its output layer weight parameters. Unlike classical deep learning, which uses cumbersome iterative adjustment strategies for all network parameters, the hidden layer node parameters of the ELM network need not be adjusted, meaning that the output layer parameter solution is transformed into a simple linear optimization problem. This linear convex optimization method reduces the training time and reduces the dependence on sample size. (2) We adopt unsupervised layer-by-layer training methods, such as autoencoders, for the multi-layer network structure. The layer-by-layer training strategy helps us to determine the "good" initial values of the parameters to be optimized, which facilitates rapid convergence in the subsequent global iteration process. The sample size required for the parameter adjustment of the layer-by-layer pre-training strategy is considerably smaller than that for the BP optimization method. (3) In terms of feature expression, the proposed method not only draws on the mechanism of the local receptive field of CNNs but also on popular regularization and bilinear pooling ideas to enhance its feature expression ability.

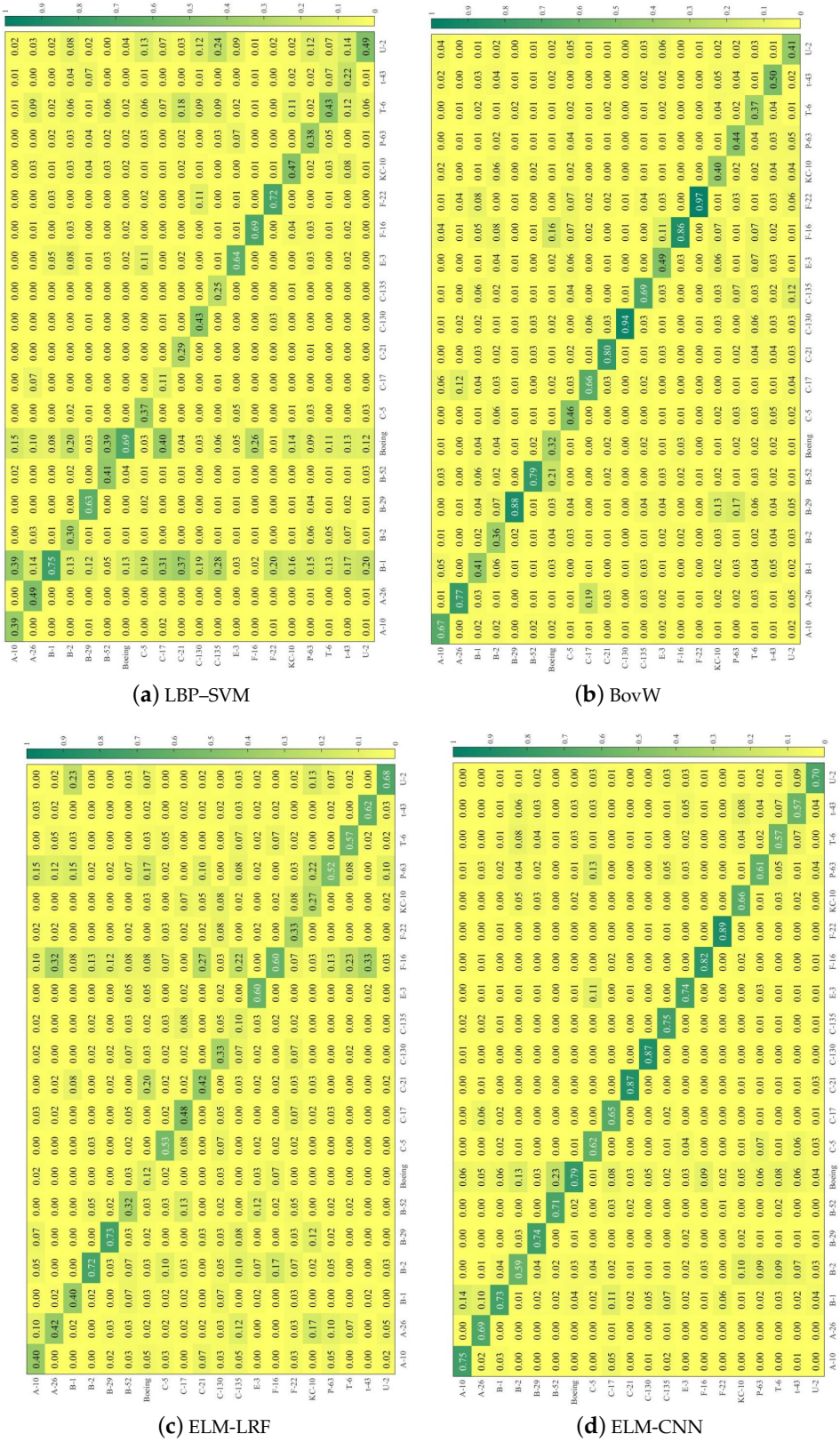
Compared to deep learning that uses only convolutional feature extraction, the feature expression ability of the proposed method is slightly superior.

4.7. Analysis of the Image Features in MTARSI

Considering the results of the experiments involving different classification algorithms, we investigated the type of aircraft in the MTARSI dataset that is most likely to be misidentified. First, we considered the recognition rate for each airplane using various categorization techniques, and then we determined the kinds of aircraft that were most often confused. Figure 6 provides the confusion matrices for the LBP–SVM, BovW, ELM-LRF, ELM-CNN, PCANet, SqueezeNet, AlexNet, and BD-ELMNet on the MTARSI dataset.

The methods based on manual features and shallow learning exhibit a certain recognition performance; however, many misclassifications of similar aircraft types (such as C-5 and Boeing or B-52) occur. This phenomenon occurs because the method based on manual features lacks discriminative representation, thereby rendering it difficult to distinguish airplanes with similar shapes when using manual feature methods.

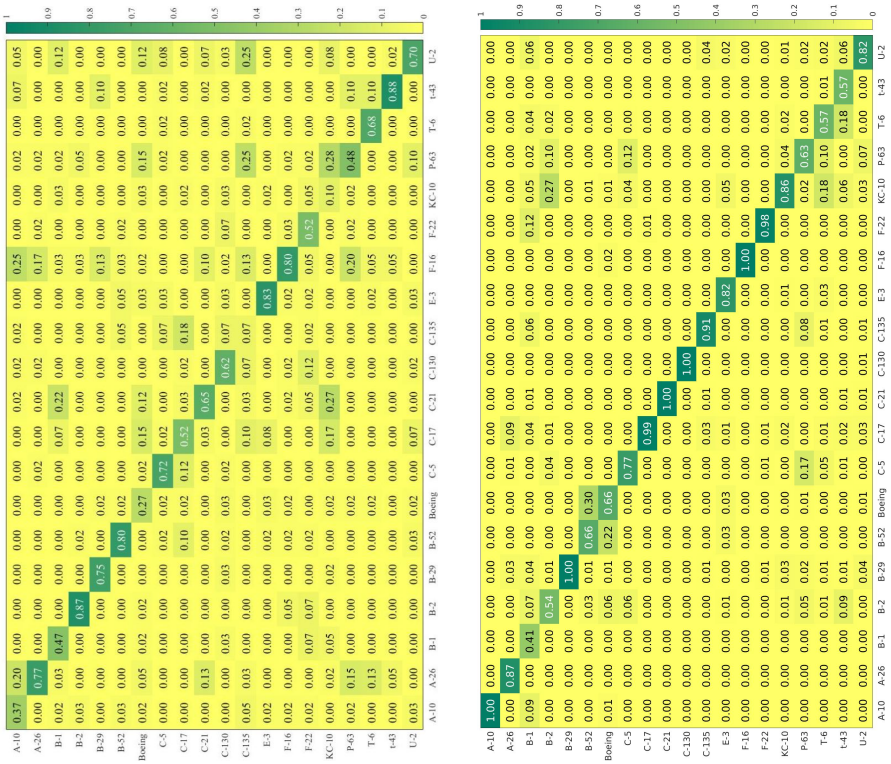
Moreover, the shallow learning method has limited learning feature patterns, and it is challenging to cover multiple aircraft types. Therefore, the associated recognition ability is inferior to that of the deep learning network. In comparison to the two techniques described above, the deep learning method based on multi-layer network feature learning can learn numerous templates related to aircraft structural characteristics using many convolution kernels, improving the deep learning method's feature expression capabilities. However, AlexNet and ELM-CNN methods still struggle to distinguish similar aircraft. These deep-neural-network-based methods do not consider the essential characteristics of the data points, such as the local and global geometries. According to Figure 6h, the proposed method demonstrates an excellent performance except on certain extremely similar aircraft. The main reason is that our method preserves the local geometry and exploits the local discrimination information from the input data. The study experimentally demonstrates that the proposed method can learn data representations with a maximized within-class compactness and between-class separability.



(**a**) LBP–SVM (**b**) BovW

(**c**) ELM-LRF (**d**) ELM-CNN

Figure 6. *Cont.*



(**e**) PCANet (**f**) SqueezeNet

(**g**) AlexNet (**h**) BD-ELMNet

Figure 6. Confusion matrices for the LBP + SVM, BovW, ELM, CRF-ELM, PCANet, SqueezeNet, AlexNet and BD-ELMNet on the MTARSI dataset.

5. Conclusions

This paper proposes a new method known as BD-ELMNet that can extract discrimination features to distinguish aircraft types with similar shapes in remote sensing images. Compared with the existing deep learning methods, the BD-ELMNet efficiently learns representations with two major advantages: (1) because it inherits the CNN feature representation and ELM rapid learning capabilities, the proposed method can realize efficient learning and exhibits an excellent generalization capability without BP fine-tuning; (2) because it can preserve the local geometry and exploit the local discrimination information from the input data by maximizing the within-class compactness and between-class separability, the proposed method can learn strong discriminative features.

Experiments conducted on the benchmark aircraft recognition MTARSI dataset show that the proposed method outperforms state-of-the-art image classification methods such as Bows, PCANet, and AlexNet. Moreover, the proposed method accelerates the training by up to three times compared with popular deep learning algorithms such as AlexNet. Thus, the proposed approach represents a useful tool for accurately recognizing aircraft types.

Author Contributions: Conceptualization, W.W.; Data curation, Y.P.; Formal analysis, B.Z. and Y.P.; Funding acquisition, B.Z. and W.W.; Methodology, W.T.; Software, W.T.; Writing—original draft, W.T.; Writing—review and editing, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 91838303 and in part by the National Natural Science Foundation of China (NSFC) under Grant 91738302.

Acknowledgments: The authors would like to thank all reviewers and editors for their constructive comments for this study.

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

References

1. Zhao, A.; Fu, K.; Wang, S.; Zuo, J.; Zhang, Y.; Hu, Y.; Wang, H. Aircraft Recognition Based on Landmark Detection in Remote Sensing Images. *IEEE Geoence Remote Sens. Lett.* **2017**, *14*, 1413–1417. [CrossRef]
2. Fu, K.; Dai, W.; Zhang, Y.; Wang, Z.; Yan, M.; Sun, X. MultiCAM: Multiple Class Activation Mapping for Aircraft Recognition in Remote Sensing Images. *Remote Sens.* **2019**, *11*, 544. [CrossRef]
3. Zuo, J.; Xu, G.; Fu, K.; Sun, X.; Sun, H. Aircraft Type Recognition Based on Segmentation with Deep Convolutional Neural Networks. *IEEE Geoence Remote Sens. Lett.* **2018**, *15*, 282–286. [CrossRef]
4. Diao, W.; Sun, X.; Dou, F.; Yan, M.; Wang, H.; Fu, K. Object recognition in remote sensing images using sparse deep belief networks. *Remote Sens. Lett.* **2015**, *6*, 745–754. [CrossRef]
5. Yuhang, Z.; Hao, S.; Jiawei, Z.; Hongqi, W.; Guangluan, X.; Xian, S. Aircraft Type Recognition in Remote Sensing Images Based on Feature Learning with Conditional Generative Adversarial Networks. *Remote Sens.* **2018**, *10*, 1123.
6. Wu, Q.; Sun, H.; Sun, X.; Zhang, D.; Fu, K.; Wang, H. Aircraft Recognition in High-Resolution Optical Satellite Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2014**,*12*, 112–116
7. Rong, H.J.; Jia, Y.X.; Zhao, G.S. Aircraft recognition using modular extreme learning machine. *Neurocomputing* **2014**, *128*, 166–174. [CrossRef]
8. Hsieh, J.W.; Chen, J.M.; Chuang, C.H.; Fan, K.C. Aircraft type recognition in satellite images. *IEE Proc. Vision Image Signal Process.* **2005**, *152*, 307–315. [CrossRef]
9. Xu, C.; Duan, H. Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft. *Pattern Recognit. Lett.* **2010**, *31*, 1759–1772. [CrossRef]
10. Lindeberg, T. Scale Invariant Feature Transform. 2012. Available online: http://https://www.diva-portal.org/smash/record.jsf (accessed on 24 May 2012).
11. Wang, X.; Han, T.X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 32–39.
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
13. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
14. Chakraborty, S.; Roy, M. A neural approach under transfer learning for domain adaptation in land-cover classification using two-level cluster mapping. *Appl. Soft Comput.* **2018**, *64*, 508–525. [CrossRef]

15. Dang, L.M.; Hassan, S.I.; Suhyeon, I.; kumar Sangaiah, A.; Mehmood, I.; Rho, S.; Seo, S.; Moon, H. UAV based wilt detection system via convolutional neural networks. *Sustain. Comput. Inform. Syst.* **2018**, *28*, 100250. [CrossRef]
16. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic Ship Detection of Remote Sensing Images from Google Earth in Complex Scenes Based on Multi-Scale Rotation Dense Feature Pyramid Networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
17. Chen, Y.; Yang, X.; Zhong, B.; Pan, S.; Chen, D.; Zhang, H. CNNTracker: Online discriminative object tracking via deep convolutional neural network. *Appl. Soft Comput.* **2016**, *38*, 1088–1098. [CrossRef]
18. Han, Y.; Deng, C.; Zhao, B.; Tao, D. State-aware anti-drift object tracking. *IEEE Trans. Image Process.* **2019**, *28*, 4075–4086. [CrossRef] [PubMed]
19. Han, Y.; Deng, C.; Zhao, B.; Zhao, B. Spatial-temporal context-aware tracking. *IEEE Signal Process. Lett.* **2019**, *26*, 500–504. [CrossRef]
20. Han, Y.; Deng, C.; Zhang, Z.; Li, J.; Zhao, B. Adaptive feature representation for visual tracking. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 1867–1870.
21. Zhao, Z.; Han, Y.; Xu, T.; Li, X.; Song, H.; Luo, J. A Reliable and Real-Time Tracking Method with Color Distribution. *Sensors* **2017**, *17*, 2303. [CrossRef]
22. Gao, X.; Sun, X.; Zhang, Y.; Yan, M.; Xu, G.; Sun, H.; Jiao, J.; Fu, K. An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network. *IEEE Access* **2018**, *6*, 39401–39414. [CrossRef]
23. Mittal, M.; Goyal, L.M.; Kaur, S.; Kaur, I.; Verma, A.; Hemanth, D.J. Deep learning based enhanced tumor segmentation approach for MR brain images. *Appl. Soft Comput.* **2019**, *78*, 346–354. [CrossRef]
24. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [CrossRef]
25. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990. [CrossRef]
26. Huang, G.B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef]
27. Huang, G.B.; Bai, Z.; Kasun, L.L.C.; Vong, C.M. Local receptive fields based extreme learning machine. *IEEE Comput. Intell. Mag.* **2015**, *10*, 18–29. [CrossRef]
28. Zhu, W.; Miao, J.; Qing, L.; Huang, G.B. Hierarchical extreme learning machine for unsupervised representation learning. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.
29. Kasun, L.L.C.; Zhou, H.; Huang, G.B.; Vong, C.M. Representational learning with extreme learning machine for big data. *IEEE Intell. Syst.* **2013**, *28*, 31–34.
30. Zong, W.; Huang, G.B.; Chen, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing* **2013**, *101*, 229–242. [CrossRef]
31. Zhou, T.; Yao, L.; Zhang, Y. Graph regularized discriminant analysis and its application to face recognition. In Proceedings of the IEEE International Conference on Image Processing, Quebec City, QC, Canada, 27–30 September 2015.
32. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01), Vancouver, BC, Canada, 3–8 December 2001; MIT Press: Cambridge, MA, USA, 2001; pp. 585–591.
33. Sun, K.; Zhang, J.; Zhang, C.; Hu, J. Generalized extreme learning machine autoencoder and a new deep neural network. *Neurocomputing* **2017**, *230*, 374–381. [CrossRef]
34. Ge, H.; Sun, W.; Zhao, M.; Yao, Y. Stacked Denoising Extreme Learning Machine Autoencoder Based on Graph Embedding for Feature Representation. *IEEE Access* **2019**, *7*, 13433–13444. [CrossRef]
35. Wu, Z.Z.; Wan, S.H.; Wang, X.F.; Tan, M.; Zou, L.; Li, X.L.; Chen, Y. A benchmark data set for aircraft type recognition from remote sensing images. *Appl. Soft Comput.* **2020**, *89*, 106132. [CrossRef]
36. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
37. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
38. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
39. 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, 7–12 June 2015; pp. 1–9.
40. 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.
41. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
44. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
45. Hoiem, D.; Divvala, S.K.; Hays, J.H. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Development Kit. *World Lit. Today*. 2009. Available online: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (accessed on 15 October 2007).
46. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
47. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
48. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
49. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
50. Boureau, Y.L.; Ponce, J.; LeCun, Y. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 111–118.
51. Yu, D.; Wang, H.; Chen, P.; Wei, Z. Mixed pooling for convolutional neural networks. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Shanghai, China, 24–26 October 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 364–375.
52. Lee, C.Y.; Gallagher, P.W.; Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Cadiz, Spain, 9–11 May 2016; pp. 464–472.
53. Gulcehre, C.; Cho, K.; Pascanu, R.; Bengio, Y. Learned-norm pooling for deep feedforward and recurrent neural networks. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 15–19 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 530–546.
54. Tenenbaum, J.B.; Freeman, W.T. Separating style and content with bilinear models. *Neural Comput.* **2000**, *12*, 1247–1283. [CrossRef]
55. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv* **2016**, arXiv:1606.01847.
56. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
57. Hussain, Z.; Gimenez, F.; Yi, D.; Rubin, D. Differential Data Augmentation Techniques for Medical Imaging Classification Tasks. *AMIA Annu. Symp. Proc.* **2017**, *2017*, 979–984.
58. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning augmentation strategies from data. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
59. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv* **2017**, arXiv:1710.09412.
60. Liang, D.; Yang, F.; Zhang, T.; Yang, P. Understanding mixup training methods. *IEEE Access* **2018**, *6*, 58774–58783. [CrossRef]
61. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; pp. 1–6.
62. Inoue, H. Data augmentation by pairing samples for images classification. *arXiv* **2018**, arXiv:1801.02929.
63. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. Available online: http://dl.acm.org/doi/10.5555/2969033.2969125 (accessed on 8 December 2014).
64. Leng, B.; Yu, K.; Jingyan, Q. Data augmentation for unbalanced face recognition training sets. *Neurocomputing* **2017**, *235*, 10–14. [CrossRef]
65. Zhu, X.; Liu, Y.; Li, J.; Wan, T.; Qin, Z. Emotion classification with data augmentation using generative adversarial networks. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Melbourne, VIC, Australia, 3–6 June; Springer: Berlin/Heidelberg, Germany, 2018; pp. 349–360.
66. Frid-Adar, M.; Diamant, I.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing* **2018**, *321*, 321–331. [CrossRef]
67. Fawzi, A.; Samulowitz, H.; Turaga, D.; Frossard, P. Adaptive data augmentation for image classification. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3688–3692.
68. Lemley, J.; Bazrafkan, S.; Corcoran, P. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access* **2017**, *5*, 5858–5869. [CrossRef]

69. Ratner, A.J.; Ehrenberg, H.R.; Hussain, Z.; Dunnmon, J.; Ré, C. Learning to compose domain-specific transformations for data augmentation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3239.
70. Tran, T.; Pham, T.; Carneiro, G.; Palmer, L.; Reid, I. A bayesian data augmentation approach for learning deep models. *arXiv* **2017**, arXiv:1710.10564.
71. Yan, D.; Chu, Y.; Zhang, H.; Liu, D. Information discriminative extreme learning machine. *Soft Comput. A Fusion Found. Methodol. Appl.* **2018**, *22*, 677–689. [CrossRef]
72. Peng, Y.; Lu, B.L. Discriminative extreme learning machine with supervised sparsity preserving for image classification. *Neurocomputing* **2017**, *261*, 242–252. [CrossRef]
73. Yan, S.; Xu, D.; Zhang, B.; Zhang, H.; Yang, Q.; Lin, S. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 40–51. [CrossRef] [PubMed]
74. Pedagadi, S.; Orwell, J.; Velastin, S.; Boghossian, B. Local fisher discriminant analysis for pedestrian re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3318–3325.
75. Atmane, K.; Hongbin, M.; Qing, F. Convolutional Neural Network Based on Extreme Learning Machine for Maritime Ships Recognition in Infrared Images. *Sensors* **2018**, *18*, 1490.
76. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear cnn models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1449–1457.
77. Charikar, M.; Chen, K.; Farach-Colton, M. Finding frequent items in data streams. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Malaga, Spain, 8–13 July 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 693–703.
78. Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In Proceedings of the Workshop on Statistical Learning in Computer Vision, ECCV, Prague, Czech Republic, 16 May 2004; Volume 1, pp. 1–2.
79. Heikkilä, M.; Pietikäinen, M.; Schmid, C. Description of interest regions with local binary patterns. *Pattern Recognit.* **2009**, *42*, 425–436. [CrossRef]